Completely Rational $\mathrm{SO}(n)$ Orthonormalization

Jin Wu[1], Soheil Sarabandi[2], Jianhao Jiao[1], Huaiyang Huang[1], Bohuan Xue[1], Ruoyu Geng[1], Lujia Wang[1], and Ming Liu[1,3,4]

***Abstract*— The rotation orthonormalization on the special orthogonal group $\mathrm{SO}(n)$, also known as the high dimensional nearest rotation problem, has been revisited. A new generalized simple iterative formula has been proposed that solves this problem in a completely rational manner. Rational operations allow for efficient implementation on various platforms and also significantly simplify the synthesis of large-scale circuitization. The developed scheme is also capable of designing efficient fundamental rational algorithms, for example, quaternion normalization, which outperforms long-exisiting solvers. Furthermore, an $\mathrm{SO}(n)$ neural network has been developed for further learning purpose on the rotation group. Simulation results verify the effectiveness of the proposed scheme and show the superiority against existing representatives. Applications show that the proposed orthonormalizer is of potential in robotic pose estimation problems, e.g., hand-eye calibration.**

I. Introduction

Rotation is one of the most important motion parameters involved in modern robotics. The 3-D rotation directly relates to the real world that we live in. Thus many robotics problems involve obtaining an improper rotation based on algebraic techniques. The improper rotation matrix must be refined to a nearest one that exactly distributes on the rotation group, which defines the nearest rotation problem [1]. A practical problem is that, when conducting certain estimation tasks, one rotation matrix may not satisfy these constraints due to word-length limit, communication bandwidth, transmission dropouts, etc. Therefore, engineers tend to find out the nearest rotation to these improper ones, such is the purpose of the $\mathrm{SO}(3)$ rotation orthonormalization. In abstract algebra and manifold theory, a matrix is considered as a 3-D rotation if and only if it is on the special orthogonal group, such that $\mathrm{SO}(3) := \left\{\boldsymbol{R} \in \mathbb{R}^{3\times 3} \,\middle|\, \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}, \det(\boldsymbol{R}) = 1\right\}$. Note that there are two nonlinear constraints for $\boldsymbol{R}$ in the definition of $\mathrm{SO}(3)$. In the 3-D world, it is convenient for us to characterize one rotation via three explicit Euler angles. According to the trigonometric relationships between these angles, it will also be evident that the mapping from angles to a rotation matrix is nonlinear. Therefore, many rotation estimation problems are challenging and time-consuming due to the existing non-convexity.

3-D rotation comes always together with a translation in many robotics problems, which forms a compact homogenous pose. In the way of understanding 3-D pose, researchers come to a consensus that maybe higher-dimensional rotation is helpful. Extending from $\mathrm{SO}(3)$, the $\mathrm{SO}(n)$ contains all n-dimensional rotation matrices such that $\mathrm{SO}(n) := \left\{\boldsymbol{R} \in \mathbb{R}^{n\times n} \,\middle|\, \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}, \det(\boldsymbol{R}) = 1\right\}$. Adding the translation raises new challenges, since in the new problem, the rotation and translation will be coupled together. Thus, mapping from 3-D pose to higher-dimensional space becomes a possible choice. One n-D pose is embedded as element $\boldsymbol{X}$ in the special Euclidean group $\mathrm{SE}(n)$ with given rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$:

$$\mathrm{SE}(n) := \left\{ \boldsymbol{X} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix} | \boldsymbol{R} \in \mathrm{SO}(n), \boldsymbol{t} \in \mathbb{R}^n \right\} \quad (1)$$

Thomas pointed out that a 3-D pose can be coupled into a 4-D transformation [2]. This has been later introduced for an analytical 4-D Procrustes-based hand-eye calibration [3]. An imperfect conversion from $\mathrm{SE}(n)$ to $\mathrm{SO}(n+1)$, named as SE(n)++, has been proposed [4]. Ozyesil et al. proposed a perfect framework that maps n-D pose to $(n+1)$-D space via contraction [5]. The results are quite helpful for fundamental problems like pose graph optimization (PGO) [6]. Recently, Soheil *et al.* also reveal that such mapping from $\mathrm{SE}(3)$ to $\mathrm{SO}(4)$ is beneficial for point-cloud registration [7]. All these developments reflect that estimation of higher-dimensional rotation space might be essential. Thus the orthonormalization on the $\mathrm{SO}(n)$ space is also required. Conventionally, the $\mathrm{SO}(n)$ orthonormalization problem can be solved via singular value decomposition (SVD). However, the SVD algorithm suffers from uncontrollable convergence due to many specific designs and different matrix structures. Therefore, for industrial parallel and circuit implementation, SVD is not so friendly and may require many additional modules and cores. The sophisticated internal mechanism of SVD also sets up an obstacle for uncertainty propagation between real space $\mathbb{R}^{n\times n}$ and $\mathrm{SO}(n)$. Therefore, we would like to explore one framework that is simple and rational, leading to friendly implementation for most of the known platforms. Based on the previous related work of ours on $\mathrm{SO}(3)$ orthonormalization [8], [9], this paper aims to give primitive answers to *1)* What the n-dimensional rotation space $\mathrm{SO}(n)$ looks like given a Euclidean perspective? *2)* How is the Euclidean matrix space $\mathbb{R}^{n\times n}$ rationally con-

This work was supported by Guangdong Basic and Applied Basic Research Foundation, under project 2021B1515120032, Foshan-HKUST Project no. FSUST20-SHCIRI06C, and Project of Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone (HZQB-KCZYB-2020083), awarded to Prof. Ming Liu.

[1]J. Wu, J. Jiao, H. Huang, B. Xue, R. Geng, L. Wang and M. Liu are with Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong SAR, China. eelium@ust.hk

[2]S. Srabandi is with the Research Center "E.Piaggio", School of Engineering, University of Pisa, Italy. soheil.sarabandi@gmail.com

[3]HKUST Guangzhou, Nansha, Guangzhou, 511400, Guangdong, China.

[4]HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen, China.

nected to the special orthogonal group $\mathrm{SO}(n)$?

II. Rational $\mathrm{SO}(n)$ Orthonormalization

A. Problem Formulation

Finding the nearest rotation of a noisy matrix $\boldsymbol{B}$ of the size $n \times n$ can be achieved via the following Frobenius-norm optimization

$$\mathop{\arg\min}_{\boldsymbol{R} \in \mathrm{SO}(n)} \ \|\boldsymbol{R} – \boldsymbol{B}\|_{\mathrm{F}}^2 = \mathrm{tr}\left[(\boldsymbol{R} - \boldsymbol{B})^\top (\boldsymbol{R} - \boldsymbol{B})\right] \tag{2}$$

which is simplified using the identities of $\mathrm{SO}(n)$, $\boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}, \det(\boldsymbol{R}) = 1$, such that

$$\mathop{\arg\max}_{\boldsymbol{R} \in \mathrm{SO}(n)} \ \mathrm{tr}\left(\boldsymbol{R}\boldsymbol{B}^\top\right) \tag{3}$$

This can also be derived in terms of the $\mathrm{SO}(n)$ geodesic error, namely the optimization

$$\mathop{\arg\min}_{\boldsymbol{R} \in \mathrm{SO}(n)} \ \mathrm{tr}\left[(\boldsymbol{R}\boldsymbol{B}^\top - \boldsymbol{I})^\top (\boldsymbol{R}\boldsymbol{B}^\top - \boldsymbol{I})\right] \tag{4}$$

in which $\boldsymbol{R}\boldsymbol{B}^\top - \boldsymbol{I}$ denotes the $\mathrm{SO}(n)$ distance of the error rotation term $\boldsymbol{R}\boldsymbol{B}^\top$ with respect to the $\mathrm{SO}(n)$ origin $\boldsymbol{I}$, i.e. an identity matrix. (3) can be solved using SVD as follows

$$\boldsymbol{R} = \boldsymbol{U} \mathrm{diag}\left[1, 1, \cdots, \det(\boldsymbol{U}\boldsymbol{V})\right] \boldsymbol{V}^\top \tag{5}$$

where $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \boldsymbol{B}$ is the SVD of $\boldsymbol{B}$. The method was first officially introduced by Arun *et al.* in [10] and refined by Umeyama in [11].

B. A Perspective of Optimal Transport

In fact, from a viewpoint of manifold theory, finding the nearest $\mathrm{SO}(n)$ matrix is an optimal transport problem [12], [13], such that it optimally transports the matrix in n-dimensional real space to the one on $\mathrm{SO}(n)$

$$\begin{aligned} &\mathop{\arg\min}_{\mathcal{T}: \mathbb{R}^{n \times n} \to \mathrm{SO}(n)} \int_{\mathbb{R}^{n \times n}} c\left[\boldsymbol{X}, \mathcal{T}(\boldsymbol{X})\right] \mu(\boldsymbol{X}) \mathrm{d}\boldsymbol{X} \\ &\text{s.t.} \ \int_{\boldsymbol{R}} \gamma(\boldsymbol{Y}) \mathrm{d}\boldsymbol{Y} = \int_{\mathcal{T}^{-1}(\boldsymbol{R})} \mu(\boldsymbol{X}) \mathrm{d}\boldsymbol{X}, \ \forall \boldsymbol{R} \in \mathrm{SO}(n) \end{aligned} \tag{6}$$

in which the transform $\mathcal{T}$ transports $\mathbb{R}^{n \times n}$ to $\mathrm{SO}(n)$; $c\left[\boldsymbol{X}, \mathcal{T}(\boldsymbol{X})\right]$ denotes the cost function of transporting $\boldsymbol{X} \in \mathbb{R}^{n \times n}$ to $\mathrm{SO}(n)$ and $\mu(\boldsymbol{X})$ is the probability density function (PDF) of $\boldsymbol{X}$, namely, the von-Mises-Fisher distribution [14]. The optimal transport is constrained by the PDF equality such that the volumes of two manifolds are identical, where $\gamma(\boldsymbol{Y})$ denotes the PDF on $\mathrm{SO}(n)$. For $\mathrm{SO}(n)$ orthonormalization, there is no such a $\mathcal{T}^{-1}$ since $\mathrm{SO}(n)$ is a subspace of $\mathbb{R}^{n \times n}$. Therefore the $\mathcal{T}$ studied here is a compression mapping. For instance, for any non-singular symmetric matrix $\boldsymbol{B} \in \mathbb{S}$, it can be proven that $\boldsymbol{U}^{-1} = \boldsymbol{V}^\top$ for $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \boldsymbol{B}$, which gives the fact that the nearest rotation matrix to any $\boldsymbol{B} \in \mathbb{S}$ is the identity matrix $\boldsymbol{I}$. Problem (6) was originated from a general one proposed by Monge, so we also call it as the Monge problem hereinafter. The Monge problem is not guaranteed to have a solution. It is obvious that (6) is unsolvable because of the non-existed $\mathcal{T}^{-1}$. However, if we relax the PDF equality as

$$\int_{\mathrm{SO}(n)} \gamma(\boldsymbol{Y}) \mathrm{d}\boldsymbol{Y} = \int_{\mathbb{R}^{n \times n}} \mu(\boldsymbol{X}) \mathrm{d}\boldsymbol{X} \tag{7}$$

the optimization (6) will be solvable. Actually, such a pair of (μ, γ) can be selected as the following normalized PDFs

$$\begin{aligned} \mu(\boldsymbol{X}) = \mu^\star(\boldsymbol{x}) = (2\pi)^{-\frac{n^2}{2}} \sqrt{\det(\boldsymbol{\Sigma})} \ \times& \\ \exp\left[-\frac{1}{2}(\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \bar{\boldsymbol{x}})\right]& \end{aligned} \tag{8}$$

$$\gamma(\boldsymbol{Y}) = \frac{1}{c_d(\kappa)} \exp\left[\kappa \ \mathrm{tr}\left(\boldsymbol{R}^\top \boldsymbol{Y}\right)\right] \tag{9}$$

in which $\boldsymbol{x} = \mathrm{vec}(\boldsymbol{X}) \sim \mathcal{N}(\bar{\boldsymbol{x}}, \boldsymbol{\Sigma})$ and c_d is the normalization factor. Thus, the target optimal transport problem is

$$\mathop{\arg\max}_{\mathcal{T}: \mathbb{R}^{n \times n} \to \mathrm{SO}(n)} \int_{\mathbb{R}^{n \times n}} \mathrm{tr}\left[\mathcal{T}(\boldsymbol{X}) \boldsymbol{X}^\top\right] \mu(\boldsymbol{X}) \mathrm{d}\boldsymbol{X} \tag{10}$$

Problem (10) is still unsolvable since there is no infimum or supremum of $\boldsymbol{X}$. We therefore give a constraint to the problem so that $\boldsymbol{X}$ is bounded within an n^2-dimensional sphere $\Omega := \left\{\boldsymbol{z}^\top \boldsymbol{z} = m^2, m > 0, \boldsymbol{z} \in \mathbb{R}^{n^2}\right\}$ such that $\sup(\boldsymbol{x}) = \Omega$, $\inf(\boldsymbol{x}) = \boldsymbol{0}$ in which $\boldsymbol{x} = \mathrm{vec}(\boldsymbol{X})$. The supremum boundary problem is

$$\mathop{\arg\max}_{\mathcal{T}^\star: \mathbb{R}^{n^2} \to \mathrm{SO}(n)} \int_{\Omega} \mathrm{tr}\left[\mathcal{T}^\star(\boldsymbol{x}) \boldsymbol{X}^\top\right] \mu^\star(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{11}$$

where $\mathcal{T}^\star$ is the vectorized version of $\mathcal{T}$ such that $\mathcal{T}^\star(\boldsymbol{x}) \in \mathrm{SO}(n)$. More specifically, to deal with many implementation problems in hardware, we expect that the transformation $\mathcal{T}^\star$ is rational, noted as $\mathcal{T}_{\mathbb{Q}}^\star$, so that

$$\mathop{\arg\max}_{\mathcal{T}_{\mathbb{Q}}^\star: \mathbb{R}^{n^2} \to \mathrm{SO}(n)} \int_{\Omega} \mathrm{tr}\left[\mathcal{T}_{\mathbb{Q}}^\star(\boldsymbol{x}) \boldsymbol{X}^\top\right] \mu^\star(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{12}$$

where $\mathbb{Q}$ denotes the set of all rational numbers. This paper proposes such a rational transformation $\mathcal{T}_{\mathbb{Q}}^\star$.

C. Rational Solution

Arun's method has been extensively applied by the community. The SVD also behaves as a quite robust algorithm upon deployment. However, SVD still has its shortcoming in industrial implementation. The principle of computing SVD mostly relies on the concept of the Jacobi rotation comprising the trigonometric functions. For the digital implementation, as all the internal processing procedures are based on integers, the Jacobi rotation needs to be approximated using the power series or compact solution like CORDIC and its variants. These approximations, however, will induce inevitable round-off errors which may cause numerical instability problems. Also, according to such nonlinearity remaining in the SVD, it is almost unlikely for practitioners to conduct the SVD with analog circuits. The following contents will involve the derivation of a brand-new method for n-dimensional registration, which is circuitization-friendly.

The SVD solution of (3) is in fact related to the following point-cloud registration problem [15]

$$\mathop{\arg\min}_{\boldsymbol{R}\in \mathrm{SO}(n),\boldsymbol{t}\in\mathbb{R}^n} \sum_{i=1}^{N} \|\boldsymbol{b}_i - \boldsymbol{R}\boldsymbol{p}_i - \boldsymbol{t}\|^2 \Rightarrow \mathop{\arg\max}_{\boldsymbol{R}\in\mathrm{SO}(n)} \ \mathrm{tr}(\boldsymbol{R}\boldsymbol{B}^\top) \tag{13}$$

which optimizes the $\boldsymbol{R}$ with the geodesic error on the $\mathrm{SO}(n)$. (13) shows that the registration is equivalent to an orthonormalization problem of an optimal $\boldsymbol{R}$ that best approximates $\boldsymbol{B}$. Our proposed algorithm will then be focused on such orthonormalization problem: given the column vectors of $\boldsymbol{B}$ such that $\boldsymbol{B} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_n)$ to solve the orthonormalized $\boldsymbol{B}$, we propose the following iterative mechanism

$$\begin{cases} \boldsymbol{r}_{1,k} = \varrho_k \left(\alpha \boldsymbol{r}_{1,k-1} + s_1 \beta \boldsymbol{r}^{\otimes}_{1,k-1}\right) \\ \boldsymbol{r}_{2,k} = \varrho_k \left(\alpha \boldsymbol{r}_{2,k-1} + s_2 \beta \boldsymbol{r}^{\otimes}_{2,k-1}\right) \\ \qquad\qquad \vdots \\ \boldsymbol{r}_{n,k} = \varrho_k \left(\alpha \boldsymbol{r}_{n,k-1} + s_3 \beta \boldsymbol{r}^{\otimes}_{n,k-1}\right) \end{cases} \tag{14}$$

in which k denotes the iteration index and the initial condition is set to $\boldsymbol{r}_{j,0} = \boldsymbol{r}_j$, for $j = 1, 2, \ldots, n$; ρ_k is a rational normalization factor while α and β are tuning parameters that control the convergence rate; $\boldsymbol{r}^{\otimes}_{j,k}$ denotes the orthonormal error other than the j-th column of updated $\boldsymbol{B}$; $s_j \in \{-1, 1\}$ denotes the sign for j-th correction.

Challenges: This algorithm is largely based on our previous works [8], [9], which is a special case in 3-D space. The process of obtaining (14) is challenging because there is a fact that in higher dimensional space, the geometric error between two vectors can hardly be obtained in the way in 3-D space, i.e., the cross product. In fact, if the dimension $n = 3$ while one ignores the translation $\boldsymbol{t}$ in (13), the problem is considered for pure attitude matrix and is called the Wahba's problem. In this way, (14) well corresponds to the correction step of a well-known complementary filter proposed by Mahony *et al.* [16]. The global and local stability of (14) is also challenging, according to the discrete nature of the formula, which reveals that the construction of the Lyapunov potential of the corresponding continuous system will be complex. These works are going to be shown in a future journal version. In the current version, we mainly show the usefulness of this formula in robotics and applications. The proposed solution has been empirically verified to be completely convergent over multiple tests.

The cross product of two vectors describes the orthonormal error between the vectors. For the 3D case, when two vectors are collinear, their cross product will be zero and when they are perpendicular to each other, the norm of cross product achieves its maximum. Unfortunately, the cross product has only been proven unique in the 3-D and 7-D spaces. Therefore, the search for an intuitive formulation of the generalized n-dimensional cross product will be trivial. However, according to the Grassmannian theory and the exterior algebra, for the n-dimensional space, a subset of $n-1$ vectors will own a unique cross product. This can be explicitly given by

$$\boldsymbol{r}^{\otimes}_j = \boldsymbol{r}_1 \times \boldsymbol{r}_2 \times \cdots \times \boldsymbol{r}_{j-1} \times \boldsymbol{r}_{j+1} \times \cdots \times \boldsymbol{r}_n \tag{15}$$

Since it is unique, one can also write it into the compact matrix form so that $\boldsymbol{r}^{\otimes}_j = \boldsymbol{Q}\boldsymbol{r}_n$, where $\boldsymbol{Q}$ is constituted by the remaining $n-2$ vectors other than j-th and n-th vectors. $\boldsymbol{Q}$ here is skew-symmetric and owns the following structure

$$\boldsymbol{Q}_{ij} = (-1)^{n-i-j} \det(\boldsymbol{P}^{ij}), \ \boldsymbol{Q}_{ij} = -\boldsymbol{Q}_{ji}, \ \boldsymbol{Q}_{ii} = 0 \tag{16}$$

for $i, j = 1, 2, \cdots, n$ and $\boldsymbol{P}^{ij}$ denotes an modification of $\boldsymbol{P} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_n)$ with elimination of its i-th and j-th columns. For a well-orthonormalized matrix $\boldsymbol{B}$, its columns should satisfy

$$\boldsymbol{r}_j \times (\boldsymbol{r}_1 \times \cdots \times \boldsymbol{r}_{j-1} \times \boldsymbol{r}_{j+1} \times \cdots \times \boldsymbol{r}_n) = \boldsymbol{0} \tag{17}$$

i.e. one column should be collinear with the cross-product, such that the sign is to be determined

$$\boldsymbol{r}_j = s_j \cdot (\boldsymbol{r}_1 \times \cdots \times \boldsymbol{r}_{j-1} \times \boldsymbol{r}_{j+1} \times \cdots \times \boldsymbol{r}_n) \tag{18}$$

By symbolic computation engines using MATLAB or Mathematica, it is rather easy for us to obtain the empirical law $s_j = (-1)^{j+1}$. Another key problem is to find out a proper formulation of the ρ_k so that during the iterations, (14) will be globally stable with arbitrary initial conditions. Here we assume that the norms of the columns of $\boldsymbol{B}$ are less than or equal to 1, i.e., $\|\boldsymbol{r}_j\| \le 1, \quad j = 1, 2, \cdots, n$. This can be achieved by pre-dividing the maximum absolute value of matrix $\boldsymbol{B}$ to itself, i.e.,

$$\boldsymbol{B} = \frac{1}{n \max(|\boldsymbol{B}|)} \boldsymbol{B} \tag{19}$$

We further propose the following ϱ_k

$$\varrho_k = (n-1)/(n - 2 + \sum_{j=1}^{n} \|\boldsymbol{r}_{j,k-1}\|^2) \tag{20}$$

This choice is able to maintain the global convergence of the problem (3). Furthermore, seen from all related equations, there is no algebraic operation other than addition, subtraction, multiplication and division. That is to say, all the operations are rational, which guarantees the simple implementation of the iterative mechanism (14). Taking the 4-D orthonormalization as an example, if we write $\boldsymbol{B}$ into

$$\boldsymbol{B} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & R_{14} \\ R_{21} & R_{22} & R_{23} & R_{24} \\ R_{31} & R_{32} & R_{33} & R_{34} \\ R_{41} & R_{42} & R_{43} & R_{44} \end{pmatrix} \tag{21}$$

the orthonormal error terms can be computed symbolically in (23). From the result, it is able for us to observe that the mechanism can be depicted as a circuit system, which is shown in Fig. 1. In the designed circuit, the red rectangle specifies the 3-D part when the 4-D rotation degenerates to

$$\boldsymbol{B} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{22}$$

$$\boldsymbol{r}_1^{\otimes} = \begin{bmatrix} R_{44}(R_{22}R_{33} - R_{23}R_{32}) - R_{34}(R_{22}R_{43} - R_{23}R_{42}) + R_{24}(R_{32}R_{43} - R_{33}R_{42}) \\ R_{34}(R_{12}R_{43} - R_{13}R_{42}) - R_{44}(R_{12}R_{33} - R_{13}R_{32}) - R_{14}(R_{32}R_{43} - R_{33}R_{42}) \\ R_{44}(R_{12}R_{23} - R_{13}R_{22}) - R_{24}(R_{12}R_{43} - R_{13}R_{42}) + R_{14}(R_{22}R_{43} - R_{23}R_{42}) \\ R_{24}(R_{12}R_{33} - R_{13}R_{32}) - R_{34}(R_{12}R_{23} - R_{13}R_{22}) - R_{14}(R_{22}R_{33} - R_{23}R_{32}) \end{bmatrix}$$
$$\boldsymbol{r}_2^{\otimes} = \begin{bmatrix} R_{44}(R_{21}R_{33} - R_{23}R_{31}) - R_{34}(R_{21}R_{43} - R_{23}R_{41}) + R_{24}(R_{31}R_{43} - R_{33}R_{41}) \\ R_{34}(R_{11}R_{43} - R_{13}R_{41}) - R_{44}(R_{11}R_{33} - R_{13}R_{31}) - R_{14}(R_{31}R_{43} - R_{33}R_{41}) \\ R_{44}(R_{11}R_{23} - R_{13}R_{21}) - R_{24}(R_{11}R_{43} - R_{13}R_{41}) + R_{14}(R_{21}R_{43} - R_{23}R_{41}) \\ R_{24}(R_{11}R_{33} - R_{13}R_{31}) - R_{34}(R_{11}R_{23} - R_{13}R_{21}) - R_{14}(R_{21}R_{33} - R_{23}R_{31}) \end{bmatrix}$$
$$\boldsymbol{r}_3^{\otimes} = \begin{bmatrix} R_{44}(R_{21}R_{32} - R_{22}R_{31}) - R_{34}(R_{21}R_{42} - R_{22}R_{41}) + R_{24}(R_{31}R_{42} - R_{32}R_{41}) \\ R_{34}(R_{11}R_{42} - R_{12}R_{41}) - R_{44}(R_{11}R_{32} - R_{12}R_{31}) - R_{14}(R_{31}R_{42} - R_{32}R_{41}) \\ R_{44}(R_{11}R_{22} - R_{12}R_{21}) - R_{24}(R_{11}R_{42} - R_{12}R_{41}) + R_{14}(R_{21}R_{42} - R_{22}R_{41}) \\ R_{24}(R_{11}R_{32} - R_{12}R_{31}) - R_{34}(R_{11}R_{22} - R_{12}R_{21}) - R_{14}(R_{21}R_{32} - R_{22}R_{31}) \end{bmatrix} \qquad (23)$$
$$\boldsymbol{r}_4^{\otimes} = \begin{bmatrix} R_{43}(R_{21}R_{32} - R_{22}R_{31}) - R_{33}(R_{21}R_{42} - R_{22}R_{41}) + R_{23}(R_{31}R_{42} - R_{32}R_{41}) \\ R_{33}(R_{11}R_{42} - R_{12}R_{41}) - R_{43}(R_{11}R_{32} - R_{12}R_{31}) - R_{13}(R_{31}R_{42} - R_{32}R_{41}) \\ R_{43}(R_{11}R_{22} - R_{12}R_{21}) - R_{23}(R_{11}R_{42} - R_{12}R_{41}) + R_{13}(R_{21}R_{42} - R_{22}R_{41}) \\ R_{23}(R_{11}R_{32} - R_{12}R_{31}) - R_{33}(R_{11}R_{22} - R_{12}R_{21}) - R_{13}(R_{21}R_{32} - R_{22}R_{31}) \end{bmatrix}$$

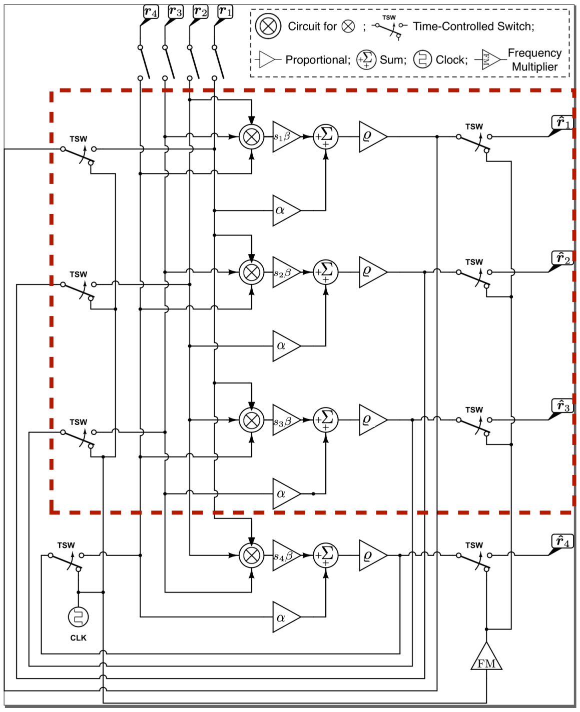

Fig. 1. The 4-D rotation orthonormalization circuit. The red rectangle specifies the sub-circuit of 3-D rotation orthonormalization.

The design shows that there is no need to generate new circuit models when encountering a shift from a higher dimension to a lower dimension. Thus, the designed circuit will be flexible for multi-dimensional orthonormalization.

III. Applications

A. *Rational* SO(n) *Network*

The iterative algorithm proposed in (14) can be treated as a recursive network as shown in Fig. 1. We name this single-layered orthonormalization network as COmpletely Rational Othornomalizer (CORO). Multiple COROs can be cascaded to form a multi-layer perceptron (MLP) from real space $\mathbb{R}^{n\times n}$ to n-dimensional rotation space SO(n). Fig. 2 shows the structure of this SO(n) MLP. The cascaded mechanism can be tuned as a weighted structure. Furthermore, we may add activation functions to the network, such that the i-th layer of the CORO neural network (CORO-NN) is

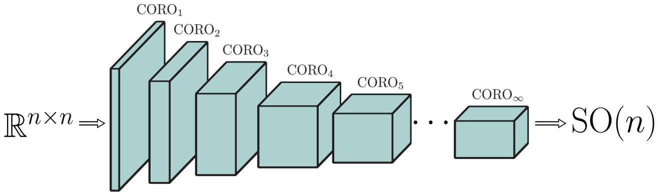
CORO$_1$
CORO$_2$
CORO$_3$
CORO$_4$
CORO$_5$
CORO$_\infty$
$\mathbb{R}^{n\times n} \Longrightarrow$
$\Longrightarrow \mathrm{SO}(n)$

Fig. 2. The structure of a multi-layer version of cascaded CORO as an SO(n) rotation learning machine.

$$\begin{cases} \boldsymbol{r}_{1,i} = \varrho_k \left[\tilde{\alpha}\boldsymbol{r}_{1,i-1} + s_1\tilde{\beta}\zeta\left(\boldsymbol{r}_{1,i-1}^{\otimes}\right)\right] \\ \boldsymbol{r}_{2,i} = \varrho_k \left[\tilde{\alpha}\boldsymbol{r}_{2,i-1} + s_2\tilde{\beta}\zeta\left(\boldsymbol{r}_{2,i-1}^{\otimes}\right)\right] \\ \qquad\qquad\vdots \\ \boldsymbol{r}_{n,i} = \varrho_k \left[\tilde{\alpha}\boldsymbol{r}_{n,i-1} + s_3\tilde{\beta}\zeta\left(\boldsymbol{r}_{n,i-1}^{\otimes}\right)\right] \end{cases} \qquad (24)$$

where $\zeta(\cdot)$ denotes the activation function to be chosen; $\tilde{\alpha}$ and $\tilde{\beta}$ act as two equivalent weights to be determined by the training process. Since CORO-NN (24) is strictly rational, the differential dynamics will also be explicit. Therefore, training such a network follows conventional ones of classical MLPs.

B. *Fast Rational Quaternion Normalization*

For an arbitrary unnormalized quaternion $\boldsymbol{q} \in \mathbb{R}^4$, the normalization is conducted as $\boldsymbol{q} = \boldsymbol{q}/\sqrt{\boldsymbol{q}^\top\boldsymbol{q}}$, which requires square-root operation. In FPGA implementation, the floating-number processing is not easy, especially for square-root computation. Some techniques have been proposed to accelerate this step on embedded platforms [17], [18]. In this paper, based on the orthonormalization (14), we will propose a new rational quaternion normalization method. In fact, for one quaternion $\boldsymbol{q} = (q_0, q_1, q_2, q_3)^\top$, the left and right quaternion multiplication can be achieved via the post matrix multiplication [19]. $\boldsymbol{R}_L$ and $\boldsymbol{R}_R$ are denoted as the left and right quaternion matrix of $\boldsymbol{q}$ respectively. For one normalized $\boldsymbol{q}$ satisfying $\boldsymbol{q}^\top\boldsymbol{q} = 1$, it can be verified that both $\boldsymbol{R}_L$ and $\boldsymbol{R}_R$ are on SO(4). Therefore, normalizing $\boldsymbol{q}$

is tantamount to orthonormalizing improper $\boldsymbol{R}_L$ and $\boldsymbol{R}_R$ to $\mathrm{SO}(4)$. By inserting matrix elements into (14), the following rational quaternion normalizer can be obtained

$$\boldsymbol{q}_k = \frac{5 + \mathfrak{N}_{k-1}}{2 + 4\mathfrak{N}_{k-1}} \boldsymbol{q}_{k-1} \tag{25}$$

where $\mathfrak{N}_{k-1} = \boldsymbol{q}_{k-1}^\top \boldsymbol{q}_{k-1}$. By first applying large-number suppression of (19), conducting (25) for only 3 times will give very accurate quaternion normalization results.

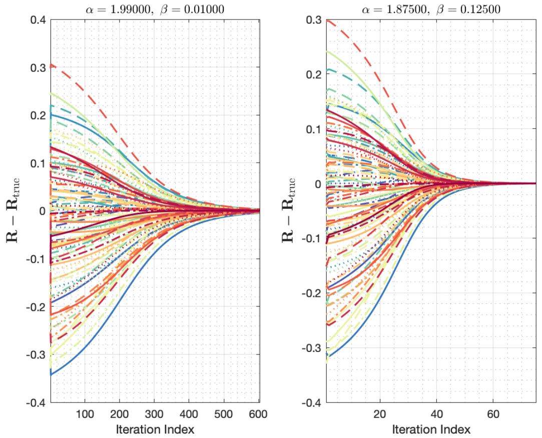

Fig. 3. The convergence performances of orthonormalization of $\mathrm{SO}(10)$ rotation with different combinations of α and β. Each line represents an error matrix element of $\boldsymbol{R} - \boldsymbol{R}_{\mathrm{true}}$.

IV. Experimental Results

A. Simulation Results

We simulate multiple orthonormalization cases under different dimensions. The true orthonormalized result can be obtained by using the SVD solver in (5). In the first case, a random matrix $\boldsymbol{B}$ of the size 10×10 is produced. Then it is orthonormalized to $\mathrm{SO}(n)$ via the proposed method. Via empirical tests, we find out that if $\alpha + \beta = 2$, the orthonormalizer (14) is always stable. We also observe that when $\beta = 1/(n-2)$, the statistical convergence rate becomes the best. In Fig. 3, the convergence results with different parameter combinations are shown. It can be seen that different pairs of α and β lead to quite different convergence rates. However, no matter how α and β vary, the convergence is always exponential-like. Next, we choose the cases of $\mathrm{SO}(6), \mathrm{SO}(8), \mathrm{SO}(10)$ orthonormalization for inspection. Three different $\boldsymbol{B}$ matrices are generated of sizes $6 \times 6, 8 \times 8, 10 \times 10$. Using the Jacobi SVD method and the proposed algorithm, the convergence results are presented in Fig. 4. It can be visualized that the Jacobi SVD solver does not have smooth convergence curves. In contrast, the proposed method achieves smooth and asymptotic convergence performance. Using the empirical optimal pair of α and β, we may see that the consumed iteration numbers for zeroing errors of the proposed method are lower than that of the Jacobi SVD. A detailed Frobenius-norm convergence plot of another $\mathrm{SO}(16)$ case is shown in Fig. 5. The Frobenius-norm convergence rates also reflect that the proposed method is superior to Jacobi SVD. This shows that the new rational mechanism may be more suitable for a simple implementation of $\mathrm{SO}(n)$ orthonormalization.

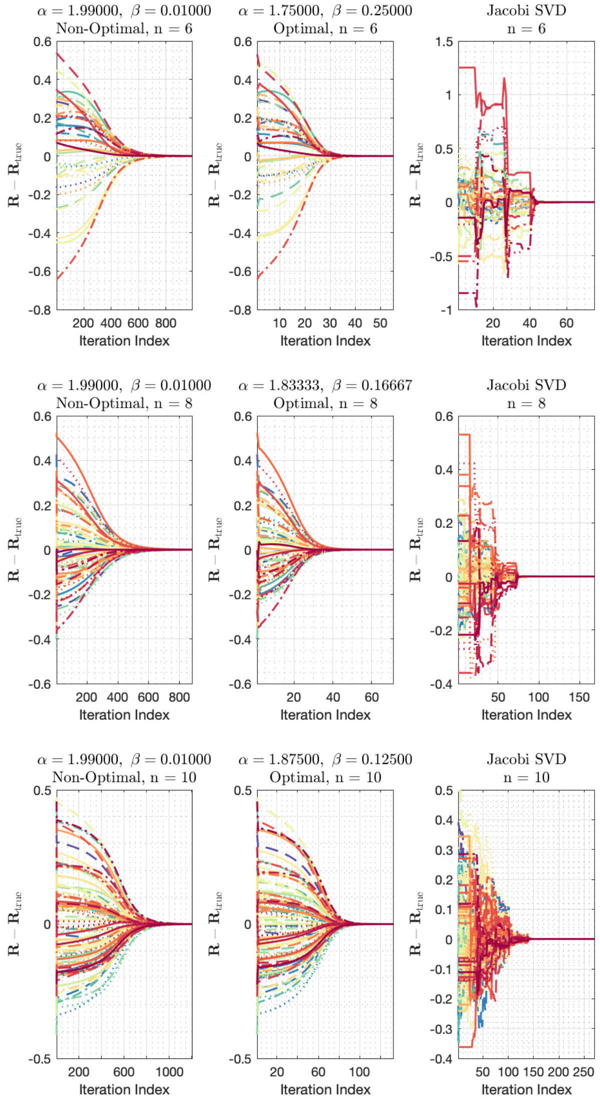

Fig. 4. The convergence performances of orthonormalization of $\mathrm{SO}(6), \mathrm{SO}(8), \mathrm{SO}(10)$ rotations with different combinations of α and β. Each line represents an error matrix element of $\boldsymbol{R} - \boldsymbol{R}_{\mathrm{true}}$.

Considering the rational quaternion orthonormalization developed in (25), a simulation study has been performed. The open-source codes are available[1]. Via the implementation using C++, the simulation has been conducted on a computer with an i7-4core processor. Compared with builtin `sqrt` function of `gcc` library and the fast `invsqrt` algorithm [18], the run-time stats indicate that the proposed method is at least 1.21 times faster than these representatives, with $< 10^{-11}$ accuracy loss with respect to the true values. This shows that rational normalizer indeed brings convenience in scientific computing.

B. Experimental Results: Hand-eye Calibration

We perform a hand-eye calibration work to verify the effectiveness of the proposed orthonormalizer. The experimen-

[1] `https://github.com/zarathustr/quat_norm`

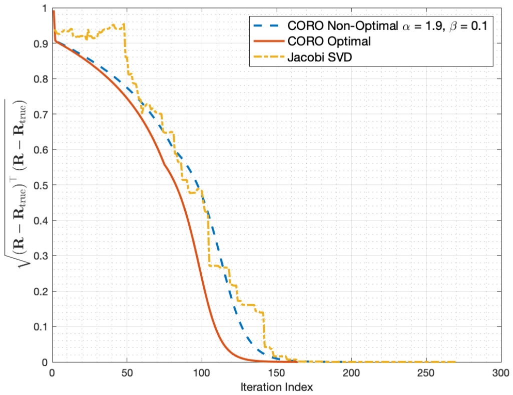

Fig. 5. The Frobenius-norm convergence performances of orthonormalization of a $\mathrm{SO}(16)$ case.

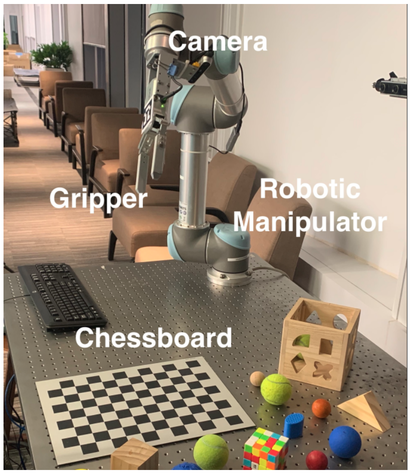

Fig. 6. The experimental setup of the hand-eye calibration for an industrial robot and attached camera.

tal setup is shown in Fig. 6, where an UR-5 industrial robotic manipulator, an Intel Realsense D435 camera, along with a robotic gripper are installed rigidly together on the table. The hand-eye calibration aims to compute the transformation between the gripper (hand) and camera (eye), so that the visual-aided automatic manipulation can be conducted. The hand-eye calibration is commonly formulated as

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B} \tag{26}$$

where $\boldsymbol{A}$ denotes the relative transformation of the gripper from forward kinematics, $\boldsymbol{B}$ is the relative transformation of the camera from perspective-n-points (PnP) and $\boldsymbol{X}$ is the unknown hand-eye transformation (see [20], [21] for details). But this single equation does not guarantee solvable $\boldsymbol{X}$. Therefore, in engineering, stacking measurements in different poses can give accurate estimate of $\boldsymbol{X}$, namely, this can be regarded as an optimization problem on $\mathrm{SE}(3)$, such that

$$\mathop{\arg\min}_{\boldsymbol{X}\in\mathrm{SE}(3)} \sum_{i=1}^{N} \mathrm{tr}\left[(\boldsymbol{A}_i\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B}_i)^\top (\boldsymbol{A}_i\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B}_i)\right] \tag{27}$$

To design an online estimator for (27), we treat (26) as an $\mathrm{SE}(3)$ Sylvester equation and it has the following time-differential form:

TABLE I

Comparison for Hand-eye Calibration

Algorithm	Loss Function	Computation Time	Success Rate
Method in [4]	0.09886	0.06709s	28.2%
Method in [22]	0.33043	0.41395s	33.7%
Proposed method	0.065569	0.139807	65.4%

$$\dot{\boldsymbol{A}}\boldsymbol{X} + \boldsymbol{A}\dot{\boldsymbol{X}} = \dot{\boldsymbol{X}}\boldsymbol{B} + \boldsymbol{X}\dot{\boldsymbol{B}} \tag{28}$$

in which $\dot{()}$ denotes the time-derivative. Defining the error term as $\boldsymbol{E}(\boldsymbol{X}, t, \boldsymbol{A}, \boldsymbol{B}) = \boldsymbol{A}\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B}$, we can obtain the error time-differential equation as

$$\frac{\mathrm{d}\boldsymbol{E}(\boldsymbol{X}, t, \boldsymbol{A}, \boldsymbol{B})}{\mathrm{d}t} = -\boldsymbol{E}(\boldsymbol{X}, t, \dot{\boldsymbol{A}}, \dot{\boldsymbol{B}}) \tag{29}$$

where $\dot{\boldsymbol{A}}, \dot{\boldsymbol{B}}$ are considered to be measurable. The ordinary differential equation (ODE) in (29) has a specialty that all input and output are connected in a rational manner. Therefore, we transform (29) into a rational deterministic network and then cascade one CORO network proposed in (14) to obtain orthonormalized output. As the input and output of such a cascaded network are connected recursively in time, the designed network is also recursive. We use the Euler method to solve the differential cascaded network.

Previously, gradient methods are selected for direct $\mathrm{SE}(3)$ solution to (27) [4], [22]. As both the objective and constraints in (27) are nonlinear, the solutions in [4], [22] are sensitive to the selection of initial guess. After grabbing 6 pairs of $\boldsymbol{A}, \boldsymbol{B}$, we compare the proposed method with that in [4], [22]. We generate 20 initial guess values on $\mathrm{SE}(3)$ via random sampling. These values are put into the use for the two methods. The computed transformation with least loss function value will be considered as the best one. We summarize the working loss function values, computational time and success rate of these methods in Table I. The results show that the proposed method owns the least loss function value with highest success rate. However, since the proposed method needs precise integral of the cascaded differential network, the computational time largely depends on the integration method, which is higher that method from [4]. Therefore, an appropriate integral method will be investigated in further studies.

V. Conclusions

We revisit the rotation orthonormalization problem for n-dimensional space. A new completely rational framework is proposed for solving this problem. Theoretical results show that the new mechanism is completely rational, thus making the implementation much easier on embedded platforms. Simulation studies reflect that the proposed method has a smoother convergence rate and is superior to the previous SVD-based solver. Experimental results on hand-eye calibration show the effectiveness of the proposed CORO network. In the future, we expect to use the new $\mathrm{SO}(n)$ orthonormalizer to solve existing essential robotics problems, where theoretical global convergence analysis will be performed.

REFERENCES

[1] S. Sarabandi, A. Shabani, J. M. Porta, and F. Thomas, "On Closed-Form Formulas for the 3-D Nearest Rotation Matrix Problem," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1333–1339, 2020.

[2] F. Thomas, "Approaching dual quaternions from matrix algebra," *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1037–1048, 2014.

[3] J. Wu, Y. Sun, M. Wang, and M. Liu, "Hand-Eye Calibration: 4-D Procrustes Analysis Approach," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 2966–2981, 2020.

[4] J. Wu, M. Liu, Y. Huang, C. Jin, Y. Wu, and C. Yu, "SE(n)++: An Efficient Solution to Multiple Pose Estimation Problems," *IEEE Trans. Cybern.*, pp. 1–12, 2020.

[5] O. Özyeşil, N. Sharon, and A. Singer, "Synchronization over Cartan Motion Groups via Contraction," *SIAM J. Appl. Algebr. Geom.*, vol. 2, no. 2, pp. 207–241, 2018.

[6] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group," *Int. J. Rob. Res.*, vol. 38, no. 2-3, pp. 95–125, 2019.

[7] S. Sarabandi and F. Thomas, "Approximating displacements in $\mathbb{R}^3$ by rotations in $\mathbb{R}^4$ and its application to pointcloud registration," *IEEE Trans. Robotics*, 2021.

[8] J. Wu, "Rigid 3-D Registration: A Simple Method Free of SVD and Eigendecomposition," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 8288–8303, 2020.

[9] J. Wu, Y. Jiang, C. Li, R. Sun, C. Zhang, Y. Yu, Y. Zhu, and M. Liu, "Circuit Synthesis of 3-D Rotation Orthonormalization," *IEEE Trans. Circuits Syst. II Express Briefs*, no. 2, pp. 1–1, 2020.

[10] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[11] S. Umeyama, "Least-Squares Estimation of Transformation Parameters Between Two Point Patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, 1991.

[12] T. Birdal, M. Arbel, U. Simsekli, and L. J. Guibas, "Synchronizing probability measures on rotations via optimal transport," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1566–1576, 2020.

[13] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent, "Optimal mass transport for registration and warping," *Int. J. Comput. Vis.*, vol. 60, no. 3, pp. 225–240, 2004.

[14] G. To and M. R. Mahfouz, "Quaternionic attitude estimation for robotic and human motion tracking using sequential monte carlo methods with von mises-fisher and nonuniform densities simulations," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 11, pp. 3046–3059, 2013.

[15] J. Wu, M. Liu, Z. Zhou, and R. Li, "Fast Symbolic 3-D Registration Solution," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 2, pp. 761–770, 2020.

[16] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[17] J. F. Blinn, "Floating-Point Tricks," *IEEE Comput. Graph. Appl.*, no. August, pp. 80–84, 1997.

[18] L. V. Moroz, C. J. Walczyk, A. Hrynchyshyn, V. Holimath, and J. L. Cieśliński, "Fast calculation of inverse square root with the use of magic constant – analytical approach," *Appl. Math. Comput.*, vol. 316, pp. 245–255, 2018.

[19] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[20] J. Wu, Y. Zheng, Z. Gao, Y. Jiang, X. Hu, Y. Zhu, J. Jiao, and M. Liu, "Quadratic pose estimation problems: Globally optimal solutions, solvability/observability analysis, and uncertainty description," *IEEE Trans. Robotics*, 2022.

[21] J. Wu, M. Liu, Y. Zhu, Z. Zou, M.-Z. Dai, C. Zhang, Y. Jiang, and C. Li, "Globally optimal symbolic hand-eye calibration," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 3, pp. 1369–1379, 2020.

[22] S. Gwak, J. Kim, and F. C. Park, "Numerical optimization on the euclidean group with applications to camera calibration," *IEEE Trans. Robot. Autom.*, vol. 19, pp. 65–74, 2003.