A Survey on the Computation of Quaternions from Rotation Matrices

Soheil Sarabandi and Federico Thomas

Institut de Robòtica i Informàtica Industrial (CSIC-UPC)
Llorens Artigas 4-6, 08028 Barcelona, Spain
{ssarabandi, fthomas}@iri.upc.edu

Abstract

The parameterization of rotations is a central topic in many theoretical and applied fields such as rigid body mechanics, multibody dynamics, robotics, spacecraft attitude dynamics, navigation, 3D image processing, computer graphics, etc. Nowadays, the main alternative to the use of rotation matrices, to represent rotations in $\mathbb{R}^3$, is the use of Euler parameters arranged in quaternion form. Whereas the passage from a set of Euler parameters to the corresponding rotation matrix is unique and straightforward, the passage from a rotation matrix to its corresponding Euler parameters has been revealed to be somewhat tricky if numerical aspects are considered. Since the map from quaternions to 3×3 rotation matrices is a 2-to-1 covering map, this map cannot be smoothly inverted. As a consequence, it is erroneously assumed that all inversions should necessarily contain singularities that arise in the form of quotients where the divisor can be arbitrarily small. This misconception is herein clarified.

This paper reviews the most representative methods available in the literature, including a comparative analysis of their computational costs and error performances. The presented analysis leads to the conclusion that Cayley's factorization, a little-known method used to compute the double quaternion representation of rotations in four dimensions from 4×4 rotation matrices, is the most robust method when particularized to three dimensions.

Keywords: Euler parameters, quaternions, rotation matrices, numerical accuracy.

1 Introduction

Arbitrary rotations in $\mathbb{R}^3$ can be represented using proper orthogonal 3×3 matrices (*rotation matrices* for short) of the form:

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}. \tag{1}$$

Since these matrices have 9 elements, while only 3 are independent, rotations represented in this way are somewhat cumbersome to manipulate. As a consequence, sets of fewer parameters have been proposed to represent rotations. These sets include Euler parameters, Rodrigues parameters, Euler angles, Cayley-Klein parameters, etc. [1, 2]. Euler and Cayley-Klein parameters require only 4 elements, which is the minimum number for a representation of rotations in $\mathbb{R}^3$ to be non-singular. Moreover, these two parameterizations are quite convenient because, when concatenating rotations, they are manipulated using the algebra of quaternions or the algebra of spinors, respectively.

Euler parameters arranged in quaternion form have gained the favor of the engineering community and, hence, the interest of computing them in a simple, fast, and numerically stable way.

Euler's theorem of rigid-body rotations states that the orientation of a body after having undergone any sequence of rotations is equivalent to a single rotation of that body through an angle θ about an axis that we will represent by the unit vector $\mathbf{n} = (n_x\, n_y\, n_z)^T$ (see [3, pp. 118-123] for a proof of this theorem in terms of rotation matrices). The rotation matrix in (1), expressed in terms of $\mathbf{n}$ and θ, has the following form (see [4, p. 30] for an elementary deduction):

$$\mathbf{R}(\hat{\mathbf{n}}, \theta) = \begin{pmatrix} c + n_x^2(1-c) & n_x n_y(1-c) - n_z s & n_x n_z(1-c) + n_y s \\ n_y n_x(1-c) + n_z s & c + n_y^2(1-c) & n_y n_z(1-c) - n_x s \\ n_z n_x(1-c) - n_y s & n_z n_y(1-c) + n_x s & c + n_z^2(1-c) \end{pmatrix} \tag{2}$$

where $s = \sin\theta$ and $c = \cos\theta$. Now, if we introduce the following change of variables

$$e_0 = \cos(\theta/2), \tag{3}$$

$$e_1 = n_x \sin(\theta/2), \tag{4}$$

$$e_2 = n_y \sin(\theta/2), \tag{5}$$

$$e_3 = n_z \sin(\theta/2), \tag{6}$$

then (2) can be rewritten as

$$\mathbf{R}(e_0, e_1, e_2, e_3) = \begin{pmatrix} 2(e_0^2+e_1^2)-1 & 2(e_1e_2-e_0e_3) & 2(e_1e_3+e_0e_2) \\ 2(e_1e_2+e_0e_3) & 2(e_0^2+e_2^2)-1 & 2(e_1e_3-e_0e_1) \\ 2(e_1e_3-e_0e_2) & 2(e_2e_3+e_0e_1) & 2(e_0^2+e_3^2)-1 \end{pmatrix}. \tag{7}$$

The parameters e_0, e_1, e_2, e_3 are defined as the *Euler parameters*. As expected, these parameters are not independent because only three are needed to represent an arbitrary rotation in $\mathbb{R}^3$. They are related through the following equation:

$$e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1. \tag{8}$$

In practice, this condition can be relaxed so that $(e_0, \ldots, e_3)$ is treated as a vector of homogeneous coordinates [5]. In this case, when (8) is not satisfied, a normalization is required prior to obtaining the corresponding rotation matrix using (7).

In what follows, we use the following notation:

$$\mathbf{e} = (e_1, e_2, e_3)^T, \tag{9}$$

$$\bar{\mathbf{e}} = (e_0\ \mathbf{e})^T = (e_0\ e_1\ e_2\ e_3)^T. \tag{10}$$

It is easy to conclude, by simply observing (7), that the Euler parameters provide a double covering of the space of rotations in the sense that $\bar{\mathbf{e}}$ and $-\bar{\mathbf{e}}$ represent the same rotation matrix. This fact is obviously present in all methods that compute these parameters: they all give the same solution within an undetermined overall sign.

As we have already said, there are no singularities associated with Euler parameters, contrarily to what happens for example with Euler angles [2]. However, this does not mean that the methods to compute them could not introduce their own singularities, as we will see below.

The most straightforward way to obtain the set of Euler parameters consists in solving the system of nonlinear equations resulting from equating the matrices in (1) and (7). From an algebraic point of view, the solution to these non-linear equations must avoid dividing by zero and taking the square root of negatives numbers. Nevertheless, from a computational point of view, the conditions are more strict: the solution must minimize possible floating-point rounding errors, which might be relevant, for example, when dividing by (or when taking the square root of) a very small number [6]. As we will see, this loss of accuracy near singularities is the essential problem in many methods despite their algebraic correctness.

The available methods for computing the Euler parameters from a rotation matrices are varied: they are based on trigonometric, algebraic, or numerical techniques. In aerial navigation, some of them have been named for their inventors. This is the case of Hughes', Shepperd's, and Bar-Itzhack's methods. Although, in general, the origin of most methods and their variations is actually uncertain because of their trivial derivation, we mostly adhere to the names given to the different methods by the aerial navigation community.

This paper is organized as follows. We start in Section 2 with two trigonometric methods which can be seen as indirect methods because they consist in obtaining a different set of parameters to represent an orientation which is then converted into Euler parameters. Then, in Section 3, we review the algebraic methods such as the well-known Hughes' and Shepperd's methods. In Section 4, we review two numerical methods. In Section 5, a detailed comparison in terms of computational cost and error performance of the reviewed methods is presented. Finally, Section 6 summarizes the main conclusions.

2 Trigonometric methods

To the best of our knowledge, although trigonometric methods are implicit in the first documents dealing with orientation representation in aeronautics [7], it seems to have attracted little subsequent attention.

2.1 Trigonometric method 1

This method consists in first extracting a set of Euler angles from $\mathbf{R}$, and then converting the result to Euler parameters. Different Euler angle conventions can be used to this end. For example, in [8], the $x-y-z$ convention is used. Nevertheless, here we propose to use the $z-x-z$ convention because the resulting formulas are much simpler.

Let us denote θ_1, θ_2, and θ_3 a sequence of angles rotated, in local reference frames, about the z, x, and z axes, respectively. Then, multiplying the corresponding three rotations matrices yields [9]:

$$\mathbf{R}_z(\theta_1)\mathbf{R}_x(\theta_2)\mathbf{R}_z(\theta_3) = \begin{pmatrix} -s_1\,c_2\,s_3 + c_1\,c_3 & -s_1\,c_2\,c_3 - c_1\,c_3 & s_2\,s_1 \\ c_1\,c_2\,s_3 + s_1\,c_3 & c_2\,c_1\,c_3 - s_1\,s_3 & -s_2\,c_1 \\ s_2\,s_3 & s_2\,c_3 & c_2 \end{pmatrix}, \tag{11}$$

where s_i and c_i stand for $\sin\theta_i$ and $\cos\theta_i$, respectively. By equating this matrix to (7), we can obtain θ_1, θ_2, and θ_3 as a function of r_{ij}, $1 \leq i, j \leq 3$. The simplest term to work with is $\cos\theta_2 = r_{33}$, so $\theta_2 = \arccos r_{33}$. Then, there are three cases to consider (see [9] for details):

- If $\theta_2 \in (0, \pi)$, then $\theta_1 = \text{atan2}(r_{13}, -r_{23})$, and $\theta_3 = \text{atan2}(r_{31}, r_{32})$.
- If $\theta_2 = 0$, then $\theta_3 + \theta_1 = \text{atan2}(-r_{12}, r_{11})$.
- If $\theta_2 = \pi$, then $\theta_3 - \theta_1 = \text{atan2}(-r_{12}, r_{11})$.

Now, the rotation described as a combination of the above Euler angles can be expressed as Euler parameters as follows (see Appendix A in [7]):

$$e_0 = \cos\left(\frac{\theta_2}{2}\right)\cos\left(\frac{\theta_1+\theta_3}{2}\right), \tag{12}$$

$$e_1 = \sin\left(\frac{\theta_2}{2}\right)\cos\left(\frac{\theta_1-\theta_3}{2}\right), \tag{13}$$

$$e_2 = \sin\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1-\theta_3}{2}\right), \tag{14}$$

$$e_3 = \cos\left(\frac{\theta_2}{2}\right)\sin\left(\frac{\theta_1+\theta_3}{2}\right). \tag{15}$$

This method requires only four additions, four multiplications, and the evaluation of eleven trigonometric functions or their inverses. However, observe that it uses neither square roots nor divisions by variable quantities, which is a good indication of numerical stability.

2.2 Trigonometric method 2

This other indirect method consist in first computing $\mathbf{n}$ and θ, and then converting them to Euler parameters, which is a common practice in Robotics. Here we will follow the description given in [4].

From equating the matrices in (1) and (2), it is possible to verify that

$$\sin\theta = \frac{1}{2}\sqrt{(r_{32} - r_{23})^2 + (r_{13} - r_{31})^2 + (r_{21} - r_{12})^2}, \tag{16}$$

and

$$\cos\theta = \frac{1}{2}(r_{11} + r_{22} + r_{33} - 1). \tag{17}$$

Therefore,

$$\theta = \arctan\left(\frac{\sqrt{(r_{32} - r_{23})^2 + (r_{13} - r_{31})^2 + (r_{21} - r_{12})^2}}{r_{11} + r_{22} + r_{22} - 1}\right). \tag{18}$$

Moreover, it is possible to conclude that

$$n_x = \frac{r_{32} - r_{23}}{2\sin\theta}, \tag{19}$$

$$n_y = \frac{r_{13} - r_{31}}{2\sin\theta}, \tag{20}$$

$$n_z = \frac{r_{21} - r_{12}}{2\sin\theta}. \tag{21}$$

When the angle of rotation is very small, the axis of rotation is physically not well defined due to the small magnitude of both numerator and denominator in (19)-(21). In this case, it is important to ensure that **n** is a unit vector by renormalizing it. When the angle of rotation approaches π radians, the vector **n** is once again poorly defined. In this case, for $\theta > \pi/2$, **n** is determined as follows:

$$n_x = \text{sign}(r_{32} - r_{23})\sqrt{\frac{r_{11} - \cos\theta}{1 - \cos\theta}}, \tag{22}$$

$$n_y = \text{sign}(r_{13} - r_{31})\sqrt{\frac{r_{22} - \cos\theta}{1 - \cos\theta}}, \tag{23}$$

$$n_z = \text{sign}(r_{21} - r_{12})\sqrt{\frac{r_{33} - \cos\theta}{1 - \cos\theta}}. \tag{24}$$

The idea here is that we only keep the largest of the three components of **n** given above, and the other two are computed according to the following rules:

- If n_x is the largest, then

$$n_y = \frac{r_{21} + r_{12}}{2n_x(1 - \cos\theta)}, \tag{25}$$

$$n_z = \frac{r_{13} + r_{31}}{2n_x(1 - \cos\theta)}. \tag{26}$$

- If n_y is the largest, then

$$n_x = \frac{r_{21} + r_{12}}{2n_y(1 - \cos\theta)}, \tag{27}$$

$$n_z = \frac{r_{32} + r_{23}}{2n_y(1 - \cos\theta)}. \tag{28}$$

- If n_z is the largest, then

$$n_x = \frac{r_{31} + r_{13}}{2n_z(1 - \cos\theta)}, \tag{29}$$

$$n_y = \frac{r_{32} + r_{23}}{2n_z(1 - \cos\theta)}. \tag{30}$$

This method is complicated compared to all others. We will show that, in general, it should be avoided.

3 Algebraic methods

The algebraic methods are directly based on solving the system of equations resulting from equating the matrices in (1) and (7), which can be expressed as:

$$\begin{align}
4e_0^2 &= 1 + r_{11} + r_{22} + r_{33}, \tag{31}\\
4e_1^2 &= 1 + r_{11} - r_{22} - r_{33}, \tag{32}\\
4e_2^2 &= 1 - r_{11} + r_{22} - r_{33}, \tag{33}\\
4e_3^2 &= 1 - r_{11} - r_{22} + r_{33}, \tag{34}\\
4e_2e_3 &= r_{23} + r_{32}, \tag{35}\\
4e_1e_3 &= r_{31} + r_{13}, \tag{36}\\
4e_1e_2 &= r_{12} + r_{21}, \tag{37}\\
4e_0e_1 &= r_{32} - r_{23}, \tag{38}\\
4e_0e_2 &= r_{13} - r_{31}, \tag{39}\\
4e_0e_3 &= r_{21} - r_{12}. \tag{40}
\end{align}$$

This system of equations can be organized in a more compact way by defining the *matrix of products* as:

$$\mathbf{P} := \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix} \begin{pmatrix} e_0 & e_1 & e_2 & e_3 \end{pmatrix} = \begin{pmatrix} e_0e_0 & e_0e_1 & e_0e_2 & e_0e_3 \\ e_1e_0 & e_1e_1 & e_1e_2 & e_1e_3 \\ e_2e_0 & e_2e_1 & e_2e_2 & e_2e_3 \\ e_3e_0 & e_3e_1 & e_3e_2 & e_3e_3 \end{pmatrix}. \tag{41}$$

Therefore, using equations (31)-(40), it can be verified that

$$\mathbf{P} = \frac{1}{4}\begin{pmatrix} r_{11}+r_{22}+r_{33}+1 & r_{32}-r_{23} & r_{13}-r_{31} & r_{21}-r_{12} \\ r_{32}-r_{23} & r_{11}-r_{22}-r_{33}+1 & r_{21}+r_{12} & r_{31}+r_{13} \\ r_{13}-r_{31} & r_{21}+r_{12} & r_{22}-r_{11}-r_{33}+1 & r_{32}+r_{23} \\ r_{21}-r_{12} & r_{31}+r_{13} & r_{32}+r_{23} & r_{33}-r_{11}-r_{22}+1 \end{pmatrix}. \tag{42}$$

3.1 Chiaverini-Siciliano's method

This is the most straightforward algebraic method. It is used in [18] and hence the name adopted here. From (31)-(34), we have that:

$$e_0 = \frac{1}{2}\sqrt{1+r_{11}+r_{22}+r_{33}}, \tag{43}$$

$$e_1 = \frac{1}{2}\sqrt{1+r_{11}-r_{22}-r_{33}}, \tag{44}$$

$$e_2 = \frac{1}{2}\sqrt{1-r_{11}+r_{22}-r_{33}}, \tag{45}$$

$$e_3 = \frac{1}{2}\sqrt{1-r_{11}-r_{22}+r_{33}}. \tag{46}$$

Due to the global undefined sign, if we assume that e_0 is positive, then, according to (38)-(40), we have to assign e_1, e_2 and e_3 the signs of $r_{32}-r_{23}$, $r_{13}-r_{31}$ and $r_{21}-r_{12}$, respectively.

This method has no singularities. It contains no multiplications nor divisions, and all Euler parameters are treated in a similar way. It only requires 12 additions and 4 square roots. However, this method only takes into account the elements of the diagonal of $\mathbf{R}$. We will see how considering the values off the diagonal of $\mathbf{R}$ is necessary to obtain numerically accurate results.

3.2 Hughes' method

Despite its limitations which will become clear later, Hughes' method [14] it is still commonly used in many engineering areas (see, for example, [15, pp. 122-123] and [16, p. 153]).

As in Chiaverini-Siciliano's method, the first Euler parameter is given by:

$$e_0 = \tfrac{1}{2}\sqrt{r_{11}+r_{22}+r_{33}+1}, \tag{47}$$

and if $e_0 \neq 0$, from (38)-(40), we have that:

$$e_1 = \frac{r_{32} - r_{23}}{4e_0}, \tag{48}$$

$$e_2 = \frac{r_{13} - r_{31}}{4e_0}, \tag{49}$$

$$e_3 = \frac{r_{21} - r_{12}}{4e_0}. \tag{50}$$

If $e_0 = 0$, (31)-(40) can be rewritten as:

$$e_1 = \pm\sqrt{\frac{1 + r_{11}}{2}}, \tag{51}$$

$$e_2 = \pm\sqrt{\frac{1 + r_{22}}{2}}, \tag{52}$$

$$e_3 = \pm\sqrt{\frac{1 + r_{33}}{2}}, \tag{53}$$

$$e_1 e_2 = \frac{r_{12}}{2}, \tag{54}$$

$$e_2 e_3 = \frac{r_{23}}{2}, \tag{55}$$

$$e_3 e_1 = \frac{r_{31}}{2}. \tag{56}$$

Again, equations (54)-(56) can serve to resolve the sign ambiguities in (51)-(53). First, observe from (54)-(56) that (a) r_{12}, r_{23}, and r_{31} cannot be simultaneously negative; and (b) if one of them is negative, another one has also to be negative. Therefore, assuming that e_1, e_2, and e_3 are initially positive, their signs have to be changed according to the following rules:

- If $r_{31} < 0$ and $r_{12} < 0$, then we have to change the sign of e_1.
- If $r_{12} < 0$ and $r_{23} < 0$, then we have to change the sign of e_2.
- If $r_{23} < 0$ and $r_{31} < 0$, then we have to change the sign of e_3.

To avoid this sign disambiguation, an alternative formulation can be found in [16, p. 153], which is adapted from [15, p. 122], where the following formulas are given for the case in which $e_0 = 0$:

$$e_1 = \frac{r_{13} r_{12}}{\sqrt{r_{12}^2 r_{13}^2 + r_{12}^2 r_{23}^2 + r_{13}^2 r_{23}^2}}, \tag{57}$$

$$e_2 = \frac{r_{12} r_{23}}{\sqrt{r_{12}^2 r_{13}^2 + r_{12}^2 r_{23}^2 + r_{13}^2 r_{23}^2}}, \tag{58}$$

$$e_3 = \frac{r_{13} r_{23}}{\sqrt{r_{12}^2 r_{13}^2 + r_{12}^2 r_{23}^2 + r_{13}^2 r_{23}^2}}. \tag{59}$$

Unfortunately, these formulas fail if $r_{12} = r_{13} = 0$, or $r_{12} = r_{23} = 0$, or $r_{23} = r_{13} = 0$, because they lead to indeterminations of the form $0/0$. They correspond precisely to the cases in which the input rotation matrix represents a pure rotation about the x, y, or z axis, respectively. Since (57)-(59) are only used in the case in which $e_0 = 0$, these situations are already excluded except in those cases in which these rotations about the coordinate axes are of π radians.

The essential problem with Hughes' method is its poor behavior when $1 + r_{11} + r_{22} + r_{33} \to 0$. To alleviate this situation, Grubin [17] proposed an algorithm that consisted in computing the following three alternative solutions:

$$\bar{\mathbf{e}}_1 = \frac{1}{2}\begin{pmatrix} \sqrt{r_{11} + r_{22} + r_{33} + 1} \\ \sqrt{2(1 + r_{11})} \\ (r_{11} + r_{21})/\sqrt{2(1 + r_{11})} \\ (r_{31} + r_{13})/\sqrt{2(1 + r_{11})} \end{pmatrix}, \tag{60}$$

$$\bar{\mathbf{e}}_2 = \frac{1}{2}\begin{pmatrix} \sqrt{r_{11} + r_{22} + r_{33} + 1} \\ (r_{11} + r_{21})/\sqrt{2(1 + r_{2,2})} \\ \sqrt{2(1 + r_{22})} \\ (r_{32} + r_{23})/\sqrt{2(1 + r_{2,2})} \end{pmatrix}, \tag{61}$$

$$\bar{\mathbf{e}}_2 = \frac{1}{2}\begin{pmatrix} \sqrt{r_{11} + r_{22} + r_{33} + 1} \\ (r_{31} + r_{13})/\sqrt{2(1 + e_{3,3})} \\ (r_{32} + r_{23})/\sqrt{2(1 + e_{3,3})} \\ \sqrt{2(1 + r_{33})} \end{pmatrix}. \tag{62}$$

Then, if we determine the ordinal number i of the largest element in of the following vector

$$\begin{pmatrix} r_{11} \\ r_{22} \\ r_{33} \end{pmatrix}, \tag{63}$$

it is not difficult to prove that the best solution, from the numerical point of view, is $\bar{\mathbf{e}}_i$. Using Grubin's improvement, it is possible to compute the Euler parameters using Hughes' method for rotation matrices whose equivalent rotated angle is $\pi - \epsilon$, where ϵ is as small as 2×10^{-4} [17].

Grubin introduced an idea further exploited by Shepperd's method: we can choose from several alternative solutions using a voting scheme.

3.3 Shepperd's method

Since it was first proposed in [19], Shepperd's method remains as one of the most popular methods. It can be seen as an improvement on Hughes' method in which the Euler parameters are computed without numerical instabilities. It can be seen as an evolution of Grubin's [10], Klumpp's [11], Spurrier's[12], and Klumpp's [13] methods.

Observe that, in Hughes' method, e_0 is calculated first and then it is treated very differently from the remaining three parameters. Since we can solve the system of equations (31)-(40) for any of the four Euler parameters, there are four different formulas for computing the Euler parameters as a function of the input rotation matrix, all of them formally equivalent. Numerically, however, these four formulas are not identical and, depending on the rotation matrix, one of them is numerically better conditioned than the others.

From the system of equations (31)-(40), we arrive at these four different solutions:

$$\bar{\mathbf{e}}_1 = \frac{1}{2}\begin{pmatrix} (1+r_{11}+r_{22}+r_{33})^{\frac{1}{2}} \\ (r_{32}-r_{23})/(1+r_{11}+r_{22}+r_{33})^{\frac{1}{2}} \\ (r_{13}-r_{31})/(1+r_{11}+r_{22}+r_{33})^{\frac{1}{2}} \\ (r_{21}-r_{12})/(1+r_{11}+r_{22}+r_{33})^{\frac{1}{2}} \end{pmatrix}, \tag{64}$$

$$\bar{\mathbf{e}}_2 = \frac{1}{2}\begin{pmatrix} (r_{32}-r_{23})/(1+r_{11}-r_{22}-r_{33})^{\frac{1}{2}} \\ (1+r_{11}-r_{22}-r_{33})^{\frac{1}{2}} \\ (r_{12}+r_{21})/(1+r_{11}-r_{22}-r_{33})^{\frac{1}{2}} \\ (r_{31}+r_{13})/(1+r_{11}-r_{22}-r_{33})^{\frac{1}{2}} \end{pmatrix}, \tag{65}$$

$$\bar{\mathbf{e}}_3 = \frac{1}{2}\begin{pmatrix} (r_{13}-r_{31})/(1-r_{11}+r_{22}-r_{33})^{\frac{1}{2}} \\ (r_{12}+r_{21})/(1-r_{11}+r_{22}-r_{33})^{\frac{1}{2}} \\ (1-r_{11}+r_{22}-r_{33})^{\frac{1}{2}} \\ (r_{23}+r_{32})/(1-r_{11}+r_{22}-r_{33})^{\frac{1}{2}} \end{pmatrix}, \tag{66}$$

$$\bar{\mathbf{e}}_4 = \frac{1}{2}\begin{pmatrix} (r_{21}-r_{12})/(1-r_{11}-r_{22}+r_{33})^{\frac{1}{2}} \\ (r_{31}+r_{13})/(1-r_{11}-r_{22}+r_{33})^{\frac{1}{2}} \\ (r_{32}+r_{23})/(1-r_{11}-r_{22}+r_{33})^{\frac{1}{2}} \\ (1-r_{11}-r_{22}+r_{33})^{\frac{1}{2}} \end{pmatrix}. \tag{67}$$

Depending on the entries of $\mathbf{R}$, some of these functions can even lead to complex solutions. To avoid such a situation, we determine the ordinal number i of the largest element in the following vector

$$\begin{pmatrix} r_{11}+r_{22}+r_{33} \\ r_{11} \\ r_{22} \\ r_{33} \end{pmatrix}. \tag{68}$$

Then, the best solution, from the numerical point of view, is considered to be $\bar{\mathbf{e}}_i$. The result is a method without numerical instabilities.

This four-fold multiplicity of the solution arises in other methods. For example, the one presented in [20], based on geometric arguments, was shown to be equivalent to this method.

3.4 Sarabandi-Thomas' method

This method, recently presented in [22], moves the voting scheme to the computation of each Euler parameter.

If we only want to compute e_0, we can directly use (43). That is,

$$e_0 = \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}}. \tag{69}$$

The term inside the square root lies in the interval $[0, 4]$. Indeed, observe that $\text{Trace}(\mathbf{R}) = r_{11}+r_{22}+r_{33} = 2\cos\theta+1$ [49, Section 2.3]. Unfortunately, numerical problems arise when this term gets close to zero. In practice, it can even become negative due to rounding errors. Since this term coincides with $2+2\cos\theta$, (69) becomes ill-conditioned when $\theta \to \pi$. Observe that (69) only takes into account the diagonal entries of $\mathbf{R}$. To obtain an alternative formula involving all the elements of the rotation matrix, let us substitute in (8) the values of e_0^2, e_1^2, e_2^2, and e_3^2 obtained from (31), (38), (39), and (40), respectively. The result is:

$$\frac{1+r_{11}+r_{22}+r_{33}}{4} + \left(\frac{r_{32}-r_{23}}{4e_0}\right)^2 + \left(\frac{r_{13}-r_{31}}{4e_0}\right)^2 + \left(\frac{r_{21}-r_{12}}{4e_0}\right)^2 = 1 \tag{70}$$

Solving the above equation for e_0, we obtain

$$e_0 = \frac{1}{2}\sqrt{\frac{(r_{32}-r_{23})^2+(r_{13}-r_{31})^2+(r_{21}-r_{12})^2}{3-r_{11}-r_{22}-r_{33}}}. \tag{71}$$

Now, the term in the denominator of (71) also lies in the interval $[0, 4]$. Since this denominator coincides with $2-2\cos\theta$, (71) is ill-conditioned for $\theta \to 0$.

When this happens, the diagonal of $\mathbf{R}$ is dominant and, as a consequence, the numerator in (71) tends also to be small. Thus, (43) and (71) can be seen as complementary. As a consequence, it is reasonable to establish a threshold for the trace of $\mathbf{R}$, whose optimal value is found to be equal to 0 in [22], above which it is preferable to use (71) instead of (43). In other words, we have that

$$e_1 = \begin{cases} \frac{1}{2}\sqrt{1{+}r_{11}{+}r_{22}{+}r_{33}}, & \text{if } r_{11}{+}r_{22}{+}r_{33} > 0, \\ \frac{1}{2}\sqrt{\dfrac{(r_{32}{-}r_{23})^2{+}(r_{13}{-}r_{31})^2{+}(r_{21}{-}r_{12})^2}{3{-}r_{11}{-}r_{22}{-}r_{33}}}, & \text{otherwise.} \end{cases} \tag{72}$$

Extending this reasoning to the computation of the other elements of the quaternion, the result is:

$$e_2 = \begin{cases} \frac{1}{2}\sqrt{1{+}r_{11}{-}r_{22}{-}r_{33}}, & \text{if } r_{11}{-}r_{22}{-}r_{33} > 0, \\ \frac{1}{2}\sqrt{\dfrac{(r_{32}{-}r_{23})^2{+}(r_{12}{+}r_{21})^2{+}(r_{31}{+}r_{13})^2}{3{-}r_{11}{+}r_{22}{+}r_{33}}}, & \text{otherwise.} \end{cases} \tag{73}$$

$$e_3 = \begin{cases} \frac{1}{2}\sqrt{1{-}r_{11}{+}r_{22}{-}r_{33}}, & \text{if } {-}r_{11}{+}r_{22}{-}r_{33} > 0, \\ \frac{1}{2}\sqrt{\dfrac{(r_{13}{-}r_{31})^2{+}(r_{12}{+}r_{21})^2{+}(r_{23}{+}r_{32})^2}{3{+}r_{11}{-}r_{22}{+}r_{33}}}, & \text{otherwise.} \end{cases} \tag{74}$$

$$e_4 = \begin{cases} \frac{1}{2}\sqrt{1{-}r_{11}{-}r_{22}{+}r_{33}}, & \text{if } {-}r_{11}{-}r_{22}{+}r_{33} > 0, \\ \frac{1}{2}\sqrt{\dfrac{(r_{21}{-}r_{12})^2{+}(r_{31}{+}r_{13})^2{+}(r_{32}{+}r_{23})^2}{3{+}r_{11}{+}r_{22}{-}r_{33}}}, & \text{otherwise.} \end{cases} \tag{75}$$

Due to the presence of square roots, the signs of e_i, $i = 0, \ldots, 3$ are undefined. As in Chiaverini-Siciliano's method where these signs are undefined, if we assume that e_0 is positive, we have to assign e_2, e_3, and e_4, the signs of $r_{32}{-}r_{23}$, $r_{13}{-}r_{31}$, and $r_{21}{-}r_{12}$, respectively.

Observe that this method computes two alternative solutions for each Euler parameter. This implicitly means that this approach works with up to 16 alternative solutions for each set of Euler parameters, which should obviously lead to a global better numerical behavior than Shepperd's method.

3.5 Cayley's method

This method is actually used for the computation of the double quaternion representation of rotations in four dimensions [21]. It is based on the analysis of the matrix of products given in (41), where it is already particularized to three dimensions (see [23] for details).

Observe that the norm of the row or column vector i of $\mathbf{P}$ in (41) equals $e_{i-1}^2(e_0^2 + e_1^2 + e_2^2 + e_3^2) = e_{i-1}^2$. As a consequence, computing the norms of the row of the matrix in (42), we straightforwardly obtain the following formulas:

$$e_0 = \frac{1}{4}\sqrt{(r_{11}+r_{22}+r_{33}+1)^2 + (r_{32}-r_{23})^2 + (r_{13}-r_{31})^2 + (r_{21}-r_{12})^2}, \quad (76)$$

$$e_1 = \frac{1}{4}\sqrt{(r_{32}-r_{23})^2 + (r_{11}-r_{22}-r_{33}+1)^2 + (r_{21}+r_{12})^2 + (r_{31}+r_{13})^2}, \quad (77)$$

$$e_2 = \frac{1}{4}\sqrt{(r_{13}-r_{31})^2 + (r_{21}+r_{12})^2 + (r_{22}-r_{11}-r_{33}+1)^2 + (r_{32}+r_{23})^2}, \quad (78)$$

$$e_3 = \frac{1}{4}\sqrt{(r_{21}-r_{12})^2 + (r_{31}+r_{13})^2 + (r_{32}+r_{23})^2 + (r_{33}-r_{11}-r_{22}+1)^2}. \quad (79)$$

As in Chiaverini-Siciliano's method, if we assume that e_0 is positive, we can give a consistent set of signs to the other Euler parameters by simply assigning e_1, e_2, and e_3 the signs of $(r_{32}-r_{23})$, $(r_{13}-r_{31})$, and $(r_{21}-r_{12})$, respectively.

This method has important advantages with respect to all other algebraic methods:

1. It involves a single mapping. There is no voting scheme to select the best solution from a set of possible solutions.

2. It requires no divisions.

3. The sum of terms under the square root symbol can never be negative independently of any rounding error.

4. It involves all the elements of the rotation matrix in the computation of each Euler parameter.

According to these characteristics, it should perform much better than all other algebraic methods.

Since this method provides us with a single mapping that depends on all the entries of the rotation matrix, it allows us to straightforwardly obtain the derivatives of any Euler parameter with respect to any entry of the rotation matrix. For example,

$$\frac{\partial e_0}{\partial r_{11}} = \frac{r_{11} + r_{22} + r_{33} + 1}{8\, e_0}, \tag{80}$$

and

$$\frac{\partial e_0}{\partial r_{32}} = \frac{r_{32} - r_{23}}{8\, e_0}. \tag{81}$$

Thus, it can be easily checked that the derivative of e_i, $i = 0, \ldots, 3$, with respect to any of the entries of the rotation matrix tends to infinity as e_i tends to 0.

4 Numerical methods

All numerical methods reduce the problem of computing the quaternion representation of a given rotation matrix to obtain the eigenvector corresponding to a known eigenvalue, which in turn reduces to finding a matrix null space. This problem can be numerically solved in many different ways. To show how dependent these methods are on the adopted numerical method, two alternatives will be considered in the analysis given in Section 5, one based on the singular value decomposition (SVD) and the other on Gaussian elimination.

4.1 Coope et al.'s method

This method was proposed in [24]. In this case, e_0 is initially computed using (47). If e_0 is non-zero but below a certain threshold, for which numerical instabilities arise, the following refinement for e_0 is used

$$e_0 \leftarrow \frac{1}{4\sqrt{1 - e_0^2}} \operatorname{norm} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}. \tag{82}$$

This operation can actually be seen as a single step of a Newton-Raphson method to find a better approximation of a root.

In [24], the threshold is set at $e_0 = 0.1$, which corresponds to values of trace($\mathbf{R}$) $= -0.96$. Nevertheless, after some experiments using single-precision floating-point numbers, better results are obtained by increasing this threshold to trace($\mathbf{R}$) $= -0.3$.

To compute $\mathbf{e} = (e_1, e_2, e_3)^T$, this method relies on the property

$$\mathbf{R}\mathbf{e} = \mathbf{e}. \tag{83}$$

Thus, the problem is reduced to compute the one-dimensional null space basis of $\mathbf{R}{-}\mathbf{I}$. The approach adopted in [24] is slightly different because (83) can be rewritten as $(\mathbf{R} + \mathbf{R}^T)\mathbf{e} = 2\,\mathbf{e}$ to transform the problem into a real well-conditioned symmetric eigenvector problem for the eigenvalue equal to 2. Nevertheless, we have observed that the results are numerically more accurate when this extra transformation is not introduced when using Gaussian elimination.

If $\mathbf{v}$ is the obtained eigenvector for the eigenvalue 1, then

$$e_i = \sqrt{1 - e_0^2}\ v_i,\ i = 1, 2, 3. \tag{84}$$

Due to the presence of square roots, the signs of e_i, $i = 1, 2, 3$ are undefined. Again, as in Chiaverini-Siciliano's if e_0 is assumed to be positive, e_1, e_2, and e_3 have to be assigned the signs of $r_{32}{-}r_{23}$, $r_{13}{-}r_{31}$, and $r_{21}{-}r_{12}$, respectively.

4.2 Bar-Itzhack's method

Almost all algorithms for estimating spacecraft attitude (orientation) from vector measurements consists in finding the rotation matrix $\mathbf{R}$ that minimizes the function

$$\sum_i |\mathbf{b}_i - \mathbf{R}\mathbf{r}_i|^2\,, \tag{85}$$

where $\mathbf{r}_i$ and $\mathbf{b}_i$ are unit vectors in the global reference frame and in the body reference frame, respectively. This problem was first proposed by Wahba in 1965 [26]. The original formulation included the possibility of weighting each measurement which is removed here for the sake of simplicity.

Some early methods to solve this minimization problem can be found in [27]. The know most robust ones are Davenport's q-method [28, 29] and the SVD method [30]. Markley and Morari showed that the q-method performs better than the SVD method [31]. According to the q-method, the quaternion corresponding to the sought rotation matrix $\mathbf{R}$ is the eigenvector corresponding to the largest eigenvalue of the following 4×4 matrix [32, 33]:

$$\mathbf{K} = \begin{pmatrix} \sigma & \mathbf{z}^T \\ \mathbf{z} & \mathbf{S}-\sigma\mathbf{I}_3 \end{pmatrix}, \tag{86}$$

where

$$\sigma = \sum_i \mathbf{r}_i^T \mathbf{b}_i, \tag{87}$$

$$\mathbf{S} = \sum_i \mathbf{r}_i \mathbf{b}_i^T + \sum_i \mathbf{b}_i \mathbf{r}_i^T, \tag{88}$$

$$\mathbf{z} = \sum_i \mathbf{r}_i \times \mathbf{b}_i. \tag{89}$$

The definition of the matrix in (86) is not the same as the one given in [32, 33] because its entries have been permuted, and an irrelevant change of sign has been introduced here, to provide a neat connection with the other methods reviewed in this paper.

Several algorithms were presented to bypass the need of obtaining the maximum eigenvalue and the corresponding eigenvector like the QUEST method [34], and its evolutions [35, 36, 37, 38, 39], the ESOQ method [40, 41], the ESOQ2 method [42, 43], the FOAM method [44], and the TRIAD method [45, 46]. According to the analyses of Markley [31] and Duarte [47], the QUEST method is the fastest one and performs nearly equally to the ESOQ and the ESOQ2 methods.

Bar-Itzhack proposed a method for the computation of the quaternion corresponding to a given rotation matrix based on the above results, in particular on the q-method and the QUEST method [48].

In Bar-Itzhack method, the matrix in (86) is constructed using the elements of the rotation matrix and then the quaternion is computed either using an eigenvector computation routine or the QUEST method. If the given rotation matrix is not noisy (*i.e.*, it is perfectly orthogonal), then there is no need to compute the eigenvalues of $\mathbf{K}$ and thus the voting process, necessary in other algorithms, can be avoided.

Since two vector measurements are enough to determine the matrix that describes the rotation from the global reference frame to the body reference frame, we can take

$$\mathbf{r}_1 = (1,0,0)^T, \tag{90}$$

$$\mathbf{r}_2 = (0,1,0)^T. \tag{91}$$

Then, since $\mathbf{b}_i = \mathbf{R}\mathbf{r}_i$, we have that

$$\mathbf{b}_1 = (r_{11}, r_{21}, r_{31})^T, \tag{92}$$

$$\mathbf{b}_2 = (r_{12}, r_{22}, r_{32})^T. \tag{93}$$

As a consequence, the problem of computing the quaternion representation of $\mathbf{R}$ reduces to computing the eigenvector associated with the eigenvalue equal to 2 of the following matrix:

$$\mathbf{K}_1 = \begin{pmatrix} r_{11}+r_{22} & r_{32} & -r_{31} & r_{21}-r_{12} \\ r_{32} & r_{11}-r_{22} & r_{21}+r_{12} & r_{31} \\ -r_{31} & r_{21}+r_{12} & r_{22}-r_{11} & r_{32} \\ r_{21}-r_{12} & r_{31} & r_{32} & -r_{11}-r_{22} \end{pmatrix}. \tag{94}$$

We could use other couples of vectors, instead of $(1, 0, 0)$ and $(0, 1, 0)$, thus leading to a different expressions for the matrix in (94). With the taken choice, the elements of the third column of $\mathbf{R}$ are not included in the computations. This column is certainty redundant, as it can be obtained as the cross product of the other two, but from the numerical point of view it is better to take all the entries of $\mathbf{R}$ into account. Therefore, it seems reasonable to use the triad $\mathbf{r}_1 = (1, 0, 0)^T$, $\mathbf{r}_2 = (0, 1, 0)^T$, and $\mathbf{r}_3 = (0, 0, 1)^T$, in which case the problem reduces to computing the eigenvector associated with the eigenvalue equal to 3 of the following matrix:

$$\mathbf{K}_2 = \begin{pmatrix} r_{11}+r_{22}+r_{33} & r_{32}-r_{23} & r_{13}-r_{31} & r_{21}-r_{12} \\ r_{32}-r_{23} & r_{11}-r_{22}-r_{33} & r_{12}+r_{21} & r_{13}+r_{31} \\ r_{13}-r_{31} & r_{12}+r_{21} & r_{22}-r_{11}-r_{33} & r_{23}+r_{32} \\ r_{21}-r_{12} & r_{13}+r_{31} & r_{23}+r_{32} & r_{33}-r_{11}-r_{22} \end{pmatrix}. \tag{95}$$

In other words, the problem reduces to finding the base vector for the one-dimensional null space either of $\mathbf{K}_1-2\mathbf{I}$ or of $\mathbf{K}_2-3\mathbf{I}$. We analyze the behavior of both alternatives in the next section where are referenced to as Bar-Itzhack-1 and Bar-Itzhack-2, respectively.

5 A Comparison

This paper has supplementary downloadable multimedia material. This material includes MATLAB® functions implementing all methods described in this paper, and several scripts needed to reproduce the comparisons included below. No particular requirements, except for an installed copy of MATLAB® version R2013 or higher, are needed.

All comparisons have been implemented using single-precision floating-point numbers according to IEEE Standard 754. Using this representation, a number greater than approximately 3.4×10^{38} or less than approximately -3.4×10^{-38} cannot be represented.

The following comparison is based on an in-depth statistical analysis. To this end, we first need to generate random Euler parameters. Since this is equivalent to generate random points uniformly distributed in $\mathbb{S}^3$, we can use the algorithm described in [50]. For each generated set of Euler parameters, we can generate a rotation matrix using (7), and then recover the original Euler parameters using the reviewed methods. The committed error is evaluated as the the norm of the vector difference between the original and recovered parameters. In general, this is not a good way to compute the distance between two orientations. Nevertheless, since in our case the error is assumed to be very small, the length of the vector connecting both orientations in $\mathbb{S}^3$ is going to coincide with the value of the angle formed by them as seen from the center of $\mathbb{S}^3$. Now, observe that this angle can be taken as a distance between any two elements of the 3D rotation group $SO(3)$ [51].

The time and error performances of the described methods for 10^6 random orientations are compiled in Table 1 and Table 2, respectively. These results have been obtained for a MATLAB® implementation, included as downloadable multimedia material, running on an Intel® Core™i7 with 32 GB of RAM.

Table 1: Time performance in microseconds

Method	Average	Best-case
Trigonometric-1	16.5	7.5
Trigonometric-2	21.4	12.9
Chiaverini-Siciliano	13.2	5.3
Hughes	9.2	4.3
Shepperd	13.6	7.5
Sarabandi-Thomas	10.5	6.0
Cayley	6.7	3.9
Coope *et al.* Gaussian	702.1	403.2
Coope *et al.* SVD	35.6	19.7
Bar-Itzhack-1 Gaussian	1094.0	608.3
Bar-Itzhack-1 SVD	36.7	21.7
Bar-Itzhack-2 Gaussian	1064.3	623.9
Bar-Itzhack-2 SVD	28.5	19.3

Table 2: Error performance

Method	Orientations recovered without error	Worst-case $\times 10^{-6}$	Average $\times 10^{-6}$	Standard deviation $\times 10^{-6}$
Trigonometric-1	22703	2749.20	0.0732	2.750
Trigonometric-2	48641	65447.00	0.2252	72.775
Chiaverini-Siciliano	51202	203.70	NaN	NaN
Hughes	153211	1178700.01	NaN	NaN
Shepperd	244191	0.17	0.0304	0.0407
Sarabandi-Thomas	254643	0.12	0.0248	0.0346
Cayley	318168	0.18	0.0247	0.0361
Coope *et al.* Gaussian	27677	24911.03	0.0992	43.5392
Coope *et al.* SVD	16997	9.20	0.0929	0.0110
Bar-Itzhack-1 Gaussian	17909	59694.31	0.7151	79.7434
Bar-Itzhack-1 SVD	1448	0.88	0.0135	0.0150
Bar-Itzhack-2 Gaussian	26005	77301.52	0.6476	105.1176
Bar-Itzhack-2 SVD	8276	0.47	0.0100	0.0113

In Table 1, the first column gives the average time required for each method to compute a set of Euler parameters; and the second column, the time required in the best of the cases. The time for the worst case is meaningless on a multitasking computer and hence it is excluded. In Table 2, we have four columns. The first shows the number of cases, out of the 10^6 orientations, in which the original orientation is recovered without error. The other three correspond to the error committed in the worst-case, the average error, and the standard deviation of the error, respectively. Observe that, for Chiaverini-Siciliano's and Hughes' methods, some orientations could not be recovered because they lead to negative radicands in their formulations, and hence the NaNs —standing for "not a number"— appearing in the corresponding rows.

We can draw two important conclusions from these results:

- Although they are used in Robotics, the second trigonometric method, Chiaverini-Siciliano's and Hughes's method should be avoided.

- Cayley's method, besides being the simplest one, is superior in terms of accuracy and speed.

Among the numerical methods, the second version of Bar-Itzhack's method is the only one that deserves some attention as it can be used to obtain the quaternion corresponding to a non-perfectly orthogonal rotation matrix [48].

In this case, the quaternion corresponds to the nearest orthogonal matrix to the input non-orthogonal matrix, where closeness is expressed in the Frobenius norm [52].

6 Conclusion

We have reviewed 13 methods proposed in the literature to compute the quaternion corresponding to a given rotation matrix. These methods have been organized in three groups (trigonometric, algebraic and numerical) and they have been compared based on their time and error performance. From this review and comparison, we have concluded that Cayley's method is the simplest, and yet the best, in terms of time and error performance. Cayley's method was previously used to obtain the double quaternion representation of rotations in four dimensions. In this survey, we have simply particularized it to the three dimensional case.

So far, the most common method used in most applications —Shepperd's method— introduces four different mappings, being thus necessary to select the one that is numerically most stable in every case. We have shown that this strategy is not necessary as Cayley's method provides a single mapping that works well in all cases.

7 Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through projects DPI2014-57220-C2-2-P, DPI2017-88282-P, and MDM-2016-0656.

References

[1] K.W. Spring, "Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: A review," *Mechanism and Machine Theory*, Vol. 21, No. 5, pp. 365-373, 1986.

[2] M.D. Shuster, "A survey of attitude representations," *The Journal of Astronautical Sciences*, Vol. 41, no. 4, pp. 439-517, 1993.

[3] H. Goldstein, *Classical Mechanics*, Addison-Wesley, Cambridge, MA, 1951.

[4] R.P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press Cambridge, MA, 1982.

[5] A. Purwar and Q. J. Ge, "On the effect of dual weights in Computer Aided Design of rational motions," *ASME Journal of Mechanical Design*, Vol. 127, No. 5, pp. 967-972, 2005.

[6] D. Goldberg, "What every computer scientist should know about floating-point arithmetic," *ACM Computing Surveys,* Vol. 23, no. 1, pp. 5-48, 1991.

[7] D.M. Henderson, "Euler angles, quaternions, and transformation matrices - Working relationships," NASA, Mission Planning and Analysis Division, July 1977.

[8] J.L. Blanco, *A Tutorial on $SE(3)$ transformation parameterization and on-manifold optimization*, Technical report # 0112010, Univertity of Málaga, Spain, 2013.

[9] D. Eberly, "Euler angle formulas," *document available at:* https://www.geometrictools.com/Documentation/EulerAngles.pdf, 2014.

[10] C. Grubin, "Derivation of the quaternion scheme via the Euler axis and angle," *Journal of Spacecraft and Rockets*, Vol. 7, pp. 1261-1263, 1970.

[11] A.R. Klumpp, "Singularity-free extraction of a quaternion from a direction-cosine matrix," *Journal of Spacecraft and Rockets*, Vol. 13, pp. 754-755, 1976.

[12] R.A. Spurrier, "Comment on singularity-free extraction of a quaternion from a direction-cosine matrix ," *Journal of Spacecraft and Rockets*, Vol. 15, No. 4, pp. 255-255, 1978.

[13] A.R. Klumpp, "Reply to comment on singularity-free extraction of a quaternion from a direction-cosine matrix," *Journal of Spacecraft and Rockets*, Vol. 13, No. 4,pp. 256-256, 1978.

[14] P.C. Hughes, *Spacecraft Attitude Dynamics*, John Wiley & Sons, Inc, 1986.

[15] G.S. Chirikjian and A.B. Kyatkin, *Engineering applications of Noncommutative Harmonic Analysis: with Emphasis on Rotation and Motion Groups,* CRC Press, Boca Raton, FL, USA, 2001.

[16] S.M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006.

[17] C. Grubin, "Quaternion singularity revised," *Journal of Guidance and Control*, Vol. 2,No. 3, pp. 255-256, 1979.

[18] S. Chiaverini and B. Siciliano, "The unit quaternion: a useful tool for inverse kinematics of robot manipulators," *Systems Analysis Modelling Simulation*, Vol. 35, No. 1, pp. 45-60, 1999.

[19] S.W. Sheppard, "Quaternion from rotation matrix," *Journal of Guidance and Control*, Vol. 1, No. 3, pp. 223-224, 1978.

[20] M.D. Shuster and G.A. Natanson, "Quaternion computation from a geometric point of view," *The Journal of the Astronautical Sciences*, Vol. 41, No. 4, pp. 545-556, 1993.

[21] F. Thomas, "Approaching dual quaternions from matrix algebra," *IEEE Transactions on Robotics*, Vol. 30, No. 5, pp. 1037-1048, 2014.

[22] S. Sarabandi and F. Thomas, "Accurate computation of quaternion from rotation matrices," In: Lenarcic J., Parenti-Castelli V. (eds) Advances in Robot Kinematics 2018 (ARK 2018), Springer Proceedings in Advanced Robotics, Vol 8., Springer, Cham, 2018.

[23] S. Sarabandi, A. Perez-Gracia, and F. Thomas, "Singularity-free computation of quaternions from rotation matrices in E^4 and E^3," *7th Conference on Applied Geometric Algebras in Computer Science and Engineering* (AGACSE 2018), Campinas, Brazil, 23-27 July, 2018.

[24] I.D. Coope, A.B. Lincott, G.R. Dunlop, and M.I. Vuskovic, "Numerically stable methods for converting rotation matrices to Euler parameters," in *Advances in Robot Kinematics*, J. Lenarčič, M. M. Stanišić (eds.), Springer, 2000.

[25] N. Rosen, "Note on the general Lorentz transformation," *Journal of Mathematics and Physics*, Vol. 9, pp. 181-187, 1930.

[26] G. Wahba, "A least squares estimate of spacecraft attitude," *SIAM Review*, Vol 7, No. 3, p. 409, 1965.

[27] J.L. Farrell, J.C. Stuelpnagel, R.H. Wessner, J.R. Velman et al., "A least squares estimate of spacecraft attitude," Solution 65-1, *SIAM Review*, Vol. 8, No. 3, pp. 384-386, 1966.

[28] J. Keat, “Analysis of least-squares attitude determination routine DOAOP,” *Computer Science Corp*, CSC/TM-77/6034, Silver Spring, MD, USA, 1977.

[29] G.M Lerner, “Three-axis attitude determination,” *Spacecraft Attitude Determination and Control*, J.R. Wertz (editor), Springer, 1978.

[30] F.L Markley, “Attitude determination using vector observations and the singular value decomposition,” *Journal of the Astronautical Sciences*, Vol. 36, No. 3, pp. 245-258, 1988.

[31] F.L Markley and D. Mortari, “How to estimate attitude from vector observations,” *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, Alaska, USA, 1999.

[32] F.L. Markley, “Parameterizations of the attitude,” *Spacecraft Attitude Determination and Control*, J.R. Wertz (editor), Springer, 1978.

[33] M.D. Shuster, “A survey of attitude representations,” *Journal of the Astronautical Sciences*, Vol 41, No. 4, pp. 439-517, 1993.

[34] M.D. Shuster, “Approximate algorithms for fast optimal attitude computation,” *AIAA Guidance and Control Conference*, pp. 7-9, 1978.

[35] I. Bar-Itzhack, “REQUEST: A recursive QUEST algorithm for sequential attitude determination,” *Journal Of Guidance, Control, and Dynamics*, Vol. 19, No. 5, pp. 1034-1038, 1996.

[36] M.D. Shuster and S. D. Oh, “A simple Kalman filter and smoother for spacecraft attitude,” *Journal of the Astronautical Sciences*, Vol. 37, No. 1, pp. 89-106, 1989.

[37] M.D. Shuster, “Kalman filtering of spacecraft attitude and the QUEST model,” *The Journal of the Astronautical Sciences*, Vol. 38, No. 3, pp. 377-393, 1990. Erratum: Vol. 51, No. 3, p. 359, 2003.

[38] M.D. Shuster, “The quest for better attitudes,” *The Journal of the Astronautical Sciences*, Vol. 54, No. 3, pp. 657-683, 2006.

[39] M.D. Shuster and S.D. Oh, “An improvement to the implementation of the QUEST algorithm,” *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, pp. 301-305, 2014.

[40] D. Mortari, “ESOQ: A closed-form solution to the Wahba problem,” *Journal of the Astronautical Sciences*, Vol. 45, No. 2, pp. 195-204, 1997.

[41] D. Mortari, “N-dimensional cross product and its application to matrix eigenanalysis,” *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, pp. 509-515, 1997.

[42] D. Mortari, “ESOQ2 single-point algorithm for fast optimal attitude determination,” *AAS/AIAA Space Flight Mechanics Meeting,* Huntsville, AL, USA, 1997.

[43] D. Mortari, “Second estimator of the optimal quaternion,” *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 5, pp. 885-888, 2000.

[44] F.L. Markley, “Attitude determination using vector observations: a fast optimal matrix algorithm,” *Journal of the Astronautical Sciences*, Vol. 41, No. 2, pp. 261-280, 1993.

[45] M.D. Shuster, and S.D. Oh, “Three-axis attitude determination from vector observations,” *Journal of Guidance and Control*, Vol. 4, No. 1, pp. 70-77, 1981.

[46] M.D. Shuster,“The Optimization of TRIAD,” *The Journal of the Astronautical Sciences*, Vol. 55, No. 2, pp. 245-257, 2007.

[47] R. Duarte, L.S. Martins, and H.K. Kuga,“Performance comparison of attitude determination algorithms developed to run in a micro processor environment,” *20th International Congress of Mechanical Engineering*, Gramado, RS, Brazil, 2009.

[48] I. Bar-Itzhack, “New Method for extracting the quaternion from a rotation matrix,” *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 6, pp. 1085-1087, 2000.

[49] J. Angeles, *Fundamentals of Robotic Mechanical Systems. Theory, Methods, and Algorithms*, Springer, 1997.

[50] G. Marsaglia, “Choosing a point from the surface of a sphere,” *Annals of Mathematical Statistics*, Vol. 43, pp. 645-646, 1972.

[51] D.Q. Huynh, “Metrics for 3D rotations: comparison and analysis,” *Journal of Mathematical Imaging and Vision*, Vol. 35, No. 2, pp 155-164, 2009.

[52] I. Bar-Itzhack, “Iterative optimal orthogonalization of the strap-down matrix,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 11, No. 1, pp. 30-37, 1975.