Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Solution Methods to the Nearest Rotation Matrix Problem in $\mathbb{R}^3$: A Comparative Survey

S. Sarabandi | F. Thomas*

Institut de Robòtica i Informàtica Industrial (CSIC-UPC), ETSEIB, Diagonal 647, Pavelló E, planta 1, 08028 Barcelona, Spain

Correspondence
*F. Thomas, Email: f.thomas@csic.es

Summary

Nowadays, the Singular Value Decomposition (SVD) is the standard method of choice for solving the nearest rotation matrix problem. Nevertheless, many other methods are available in the literature for the 3D case. This paper reviews the most representative ones, proposes alternative ones, and presents a comparative analysis to elucidate their relative computational costs and error performances. This analysis leads to the conclusion that some algebraic closed-form methods are as robust as the SVD, but significantly faster and more accurate.

KEYWORDS:
Rotation matrices, quaternions, singular value decomposition.

1 | INTRODUCTION

One of the most common ways of representing 3D rotations consists in using proper orthogonal matrices. As a consequence, proper orthogonal matrices are also known as rotation matrices. A 3×3 matrix, say $\mathbf{R}$, is said to be orthogonal if $\mathbf{R}\mathbf{R}^T = \mathbf{I}$, with $\mathbf{I}$ the 3×3 identity matrix, and to be proper if, in addition, $\det(\mathbf{R}) = 1$. In other words, the three row and column vectors of $\mathbf{R}$ represent a right-handed orthonormal reference frame.

In some applications — arising in different areas such as aeronautics, robotics, computer vision, and computer graphics — the obtained rotation matrices do not exactly satisfy the two aforementioned conditions due to numerical or/and measurement errors. It is then desired to find the nearest rotation matrix to a given noisy rotation matrix, in a given metrics, that exactly satisfy them.

The available methods for computing the nearest rotation matrices are varied, but a coarse classification permits to divide them into geometric or algebraic. The organization of this paper reflects this classification. Sections 2 and Section 3 review the geometric methods and algebraic methods, respectively. Section 4 compares the performance of the described algebraic closed-form methods — i.e., those that do not rely on any iterative numerical procedure — with respect to the method based on the SVD. The analysis is limited to algebraic closed-form methods because they permit analyzing symbolically the influence of each variable on the result; and their computational cost (in terms of the number of arithmetic operations) is constant, which is important in real-time control loops. Geometric methods are excluded, despite they are also closed-form methods, because they are mostly based on heuristic arguments. Nevertheless, as it shown in Section 3, they are of practical interest as preconditioners. Finally, Section 5 summarizes the main conclusions.

[1]This study does not have any conflicts to disclose. It was partially supported by the Spanish Government through project PID2020-117509GB-I00/AEI/10.13039/50110001103. This paper has supplementary downloadable material available at http://www.iri.upc.edu/people/thomas/Soft/ComparativeSurvey.zip. This material consists of several Matlab® scripts that permit to reproduce the results presented in Section 4. No particular requirements, except for an installed copy of Matlab®, release 2019b or higher, are needed. Contact F. Thomas for questions about this material.

2 | GEOMETRIC METHODS

These methods are simple and intuitive. They consist in restoring the orthogonality of a noisy rotation matrix by using geometric constructions built on the reference frame defined by either its row or column vectors. Although these methods do not use what can be classified as a meaningful optimality criteria, their main virtue is that they provide a fast solution. Thus, the results they provide can be used as a preconditioner for a more sophisticated method. Indeed, if $\bar{\mathbf{R}}$ is the orthonormalized matrix resulting from the noisy rotation matrix $\mathbf{R}$ using a geometric method, then we can find the nearest rotation matrix to $\bar{\mathbf{R}}^T\mathbf{R} \approx \mathbf{I}$ using any other method which is known to behave well for matrices close to the identity. If $\hat{\mathbf{R}}$ is the orthonormalized matrix resulting from this refinement, and the used method is invariant with respect to the reference frame, then the sought orthonormalized matrix is clearly given by $\bar{\mathbf{R}}\hat{\mathbf{R}}$.

2.1 | Dot product method and QR factorization

This method can be seen as the particularization of the Gram-Schmidt orthonormalization process[1] to three dimensions. If we denote $\mathbf{R} = (\mathbf{n}\ \mathbf{o}\ \mathbf{a})$ the noisy rotation matrix (we adhere here to the standard robotics notation[2] p. 26), this method takes $\mathbf{n}$ as a reference vector, then it subtracts from $\mathbf{o}$ its projection onto $\mathbf{n}$, then subtracts from $\mathbf{a}$ its projections onto $\mathbf{n}$ and $\mathbf{o}$, and, finally, the three vectors are normalized. In algebraic terms, this reads

$$\hat{\mathbf{n}} = \frac{\mathbf{n}}{\|\mathbf{n}\|}, \tag{1}$$

$$\hat{\mathbf{o}} = \frac{\mathbf{o} - (\mathbf{o}\cdot\hat{\mathbf{n}})\,\hat{\mathbf{n}}}{\|\mathbf{o} - (\mathbf{o}\cdot\hat{\mathbf{n}})\,\hat{\mathbf{n}}\|}, \tag{2}$$

$$\hat{\mathbf{a}} = \frac{\mathbf{a} - (\mathbf{a}\cdot\hat{\mathbf{n}})\,\hat{\mathbf{n}} - (\mathbf{a}\cdot\hat{\mathbf{o}})\,\hat{\mathbf{o}}}{\|\mathbf{a} - (\mathbf{a}\cdot\hat{\mathbf{n}})\,\hat{\mathbf{n}} - (\mathbf{a}\cdot\hat{\mathbf{o}})\,\hat{\mathbf{o}}\|}. \tag{3}$$

Then, we have that the orthonormalized rotation matrix is given by $\hat{\mathbf{R}} = (\hat{\mathbf{n}}\ \hat{\mathbf{o}}\ \hat{\mathbf{a}})$. It is not difficult to prove that the original rotation matrix $\mathbf{R}$ and the resulting orthogonal matrix $\hat{\mathbf{R}}$ are related through the expression

$$\mathbf{R} = \hat{\mathbf{R}}\mathbf{U}, \tag{4}$$

where $\mathbf{U}$ is an upper triangular matrix with positive diagonal elements. Expression (4) is technically known as the QR factorization of $\mathbf{R}$[3]. There are other methods to compute this decomposition, besides the just described one based on the Gram-Schmidt orthonormalization process. They include the modified Gramm-Schmidt method, and the methods based on Householder transformations, or Givens rotations. All of them have a direct geometric interpretation. Each has several advantages and disadvantages[4,5]. The algorithm resulting from using Householder transformations is considered superior in terms of the orthogonality of the resulting matrix, especially if $\mathbf{R}$ is close to be singular.

Observe that in this method the sign of $\det(\hat{\mathbf{R}})$ equals that of $\det(\mathbf{R})$ because the diagonal elements of the upper triangular matrix $\mathbf{U}$ are positive. Thus, this method preserves the orientation of the reference frame defined by $\mathbf{R}$. When the result is desired to be proper orthogonal even when $\det(\mathbf{R}) < 0$, the following method is preferable.

2.2 | Cross product method

Alternatively to the previous method, using cross products, we have that

$$\hat{\mathbf{n}} = \frac{\mathbf{o}\times\mathbf{a}}{\|\mathbf{o}\times\mathbf{a}\|}, \tag{5}$$

$$\hat{\mathbf{o}} = \frac{\mathbf{a}\times\hat{\mathbf{n}}}{\|\mathbf{a}\times\hat{\mathbf{n}}\|}, \tag{6}$$

$$\hat{\mathbf{a}} = \hat{\mathbf{n}}\times\hat{\mathbf{o}}. \tag{7}$$

Both, the dot and the cross product methods, have the disadvantage that the result is biased by the order in which the column vectors are taken. Altering this order yields a different result. The cross product method is slightly less asymmetrical than the dot product method and it is why it is, in general, preferred. The following two methods were designed to give an equal treatment of the three vectors without preference to any one of them.

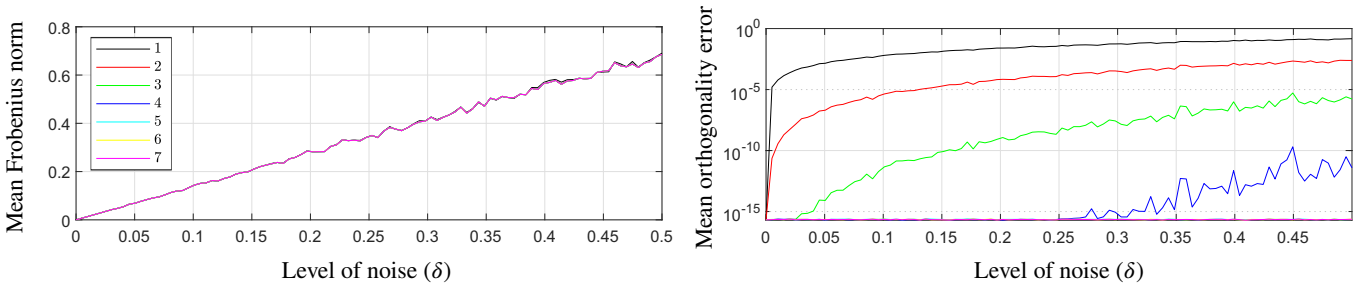

FIGURE 1 Frobenius norm of $\hat{\mathbf{R}}-\mathbf{R}$ and orthogonality error of $\hat{\mathbf{R}}$ for the iterative cross product method as a function of the level of noise, and for a different number of iterations. These plots result from averaging the estimations obtained using 10^5 random matrices for each value of δ. Observe how all the plots for the averaged Frobenius norm overlap thus indicating its independence on the number of iterations.

2.3 | Iterative cross product method

This method was proposed in[6]. It consists in obtaining each column from the other two and averaging the result with the original value of the column. That is,

$$\hat{\mathbf{n}} = \frac{(\mathbf{o}\times\mathbf{a})+\mathbf{n}}{2}, \tag{8}$$

$$\hat{\mathbf{o}} = \frac{(\mathbf{a}\times\mathbf{n})+\mathbf{o}}{2}, \tag{9}$$

$$\hat{\mathbf{a}} = \frac{(\mathbf{n}\times\mathbf{o})+\mathbf{a}}{2}. \tag{10}$$

Then,

$$\hat{\mathbf{R}} = \left(\frac{\hat{\mathbf{n}}}{\|\hat{\mathbf{n}}\|}\ \frac{\hat{\mathbf{o}}}{\|\hat{\mathbf{o}}\|}\ \frac{\hat{\mathbf{a}}}{\|\hat{\mathbf{a}}\|}\right) \tag{11}$$

is clearly a new rotation matrix closer to be orthogonal. Then, the idea is simple: this operation can be iteratively repeated until no relevant improvement is made.

In this method, an average of five iterations is enough to convergence with a variation in each element of the matrix lower than 10^{-8} (see[6] for details). To verify this result, a set of 10^5 random rotation matrices are generated[1] whose elements are contaminated with additive uncorrelated uniformly distributed noise in the interval $[-\delta,\delta]$. Then, we evaluate the mean Frobenius norm of the difference between these noisy matrices and the estimated rotation matrices, and the mean orthogonality error, defined as the Frobenius norm of $\hat{\mathbf{R}}\hat{\mathbf{R}}^T-\mathbf{I}$, using an increasing number of iterations of this method. The result of this experiment appears in Fig. 1 for values of δ ranging from 0 to 0.5. We can see how the use of five iterations is enough to obtain excellent results, thus concurring with the conclusions presented in[6], where it is also concluded that this simple geometric iterative method outperforms the two quadratically convergent methods explained in Section 3.

2.4 | Equal mean direction method

This method was originally proposed in[8] and rediscovered in[9]. It consists in first computing the mean direction of the column vectors, that is,

$$\mathbf{c} = \frac{1}{3}\left(\frac{\mathbf{n}}{\|\mathbf{n}\|}+\frac{\mathbf{o}}{\|\mathbf{o}\|}+\frac{\mathbf{a}}{\|\mathbf{a}\|}\right). \tag{12}$$

Then, the goal is to find the proper rotation matrix that has the same mean direction but whose columns are as "close" as possible to the columns of $\mathbf{R}$.

Fist of all, let us define the plane $\Pi:\{\mathbf{x}\mid(\mathbf{x}-\mathbf{p})\cdot\mathbf{p}=0\}$, where

$$\mathbf{p} = \frac{\mathbf{c}}{\|\mathbf{c}\|}\frac{1}{\sqrt{3}}. \tag{13}$$

[1]To this end, random points in $\mathbb{S}^3$ are generated using the algorithm described in[7], which are then considered as unit quaternions that are translated into rotation matrices.

Then,

$$\mathbf{x}_p = \frac{\|\mathbf{c}\|}{\mathbf{n}\cdot\mathbf{c}}\frac{1}{\sqrt{3}}\mathbf{n},$$
$$\mathbf{y}_p = \frac{\|\mathbf{c}\|}{\mathbf{o}\cdot\mathbf{c}}\frac{1}{\sqrt{3}}\mathbf{o},$$
$$\mathbf{z}_p = \frac{\|\mathbf{c}\|}{\mathbf{a}\cdot\mathbf{c}}\frac{1}{\sqrt{3}}\mathbf{a},$$

are the position vectors of the intersections of the lines defined by $\mathbf{n}$, $\mathbf{o}$, and $\mathbf{a}$ with Π. Now, we can define the angles

$$\phi_{xy} = \arccos\left(\frac{(\mathbf{x}_p-\mathbf{p})\cdot(\mathbf{y}_p-\mathbf{p})}{\left\|\mathbf{x}_p-\mathbf{p}\right\|\left\|\mathbf{y}_p-\mathbf{p}\right\|}\right),$$
$$\phi_{yz} = \arccos\left(\frac{(\mathbf{y}_p-\mathbf{p})\cdot(\mathbf{z}_p-\mathbf{p})}{\left\|\mathbf{y}_p-\mathbf{p}\right\|\left\|\mathbf{z}_p-\mathbf{p}\right\|}\right),$$

and

$$\theta_x = \frac{2\phi_{xy}+\phi_{yz}-2\pi}{3}. \tag{14}$$

This allows us to build the auxiliary rotation matrix

$$\mathbf{R}' = \left(\frac{\mathbf{x}_p-\mathbf{p}}{\left\|\mathbf{x}_p-\mathbf{p}\right\|} \quad \frac{\mathbf{c}}{\|\mathbf{c}\|} \quad \mathbf{a}'\times\mathbf{n}'\right). \tag{15}$$

Finally, we obtain

$$\hat{\mathbf{R}} = (\hat{\mathbf{n}}\ \hat{\mathbf{o}}\ \hat{\mathbf{a}}), \tag{16}$$

where

$$\hat{\mathbf{n}} = \mathbf{p}+\sqrt{\frac{2}{3}}\mathbf{R}'\left(\cos\theta_x \ \ \sin\theta_x \ \ 0\right)^T,$$
$$\hat{\mathbf{o}} = \mathbf{p}+\sqrt{\frac{2}{3}}\mathbf{R}'\left(\cos\left(\theta_x+\frac{2\pi}{3}\right) \ \ \sin\left(\theta_x+\frac{2\pi}{3}\right) \ \ 0\right)^T,$$
$$\hat{\mathbf{a}} = \mathbf{p}+\sqrt{\frac{2}{3}}\mathbf{R}'\left(\cos\left(\theta_x+\frac{4\pi}{3}\right) \ \ \sin\left(\theta_x+\frac{4\pi}{3}\right) \ \ 0\right)^T.$$

3 | ALGEBRAIC METHODS

The algebraic methods are based on the minimization of a measure of closeness between the noisy rotation matrix $\mathbf{R}$ and the estimated proper orthogonal matrix $\hat{\mathbf{R}}$. The usual practice is to adopt the Frobenius norm of their difference as the measure of closeness which is denoted as $\left\|\mathbf{R}-\hat{\mathbf{R}}\right\|_F$ (see [10] for a geometric justification of this choice). Thus, the problem is stated as that of finding $\hat{\mathbf{R}}$ that minimizes

$$\left\|\hat{\mathbf{R}}-\mathbf{R}\right\|_F^2 = \mathrm{Tr}\left((\hat{\mathbf{R}}-\mathbf{R})(\hat{\mathbf{R}}-\mathbf{R})^T\right) = 3+\|\mathbf{R}\|_F^2-2\,\mathrm{Tr}\left(\hat{\mathbf{R}}\mathbf{R}^T\right), \tag{17}$$

subject to $\hat{\mathbf{R}}^T\hat{\mathbf{R}} = \mathbf{I}$. Therefore, the problem boils down to maximizing

$$\mathrm{Tr}\left(\hat{\mathbf{R}}\mathbf{R}^T\right), \tag{18}$$

subject to $\hat{\mathbf{R}}^T\hat{\mathbf{R}} = \mathbf{I}$.

Using Lagrange multipliers, it can be proved that the optimal solution to this constrained optimization problem, in the case that $\mathbf{R}$ is not singular, is given by [11, 12]:

$$\hat{\mathbf{R}} = \mathbf{R}\left(\mathbf{R}^T\mathbf{R}\right)^{-\frac{1}{2}} = \mathbf{R}\left(\mathbf{I}+\mathbf{E}\right)^{-\frac{1}{2}}, \tag{19}$$

where

$$\mathbf{E} = \mathbf{R}^T\mathbf{R}-\mathbf{I} \tag{20}$$

can be seen as an error matrix.

It is easy to verify that $\hat{\mathbf{R}}$ thus obtained is orthonormal, i.e., $\hat{\mathbf{R}}^T\hat{\mathbf{R}} = \mathbf{I}$. However, there is no guarantee that $\det(\hat{\mathbf{R}}) = +1$. To represent a proper rotation, the orthonormal matrix $\hat{\mathbf{R}}$ has to satisfy this condition as well. Otherwise it represents a reflection, not a rotation. There is no easy way to enforce this condition, and with poor measurements, it might happen that $\det(\hat{\mathbf{R}}) = -1$.

Alternatively, (19) can also be expressed as:

$$\hat{\mathbf{R}} = \left(\mathbf{R}\mathbf{R}^T\right)^{\frac{1}{2}}\left(\mathbf{R}^T\right)^{-1} = \left(\mathbf{I} + \bar{\mathbf{E}}\right)^{\frac{1}{2}}\left(\mathbf{R}^T\right)^{-1}, \tag{21}$$

where

$$\bar{\mathbf{E}} = \mathbf{R}\mathbf{R}^T - \mathbf{I} \tag{22}$$

can also be seen as an error matrix.

While (19) is called the primal solution, (21) is referred to as the dual solution. A proof of equivalence between them can be found in [13] or [14].

3.1 | Series expansion method

This technique was first proposed in [15]. To obtain an approximate value of the primal solution in (19), we can compute some terms of its Maclaurin series of $(\mathbf{I} + \mathbf{E})^{-\frac{1}{2}}$ as follows:

$$\hat{\mathbf{R}} = \mathbf{R}(\mathbf{I} + \mathbf{E})^{-\frac{1}{2}} = \mathbf{R}\left(\mathbf{I} - \frac{1}{2}\mathbf{E} + \frac{3}{8}\mathbf{E}^2 - \frac{5}{16}\mathbf{E}^3 + \frac{35}{128}\mathbf{E}^4 + \dots\right). \tag{23}$$

Likewise, to obtain an approximate value of the dual solution in (21), we can also compute some terms of its Maclaurin series expansion of $\left(\mathbf{I} + \bar{\mathbf{E}}\right)^{\frac{1}{2}}$ as follows:

$$\hat{\mathbf{R}} = (\mathbf{I} + \bar{\mathbf{E}})^{\frac{1}{2}}(\mathbf{R}^T)^{-1} = \left(\mathbf{I} + \frac{1}{2}\bar{\mathbf{E}} - \frac{1}{8}\bar{\mathbf{E}}^2 + \frac{1}{16}\bar{\mathbf{E}}^3 - \frac{5}{128}\bar{\mathbf{E}}^4 + \dots\right)(\mathbf{R}^T)^{-1}. \tag{24}$$

Obviously, in both cases, only the first few terms are worth using. In many applications, only one term of the series expansion suffices to get the desired accuracy [15]. However, for highly noisy systems, this method can not guarantee to converge to the solution.

By taking up to the linear, quadratic, and cubic terms in (23) and (24), we obtain different levels of approximation. As explained in [16], the resulting formulas can also be applied iteratively in the hope that the result converges to the solution. They are compiled, after simplification, in Table 1 and Table 2, respectively. The iterative application of these formulas was rediscovered in [17, 18] as the result of reformulating the problem as a dynamical system. However, solving the nearest rotation matrix problem using these iterative methods is, in general, computationally costly.

To see the influence of the number of iterations in the quality of the result, we can proceed as in Section 2.3. Figs. 2, 3, and 4 show the results for the quadratically, cubically and quadrically convergent formulas, respectively. Two conclusions can be drawn from these plots:

- The dual formulas perform much better than their primal counterparts. However, this does not come without a cost: the dual formulas require the computation of the inverse of $\mathbf{R}^T$.
- The quadrically convergent formulas do not provide an improvement, with respect to the cubically convergent ones, deserving the effort of their computation.

There are other iterative methods, derived using other algebraic arguments. One example is the one described in [12] resulting from a gradient projection technique. Unfortunately, its rate of convergence was shown to be linear [19] and hence its little practical interest. Another example is the two-step iterative algorithm proposed in [6] where it was also shown to be inferior to the quadratically convergent primal and dual methods. A particularization of this latter method reappeared in [26] using the matrix sign function to compute the square root of positive definite matrices. Next, we analyze three other methods that deserve some attention due to their simplicity: the matrix sign function method, the Padé approximant method, and the continued fraction method. They can also be implemented in iterative form.

TABLE 1 Series expansion of the primal solution and derived iterative methods.

Closed-form solution	$\hat{\mathbf{R}} = \mathbf{R}(\mathbf{R}^T\mathbf{R})^{-\frac{1}{2}}$
Error matrix	$\mathbf{E} = \mathbf{R}^T\mathbf{R} - \mathbf{I}$
Series solution	$\hat{\mathbf{R}} = \mathbf{R}\left(\mathbf{I} - \frac{1}{2}\mathbf{E} + \frac{3}{8}\mathbf{E}^2 - \frac{5}{16}\mathbf{E}^3 + \dots\right)$
Quadratically convergent iterative solution	$\mathbf{R}_0 = \mathbf{R}$ $\mathbf{R}_{n+1} = \frac{1}{2}\mathbf{R}_n(3\mathbf{I} - \mathbf{R}_n^T\mathbf{R}_n)$
Convergence conditions	$\det(\mathbf{R}) \neq 0, 0 \leq \max\{\lambda_i\} \leq \sqrt{3}$
References	6, 13, 16–20
Cubically convergent iterative solution	$\mathbf{R}_0 = \mathbf{R}$ $\mathbf{R}_{n+1} = \frac{1}{8}\mathbf{R}_n\left(15\mathbf{I} - 10\mathbf{R}_n^T\mathbf{R}_n + 3(\mathbf{R}_n^T\mathbf{R}_n)^2\right)$
Convergence conditions	$\det(\mathbf{R}) \neq 0, 0 \leq \max\{\lambda_i\} \leq 1.5275$
References	17, 18
Quartically convergent iterative solution	$\mathbf{R}_0 = \mathbf{R}$ $\mathbf{R}_{n+1} = \frac{1}{16}\mathbf{R}_n\left(35\mathbf{I} - 35\mathbf{R}_n^T\mathbf{R}_n + 21(\mathbf{R}_n^T\mathbf{R}_n)^2 - 5(\mathbf{R}_n^T\mathbf{R}_n)^3\right)$
Convergence conditions	$\det(\mathbf{R}) \neq 0, 0 \leq \max\{\lambda_i\} \leq \sqrt{3}$
References	17, 18, 21

TABLE 2 Series expansion of the dual solution and derived iterative methods.

Closed-form solution	$\hat{\mathbf{R}} = (\mathbf{R}\mathbf{R}^T)^{\frac{1}{2}}(\mathbf{R}^T)^{-1}$
Error matrix	$\bar{\mathbf{E}} = \mathbf{R}\mathbf{R}^T - \mathbf{I}$
Series solution	$\hat{\mathbf{R}} = \left(\mathbf{I} + \frac{1}{2}\bar{\mathbf{E}} - \frac{1}{8}\bar{\mathbf{E}}^2 + \frac{1}{16}\bar{\mathbf{E}}^3 - \dots\right)\left(\mathbf{R}^T\right)^{-1}$
Quadratically convergent iterative solution	$\mathbf{R}_0 = \mathbf{R}$ $\mathbf{R}_{n+1} = \frac{1}{2}(\mathbf{R}_n^T)^{-1} + \frac{1}{2}\mathbf{R}_n$
Convergence conditions	$\det(\mathbf{R}) \neq 0$
References	6, 13, 22–24
Cubically convergent iterative solution	$\mathbf{R}_0 = \mathbf{R}$ $\mathbf{R}_{n+1} = \mathbf{R}_n(3\mathbf{I} + \mathbf{R}_n^T\mathbf{R}_n)(\mathbf{I} + 3\mathbf{R}_n^T\mathbf{R}_n)^{-1}$
Convergence conditions	$\det(\mathbf{R}) \neq 0$
References	17, 18, 24
Quartically convergent iterative solution	$\mathbf{R}_0 = \mathbf{R}$ $\mathbf{R}_{n+1} = \mathbf{R}_n(\mathbf{I} + 3\mathbf{R}_n^T\mathbf{R}_n)\left(3\mathbf{R}_n^T\mathbf{R}_n + (\mathbf{R}_n^T\mathbf{R}_n)^2\right)^{-1}$
Convergence conditions	$\det(\mathbf{R}) \neq 0$
References	17, 18, 25

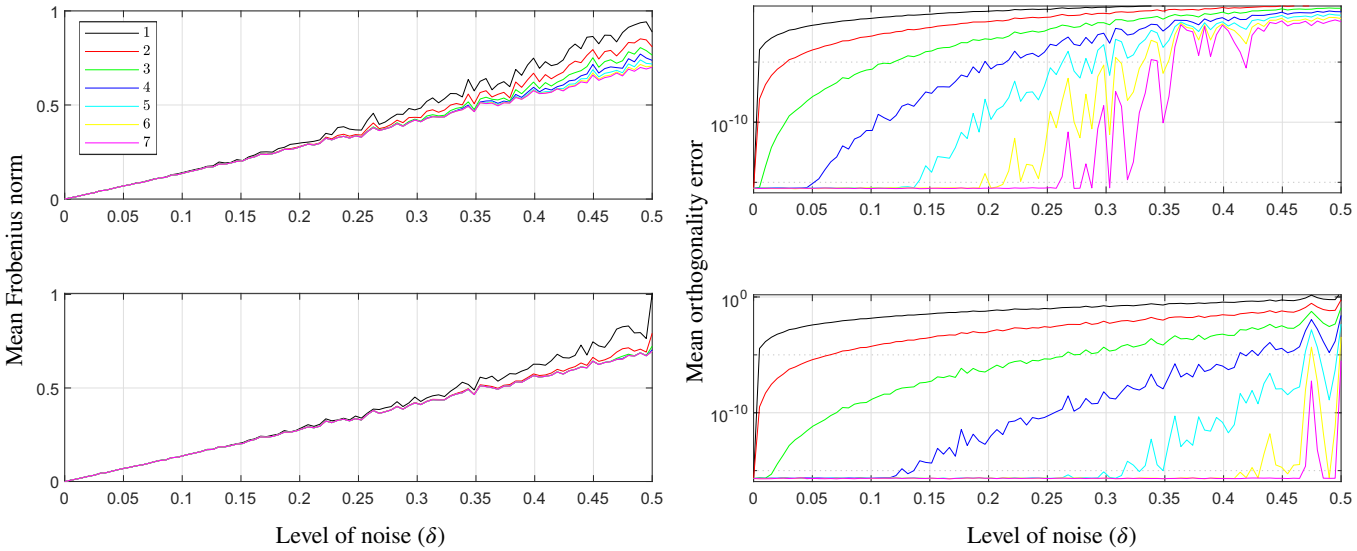

FIGURE 2 Frobenius norm of $\hat{\mathbf{R}}-\mathbf{R}$ and orthogonality error of $\hat{\mathbf{R}}$ for the primal quadratically convergent formula (top row) and its dual (bottom row) as a function of the level of noise, and for a different number of iterations. 10^6 random matrices are generated for each value of δ.

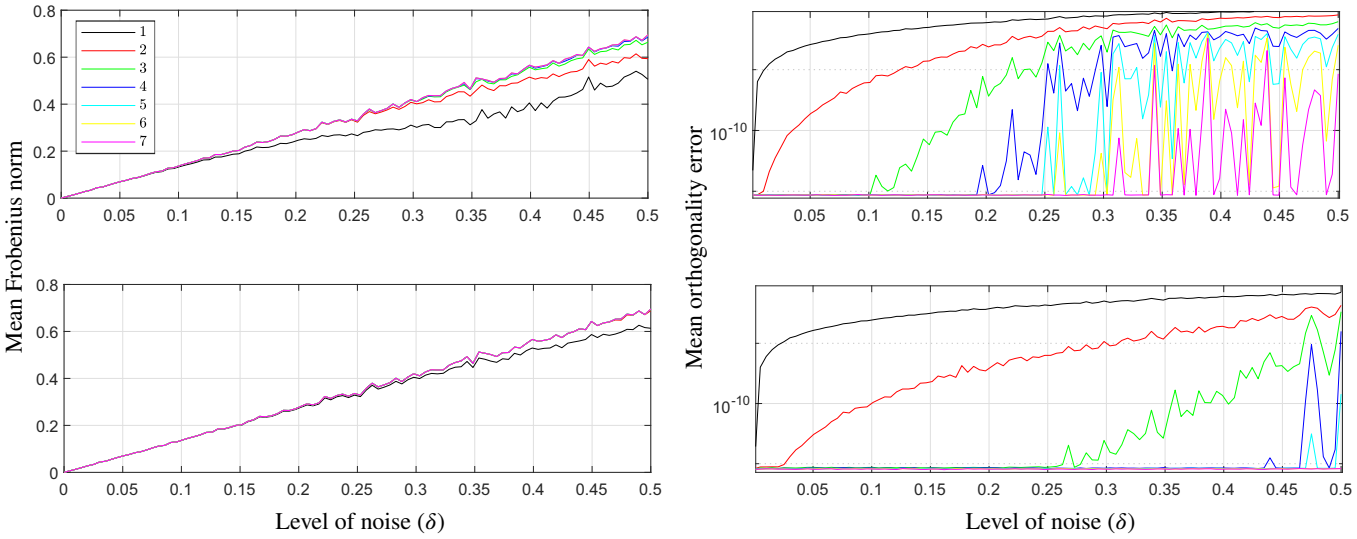

FIGURE 3 Frobenius norm of $\hat{\mathbf{R}}-\mathbf{R}$ and orthogonality error of $\hat{\mathbf{R}}$ for the primal cubically convergent formula (top row) and its dual (bottom row) as a function of the level of noise, and for a different number of iterations. 10^6 random matrices are generated for each value of δ.

3.2 | Matrix sign function method

Given the positive definite matrix $\mathbf{A}$, its sign is defined as:

$$\text{sign}(\mathbf{A}) = \mathbf{A}\left(\mathbf{A}^2\right)^{-\frac{1}{2}}. \tag{25}$$

It is possible to derive iterative methods for computing the square root of a matrix by relying on this function. In our case, the one described in[20], based on the matrix sign function algorithm described in[27], permits computing the square root of $\mathbf{R}\mathbf{R}^T$ using the iterative application of the following formula

$$\mathbf{S}_{n+1} = \frac{1}{2}\left(\mathbf{S}_n + \mathbf{S}_0\mathbf{S}_n^{-1}\right), \tag{26}$$

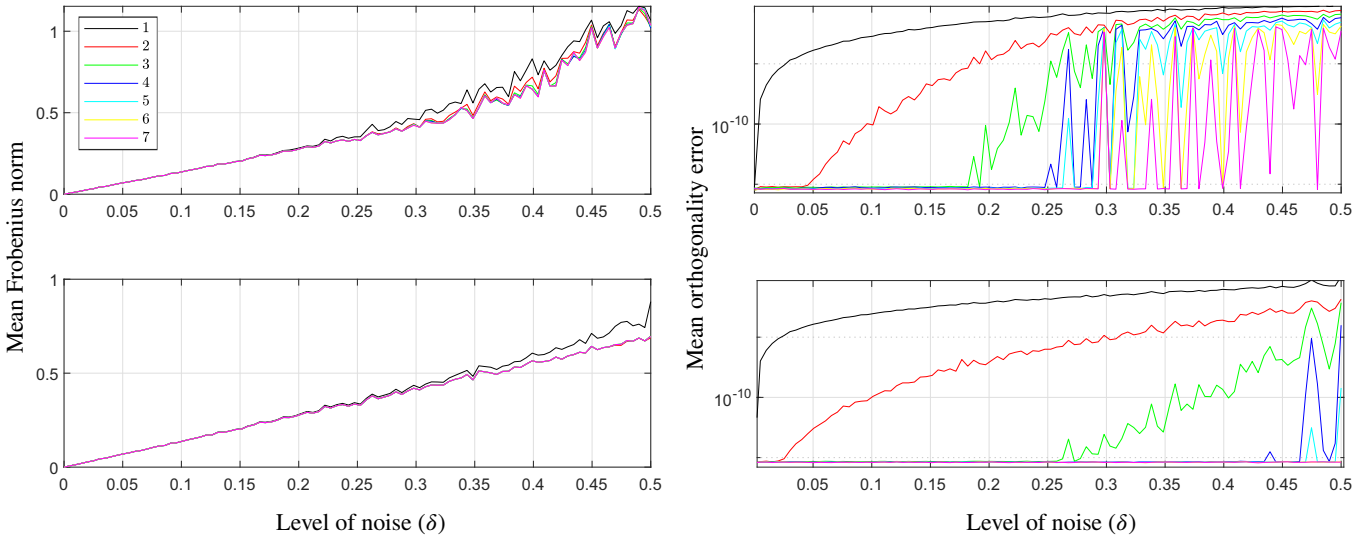

FIGURE 4 Frobenius norm of $\hat{\mathbf{R}}-\mathbf{R}$ and orthogonality error of $\hat{\mathbf{R}}$ for the primal quadrically convergent formula (top row) and its dual (bottom row) as a function of the level of noise, and for a different number of iterations. 10^6 random matrices are generated for each value of δ.

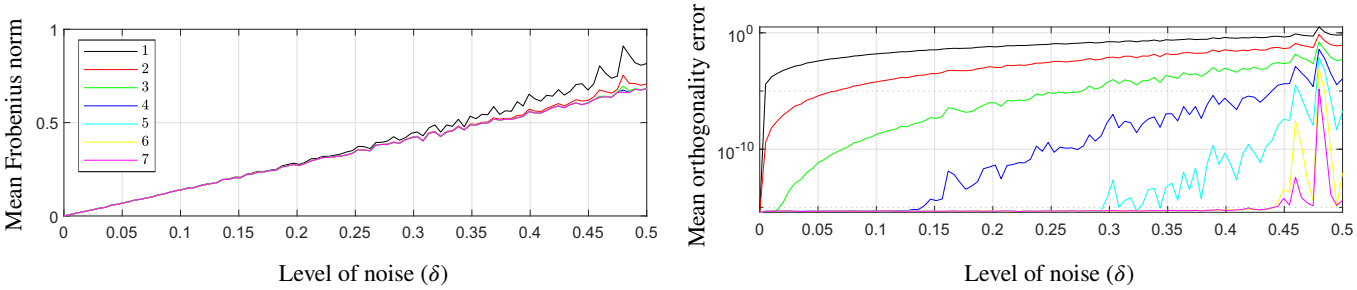

FIGURE 5 Frobenius norm of $\hat{\mathbf{R}}-\mathbf{R}$ and orthogonality error of $\hat{\mathbf{R}}$ for the sign matrix method as a function of the level of noise and for a different number of iterations. 10^6 random matrices are generated for each value of δ.

with $\mathbf{S}_0 = \mathbf{R}\mathbf{R}^T$. This iterative formula converges to $\mathbf{S} = (\mathbf{R}\mathbf{R}^T)^{\frac{1}{2}}$. As a consequence,

$$\hat{\mathbf{R}} = \mathbf{S}\left(\mathbf{R}^T\right)^{-1}. \tag{27}$$

In[20], this method is compared with the quadratically convergent primal and dual methods to conclude that it behaves better in terms of convergence. To verify this result, we can perform a similar analysis to that in Section 2.3. The result appears in Fig. 5. If we compare it with that in Fig. 2, it is easy to conclude that, while this method clearly performs better than the quadratically convergent primal method, it performs similarly to its dual counterpart. As a consequence, the improvement reported in[20] is not remarkable.

3.3 | Padé approximant method

Using Padé approximants, we have that[28, 29]

$$\left(\mathbf{R}^T\mathbf{R}\right)^{\frac{1}{2}} = (\mathbf{I}+\mathbf{E})^{\frac{1}{2}} \approx \mathbf{I} + \sum_{j=1}^{m} a_j^{(m)} \left(\mathbf{I} + b_j^{(m)}\mathbf{E}\right)^{-1} \mathbf{E}, \tag{28}$$

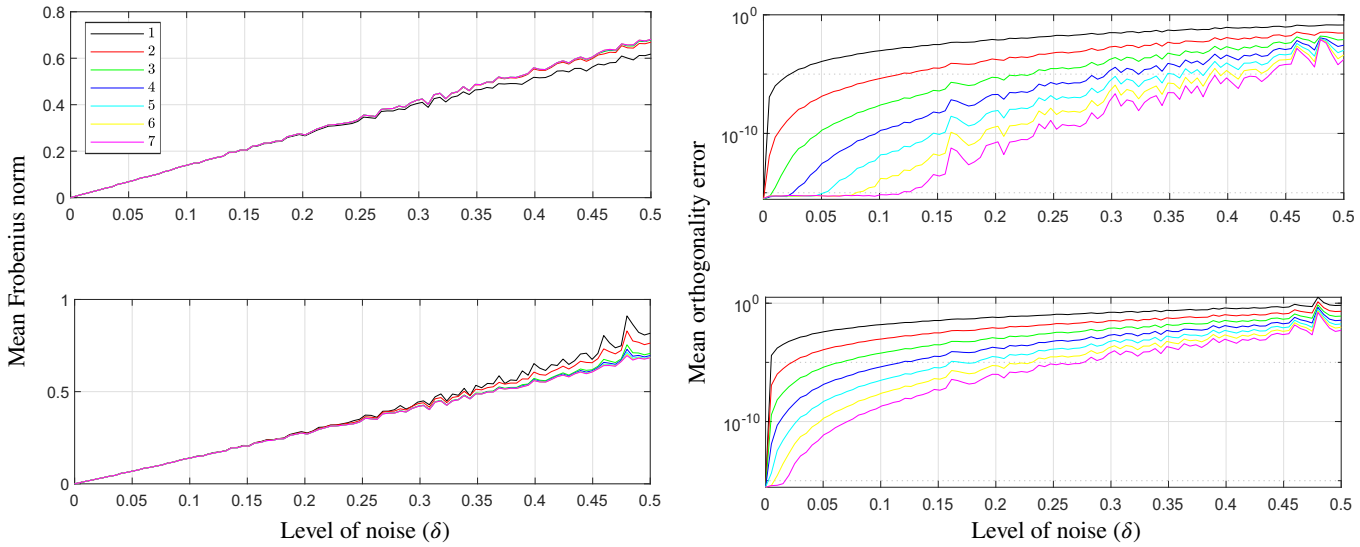

FIGURE 6 Frobenius norm of $\hat{\mathbf{R}}-\mathbf{R}$ and orthogonality error of $\hat{\mathbf{R}}$ for the Padé approximant method (top row) and the continued fraction method (bottom row) as a function of the level of noise and for different approximation orders. 10^6 random matrices are generated for each value of δ.

where

$$a_j^{(m)} = \frac{2}{2m+1}\sin^2\frac{j\pi}{2m+1}, \tag{29}$$

$$b_j^{(m)} = \cos^2\frac{j\pi}{2m+1}. \tag{30}$$

Observe that in this method the values of the coefficients of the expansion vary with the number of taken terms, that is, with the order of the approximation.

3.4 | Continued fraction method

Since $\mathbf{S}^2 = \mathbf{R}^T\mathbf{R}$, we have that

$$(\mathbf{S}-\mathbf{I})(\mathbf{S}+\mathbf{I}) = \mathbf{S}^2 - \mathbf{I} = \mathbf{E}. \tag{31}$$

Then,

$$\mathbf{S} = \mathbf{I} + \mathbf{E}(\mathbf{I}+\mathbf{S})^{-1}. \tag{32}$$

Therefore, using rational notation and recursively substituting the value of $\mathbf{S}$ in the right-hand side of (32) by itself, we have that

$$(\mathbf{R}^T\mathbf{R})^{\frac{1}{2}} = \mathbf{S} = \mathbf{I} + \cfrac{\mathbf{E}}{\mathbf{I} + \cfrac{\mathbf{E}}{\mathbf{I} + \cfrac{\mathbf{E}}{\mathbf{I}+\dots}}} \tag{33}$$

which leads to the following elegant formula

$$\hat{\mathbf{R}} = \cfrac{\mathbf{R}}{\mathbf{I} + \cfrac{\mathbf{E}}{\mathbf{I} + \cfrac{\mathbf{E}}{\mathbf{I} + \cfrac{\mathbf{E}}{\mathbf{I}+\dots}}}} \tag{34}$$

By truncating this continued fraction, we get different approximation orders.

To observe the influence of the approximation order in the Padé approximant method and this method on the result, we can perform a similar analysis to that in Section 2.3 where the number of iterations is substituted with the approximation order.

The results appear in Fig. 6. It can be concluded that the Padé approximant method performs better than the continued fraction method for the same approximation order.

3.5 | Logarithm method

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[30] pp. 118-123 for a proof of this theorem in terms of rotation matrices). We can associate the following 3×3 skew-symmetric matrix with this unit vector

$$\mathbf{N} = \begin{pmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{pmatrix}. \tag{35}$$

It is easy to verify that, for $\mathbf{v} \in \mathbb{R}^3$,

$$\mathbf{n} \times \mathbf{v} = \mathbf{N}\mathbf{v}, \tag{36}$$

where $\times$ stands for the vector cross product.

Now, consider the following problem: given the unit vector $\mathbf{n} \in \mathbb{R}^3$ and an angle $\theta \in \mathbb{R}$, find the rotation matrix $\mathbf{R}$ that rotates any vector through the angle θ about an axis given by $\mathbf{n}$. The matrix exponential gives the elegant solution[31] p. 29

$$\mathbf{R} = e^{\theta\mathbf{N}}, \tag{37}$$

which can be rewritten, using the series expansion of the exponential, as

$$\begin{aligned} \mathbf{R} &= \sum_{k=0}^{\infty} \frac{(\theta\mathbf{N})^k}{k!} \\ &= \mathbf{I} + \theta\mathbf{N} + \frac{1}{2!}(\theta\mathbf{N})^2 + \frac{1}{3!}(\theta\mathbf{N})^3 + \dots \\ &= \mathbf{I} + (\sin\theta)\mathbf{N} + (1 - \cos\theta)\mathbf{N}^2, \end{aligned} \tag{38}$$

which is commonly known as Rodrigues' formula.

Now, decomposing equation (38) into its symmetric and skew-symmetric components, we have that

$$\frac{1}{2}(\mathbf{R}+\mathbf{R}^T) = \mathbf{I}+(1-\cos\theta)\mathbf{N}^2, \tag{39}$$

$$\frac{1}{2}(\mathbf{R}-\mathbf{R}^T) = \sin\theta\,\mathbf{N}. \tag{40}$$

Then, from (39) and (40), it can be concluded that

$$\mathrm{Tr}(\mathbf{R}) = 1 + 2\cos\theta \tag{41}$$

and

$$\mathbf{N} = \frac{1}{2\sin\theta}(\mathbf{R} - \mathbf{R}^T), \tag{42}$$

respectively. This allows us to conclude, using (37), that

$$\theta\mathbf{N} = \log(\mathbf{R}) = \frac{\theta}{2\sin\theta}(\mathbf{R} - \mathbf{R}^T). \tag{43}$$

The logarithm method can be summarized as follows. First, we compute $\mathbf{M} = \log(\mathbf{R})$. If $\mathbf{R}$ is a noisy rotation matrix, $\mathbf{M}$ is not skew-symmetric. Then, it can be decomposed into the sum of a skew-symmetric matrix and a symmetric residual matrix as follows

$$\mathbf{M} = \tfrac{1}{2}\left(\mathbf{M} + \mathbf{M}^T\right) + \tfrac{1}{2}\left(\mathbf{M} - \mathbf{M}^T\right). \tag{44}$$

Now, by simply canceling the symmetric residual, we have that

$$\widehat{\mathbf{M}} = \tfrac{1}{2}\left(\mathbf{M} - \mathbf{M}^T\right). \tag{45}$$

As a consequence,

$$\hat{\theta} = \frac{1}{\sqrt{2}}\left\|\widehat{\mathbf{M}}\right\|_F \tag{46}$$

and

$$\widehat{\mathbf{N}} = \frac{1}{\hat{\theta}}\widehat{\mathbf{M}}. \tag{47}$$

Finally, we can recover from $\widehat{\mathbf{N}}$ the sought proper orthogonal rotation matrix, $\hat{\mathbf{R}}$, using Rodrigues' formula. Thus, the logarithm method essentially reduces to the computation of $\mathbf{M} = \log(\mathbf{R})$ and hence its name. The problem is that the computation of the logarithm of a matrix is not a trivial operation as not all matrices have a logarithm, and those matrices that do have a logarithm may have more than one. Function *logm* in Matlab® implements the algorithm presented in [32]. Since the exponential function is not one-to-one for complex numbers, numbers can have multiple complex logarithms, and this is the ultimate reason why some matrices may have more than one logarithm. If $\mathbf{R}$ is singular or has an eigenvalue on the negative real axis, its logarithm is undefined [33]. Moreover, even if it is defined, it is not necessarily a real matrix. A real matrix has a real logarithm if and only if it is invertible and each Jordan block belonging to a negative eigenvalue occurs an even number of times [34].

Observe that, according to (40), even the logarithm of non-noisy rotation matrices may be numerically imprecise for $\theta \to n\pi$, $n \in \mathbb{Z}$. Nevertheless, if $\|\mathbf{R} - \mathbf{I}\|_F^2 < 1$, the logarithm of $\mathbf{R}$ can be computed by means of the following power series

$$\log(\mathbf{R}) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n}(\mathbf{R} - \mathbf{I}) = (\mathbf{R} - \mathbf{I}) - \frac{1}{2}(\mathbf{R} - \mathbf{I})^2 + \frac{1}{3}(\mathbf{R} - \mathbf{I})^3 - \dots$$

Therefore, as explained at the beginning of Section 2, we can change the reference frame so that $\|\mathbf{R} - \mathbf{I}\|_F^2 < 1$ and use few terms of the above power series to obtain a good approximation of $\log(\mathbf{R})$. For example, the computation of the logarithm of

$$\mathbf{R} = \begin{pmatrix} 0.8510 & 0.4687 & 0.2397 \\ 0.4684 & -0.8823 & 0.0602 \\ 0.2402 & 0.0598 & -0.9681 \end{pmatrix} \tag{48}$$

fails because its eigenvalues are 1.0006, −1.0011, −0.9990, and the matrix logarithm is not defined for matrices with nonpositive real eigenvalues. A way around this inconvenience is to introduce a preconditioner, that is, an estimation of the nearest rotation matrix using a fast geometric method that is used to change the reference frame.

The result of applying the dot product method to matrix (48) is

$$\bar{\mathbf{R}} = \begin{pmatrix} 0.8504 & 0.4683 & 0.2395 \\ 0.4684 & -0.8814 & 0.0602 \\ 0.2393 & 0.0609 & -0.9690 \end{pmatrix}, \tag{49}$$

and the nearest rotation matrix to $\bar{\mathbf{R}}^{-1}\mathbf{R}$, using the logarithm method, is

$$\bar{\bar{\mathbf{R}}} = \begin{pmatrix} 0.9999 & -0.0004 & 0.0004 \\ 0.0004 & 0.9999 & -0.0004 \\ -0.0004 & 0.0004 & 0.9999 \end{pmatrix}. \tag{50}$$

Then, the resulting nearest rotation matrix to $\mathbf{R}$ is

$$\hat{\mathbf{R}} = \bar{\mathbf{R}}\bar{\bar{\mathbf{R}}} = \begin{pmatrix} 0.8505 & 0.4681 & 0.2397 \\ 0.4680 & -0.8815 & 0.0609 \\ 0.2398 & 0.0604 & -0.9689 \end{pmatrix}. \tag{51}$$

3.6 | Matrix factorization methods

We have shown how the dot product method can be formalized as the QR factorization of the noisy rotation matrix, but this is not the only matrix factorization that can be useful to solve the problem.

3.6.1 | Polar decomposition method

The first use of the concept of polar decomposition in the context of the problem tackled in this paper appears in [35]. For details on this matrix factorization and its applications the reader is referred, for example, to [36].

It follows from Theorem 1 in [37] p. 169 (alternatively, see also [36]), that a square matrix $\mathbf{R}$ can be factorized as:

$$\mathbf{R} = \mathbf{W}\mathbf{Y}, \tag{52}$$

where $\mathbf{W}$ is an orthogonal matrix and $\mathbf{Y}$ is a positive semi-definite symmetric matrix. Matrix $\mathbf{Y}$ is unique, even if $\mathbf{R}$ is singular, and is given by

$$\mathbf{Y} = \left(\mathbf{R}^T\mathbf{R}\right)^{\frac{1}{2}}. \tag{53}$$

Therefore, its substitution in (52) yields

$$\mathbf{W} = \mathbf{R}\left(\mathbf{R}^T\mathbf{R}\right)^{-\frac{1}{2}}. \tag{54}$$

Now, if we compare (19) and (54), we conclude that $\mathbf{W}$ in (52) coincides with $\hat{\mathbf{R}}$. In general, to compute the polar decomposition, $\mathbf{W}$ is obtained using iterative algorithms [38–40]. These algorithms exactly correspond to the iterative formulas appearing in Table 1. In this context, these formulas are called Heron's, Halley's, and Housholder's formulas, depending on the order of convergence. Thus, the polar decomposition itself does not provide any new insight into the problem. It simply provides a more elegant formulation. The operations required to obtain it are exactly the same as those described in Section 3.

A method to obtain the polar decomposition of a 3×3 matrix using a closely related approach to that presented in Section 3.7 was presented in [41] where it was was shown to be stable and more efficient than a standard SVD. Contrarily to the method in Section 3.7, it is an iterative method because, in some cases, it relies on a Newton's method. It will be used in Section 4 for comparison purposes with the identified closed-form methods.

3.6.2 | SVD method

The singular value decomposition (SVD) was introduced in this context in [42]. The central role of the SVD in matrix nearness problems was first identified in [43] where an early description of what is now the standard algorithm for computing the SVD. In our case, the SVD of the noisy rotation matrix $\mathbf{R}$ can be expressed as

$$\mathbf{R} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^T, \tag{55}$$

where $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\boldsymbol{\Delta} = \text{diag}(\sigma_1\ \sigma_2\ \sigma_3)$. $\boldsymbol{\Delta}$. The singular values of $\mathbf{R}$, $\sigma_1 > \sigma_2 > \sigma_3$, are nonnegative square roots of the eigenvalues of $\mathbf{R}^t\mathbf{R}$. Then, if $\hat{\mathbf{R}}$ is the orthogonal matrix that minimizes the Frobenius norm error with respect to $\mathbf{R}$, we have that

$$\epsilon = \left\|\mathbf{R} - \hat{\mathbf{R}}\right\|_F^2 = \left\|\mathbf{U}\boldsymbol{\Delta}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\hat{\mathbf{R}}\mathbf{V}\mathbf{V}^T\right\|_F^2 = \left\|\boldsymbol{\Delta} - \widetilde{\mathbf{R}}\right\|_F^2. \tag{56}$$

where $\widetilde{\mathbf{R}} = \mathbf{U}^T\hat{\mathbf{R}}V$ is another orthogonal matrix. Now, observe that minimizing (56) is equivalent to minimizing $\left\|\widetilde{\mathbf{R}}^T\boldsymbol{\Delta} - \mathbf{I}\right\|_F^2$, which in turn is equivalent to maximizing $\text{Tr}(\boldsymbol{\Delta}\widetilde{\mathbf{R}})$, which is maximized for $\widetilde{\mathbf{R}} = \mathbf{I}$. Thus, the optimal rotation matrix is $\hat{\mathbf{R}} = \mathbf{U}\mathbf{V}^T$, if $\det(\mathbf{U})\det(\mathbf{V}) = +1$. In the case in which $\det(\mathbf{U})\det(\mathbf{V}) = -1$, it can be shown that the optimal rotation matrix is given by $\mathbf{R} = \mathbf{U}\ \text{diag}(1, 1, 1, -1)\ \mathbf{V}^T$ [44].

Now, observe that from the SVD of $\mathbf{R}$, we have that

$$\mathbf{R} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^T = \mathbf{U}\mathbf{V}^T\mathbf{V}\boldsymbol{\Delta}\mathbf{V}^T = \mathbf{W}\mathbf{Y} \tag{57}$$

where $\mathbf{W} = \hat{\mathbf{R}}$ and $\mathbf{Y} = \mathbf{V}\boldsymbol{\Delta}\mathbf{V}^T$. In other words, the polar decomposition can be obtained as a byproduct of the SVD.

It is finally interesting to mention that there are methods that compute the square of a matrix based on its Schur complement. In this way the problem comes down to computing the square root of an upper triangular matrix [45] p. 313. This method was first proposed in [46]. Nevertheless, since in our case the matrix is real symmetric, the computation of its Shur complement is equivalent to compute its SVD.

3.6.3 | Closed-form diagonalization method

This method was proposed in [10]. We next summarize it.

If $\det(\mathbf{R}) \neq 0$, $\mathbf{A} = \mathbf{R}^T\mathbf{R}$ is symmetric, positive definitive. Then, it can be diagonalized as follows:

$$\mathbf{A} = \mathbf{Z}^T\begin{pmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{pmatrix}\mathbf{Z}, \tag{58}$$

where $\{\lambda_i\}$ is the set of non-negative real eigenvalues of $\mathbf{A}$. Then, it can be proved that[47]

$$\mathbf{A}^{-\frac{1}{2}} = \mathbf{Z}^T \begin{pmatrix} \frac{1}{\sqrt{\lambda_1}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{\lambda_3}} \end{pmatrix} \mathbf{Z}. \tag{59}$$

A simple method to compute $\{\lambda_i\}$ can be found in[48], where it is shown that

$$\lambda_1 = m + 2\sqrt{p}\cos\theta, \tag{60}$$

$$\lambda_2 = m - 2\sqrt{p}(\cos\theta + \sqrt{3}\sin\theta), \tag{61}$$

$$\lambda_3 = m - 2\sqrt{p}(\cos\theta - \sqrt{3}\sin\theta), \tag{62}$$

where

$$m = \mathrm{Tr}(\mathbf{A})/3, \tag{63}$$

$$p = \mathrm{Tr}[(\mathbf{A} - m\mathbf{I})(\mathbf{A} - m\mathbf{I})^T]/6, \tag{64}$$

$$\theta = \frac{1}{3}\mathrm{atan2}\left(\sqrt{p^3 - q^2}, q\right), \tag{65}$$

with $q = \det(\mathbf{A} - m\mathbf{I})/2$.

The term $p^3 - q^2$ corresponds to the discriminant of the characteristic polynomial of $\mathbf{A}$[49]. When $\mathbf{A}$ has two equal eigenvalues, round-off errors might lead to a small negative value for this discriminant. Any implementation should consider this possibility.

Now, if we apply the Cayley-Hamilton theorem to the characteristic polynomial of $\mathbf{A}^{\frac{1}{2}}$[50], we have that

$$\left(\mathbf{A}^{\frac{1}{2}} - \sqrt{\lambda_1}\mathbf{I}\right)\left(\mathbf{A}^{\frac{1}{2}} - \sqrt{\lambda_2}\mathbf{I}\right)\left(\mathbf{A}^{\frac{1}{2}} - \sqrt{\lambda_3}\mathbf{I}\right) = \mathbf{A}^{\frac{3}{2}} - a_2\mathbf{A} + a_1\mathbf{A}^{\frac{1}{2}} - a_0\mathbf{I} = \mathbf{0}, \tag{66}$$

where

$$a_2 = \sqrt{\lambda_1} + \sqrt{\lambda_2} + \sqrt{\lambda_3},$$
$$a_1 = \sqrt{\lambda_1\lambda_2} + \sqrt{\lambda_1\lambda_3} + \sqrt{\lambda_2\lambda_3},$$
$$a_0 = \sqrt{\lambda_1\lambda_2\lambda_3}.$$

Then, after some simply algebraic manipulations, it can be finally proved that

$$\mathbf{A}^{-\frac{1}{2}} = b_2\mathbf{A}^2 - b_1\mathbf{A} + b_0\mathbf{I}, \tag{67}$$

where

$$b_2 = \frac{a_2}{a_0(a_2a_1 - a_0)}, \tag{68}$$

$$b_1 = \frac{a_0 + a_2(a_2^2 - 2a_1)}{a_0(a_2a_1 - a_0)}, \tag{69}$$

$$b_0 = \frac{a_2a_1^2 - a_0(a_2^2 + a_1)}{a_0(a_2a_1 - a_0)}. \tag{70}$$

Now, it is worth observing what happens for low levels of noise. In this case, $\lambda_i \approx 1$. Then, $a_2 \approx 3$, $a_1 \approx 3$, and $a_0 \approx 1$. As a consequence,

$$\hat{\mathbf{R}} \approx \frac{1}{8}\mathbf{R}(3\mathbf{R}^T\mathbf{R}\mathbf{R}^T\mathbf{R} - 10\mathbf{R}^T\mathbf{R} + 15\mathbf{I}). \tag{71}$$

It is interesting to realize that this formula coincides with the one obtained by computing the Taylor series expansion of $\mathbf{E}$ up to the third term and substituting the result in (19). This formula is actually used in[18] in an iterative algorithm intended to converge to $\hat{\mathbf{R}}$.

The above formulation fails if one of the eigenvalues is zero[10]. Moreover, since the sign of $\det(\hat{\mathbf{R}})$ is the same as that of $\det(\mathbf{R})$, it actually provides a closed-form formula for the nearest orthogonal matrix, not the nearest rotation matrix. Thus, it is only valid if $\det(\mathbf{R})>0$. Moreover, for exact rotation matrices, the three eigenvalues of $\mathbf{A}$ are equal to 1, but as noise is added these eigenvalues start to differ significantly in magnitude which leads to a loss of accuracy[51, 52]. The bound in the level of noise for this method to work correctly is evaluated in Section 4.

3.7 | Closed-form quaternion method

As already mentioned in Section 3.5, Euler's theorem of rotations states that the rotation resulting from any sequence of rotations is equivalent to a single rotation through an angle θ about an axis that we will represent by the unit vector $\mathbf{n} = (n_x\ n_y\ n_z)^T$. Then, it can be proved that a general rotation matrix can be written as

$$\mathbf{R} = \frac{1}{e_0^2+e_1^2+e_2^2+e_3^2}\begin{pmatrix} 2(e_0^2+e_1^2)-1 & 2(e_1e_2-e_0e_3) & 2(e_1e_3+e_0e_2) \\ 2(e_1e_2+e_0e_3) & 2(e_0^2+e_2^2)-1 & 2(e_2e_3-e_0e_1) \\ 2(e_1e_3-e_0e_2) & 2(e_2e_3+e_0e_1) & 2(e_0^2+e_3^2)-1 \end{pmatrix}. \tag{72}$$

where

$$e_0 = \cos(\theta/2), \tag{73}$$

$$e_1 = n_x \sin(\theta/2), \tag{74}$$

$$e_2 = n_y \sin(\theta/2), \tag{75}$$

$$e_3 = n_z \sin(\theta/2). \tag{76}$$

The vector $\mathbf{e} = (e_0\, e_1\, e_2\, e_3)^T$ is defined as the vector of Euler parameters. The elements of this vector are usually arranged in quaternion form and hence the name given to this method, despite the algebra of quaternions is not required in its derivation.

Observe that the elements of $\mathbf{e}$ are not independent as they are related through the following equation:

$$\mathbf{e}^T\mathbf{e} = 1. \tag{77}$$

Nevertheless, in practice, this condition can be relaxed by representing $\mathbf{e}$ in homogeneous coordinates. Expression (72) is actually valid for this general case (observe that $\mathbf{e}$ is implicitly normalized).

To obtain the Euler parameters corresponding to a given rotation matrix, we have to solve the system of equations resulting from equating $\mathbf{R} = (r_{ij})$, $1 \le i, j \le 3$, to (72), which can be reorganized in matrix form as

$$\mathbf{P} = \mathbf{K}, \tag{78}$$

where

$$\mathbf{P} = \mathbf{e}\ \mathbf{e}^T = \begin{pmatrix} e_0e_0 & e_0e_1 & e_0e_2 & e_0e_3 \\ e_1e_0 & e_1e_1 & e_1e_2 & e_1e_3 \\ e_2e_0 & e_2e_1 & e_2e_2 & e_2e_3 \\ e_3e_0 & e_3e_1 & e_3e_2 & e_3e_3 \end{pmatrix} \tag{79}$$

and

$$\mathbf{K} = \frac{1}{4}\begin{pmatrix} r_{11}+r_{22}+r_{33}+1 & r_{32}-r_{23} & r_{13}-r_{31} & r_{21}-r_{12} \\ r_{32}-r_{23} & r_{11}-r_{22}-r_{33}+1 & r_{21}+r_{12} & r_{31}+r_{13} \\ r_{13}-r_{31} & r_{21}+r_{12} & r_{22}-r_{11}-r_{33}+1 & r_{32}+r_{23} \\ r_{21}-r_{12} & r_{31}+r_{13} & r_{32}+r_{23} & r_{33}-r_{11}-r_{22}+1 \end{pmatrix}. \tag{80}$$

Now, let us suppose that, instead of computing directly $\hat{\mathbf{R}}$, we first compute the Euler parameters of $\mathbf{R}$, $\hat{\mathbf{e}} = (\hat{e}_0\, \hat{e}_1\, \hat{e}_2\, \hat{e}_3)^T$, from which we can then derive $\hat{\mathbf{R}}$ using (72).

Therefore, the maximization of (18) is equivalent to maximize

$$\mathrm{Tr}\left[\begin{pmatrix} 2(\hat{e}_0^2+\hat{e}_1^2)-1 & 2(\hat{e}_1\hat{e}_2-\hat{e}_0\hat{e}_3) & 2(\hat{e}_1\hat{e}_3+\hat{e}_0\hat{e}_2) \\ 2(\hat{e}_1\hat{e}_2+\hat{e}_0\hat{e}_3) & 2(\hat{e}_0^2+\hat{e}_2^2)-1 & 2(\hat{e}_1\hat{e}_3-\hat{e}_0\hat{e}_1) \\ 2(\hat{e}_1\hat{e}_3-\hat{e}_0\hat{e}_2) & 2(\hat{e}_2\hat{e}_3+\hat{e}_0\hat{e}_1) & 2(\hat{e}_0^2+\hat{e}_3^2)-1 \end{pmatrix}\begin{pmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{pmatrix}\right] = \hat{\mathbf{e}}^T\mathbf{G}\hat{\mathbf{e}}, \tag{81}$$

where

$$\mathbf{G} = 4(\mathbf{K} - \mathbf{I}), \tag{82}$$

subject to the constraint $\hat{\mathbf{e}}^T\hat{\mathbf{e}} = 1$[53].

Using Lagrange multipliers, we have that the value of $\hat{\mathbf{e}}$ that maximizes the quadratic form $\hat{\mathbf{e}}^T\mathbf{G}\hat{\mathbf{e}}$ subject to the constraint $\hat{\mathbf{e}}^T\hat{\mathbf{e}} = 1$, is obtained by solving

$$\frac{\partial(\mathbf{e}^T\mathbf{G}\mathbf{e})}{\partial\mathbf{e}} = \lambda\frac{\partial(\mathbf{e}^T\mathbf{e})}{\partial\mathbf{e}}. \tag{83}$$

That is,

$$\mathbf{G}\mathbf{e} = \lambda\mathbf{e}. \tag{84}$$

Equivalently, substituting (82) in (84),

$$\mathbf{K}\mathbf{e} = \frac{\lambda + 1}{4}\mathbf{e}. \tag{85}$$

Thus, we have four solution candidates for our optimization problem: the four eigenvectors of $\mathbf{G}$ (equivalently, the eigenvectors of $\mathbf{K}$). Nevertheless, the one that maximizes the quadratic form is clearly the one corresponding to the largest eigenvalue of $\mathbf{G}$.

Since $\mathbf{G}$ is symmetric, its eigenvalues are real. Therefore, the determination of the largest eigenvalue of $\mathbf{G}$ requires computing the largest root of the following quartic characteristic polynomial:

$$\lambda^4 + \tau_3\lambda^3 + \tau_2\lambda^2 + \tau_1\lambda + \tau_0, \tag{86}$$

where

$$\begin{aligned}
\tau_3 &= \mathrm{Tr}(\mathbf{G}) = 0,\\
\tau_2 &= -2\,\mathrm{Tr}\left(\mathbf{R}^T\mathbf{R}\right),\\
\tau_1 &= -8\det(\mathbf{R}),\\
\tau_0 &= \det(\mathbf{G}).
\end{aligned}$$

Since $\tau_3 = 0$, the polynomial in (86) is already a depressed quartic. This means that the application of Ferrari's method to obtain the sought root, as it is done in[54], is simplified.

If the absolute value of τ_1 is below a certain small threshold (in our implementation it is set to 10^{-5}), the polynomial in (86) can be safely approximated by a biquartic polynomial. In this case, the largest eigenvalue is

$$\lambda_{\max} = \sqrt{-\frac{\tau_2}{2} + \sqrt{\left(\frac{\tau_2}{2}\right)^2 - \tau_0}}. \tag{87}$$

Otherwise, in the general case, we have that the largest real root of (86) can be expressed as (see[55] for details)

$$\lambda_{\max} = \frac{1}{\sqrt{6}}\left(k_1 + \sqrt{-k_1^2 - 12\tau_2 - \frac{12\sqrt{6}\tau_1}{k_1}}\right), \tag{88}$$

where

$$\begin{aligned}
k_0 &= 2\tau_2^3 + 27\tau_1^2 - 72\tau_2\tau_0,\\
\theta &= \mathrm{atan2}\left(\sqrt{4(\tau_2^2 + 12\tau_0)^3 - k_0^2},\ k_0\right),\\
k_1 &= 2\sqrt{\left(\sqrt{\tau_2^2 + 12\tau_0}\right)\cos\frac{\theta}{3} - \tau_2}.
\end{aligned}$$

It can be proven that all the rows of the cofactor matrix of $(\mathbf{G} - \lambda_{\max}\mathbf{I})$ are proportional to the eigenvector corresponding to $\lambda_{\max}$[54]. In[55], some computational time is saved by computing only the last row of this cofactor matrix. Unfortunately, all the elements of this row are identically zero for rotations whose rotation axis lies on the xy-plane. Although, at least in theory, rotations whose rotation axes lie on the xy-plane can be seen as a set of measure zero in the space of quaternions, in practice it is enough to be close to this situation to generate large errors. Similar situations arise if we take any other row. Thus, for the sake of robustness, we have to compute all rows and take the one with the largest norm.

3.8 | Closed-form approximate quaternion methods

Observe that, for exact rotation matrices, all the columns of $\mathbf{K}$ in (80), denoted as $\mathbf{k}_i$, $i = 1, \ldots, 4$, are equal to the corresponding quaternion up to a scalar factor. This was first noticed in[54], and later independently rediscovered in[56] and[41] (the same applies to its rows as $\mathbf{K}$ is symmetric). For erroneous rotation matrices, this is no longer true. Then, assuming that the elements of the rotation matrix are contaminated by uncorrelated noise, it is reasonable to compute the sought quaternion as the one that minimizes its quadratic error with respect to the column vectors of $\mathbf{K}$. That is, the value of $\mathbf{e}$ that minimizes

$$\sum_{i=1}^{4} \left\|\mathbf{e} - \mathbf{k}_i\right\|^2 = \sum_{i=1}^{4} \left(\mathbf{e}^T\mathbf{e} + \mathbf{k}_i^T\mathbf{k}_i - 2\mathbf{e}^T\mathbf{k}_i\right), \tag{89}$$

subject to the constraint $\mathbf{e}^T\mathbf{e} = 1$. Expression (89) is minimized when its last term is maximized, which is equivalent to maximize

$$\mathbf{e}^T\mathbf{K}\mathbf{e}, \tag{90}$$

subject to the constraint $\mathbf{e}^T\mathbf{e} = 1$. We already know that the solution to this problem is the largest eigenvector of $\mathbf{K}$. Thus, this approach is equivalent to the one described in Section 3.7.

3.8.1 | Arithmetic mean method

Instead of the quaternion that minimizes its quadratic error with respect to the column vectors of $\mathbf{K}$, we can simply compute the arithmetic mean of these columns as described in [10]. Next, we briefly summarize this method. It we estimate $\mathbf{e}$ by obtaining the arithmetic mean of all rows of $\mathbf{K}$, we obtain

$$\hat{\mathbf{e}} = \sum_{i=1}^{4} \mathbf{k}_i, \tag{91}$$

where $\mathbf{k}_i$ denotes the i row of $\mathbf{K}$. Nevertheless, this simple idea has a subtle flaw. Since $\mathbf{k}_i$ and $-\mathbf{k}_i$ represent the same rotation (quaternions provide a double covering of the rotation group), changing the sign of any $\mathbf{k}_i$ should not change the average. Nevertheless, it is clear that (91) does not have this property. To fix this problem, one possibility is to homogenize the signs of $\mathbf{k}_i$ before averaging them. A simple way to implement this idea reads as follows:

$$\hat{\mathbf{e}} = \sum_{i=1}^{4} \text{sign}(\mathbf{k}_j \cdot \mathbf{k}_i)\mathbf{k}_i, \tag{92}$$

where $\mathbf{k}_j$ is chosen so that $\left\|\mathbf{k}_j\right\| \geq \left\|\mathbf{k}_i\right\|$, $i = 1, \ldots, 4$.

Therefore, the method can be summarized as follows: given the rotation matrix $\mathbf{R}$, $\mathbf{K}$ is computed using (80), then $\hat{\mathbf{e}}$ using (92), and finally $\hat{\mathbf{R}}$ from $\hat{\mathbf{e}}$ using (72). This is probably the simplest of the methods because it only requires the four basic arithmetic operations.

TABLE 3 The four consistent sets of signs for the components of $\hat{\mathbf{e}}$.

	$\text{sign}(e_0)$	$\text{sign}(e_1)$	$\text{sign}(e_2)$	$\text{sign}(e_3)$
$\text{sign}(e_0) = 1$	1	$\text{sign}(r_{32} - r_{23})$	$\text{sign}(r_{13} - r_{31})$	$\text{sign}(r_{21} - r_{12})$
$\text{sign}(e_1) = 1$	$\text{sign}(r_{32} - r_{23})$	1	$\text{sign}(r_{21} + r_{12})$	$\text{sign}(r_{13} + r_{31})$
$\text{sign}(e_2) = 1$	$\text{sign}(r_{13} - r_{31})$	$\text{sign}(r_{21} + r_{12})$	1	$\text{sign}(r_{32} + r_{23})$
$\text{sign}(e_3) = 1$	$\text{sign}(r_{21} - r_{12})$	$\text{sign}(r_{13} + r_{31})$	$\text{sign}(r_{32} + r_{23})$	1

3.8.2 | Cayley's method

This method was presented in [57] as a particularization of Cayley's factorization to three dimensions. Now, we can derive it by simply substituting the arithmetic mean in (92) by the squared mean root. The result reads as follows:

$$|e_0| = \frac{1}{4}\sqrt{r_{11}+r_{22}+r_{33}+1)^2 + (r_{32}-r_{23})^2 + (r_{13}-r_{31})^2 + (r_{21}-r_{12})^2} \tag{93}$$

$$|e_1| = \frac{1}{4}\sqrt{r_{32}-r_{23})^2 + (r_{11}-r_{22}-r_{33}+1)^2 + (r_{21}+r_{12})^2 + (r_{31}+r_{13})^2} \tag{94}$$

$$|e_2| = \frac{1}{4}\sqrt{r_{13}-r_{31})^2 + (r_{21}+r_{12})^2 + (r_{22}-r_{11}-r_{33}+1)^2 + (r_{32}+r_{23})^2} \tag{95}$$

$$|e_3| = \frac{1}{4}\sqrt{r_{21}-r_{12})^2 + (r_{31}+r_{13})^2 + (r_{32}+r_{23})^2 + (r_{33}-r_{11}-r_{22}+1)^2} \tag{96}$$

If we assume that e_0 is positive, we can give a consistent set of signs to the other elements of the quaternion by assigning the signs of $(r_{32} - r_{23})$, $(r_{13} - r_{31})$, and $(r_{21} - r_{12})$, to e_1, e_2, and e_3, respectively. Alternatively, if we assume that e_1 is positive, a consistent set of signs to the other elements of the quaternion result from assigning the signs of $(r_{32} - r_{23})$, $(r_{21} + r_{12})$, and $(r_{1,3} + r_{3,1})$ to e_0, e_2, and e_3, respectively. Table 3 summarizes the four possible alternatives.

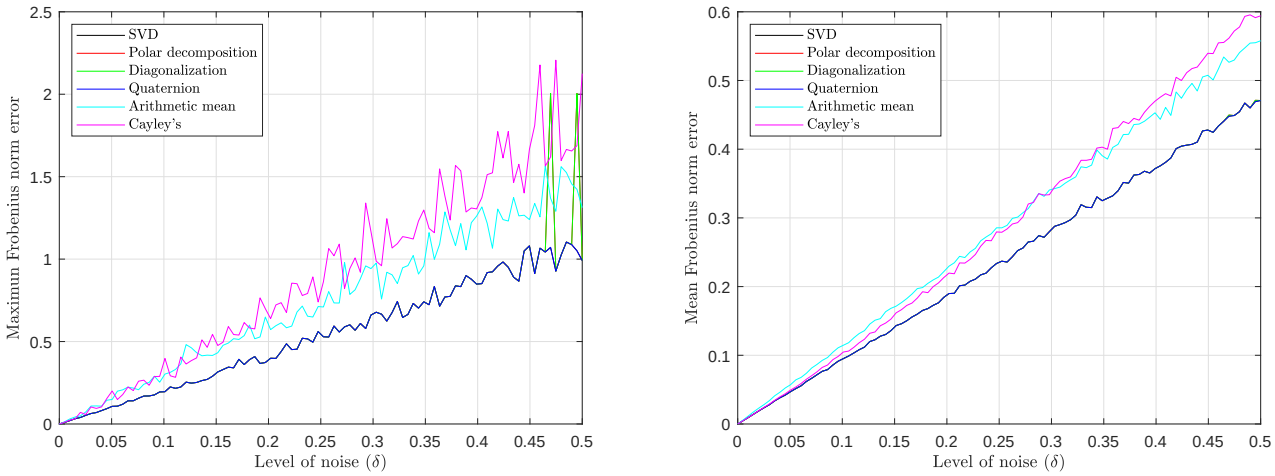

FIGURE 7 Maximum Frobenius norm (left) and mean Frobenius norm (right) of the difference between the original error-free matrices and the estimations obtained from their error-contaminated versions using the SVD method, the polar decomposition method, and the four closed-form methods. For levels of error lower than 0.45, the plots for all exact methods overlap. Only the approximate methods (the arithmetic mean method and Cayley's method) obviously exhibit a different behavior.

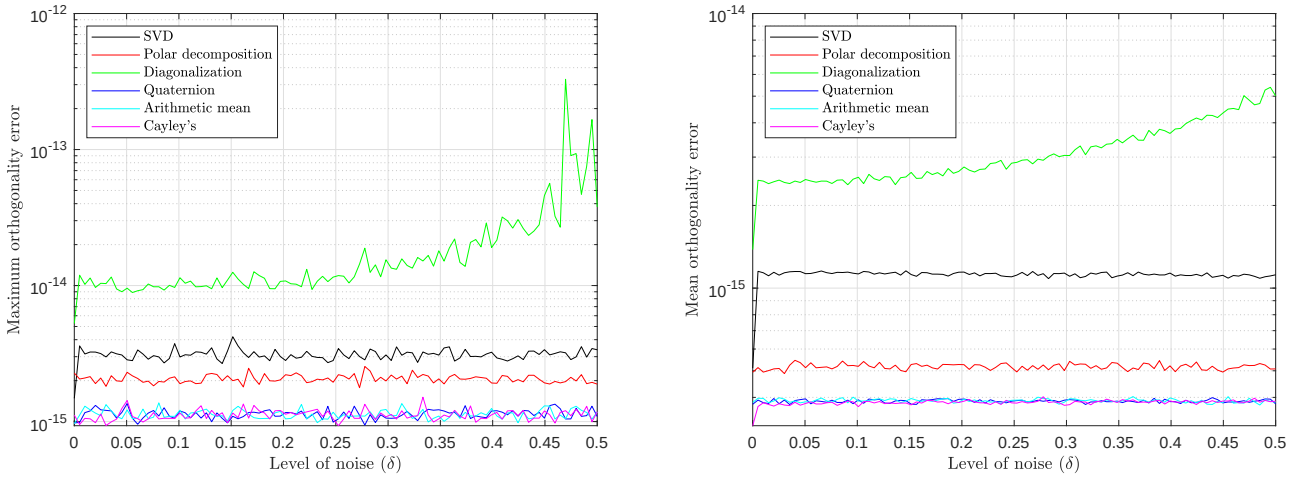

FIGURE 8 Maximum orthogonality error (left) and mean orthogonality error (right) of $\hat{\mathbf{R}}$ obtained using the SVD method, the polar decomposition, and the four closed-form methods. Even in the case of the diagonalization method, the committed orthogonality errors can be considered as negligible for most applications.

4 | PERFORMANCE COMPARISON OF THE ALGEBRAIC CLOSED-FORM METHODS

In this section, we compare all closed-form formulas described above, in terms of accuracy and computational cost, with respect to following two iterative methods:

- The SVD method described in Section 3.6.2. In this case, the Matlab® built-in function *svd* is used; and
- The polar decomposition method described in Section 3.6.1 as it is considered as an advantageous alternative to the SVD for 3×3 matrices. In this case, the implementation delivered by the authors of[41] is used.

In particular, the implemented closed-form formulas correspond to:

- The diagonalization method described in Section 3.6.3.
- The quaternion method described in Section 3.7.
- The arithmetic mean method described in Section 3.8.1.
- The Cayley's method described in Section 3.8.2.

To assess the performance of these methods, we have implemented the following procedure in Matlab® on a PC with a CoreTMi7 processor running at 3.70 GHz and 16 GB of RAM:

1. Generate 10^5 random quaternions using the algorithm detailed in[7], which permits to obtain sets of points uniformly distributed on $\mathbb{S}^3$.
2. Convert these quaternions to rotation matrices whose elements are then contaminated with additive uncorrelated uniformly distributed noise in the interval $[-\delta, \delta]$.
3. Compute the nearest rotation matrices for these 10^5 noisy rotation matrices using each of the above methods.
4. Compute the maximum and the mean Frobenius norm between the noisy matrices and the original rotation matrices using each method.
5. Compute the maximum and the mean orthogonality error of the obtained results as the Frobenius norm of $\hat{\mathbf{R}}\hat{\mathbf{R}}^T - \mathbf{I}$.

If the above procedure is repeated for δ ranging from 0 to 0.5, the plots in Figs. 7 and 8 are obtained. Fig. 7 shows the maximum and mean Frobenius norm error between the original error-free matrices and the corresponding nearest rotation matrices to their noisy versions using the six aforementioned methods implemented in single-precision arithmetic. Except for the two approximate methods, the plots for all other methods overlap for $\delta < 0.45$. For higher levels of error, the diagonalization method starts to have problems, as we already predicted in Section 3.6.3. Curiously enough, the arithmetic mean method performs better than the Cayley's method despite it results from a heuristic argument.

To assess the orthogonality of the obtained rotation matrices, we have also computed the maximum and the mean Frobenius norm of $\hat{\mathbf{R}}\hat{\mathbf{R}}^T - \mathbf{I}$ for δ ranging from 0 to 0.5. The results are plotted in Fig. 8. All orthogonality errors are always lower that 10^{-13}, which is completely negligible for most applications. Nevertheless, contrarily to all other methods, the orthogonality error for the diagonalization method increases with the level of noise. As a consequence, in general, it should be avoided for levels of noise greater than 0.45.

The most clear distinctive feature between the compared methods is their execution time. Indeed, using single-precision arithmetic, their average execution times, normalized with respect to that of the SVD, are as follows

SVD	1.00
Polar decomposition	2.22
Diagonalization	0.59
Quaternion	103.07
Arithmetic mean	0.78
Cayley's	0.89

Nevertheless, the inclusion of the SVD in the above comparison is meaningless because it is based on a Matlab®'s compiled built-in function. It was actually shown in[10] that an interpreted version of the SVD has a computational cost two orders of magnitude higher than its compiled counterpart.

5 | CONCLUSION

The Singular Value Decomposition (SVD) is probably the most important matrix factorization of the computational era. It provides a numerically stable matrix decomposition that can be used for solving a large variety of problems, including the nearest

rotation matrix problem. Nevertheless, we have shown how solving this problem using the SVD is not, in general, a good choice. A clear better approach consists in using the polar decomposition method presented in [41].

We have also shown how some little known closed-form algebraic methods perform well both in terms of computational time and accuracy. In real-time control applications, these closed-form methods are preferable than numerical ones because they permit analyzing symbolically the influence of each variable on the result, and their computational cost –in terms of arithmetic operations– is constant. Moreover, two of these closed-form methods are approximate methods that can be easily implemented in hardware due to their simplicity.

Finally, it is worth adding that our comparison has been based on randomly generated inputs. This analysis could be enhanced by including the ill-conditioned inputs of the benchmark tests provided in [41] for which the polar decomposition method presented therein was proved to be stable, whereas other methods could be unstable. This is certainly a point that deserves further attention.

References

1. Werneth CM, Dhar M, Maung KM, Sirola C, and Norbury JW. Numerical Gram–Schmidt orthonormalization. Eur J Physics. 2010;**31**(3):1058–1087.

2. Paul RP. Robot Manipulators: Mathematics, Programming, and Control. Cambridge, MA: MIT Press; 1982.

3. Stewart GW. Matrix Algorithms. Volume I: Basic Decompositions. SIAM; 1998.

4. Nugraha AS, and Basaruddin T. Analysis and comparison of QR decomposition algorithm in some types of matrix. In: FedCSIS 2012. Wroclaw, Poland; 2012. p. 561–565.

5. Moler C. Compare Gram-Schmidt and Householder orthogonalization algorithms. https://blogs.mathworks.com/cleve/2016/07/25/compare-gram-schmidt-and-householder-orthogonalization-algorithms; 2016.

6. Bar-Itzhack IY, and Fegley KA. Orthogonalization techniques of a direction cosine matrix. IEEE T Aero and Elec Sys. 1964;**AES-5**(5):798–804.

7. Marsaglia G. Choosing a point from the surface of a sphere. Ann Math Stat. 1972;**43**(2):645–646.

8. Zhuang H, Roth ZS, and Sudhakar R. Practical fusion algorithms for rotation matrices: A comparative study. J Robotic Syst. 1992;**9**(7):915–931.

9. Costandin M, Costandin B, and Dobra P. A new orthogonalization and sensor fusion algorithm for attitude estimation. In: 2018 IEEE AQTR. Cluj-Napoca, Romania; 2018. p. 1–6.

10. Sarabandi S, Shabani A, Porta JM, and Thomas F. On closed-form formulas for the 3-D nearest rotation matrix problem. IEEE T Rob. 2020;**36**(4):1333–1339.

11. Giardini CR, Bronson R, and Wallen I. An optimal normalization scheme. IEEE T Aeros Elec Sys. 1975;**AES-11**(4):443–446.

12. Bar-Itzhack IY. Iterative optimal orthogonalization of the strapdown matrix. IEEE T Aero Elec Sys. 1975;**11**(1):30–37.

13. Bar-Itzhack IY, Meyer J, and Fuhrmann PA. Strapdown matrix orthogonalization: the dual iterative algorithm. IEEE T Aero Elec Sys. 1976;**AES-12**(1):32–38.

14. Horn BK, Hilden HM, and Negahdaripour S. Closed form solution of absolute orientation using orthonormal matrices. J Opt Soc Am A. 1988;**5**(7):1127–1135.

15. Gains HT. 1965. *Attitude matrix orthonormality correction*. . Honeywell Interoffice Correspondence.

16. Björek A, and Bowie C. An iterative algorithm for computing the best estimate of an orthogonal matrix. SIAM J Numer Anal. 1971;**8**(2):358–364.

17. Hasan MA. Families of orthonormalization algorithms. In: IJCNN 2009. Atlanta, Georgia, USA; 2009. p. 14–19.

18. Hasan MA. Square-root free orthogonalization algorithms. In: ICASSP '99. Taipei, Taiwan; 2009. p. 3173–3176.

19. Bar-Itzhack IY, and Meyer J. On the convergence of iterative orthogonalization processes. IEEE T Aero Elec Sys. 1976;**AES-12**(2):1946–1951.

20. Mao J, and Yin B. An orthonormalization algorithm for inertial navigation systems by using matrix sign function. In: IFAC 12th Tri. WC. Sydney, Australia; 1993. p. 801–804.

21. Liu Z, and Yan Q. New iterative algorithm for attitude determination. In: 2010 Int. Conf. E-Prod. E-Serv. E-Entert. Henan, China; 2010. p. 1–4.

22. Priester RW, and Denman ED. Orthogonalization of a direction cosine matrix by iterative techniques. IEEE T Aero Elec Sys. 1972;**AES-8**(5):692–694.

23. Bar-Itzhack IY, Meyer J, and Fuhrmann PA. Correction to 'Strapdown matrix orthogonalization: the dual iterative algorithm'. IEEE T Aero and Elec Sys. 1976;**AES-12**(1):32–38.

24. Higham NJ. Newton's method for the matrix square root. Math Comput. 1986;**46**:537–550.

25. Hasan MA. Orthonormalization learning algorithm. In: IJCNN 2007. Orlando, Florida, USA; 2007. p. 12–17.

26. Higham NJ. Stable iterations for the matrix square root. Numer Algorithms. 1997;**15**(2):227–242.

27. Denman ED, and Beavers Jr AN. The matrix sign function and computations in systems. Appl Math Comput. 1976;**2**(1):63–94.

28. Lu YY. A Padé approximation method for square roots of symmetric positive definite matrices. SIAM J Matrix Anal A. 1998;**19**(3):833–845.

29. Jones WB, and Thron WJ. Continued Fractions, Analytic Theory and Applications. Addison-Wesley, Reading, MA; 1990.

30. Goldstein H. Classical Mechanics. Cambridge, MA: Addison-Wesley; 1951.

31. Murray RM, Li Z, and Sastry SS. A Mathematical Introduction to Robotic Manipulation. Boca Raton, FL: CRC Press; 1994.

32. Al-Mohy AH, and Higham NJ. Improved inverse scaling and squaring algorithms for the matrix logarithm. SIAM J Sci Comp. 2012;**34**(4):153–169.

33. Higham NJ. Functions of Matrices: Theory and Computation. SIAM; 2008.

34. Culver WJ. On the existence and uniqueness of the real logarithm of a matrix. P Am Math Soc. 1966;**17**(5):1146–1151.

35. Farrell JL, Stuelpnagel JC, Wessner RH, and Velman JR. A least squares estimate of spacecraft attitude. Solution 65-1. SIAM Rev. 1966;**8**(3):384–386.

36. Higham NJ. Computing the polar decomposition—with applications. SIAM J Sci Stat Comp. 1986;**7**(4):1160–1174.

37. Halmos PR. Finite Dimensional Vector Spaces. 2nd ed. Princeton, N.J: Van Nostrand; 1958.

38. Higham NJ, and Schreiber RS. Fast polar decomposition of an arbitrary matrix. SIAM J Sci Stat Comp. 1990;**11**(4):648–655.

39. Du K. The iterative methods for computing the polar decomposition of rank-deficient matrix. Appl Math Comput. 2005;**162**(1):95–102.

40. Soleymani F, Stanimirović PS, and Stojanović I. A novel iterative method for polar decomposition and matrix sign function. Discrete Dyn Nat Soc. 2015;(Article ID 649423).

41. Higham NJ, and Noferini V. An algorithm to compute the polar decomposition of a 3×3 matrix. Numer Algorithms. 2016;**73**(2):349–369.

42. Mao J. Optimal orthonormalization of the strapdown matrix by using singular value decomposition. Comp Math Appl. 1986;**12**(3):253–262.

43. Golub GH. Least squares, singular values and matrix approximations. Appl Math-Czech. 1968;**13**:44–51.

44. Eggert DW, Lorusso A, and Fisher RB. Estimating 3-D rigid body transformations: a comparison of four major algorithms. Mach Vision Appl. 1997;**9**(5):272–290.

45. Golub GH, and Van Loan CF. Matrix Computations. Johns Hopkins University Press; 1983.

46. Björck A, and Hammarling S. A Schur method for the square root of a matrix. Linear Algebra Appl. 1983;**52-53**:127–140.

47. Schweinler HC, and Wigner EP. Orthogonalization methods. J Math Phys. 1970;**11**(5):1693–1694.

48. Smith OK. Eigenvalues of a symmetric 3×3 matrix. Commun ACM. 1961;**4**(4):168.

49. Blinn JF. How to solve a cubic equation, part 1: The shape of the discriminant. IEEE Comput Graph. 2006;**26**(3):84–93.

50. Franca LP. An algorithm to compute the square root of a 3×3 positive definite matrix. Comput Math Appl. 1989;**18**(5):459–466.

51. Kahan W. Lecture notes for a numerical analysis course. https://people.eecs.berkeley.edu/~wkahan/Math128/Cubic.pdf; 1986.

52. Blinn JF. How to solve a cubic equation, part 5: Back to numerics. IEEE Comput Graph. 2007;**27**(3):78–89.

53. Keat J. 1977. *Analysis of least-squares attitude determination routine DOAOP*. CSC/TM-77/6034. Goddard Space Flight Center, Greenbelt, MD.: Computer Science Corp.

54. Horn BK. Closed-form solution of absolute orientation using unit quaternions. J Opt Soc Am. 1987;**4**(4):629–642.

55. Wu J, Liu M, Zhou Z, and Li R. Fast symbolic 3D registration solution. IEEE T Autom Sci Eng. 2019;**17**(2):761–770.

56. Bar-Itzhack IY. New method for extracting the quaternion from a rotation matrix. J Guid Control Dynam. 2000;**23**(6):1085–1087.

57. Sarabandi S, Perez-Gracia A, and Thomas F. On Cayley's factorization with an application to the orthonormalization of noisy rotation matrices. Adv Appl Clifford Al. 2019;**29**(3):49–65.

How to cite this article: S. Sarabandi, and F. Thomas (2023), Solution Methods to the Nearest Rotation Matrix, *Numerical Linear Algebra with Applications*, *2023;00:1–21*.