Robotica (2022), **40**, pp. 2409–2427
doi:10.1017/S0263574721001715

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Vectorial parameterizations of pose

Timothy D. Barfoot[1,*], James R. Forbes[2] and Gabriele M. T. D'Eleuterio[1]

[1]University of Toronto Institute for Aerospace Studies, Toronto, Canada and [2]Department of Mechanical Engineering, McGill University, Montreal, Canada
*Corresponding author. E-mail: tim.barfoot@utoronto.ca

Received: 11 June 2021; **Revised:** 8 September 2021; **Accepted:** 29 October 2021; **First published online:** 10 December 2021

Keywords: matrix Lie groups, state estimation, vector representation of rotation, pose

Abstract

Robotics and computer vision problems commonly require handling rigid-body motions comprising translation and rotation – together referred to as *pose*. In some situations, a *vectorial* parameterization of pose can be useful, where elements of a vector space are surjectively mapped to a matrix Lie group. For example, these vectorial representations can be employed for optimization as well as uncertainty representation on groups. The most common mapping is the matrix exponential, which maps elements of a Lie algebra onto the associated Lie group. However, this choice is not unique. It has been previously shown how to characterize all such vectorial parameterizations for $SO(3)$, the group of rotations. Some results are also known for the group of poses, where it is possible to build a family of vectorial mappings that includes the matrix exponential as well as the Cayley transformation. We extend what is known for these pose mappings to the 4×4 representation common in robotics and also demonstrate three different examples of the proposed pose mappings: (i) pose interpolation, (ii) pose servoing control, and (iii) pose estimation in a pointcloud alignment problem. In the pointcloud alignment problem, our results lead to a new algorithm based on the Cayley transformation, which we call *CayPer*.

1. Introduction

Describing rigid-body motion is a fundamental concept in robotics, computer vision, computer graphics, augmented reality, classical mechanics, and beyond. From Mozzi [1] and Chasles [2], we have the *Mozzi–Chasles theorem*, which says that the most general rigid-body displacement can be produced by a translation along a line (the *screw axis*) followed by (or preceded by) a rotation about that line. A complete screw theory was later developed Ball [3]. Murray et al. [4] popularized the modern approach to describing rigid-body motion (in the context of manipulator robotics) using the *special Euclidean group*, where screw motions are elegantly described by mapping elements of the Lie algebra onto the Lie group through the matrix exponential. It is through this sequence of historical events that multiple communities have come to think of the matrix exponential as the *canonical* way to map a six-degree-of-freedom *vector* to a rigid-body *pose*. Optimization and uncertainty representation for pose quantities now also commonly make use of matrix Lie group tools and the exponential map [5–7].

Yet, there are many representations of three-dimensional *rotations* in active use today: Euler angles, unit-length quaternions (a.k.a., Euler parameters), rotation vectors, Cayley–Gibbs–Rodrigues parameters, modified Rodrigues parameters, and many more [8]. Of these, only the *rotation vector* (i.e., angle multiplied with a unit-length axis) maps to an element of the *special orthogonal group* (i.e., a rotation matrix) through the matrix exponential. However, Bauchau and Trainelli [9] synthesized the class of *vectorial* parameterizations of rotation. In particular, they elegantly describe a general mapping from a three-degree-of-freedom vector to a rotation matrix that encompasses many commonly employed representations of rotation, including Cayley–Gibbs–Rodrigues and modified Rodrigues parameters.

© The Author(s), 2021. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

Bauchau and Choi [10] and later Bauchau and Li [11] expanded the idea of a general vector mapping to the special Euclidean group representing poses, where they focus on the 6×6 *adjoint* representation of pose, Ad($SE(3)$). Bauchau [12] provides an excellent summary of both the rotation and adjoint pose vectorial mappings. Interestingly, these works do not present the general vector mappings for the common 4×4 representation of pose, $SE(3)$, although some works do explore special cases [13,14]. We revisit the idea of a *vectorial parameterization of pose*, extending what is known in a few specific ways. First, we explore the more common 4×4 representation, $SE(3)$, showing that for every parameterization of rotation, there are infinite possibilities for the parameterization of pose having different properties from which one might choose; each has a different coupling between the translational and rotational variables. Second, we show that our general vector mapping provides the special cases of the exponential map and the *Cayley transformation* [15] for $SE(3)$, which has been previously discussed in the literature [13,14]. Finally, we hope that this paper serves to further shine a light on the important long-term efforts of Bauchau [12] on the vectorial parameterizations of rotations and poses, which we believe is of great interest to the robotics community.

Further to our extensions of the vectorial parameterization of pose, which is a theoretical contribution, we also demonstrate how to use the vectorial parameterization of pose in various canonical robotics applications. In particular, the classic problems of pose interpolation, pose control, and pose estimation in a pointcloud alignment framework are each addressed using different vectorial parameterizations of pose. We show using an alternative pose parameterization over the classic matrix exponential can lead to different results in terms of performance and robustness.

The paper is organized as follows. Section 2 provides some brief mathematical background. Section 3 introduces our vectorial parameterization of pose and discusses several properties thereof. While our main contribution is the theory, Section 4 provides three applications highlighting that different pose mappings may be beneficial in different situations. Section 5 concludes the paper.

2. Mathematical preliminaries

We briefly review some key concepts and notation [5] to prepare for what follows.

2.1. Matrix Lie groups

The *special orthogonal group*, representing rotations, is the set of valid rotation matrices:

$$SO(3) = \left\{ \mathbf{C} \in \mathbb{R}^{3\times 3} \mid \mathbf{C}\mathbf{C}^T = \mathbf{1}, \det \mathbf{C} = 1 \right\}, \tag{1}$$

where $\mathbf{1}$ is the identity matrix. It is common to map a vector, $\boldsymbol{\phi} \in \mathbb{R}^3$, to a rotation matrix, $\mathbf{C}$, through the matrix exponential,

$$\mathbf{C}(\boldsymbol{\phi}) = \exp\left(\boldsymbol{\phi}^\wedge\right), \tag{2}$$

where $(\,\cdot\,)^\wedge$ is the skew-symmetric operator,

$$\boldsymbol{\phi}^\wedge = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{pmatrix}^\wedge = \begin{pmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{pmatrix}. \tag{3}$$

The mapping is surjective only, meaning every $\mathbf{C}$ can be produced by many different values for $\boldsymbol{\phi}$.

The *special Euclidean group*, representing poses (i.e., translation and rotation), is the set of valid transformation matrices:

$$SE(3) = \left\{ \mathbf{T} = \begin{pmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{pmatrix} \in \mathbb{R}^{4\times 4} \;\middle|\; \mathbf{C} \in SO(3),\ \mathbf{r} \in \mathbb{R}^3 \right\}. \tag{4}$$

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

It is again common to map a vector, $\boldsymbol{\xi} \in \mathbb{R}^6$, to a transformation matrix, $\mathbf{T}$, through the matrix exponential,

$$\mathbf{T}(\boldsymbol{\xi}) = \exp\left(\boldsymbol{\xi}^{\wedge}\right), \tag{5}$$

where

$$\boldsymbol{\xi}^{\wedge} = \begin{pmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{pmatrix}^{\wedge} = \begin{pmatrix} \boldsymbol{\phi}^{\wedge} & \boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{pmatrix}. \tag{6}$$

As is common practice [5], we have broken the pose vector, $\boldsymbol{\xi}$, into a translational component, $\boldsymbol{\rho}$, and a rotational component, $\boldsymbol{\phi}$. The mapping is also surjective only, meaning every $\mathbf{T}$ can be produced by many different values for $\boldsymbol{\xi}$.

Finally, the *adjoint* of pose is given by

$$\mathcal{T}(\boldsymbol{\xi}) = \mathrm{Ad}\left(\mathbf{T}\right) = \begin{pmatrix} \mathbf{C}(\boldsymbol{\phi}) & \mathbf{r}^{\wedge}\mathbf{C}(\boldsymbol{\phi}) \\ \mathbf{0} & \mathbf{C}(\boldsymbol{\phi}) \end{pmatrix}, \tag{7}$$

which is now 6×6. We will refer to the set of adjoints as $\mathrm{Ad}(SE(3))$. We can map a vector, $\boldsymbol{\xi} \in \mathbb{R}^6$, to an adjoint transformation matrix again through the matrix exponential map:

$$\mathcal{T}(\boldsymbol{\xi}) = \exp\left(\boldsymbol{\xi}^{\curlywedge}\right), \tag{8}$$

where

$$\boldsymbol{\xi}^{\curlywedge} = \begin{pmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{pmatrix}^{\curlywedge} = \begin{pmatrix} \boldsymbol{\phi}^{\wedge} & \boldsymbol{\rho}^{\wedge} \\ \mathbf{0} & \boldsymbol{\phi}^{\wedge} \end{pmatrix}. \tag{9}$$

The mapping is again surjective only, meaning every $\mathcal{T}$ can be produced by many different values for $\boldsymbol{\xi}$.

2.2. Key identities

We will be exploring mappings (including the exponential map) from vectors to Lie groups and will have occasion to work with series expressions of rotation and transformation matrices. Each group has associated with it an identity that can be used to limit the number of terms in such series, related to the Cayley–Hamilton theorem.

For rotations, we have that the characteristic equation of $\boldsymbol{\phi}^{\wedge}$ is

$$\left|\lambda\mathbf{1} - \boldsymbol{\phi}^{\wedge}\right| = \lambda(\lambda^2 + \underbrace{\phi_1^2 + \phi_2^2 + \phi_3^2}_{\phi^2}) = \lambda^3 + \phi^2\lambda = 0, \tag{10}$$

where λ are the eigenvalues of $\boldsymbol{\phi}^{\wedge}$. From the Cayley–Hamilton theorem, we can claim that

$$\boldsymbol{\phi}^{\wedge^3} + \phi^2\boldsymbol{\phi}^{\wedge} \equiv \mathbf{0}, \tag{11}$$

since $\boldsymbol{\phi}^{\wedge}$ must satisfy its own characteristic equation.

For poses, we have that the characteristic equation of $\boldsymbol{\xi}^{\wedge}$ is

$$\left|\lambda\mathbf{1} - \boldsymbol{\xi}^{\wedge}\right| = \begin{vmatrix} \lambda\mathbf{1} - \boldsymbol{\phi}^{\wedge} & -\boldsymbol{\rho} \\ \mathbf{0}^T & \lambda \end{vmatrix} = \lambda^4 + \phi^2\lambda^2 = 0. \tag{12}$$

Again, from the Cayley–Hamilton theorem we can claim that

$$\boldsymbol{\xi}^{\wedge^4} + \phi^2\boldsymbol{\xi}^{\wedge^2} \equiv \mathbf{0}, \tag{13}$$

since $\boldsymbol{\xi}^{\wedge}$ must satisfy its own characteristic equation.

Finally, for adjoint poses we have that the characteristic equation of $\boldsymbol{\xi}^{\curlywedge}$ is

$$\left|\lambda\mathbf{1} - \boldsymbol{\xi}^{\curlywedge}\right| = \begin{vmatrix} \lambda\mathbf{1} - \boldsymbol{\phi}^{\wedge} & -\boldsymbol{\rho}^{\wedge} \\ \mathbf{0} & \lambda\mathbf{1} - \boldsymbol{\phi}^{\wedge} \end{vmatrix} = \left(\lambda^3 + \phi^2\lambda\right)^2 = \lambda^6 + 2\phi^2\lambda^4 + \phi^4\lambda^2 = 0. \tag{14}$$

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

We could again employ the Cayley–Hamilton theorem to create a similar identity to the other two cases, but it turns out that the *minimal polynomial* of $\boldsymbol{\xi}^\wedge$ is actually [5,16],

$$\boldsymbol{\xi}^{\wedge 5} + 2\phi^2\boldsymbol{\xi}^{\wedge 3} + \phi^4\boldsymbol{\xi}^\wedge \equiv \mathbf{0}, \tag{15}$$

which is one order lower than the characteristic equation in this case. This is because both the algebraic and geometric multiplicities of eigenvalue $\lambda = 0$ in (14) are two (two Jordan blocks of size one), allowing us to drop one copy when constructing the minimal polynomial; in (12), the algebraic multiplicity of $\lambda = 0$ is two, while the geometric multiplicity is one (one Jordan block of size two).

3. Vector mappings

In this section, we will show that there exist several options for mapping vectors onto the two most common matrix Lie groups used in robotics and computer vision. Our approach builds on the work of Bauchau and Trainelli [9], who worked this out for $SO(3)$, the group of rotations and Bauchau and Choi [10], Bauchau [12] who extended this work to $\mathrm{Ad}(SE(3))$, the adjoint representation of pose. We show how to extend the rotation work to $SE(3)$ directly.

3.1. Vector mappings for rotations

It is well known that there exist many different ways to parameterize the group of three-dimensional rotations, $SO(3)$. Naturally, such rotations have three degrees of freedom. Representations with three parameters are known to always have singularities, while representations with more than three parameters have constraints to keep the degrees of freedom at three [8,17]. Bauchau and Trainelli [9] synthesized previous work on three-parameter representations that can be considered *vectors*; they show that all such parameterizations can be written as

$$\boldsymbol{\phi} = \phi(\varphi)\,\mathbf{a} \in \mathbb{R}^3, \tag{16}$$

where $\mathbf{a}$ is the unit-length axis of rotation, φ the angle of rotation, and $\phi(\varphi)$ a *generating function*. The scalar generating function is an odd function of the rotation angle that satisfies

$$\lim_{\varphi \to 0} \frac{\phi(\varphi)}{\varphi} = \kappa, \tag{17}$$

with κ a real normalization constant. We will assume $\kappa = 1$ in what follows as this makes it easier to compare different generating functions directly. The generating function is essentially a nonlinear warping of one angle, φ, to new angle, ϕ.

Some common examples of vector parameterizations of rotation are

$$\underbrace{\boldsymbol{\phi} = \varphi\mathbf{a}}_{\text{rotation vector}}, \qquad \underbrace{\boldsymbol{\phi} = 2\tan\frac{\varphi}{2}\mathbf{a}}_{\text{Cayley-Gibbs-Rodrigues}}, \qquad \underbrace{\boldsymbol{\phi} = 4\tan\frac{\varphi}{4}\mathbf{a}}_{\text{modified Rodrigues}}, \tag{18}$$

where we have ensured $\kappa = 1$ in (17) by including normalization constants where necessary. Notably, the Euler angle sequences, a common three-parameter representation, do not fit the vector definition since they are not parallel to the axis of rotation. Vector parameterizations come along with singularities but these will not be a roadblock for the way in which we intend to use them. Figure 1 depicts several generating functions.

Bauchau and Trainelli [9] show that the corresponding rotation matrix, $\mathbf{C} \in SO(3)$, can be constructed from the general vector parameterization as follows:

$$\mathbf{C}(\boldsymbol{\phi}) = \mathbf{1} + \frac{\nu^2}{\epsilon}\boldsymbol{\phi}^\wedge + \frac{\nu^2}{2}\boldsymbol{\phi}^{\wedge 2}, \tag{19}$$

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

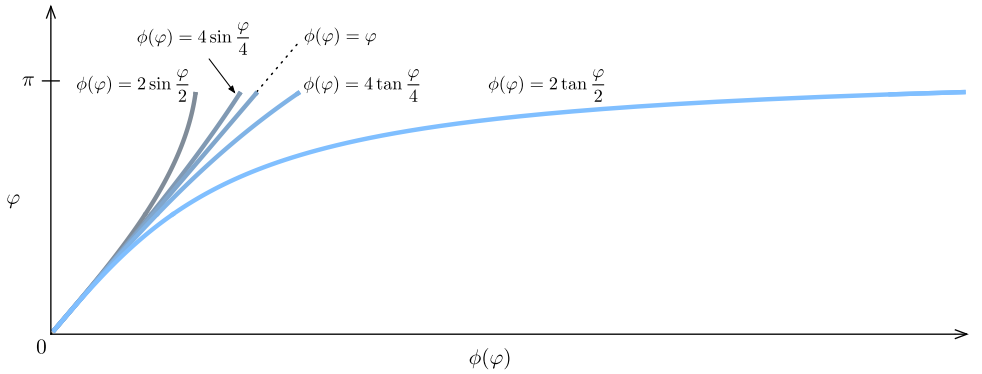


Figure 1. *Example generating functions, $\phi(\varphi)$, that map rotation angle, φ, to different vector representations of rotation and pose. The independent variable is shown on the vertical axis to save space. We see that the* $2\tan\frac{\varphi}{2}$ *function is quite a bit more extreme than the others.*

where

$$\nu(\varphi) = \frac{2\sin(\varphi/2)}{\phi(\varphi)}, \quad \epsilon(\varphi) = \frac{2\tan(\varphi/2)}{\phi(\varphi)}. \tag{20}$$

The group-mapped-from-vector function is a surjective-only mapping from $\mathbb{R}^3$ to $SO(3)$, which means every rotation matrix can be generated by many vectors (owing to the rotation angle wrapping around).

The expression for the $SO(3)$ Jacobian, $\mathbf{J}(\boldsymbol{\phi})$, that maps the derivative of the vector parameters to angular velocity,

$$\boldsymbol{\omega} = \mathbf{J}(\boldsymbol{\phi})\dot{\boldsymbol{\phi}}, \tag{21}$$

is given by ref. [9]

$$\mathbf{J}(\boldsymbol{\phi}) = \mu\mathbf{1} + \frac{\nu^2}{2}\boldsymbol{\phi}^\wedge + \frac{1}{\phi^2}\left(\mu - \frac{\nu^2}{\epsilon}\right)\boldsymbol{\phi}^{\wedge^2}, \tag{22}$$

where

$$\mu(\varphi) = \left(\frac{d\phi(\varphi)}{d\varphi}\right)^{-1}. \tag{23}$$

The inverse of this Jacobian is given by ref. [9]

$$\mathbf{J}(\boldsymbol{\phi})^{-1} = \frac{1}{\mu}\mathbf{1} - \frac{1}{2}\boldsymbol{\phi}^\wedge - \frac{1}{\phi^2}\left(\frac{1}{\epsilon} - \frac{1}{\mu}\right)\boldsymbol{\phi}^{\wedge^2}, \tag{24}$$

which we will require when dealing with $SE(3)$.

As the rotation angle, φ, becomes small, we have the following infinitesimal expressions:

$$\mathbf{C}(\boldsymbol{\phi}) \approx \mathbf{1} + \boldsymbol{\phi}^\wedge, \quad \mathbf{J}(\boldsymbol{\phi}) \approx \mathbf{1} + \frac{1}{2}\boldsymbol{\phi}^\wedge. \tag{25}$$

These will become important when considering perturbations.

The result from this section is general, but it will be instructive to consider some specific examples. Table 1 provides some common vector parameterizations and their associated mappings to $SO(3)$. The first three mappings are appealing in that they do not require calculating the angle of rotation, φ, and instead can be built directly from $\boldsymbol{\phi}$.

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

Table 1. *Example vector parameterizations of rotation and the associated mappings to SO(3). Note the connection to the Cayley transformation in the* tan *parameterizations. The entries in the table are ordered from steepest to shallowest generating function (right to left in Figure 1)*

	$\phi(\varphi)$	$\mathbf{C}(\boldsymbol{\phi})$	**Validity range**
Cayley–Gibbs–Rodrigues	$2\tan\frac{\varphi}{2}$	$\left(\mathbf{1}-\frac{1}{2}\boldsymbol{\phi}^\wedge\right)^{-1}\left(\mathbf{1}+\frac{1}{2}\boldsymbol{\phi}^\wedge\right)$	$\lvert\varphi\rvert<\pi$
Modified Rodrigues	$4\tan\frac{\varphi}{4}$	$\left(\mathbf{1}-\frac{1}{4}\boldsymbol{\phi}^\wedge\right)^{-2}\left(\mathbf{1}+\frac{1}{4}\boldsymbol{\phi}^\wedge\right)^{2}$	$\lvert\varphi\rvert<2\pi$
Rotation vector	φ	$\exp\left(\boldsymbol{\phi}^\wedge\right)$	$\lvert\varphi\rvert<2\pi$
Bauchau–Trainelli	$4\sin\frac{\varphi}{4}$	$\mathbf{1}+\cos\frac{\varphi}{2}\cos\frac{\varphi}{4}\boldsymbol{\phi}^\wedge+\frac{1}{2}\cos^2\frac{\varphi}{4}\boldsymbol{\phi}^{\wedge^2}$	$\lvert\varphi\rvert<2\pi$
Euler–Rodrigues	$2\sin\frac{\varphi}{2}$	$\mathbf{1}+\cos\frac{\varphi}{2}\boldsymbol{\phi}^\wedge+\frac{1}{2}\boldsymbol{\phi}^{\wedge^2}$	$\lvert\varphi\rvert<\pi$

3.2. Vector mappings for poses

We next extend the idea of generalized vector mappings to $SE(3)$. As pose changes have six degrees of freedom, our vector parameterization will take the form

$$\boldsymbol{\xi}=\begin{pmatrix}\boldsymbol{\rho}\\ \boldsymbol{\phi}\end{pmatrix}\in\mathbb{R}^6, \tag{26}$$

where the rotational parameters, $\boldsymbol{\phi}\in\mathbb{R}^3$, are the same as for $SO(3)$ and $\boldsymbol{\rho}\in\mathbb{R}^3$ are the translational parameters (not necessarily equivalent to the actual translation in Euclidean space, $\mathbf{r}$). We would like to generalize the $SO(3)$ result such that we have a surjective-only mapping from $\mathbb{R}^6$ to $SE(3)$.

Motivated by the Rodrigues-type form of (19) above, we define a general vector pose mapping to be of the form

$$\mathbf{T}(\boldsymbol{\xi})=\mathbf{1}+a\,\boldsymbol{\xi}^\wedge+b\,\boldsymbol{\xi}^{\wedge^2}+c\,\boldsymbol{\xi}^{\wedge^3}, \tag{27}$$

for some a, b, and c such that the result is an element of $SE(3)$. We know that higher-order powers of $\boldsymbol{\xi}^\wedge$ are unnecessary owing to (13), which comes from the Cayley–Hamilton theorem.

It is not immediately obvious that this definition of a pose vector parameterization for $SE(3)$ is consistent with Bauchau [12, Section 14.4.1], who notes that we can define such a mapping for $\mathrm{Ad}(SE(3))$ as one that is *parallel to an eigenvector of the resulting adjoint pose matrix associated with the unit eigenvalue*. It turns out that the two definitions are equivalent and we will demonstrate this shortly after first examining the conditions on a, b, and c to ensure $\mathbf{T}\in SE(3)$.

Using the definition of $\boldsymbol{\xi}^\wedge$, we can manipulate our pose mapping into the form,

$$\mathbf{T}(\boldsymbol{\xi})=\begin{pmatrix}\mathbf{C}(\boldsymbol{\phi}) & \mathbf{D}(\boldsymbol{\phi})\boldsymbol{\rho}\\ \mathbf{0}^T & 1\end{pmatrix}=\begin{pmatrix}\mathbf{1}+(a-\phi^2 c)\,\boldsymbol{\phi}^\wedge+b\,\boldsymbol{\phi}^{\wedge^2} & \left(a\mathbf{1}+b\,\boldsymbol{\phi}^\wedge+c\,\boldsymbol{\phi}^{\wedge^2}\right)\boldsymbol{\rho}\\ \mathbf{0}^T & 1\end{pmatrix}, \tag{28}$$

where we used (11) to simplify the top-left entry.

Looking to (4), the top-left entry in (28) must be the rotation matrix:

$$\mathbf{C}(\boldsymbol{\phi})=\mathbf{1}+(a-\phi^2 c)\,\boldsymbol{\phi}^\wedge+b\,\boldsymbol{\phi}^{\wedge^2}. \tag{29}$$

Comparing to (19), the following conditions must hold:

$$a-\phi^2 c=\frac{\nu^2}{\epsilon},\quad b=\frac{\nu^2}{2}. \tag{30}$$

The second condition leaves no choice, while the first provides some flexibility.

Turning to the top-right entry of (28), the matrix

$$\mathbf{D}(\boldsymbol{\phi})=a\mathbf{1}+b\,\boldsymbol{\phi}^\wedge+c\,\boldsymbol{\phi}^{\wedge^2}, \tag{31}$$

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

is the *coupling matrix* between the translational and rotational degrees of freedom. Because there is flexibility in the choice of a and c, this means there are an infinite number of possibilities for $\mathbf{D}(\boldsymbol{\phi})$ that satisfy our pose mapping definition in (27).

Under the conditions in (30), the defined properties of the generating function, and the further condition that $\mathbf{D}(\boldsymbol{\phi})$ must be invertible (except at some singularities), we can claim that our chosen mapping in (27) is surjective only from $\mathbb{R}^6$ to $SE(3)$. This is not too hard to see since every $\mathbf{C}(\boldsymbol{\phi})$ can be generated by many $\boldsymbol{\phi}$ and every $\mathbf{r}$ can be generated by some $\boldsymbol{\rho}$ provided $\mathbf{D}(\boldsymbol{\phi})$ invertible. The inverse map is discussed further in Section 3.6.

Returning to the question of consistency of our definition of pose vector parameterizations with that of Bauchau [12], we want to show that $\boldsymbol{\xi}$ *is* an eigenvector of $\mathcal{T} = \mathrm{Ad}(\mathbf{T})$ associated with a unit eigenvalue, that is,

$$\mathcal{T}\boldsymbol{\xi} = \boldsymbol{\xi}. \tag{32}$$

Filling in the details of $\mathcal{T}$ from (7), we have

$$\mathcal{T}\boldsymbol{\xi} = \begin{pmatrix} \mathbf{C} & (\mathbf{D}\boldsymbol{\rho})^{\wedge}\,\mathbf{C} \\ \mathbf{0} & \mathbf{C} \end{pmatrix} \begin{pmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{pmatrix} = \begin{pmatrix} \left(\mathbf{C} - \boldsymbol{\phi}^{\wedge}\mathbf{D}\right)\boldsymbol{\rho} \\ \boldsymbol{\phi} \end{pmatrix}, \tag{33}$$

where we have used $\mathbf{C}\boldsymbol{\phi} = \boldsymbol{\phi}$. Using (29), (31), and (11), we see that $\mathbf{C} - \boldsymbol{\phi}^{\wedge}\mathbf{D} = \mathbf{1}$ so that $\mathcal{T}\boldsymbol{\xi} = \boldsymbol{\xi}$. Thus, our $SE(3)$ definition of pose vector parameterizations is equivalent to that of Bauchau [12], for $\mathrm{Ad}(SE(3))$.

The nonuniqueness of the pose parameterization that we discussed earlier can also be explained from the eigenstructure, since there are two linearly independent eigenvectors for $\mathcal{T}$ associated with its two unit eigenvalues [12,16]; the $SE(3)$ pose vector parameterization therefore has a free parameter, as we showed by another means.

3.3. Screws, pose adjoints, and commutative mappings

As discussed in the previous section, we must choose values for the parameters, a, b, and c to construct the rotation matrix in (29). However, we have a free parameter in making this choice, as long as the parameters satisfy (30). One possibility is to choose

$$a = \mu, \quad b = \frac{\nu^2}{2}, \quad c = \frac{1}{\phi^2}\left(\mu - \frac{\nu^2}{\epsilon}\right), \tag{34}$$

which satisfy the conditions in (30) and importantly this particular choice means that we can write our vector pose parameter as [5, eq. 14.34]

$$\boldsymbol{\xi} = \phi(\varphi)\,\mathbf{s}, \tag{35}$$

where $\mathbf{s}$ is the constant *screw axis* of the Mozzi–Chasles theorem. The screw can be written in the form

$$\mathbf{s} = \begin{pmatrix} p\,\mathbf{a} + \mathbf{m}^{\wedge}\mathbf{a} \\ \mathbf{a} \end{pmatrix}, \tag{36}$$

where p is the (scalar) *pitch*, $\mathbf{a}$ is the (unit) axis of rotation, and $\mathbf{m}$ is the moment arm; $(\mathbf{m}^{\wedge}\mathbf{a}, \mathbf{a})$ are the *Plücker coordinates*. The relationship in (35) parallels the $SO(3)$ definition of ref. [9] captured in (16); we have the product of the generating function and a (screw) axis. In the case that $\phi(\varphi) = \varphi$, then $\boldsymbol{\xi} = \varphi\mathbf{s}$ is the familiar *twist*. We underscore that it is the specific selection of a and c in (34) that has made this connection; other choices for a and c will not result in a constant screw.

The selection of a and c in (34) means that the pose mapping becomes

$$\mathbf{T}(\boldsymbol{\xi}) = \mathbf{1} + \mu\,\boldsymbol{\xi}^{\wedge} + \frac{\nu^2}{2}\boldsymbol{\xi}^{\wedge^2} + \frac{1}{\phi^2}\left(\mu - \frac{\nu^2}{\epsilon}\right)\boldsymbol{\xi}^{\wedge^3}. \tag{37}$$

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

As the rotation angle, φ, becomes small, we have the infinitesimal expression,

$$\mathbf{T}(\boldsymbol{\xi}) \approx \mathbf{1} + \boldsymbol{\xi}^{\wedge}, \tag{38}$$

which will become important when considering perturbations.

Choosing a and c as in (34) also has the effect of selecting

$$\mathbf{D}(\boldsymbol{\phi}) = \mathbf{J}(\boldsymbol{\phi}), \tag{39}$$

the Jacobian of $SO(3)$ for the chosen parameterization [12], which can be verified by substituting into (22). Choosing $\mathbf{D}(\boldsymbol{\phi}) = \mathbf{J}(\boldsymbol{\phi})$ gives our $SE(3)$ mapping an important property related to pose *adjoints*. With these definitions, we can establish the following desirable commutative relationship:

$$\begin{array}{lccc} 4\times 4 & \boldsymbol{\xi}^{\wedge} \in \mathfrak{se}(3) & \xrightarrow{\text{mapping}} & \mathbf{T} \in SE(3) \\ & \Big\downarrow \text{ad} & & \Big\downarrow \text{Ad} \\ 6\times 6 & \boldsymbol{\xi}^{\curlywedge} \in \text{ad}(\mathfrak{se}(3)) & \xrightarrow{\text{mapping}} & \boldsymbol{\mathcal{T}} \in \text{Ad}(SE(3)) \end{array} \tag{40}$$

This means we can transform our vector parameterization of pose to its adjoint in two equivalent ways.

To verify this commutative property, we can write the series form for the adjoint mapping as

$$\boldsymbol{\mathcal{T}}(\boldsymbol{\xi}) = \mathbf{1} + d\,\boldsymbol{\xi}^{\curlywedge} + e\,\boldsymbol{\xi}^{\curlywedge 2} + f\,\boldsymbol{\xi}^{\curlywedge 3} + g\,\boldsymbol{\xi}^{\curlywedge 4}, \tag{41}$$

for some unknowns, d, e, f, g. We do not need higher-order terms in this Rodrigues-type series owing to (15), which can be used to reduce quintic and higher terms. Multiplying out and comparing terms between (7) and (41), we find that

$$d = a + \phi^2(f - c), \quad e = \phi^2 g + b, \quad f = \frac{1}{2}(ab - c), \quad g = \frac{1}{2}\left(b^2 - c(a - \phi^2 c)\right), \tag{42}$$

in terms of the earlier choices for a, b, and c. There may also be other choices for a and c (recall these are not unique) that impose the commutative relationship for $SE(3)$, but this was not explored further.

Owing to smoothness of the generating function and its convergence to φ in the limit of an infinitesimally small rotation, one might be able to argue that our vector mappings offer alternatives (to the exponential map) for the local diffeomorphism from the Lie algebra, $\mathfrak{se}(3)$, to the Lie group, $SE(3)$. In such a case, by changing the map we are merely "warping" how the Lie algebra maps onto the Lie group.

3.4. Cayley transformations

It is known [9,13] that for the Cayley–Gibbs–Rodrigues parameterization of rotation, $\phi(\varphi) = 2\tan\frac{\varphi}{2}$, we can write the rotation matrix in terms of the Cayley transformation:

$$\mathbf{C}(\boldsymbol{\phi}) = \left(\mathbf{1} - \frac{1}{2}\boldsymbol{\phi}^{\wedge}\right)^{-1}\left(\mathbf{1} + \frac{1}{2}\boldsymbol{\phi}^{\wedge}\right). \tag{43}$$

In fact, this can be generalized to the case where $\phi(\varphi) = 2m\tan\frac{\varphi}{2m}$ with m a positive integer using higher-order Cayley transformations [9]:

$$\mathbf{C}(\boldsymbol{\phi}) = \left(\mathbf{1} - \frac{1}{2m}\boldsymbol{\phi}^{\wedge}\right)^{-m}\left(\mathbf{1} + \frac{1}{2m}\boldsymbol{\phi}^{\wedge}\right)^{m}. \tag{44}$$

However, things become somewhat more complicated for poses.

Borri et al. [13] and later Selig [14] demonstrated that the Cayley transformation can be used to map pose vectors to $SE(3)$ according to

$$\mathbf{T}(\boldsymbol{\xi}) = \left(\mathbf{1} - \frac{1}{2m}\boldsymbol{\xi}^{\wedge}\right)^{-m}\left(\mathbf{1} + \frac{1}{2m}\boldsymbol{\xi}^{\wedge}\right)^{m}, \tag{45}$$

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

where we have generalized to higher-order transformations. Borri et al. [13] and Selig [14] also show that the Cayley transformation can be used to map pose vectors to Ad($SE(3)$) according to

$$\mathcal{T}(\boldsymbol{\xi}) = \left(\mathbf{1} - \frac{1}{2m}\boldsymbol{\xi}^{\curlywedge}\right)^{-m}\left(\mathbf{1} + \frac{1}{2m}\boldsymbol{\xi}^{\curlywedge}\right)^{m}, \tag{46}$$

where we have generalized again to higher-order transformations. Disappointingly, Selig [14] demonstrates that starting from the same $\boldsymbol{\xi}$ and applying (45) and (46) does not result in an equivalent transformation, that is, $\mathcal{T}(\boldsymbol{\xi}) \neq \mathrm{Ad}(\mathbf{T}(\boldsymbol{\xi}))$; the commutative property for adjoints does not hold.

We can actually say something more than Selig [14]. It turns out that (45) is *not* equivalent to (37), but (46) *is* equivalent to (41), for the family of rotation parameterizations, $\phi(\varphi) = 2m\tan\frac{\varphi}{2m}$. The explanation for this lies in the choice of $\mathbf{D}(\boldsymbol{\phi})$, discussed earlier. For the 6×6 case, choosing $\mathbf{D}(\boldsymbol{\phi}) = \mathbf{J}(\boldsymbol{\phi})$ results in the Cayley transformation, while for the 4×4 case it does not; to connect to the Cayley transformation in the 4×4 case for $m = 1$, we must pick $\mathbf{D}(\boldsymbol{\phi}) = \frac{1}{2}(\mathbf{C}(\boldsymbol{\phi}) + \mathbf{1}) \neq \mathbf{J}(\boldsymbol{\phi})$, as shown by Selig [14].

To show that (46) holds, consider the $m = 1$ case. If we left-multiply by the inverse of the first factor we should have

$$\left(\mathbf{1} - \frac{1}{2}\boldsymbol{\xi}^{\curlywedge}\right)\mathcal{T}(\boldsymbol{\xi}) = \mathbf{1} + \frac{1}{2}\boldsymbol{\xi}^{\curlywedge}. \tag{47}$$

Multiplying out the left side, we have

$$\mathbf{1} + \left(d - \frac{1}{2}(1-\phi^4 g)\right)\boldsymbol{\xi}^{\curlywedge} + \left(e - \frac{1}{2}d\right)\boldsymbol{\xi}^{\curlywedge^2} + \left(f - \frac{1}{2}(e - 2\phi^2 g)\right)\boldsymbol{\xi}^{\curlywedge^3} + \left(g - \frac{1}{2}f\right)\boldsymbol{\xi}^{\curlywedge^4} = \mathbf{1} + \frac{1}{2}\boldsymbol{\xi}^{\curlywedge}, \tag{48}$$

where we have made use of (15) once to reduce a quintic term. Comparing the coefficients of each term it must be that

$$d - \frac{1}{2}(1-\phi^4 g) = \frac{1}{2}, \quad e - \frac{1}{2}d = 0, \quad f - \frac{1}{2}(e - 2\phi^2 g) = 0, \quad g - \frac{1}{2}f = 0. \tag{49}$$

Substituting (42) and then (34) confirms these conditions for $\phi(\varphi) = 2\tan\frac{\varphi}{2}$.

3.5. Cayley-type factorizations

Following the previous section that focussed on the $\phi(\varphi) = 2m\tan\frac{\varphi}{2m}$ family, we next consider some novel Cayley-type mappings for different generating functions. To our knowledge, these have not been presented before.

It turns out that we can factor *any* of the vectorial rotation parameterizations according to

$$\mathbf{C}(\boldsymbol{\phi}) = \left(\mathbf{1} - \lambda\boldsymbol{\phi}^{\wedge}\right)^{-1}\left(\mathbf{1} + \gamma\boldsymbol{\phi}^{\wedge}\right) = \left(\mathbf{1} - \gamma\boldsymbol{\phi}^{\wedge}\right)^{-1}\left(\mathbf{1} + \lambda\boldsymbol{\phi}^{\wedge}\right), \tag{50}$$

where

$$\gamma = \frac{1}{2}\epsilon, \tag{51a}$$

$$\lambda = \frac{\nu^2}{\epsilon} - \frac{1}{2}\epsilon\left(1 - \frac{1}{2}\phi^2\nu^2\right). \tag{51b}$$

In the case that $\phi(\varphi) = 2\tan\frac{\varphi}{2}$, we recover $\gamma = \lambda = \frac{1}{2}$. The inverse of $\mathbf{C}(\boldsymbol{\phi})$ is obtained by flipping the signs in the factorization:

$$\mathbf{C}(\boldsymbol{\phi})^{-1} = \mathbf{C}(\boldsymbol{\phi})^{T} = \left(\mathbf{1} + \lambda\boldsymbol{\phi}^{\wedge}\right)^{-1}\left(\mathbf{1} - \gamma\boldsymbol{\phi}^{\wedge}\right) = \left(\mathbf{1} + \gamma\boldsymbol{\phi}^{\wedge}\right)^{-1}\left(\mathbf{1} - \lambda\boldsymbol{\phi}^{\wedge}\right). \tag{52}$$

The relationships can be verified by simply multiplying out and comparing to (19).

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

We can also factor some $SE(3)$ parameterizations in an analogous way:

$$\mathbf{T}(\boldsymbol{\xi}) = \left(\mathbf{1} - \lambda\boldsymbol{\xi}^{\wedge}\right)^{-1}\left(\mathbf{1} + \gamma\boldsymbol{\xi}^{\wedge}\right) = \left(\mathbf{1} - \gamma\boldsymbol{\xi}^{\wedge}\right)^{-1}\left(\mathbf{1} + \lambda\boldsymbol{\xi}^{\wedge}\right), \tag{53a}$$

$$\mathbf{T}(\boldsymbol{\xi})^{-1} = \left(\mathbf{1} + \lambda\boldsymbol{\xi}^{\wedge}\right)^{-1}\left(\mathbf{1} - \gamma\boldsymbol{\xi}^{\wedge}\right) = \left(\mathbf{1} + \gamma\boldsymbol{\xi}^{\wedge}\right)^{-1}\left(\mathbf{1} - \lambda\boldsymbol{\xi}^{\wedge}\right), \tag{53b}$$

where γ and λ are again given by (51). To achieve this factorization, we must select $c = \frac{1}{4}\nu^2\epsilon$ in (30), which consequently selects a $\mathbf{D}(\boldsymbol{\phi}) \neq \mathbf{J}(\boldsymbol{\phi})$.

Finally, the pattern holds for $\mathrm{Ad}(SE(3))$ where we have

$$\boldsymbol{\mathcal{T}}(\boldsymbol{\xi}) = \left(\mathbf{1} - \lambda\boldsymbol{\xi}^{\curlywedge}\right)^{-1}\left(\mathbf{1} + \gamma\boldsymbol{\xi}^{\curlywedge}\right) = \left(\mathbf{1} - \gamma\boldsymbol{\xi}^{\curlywedge}\right)^{-1}\left(\mathbf{1} + \lambda\boldsymbol{\xi}^{\curlywedge}\right), \tag{54a}$$

$$\boldsymbol{\mathcal{T}}(\boldsymbol{\xi})^{-1} = \left(\mathbf{1} + \lambda\boldsymbol{\xi}^{\curlywedge}\right)^{-1}\left(\mathbf{1} - \gamma\boldsymbol{\xi}^{\curlywedge}\right) = \left(\mathbf{1} + \gamma\boldsymbol{\xi}^{\curlywedge}\right)^{-1}\left(\mathbf{1} - \lambda\boldsymbol{\xi}^{\curlywedge}\right), \tag{54b}$$

where γ and λ are yet again given by (51). To achieve this, we must select

$$d = \phi^2 f + \frac{\nu^2}{\epsilon}, \quad e = \frac{1}{2}\epsilon\phi^2 f + \frac{\nu^2}{2}, \quad f = \frac{\nu^2\epsilon}{4 + \phi^2\epsilon^2}, \quad g = \frac{1}{2}\epsilon f, \tag{55}$$

in (41). As discussed in the previous section, when $\phi(\varphi) = 2\tan\frac{\varphi}{2}$ this happens to select $\mathbf{D}(\boldsymbol{\phi}) = \mathbf{J}(\boldsymbol{\phi})$, but this is not true for the other generating functions.

3.6. Inverse mappings

We briefly discuss how to go the other way from $\mathbf{T}$, or $\boldsymbol{\mathcal{T}}$, back to $\boldsymbol{\xi}$ in the general case. The first step is to extract $\mathbf{C}(\boldsymbol{\phi})$ and $\mathbf{r}$ from $\mathbf{T}$, or $\boldsymbol{\mathcal{T}}$. From $\mathbf{C}(\boldsymbol{\phi})$, we can exploit the facts that

$$\mathrm{tr}(\mathbf{C}(\boldsymbol{\phi})) = 2\cos\varphi + 1, \quad \mathbf{C}(\boldsymbol{\phi})\mathbf{a} = \mathbf{a}, \tag{56}$$

which are true independent of the chosen vector parameterization. The first gives us the rotation angle, φ, while the second is an eigenproblem that can be solved for axis $\mathbf{a}$ (unit-length eigenvector corresponding to eigenvalue of 1); see Hughes [8] for an explicit formula. Owing to the singularity of the vector parameterization, we choose $\varphi \in (-\pi, \pi]$. We can then calculate $\boldsymbol{\phi} = \phi(\varphi)\,\mathbf{a}$.

Once we have $\boldsymbol{\phi}$, we can calculate $\boldsymbol{\rho}$ according to

$$\boldsymbol{\rho} = \mathbf{D}(\boldsymbol{\phi})^{-1}\mathbf{r}, \tag{57}$$

and then assemble $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$ into $\boldsymbol{\xi}$. In the case of $\mathbf{D}(\boldsymbol{\phi}) = \mathbf{J}(\boldsymbol{\phi})$, the inverse, $\mathbf{J}(\boldsymbol{\phi})^{-1}$, is given by (24) and naturally has singularities of which to be aware.

3.7. Compounding poses

For $SO(3)$, we seek to compound two rotations such that

$$\mathbf{C}(\boldsymbol{\phi}) = \mathbf{C}(\boldsymbol{\phi}_2)\mathbf{C}(\boldsymbol{\phi}_1). \tag{58}$$

Bauchau and Trainelli [9] show that the vector parameterization can be directly compounded according to

$$\varphi = 2\cos^{-1}\left(\nu(\varphi_1)\nu(\varphi_2)\left(\frac{1}{\epsilon(\varphi_1)\epsilon(\varphi_2)} - \frac{1}{4}\boldsymbol{\phi}_1^T\boldsymbol{\phi}_2\right)\right), \tag{59a}$$

$$\boldsymbol{\phi} = \frac{\nu(\varphi_1)\nu(\varphi_2)}{\nu(\varphi)}\left(\frac{1}{\epsilon(\varphi_2)}\boldsymbol{\phi}_1 + \frac{1}{\epsilon(\varphi_1)}\boldsymbol{\phi}_2 - \frac{1}{2}\boldsymbol{\phi}_1^{\wedge}\boldsymbol{\phi}_2\right). \tag{59b}$$

The first of these is used to calculate the compound rotation angle, φ, while the second provides the complete vectorial parameterization, $\boldsymbol{\phi}$.

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

For $SE(3)$, Condurache and Ciureanu [18] recently discussed a closed-form solution for compounding two pose vectors in the case that $\boldsymbol{\phi} = \varphi \mathbf{a}$. Bauchau and Choi (2003) showed that we can easily generalize this result to any of our vectorial pose parameterizations. The compounding of two poses is

$$\begin{aligned}\mathbf{T}(\boldsymbol{\xi}) = \begin{pmatrix} \mathbf{C}(\boldsymbol{\phi}) & \mathbf{D}(\boldsymbol{\phi})\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{pmatrix} = \mathbf{T}(\boldsymbol{\xi}_2)\mathbf{T}(\boldsymbol{\xi}_1) &= \begin{pmatrix} \mathbf{C}(\boldsymbol{\phi}_2) & \mathbf{D}(\boldsymbol{\phi}_2)\boldsymbol{\rho}_2 \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{C}(\boldsymbol{\phi}_1) & \mathbf{D}(\boldsymbol{\phi}_1)\boldsymbol{\rho}_1 \\ \mathbf{0}^T & 1 \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{C}(\boldsymbol{\phi}_2)\mathbf{C}(\boldsymbol{\phi}_1) & \mathbf{C}(\boldsymbol{\phi}_2)\mathbf{D}(\boldsymbol{\phi}_1)\boldsymbol{\rho}_1 + \mathbf{D}(\boldsymbol{\phi}_2)\boldsymbol{\rho}_2 \\ \mathbf{0}^T & 1 \end{pmatrix}. \end{aligned} \tag{60}$$

Comparing entries, we see that

$$\mathbf{C}(\boldsymbol{\phi}) = \mathbf{C}(\boldsymbol{\phi}_2)\mathbf{C}(\boldsymbol{\phi}_1), \tag{61a}$$

$$\boldsymbol{\rho} = \mathbf{D}(\boldsymbol{\phi})^{-1}\mathbf{C}(\boldsymbol{\phi}_2)\mathbf{D}(\boldsymbol{\phi}_1)\boldsymbol{\rho}_1 + \mathbf{D}(\boldsymbol{\phi})^{-1}\mathbf{D}(\boldsymbol{\phi}_2)\boldsymbol{\rho}_2. \tag{61b}$$

We can use the approach of Bauchau and Trainelli [9] to calculate the compound rotation, $\boldsymbol{\phi}$, and then calculate $\boldsymbol{\rho}$ afterwards. We require an expression for $\mathbf{D}(\boldsymbol{\phi})^{-1}$, which in the case of $\mathbf{D}(\boldsymbol{\phi}) = \mathbf{J}(\boldsymbol{\phi})$ is given by (24).

3.8. Additional useful properties of vector mappings

There are a few other important properties of our vector mappings of which we frequently make use when manipulating expressions in state-estimation problems. Owing to the well-known identity [5],

$$(\mathbf{C}\mathbf{v})^\wedge \equiv \mathbf{C}\mathbf{v}^\wedge\mathbf{C}^T, \tag{62}$$

where $\mathbf{C} \in SO(3)$ and $\mathbf{v} \in \mathbb{R}^3$, we have that

$$\mathbf{C}_1\mathbf{C}_2(\boldsymbol{\phi}_2)\mathbf{C}_1^T \equiv \mathbf{C}_2\left(\mathbf{C}_1\boldsymbol{\phi}_2\right). \tag{63}$$

The proof is as follows:

$$\begin{aligned}\mathbf{C}_1\mathbf{C}_2(\boldsymbol{\phi}_2)\mathbf{C}_1^T = \mathbf{C}_1\left(\mathbf{1} + \alpha\,\boldsymbol{\phi}_2^\wedge + \beta\boldsymbol{\phi}_2^{\wedge^2}\right)\mathbf{C}_1^T &= \mathbf{1} + \alpha\,\mathbf{C}_1\boldsymbol{\phi}_2^\wedge\mathbf{C}_1^T + \beta\,\mathbf{C}_1\boldsymbol{\phi}_2^\wedge\mathbf{C}_1\,\mathbf{C}_1^T\boldsymbol{\phi}_2^\wedge\mathbf{C}_1^T \\ &= \mathbf{1} + \alpha\left(\mathbf{C}_1\boldsymbol{\phi}_2\right)^\wedge + \beta\left(\mathbf{C}_1\boldsymbol{\phi}_2\right)^{\wedge^2} = \mathbf{C}_2\left(\mathbf{C}_1\boldsymbol{\phi}_2\right), \end{aligned} \tag{64}$$

where α and β are standing in for the coefficients in (19).

For $SE(3)$, we can use the identities [5],

$$(\mathcal{T}\mathbf{x})^\wedge \equiv \mathbf{T}\mathbf{x}^\wedge\mathbf{T}^{-1}, \tag{65}$$

$$(\mathcal{T}\mathbf{x})^\curlywedge \equiv \mathcal{T}\mathbf{x}^\curlywedge\,\mathcal{T}^{-1} \tag{66}$$

with $\mathbf{T} \in SE(3)$, $\mathcal{T} = \mathrm{Ad}(\mathbf{T})$, $\mathbf{x} \in \mathbb{R}^6$ to establish

$$\mathbf{T}_1\mathbf{T}_2(\boldsymbol{\xi}_2)\mathbf{T}_1^{-1} \equiv \mathbf{T}_2\left(\mathcal{T}_1\boldsymbol{\xi}_2\right), \tag{67}$$

$$\mathcal{T}_1\,\mathcal{T}_2(\boldsymbol{\xi}_2)\,\mathcal{T}_1^{-1} \equiv \mathcal{T}_2\left(\mathcal{T}_1\boldsymbol{\xi}_2\right), \tag{68}$$

where the proofs are similar to (65).

4. Applications

In this section, we touch on three classic applications of pose vectors: (i) interpolation between two poses (useful in computer graphics, for example), (ii) a servoing control problem, and (iii) the alignment of two pointclouds. In each, we show how different choices for the vector parameterization can produce different results.

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

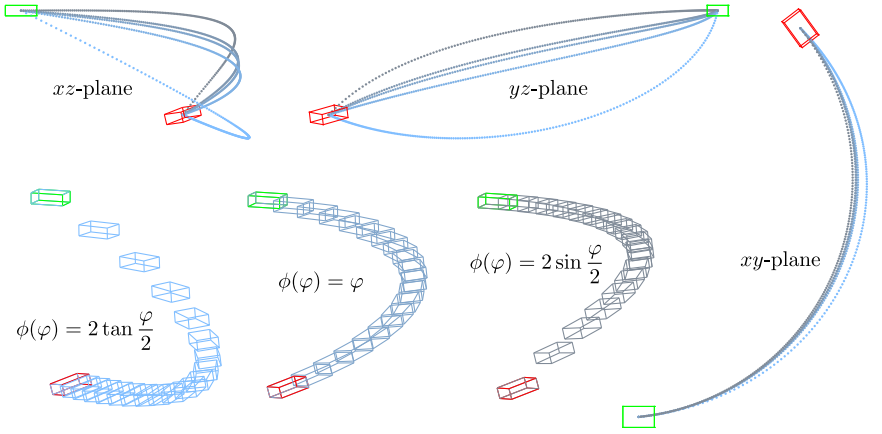


Figure 2. *An example of "linear" interpolation between an initial pose (green) and final pose (red) for different vector parameterizations of pose. The paths are straight lines in $\mathbb{R}^6$ but when mapped to SE(3) produce different curves. Each parameterization takes a unique path with the $\phi(\varphi) = 2\tan\frac{\varphi}{2}$ generating function being the most distinct. The figure compares the path taken by each of the generating functions from Figure 1 (same color scheme) as well as displays the full pose trajectory individually for three generating functions.*

4.1. Interpolation

Pose interpolation has been discussed extensively in the literature; Han and Bauchau [19,20] provide some representative examples. Here our goal is not to claim a new method of pose interpolation but rather to show the effect of simply changing the generating function on a basic interpolation scheme defined in the vector space.

We can define a "linear" interpolation scheme for poses using any of our vector parameterizations:

$$\boldsymbol{\xi} = (1 - s)\,\boldsymbol{\xi}_{\text{final}} + s\,\boldsymbol{\xi}_{\text{initial}}, \tag{69}$$

where $\boldsymbol{\xi}_{\text{initial}}$ is an initial pose, $\boldsymbol{\xi}_{\text{final}}$ a final pose, and $s \in [0, 1]$ the interpolation variable. As we vary s from 0 to 1, the vector, $\boldsymbol{\xi}$, will vary and so too will $\mathbf{T}(\boldsymbol{\xi})$. This allows the smooth transition from one pose to another.

Figure 2 provides a qualitative example of this scheme for all five generating functions, $\phi(\varphi)$, listed in Table 1 with $\mathbf{D}(\boldsymbol{\phi}) = \mathbf{J}(\boldsymbol{\phi})$. The paths taken each correspond to a *geodesic* in $\mathbb{R}^6$, but when mapped to $SE(3)$ are quite different. Notably, none of the paths are straight lines in Euclidean space owing to the coupling between translation and rotation. As the rotational motion becomes smaller, the paths followed by the different generating functions look more similar owing to (17). Looking more closely at Figure 2, we can also observe that the spacing of the steps is quite different for the different generating functions, even though we stepped s uniformly. The tan generating functions take larger steps at the beginning but smaller ones at the end, while the sin ones do the opposite, and the $\phi(\varphi) = \varphi$ function keeps the steps even throughout. Naturally, we can make the steps in the interpolation variable, s, as small as we like to obtain smooth interpolation; we chose fairly large steps to keep the plot in Figure 2 from becoming too cluttered.

4.2. Pose servoing control

Pose servoing is also a well-studied topic. Here, we simply want to show the effect of changing the pose mapping on a simple control law defined in the vector space. We define a "linear" control law to servo onto a reference path. Consider a reference pose, $\mathbf{T}_{\text{ref}}$, that we want to track. We can compare our current

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

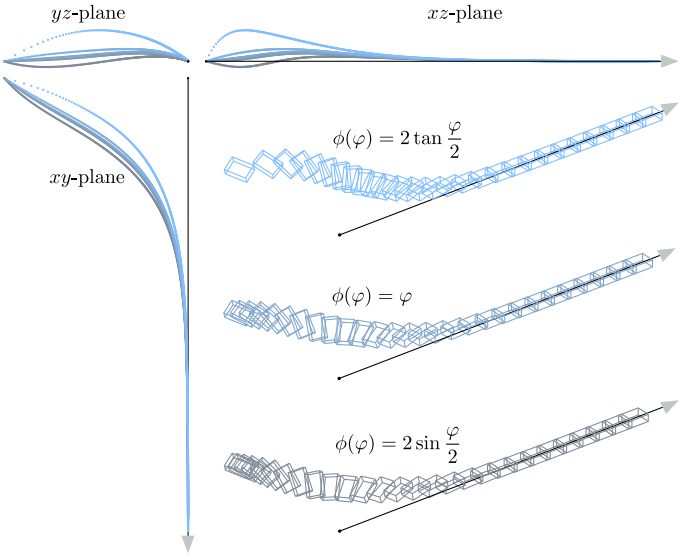


Figure 3. *An example of pose servoing control using a "linear" controller in pose vector space. Once mapped to SE(3), we see different behaviors from the different pose mappings. In this example, the reference pose is moving along the straight black line and there is an initial error. The figure compares the path taken to join the reference by each of the generating functions from Figure 1 (same color scheme) as well as displays the full pose trajectory individually for three generating functions.*

pose, $\mathbf{T}(\boldsymbol{\xi})$, to this reference and put it through the inverse mapping to create an "error vector":

$$\boldsymbol{\xi}_{\text{error}} \leftarrow \mathbf{T}(\boldsymbol{\xi})\mathbf{T}_{\text{ref}}^{-1}, \tag{70}$$

where the left arrow stands in for the inverse of any of our pose mappings. We can then adjust our pose to be slightly closer to the reference pose (which is moving) using a simple control law:

$$\dot{\boldsymbol{\xi}} = -\kappa\, \boldsymbol{\xi}_{\text{error}}. \tag{71}$$

with $\kappa > 0$ a gain. As the error $\boldsymbol{\xi}_{\text{error}}$ approaches zero, the moving pose, $\mathbf{T}(\boldsymbol{\xi})$, will approach the reference pose, $\mathbf{T}_{\text{ref}}$.

Figure 3 provides an example of servoing to a reference pose that is moving straight along the x-axis, starting from a poor initial pose that is translated and rotated from the reference. We kept the vector control law (including the gain) identical and varied only the choice of pose mapping. As expected, we see different transient behavior resulting from the different pose mappings as they seek the reference.

4.3. Pointcloud alignment

We use this section to discuss aligning two clouds of three-dimensional points. Our problem is similar to that of Barfoot et al. [21], but formulated directly on $SE(3)$.

4.3.1. Problem setup

Let us consider that we have two sets of observations of some three-dimensional points, $\mathbf{p}_j$ and $\mathbf{q}_j$, expressed in two different reference frames. We assume the points are expressed in homogeneous format,

$$\mathbf{p} = \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{72}$$

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

Our goal is to find the rigid-body transformation, $\mathbf{T}$, that minimizes the following sum-of-squares cost function,

$$J(\mathbf{T}) = \frac{1}{2} \sum_j \mathbf{e}_j^T \mathbf{W}_j \mathbf{e}_j, \tag{73}$$

where

$$\mathbf{e}_j = \mathbf{q}_j - \mathbf{T}\,\mathbf{p}_j, \tag{74}$$

and $\mathbf{W}_j > 0$ is a matrix weight for each associated point pair.

4.3.2. General perturbation solution

Owing to the matrix weights, there is no closed-form solution to this optimization problem and we must turn to an iterative scheme [5]. If we have some initial guess (or operating point) for our pose solution, $\mathbf{T}_{\text{op}}$, we can seek an optimal perturbation, $\delta\mathbf{T}(\boldsymbol{\xi})$, that iteratively reduces our cost function. Any of the pose mappings discussed earlier could be used in this regard:

$$\mathbf{T} = \delta\mathbf{T}(\boldsymbol{\xi})\,\mathbf{T}_{\text{op}} \approx \left(\mathbf{1} + \boldsymbol{\xi}^{\wedge}\right)\mathbf{T}_{\text{op}}, \tag{75}$$

where notably the infinitesimal expression is the same for all our pose mappings. Inserting this into the error, we have

$$\mathbf{e}_j \approx \mathbf{q}_j - \left(\mathbf{1} + \boldsymbol{\xi}^{\wedge}\right)\mathbf{T}_{\text{op}}\mathbf{p}_j = \underbrace{\mathbf{q}_j - \mathbf{T}_{\text{op}}\mathbf{p}_j}_{\mathbf{e}_{j,\text{op}}} - \left(\mathbf{T}_{\text{op}}\mathbf{p}_j\right)^{\odot}\boldsymbol{\xi} = \mathbf{e}_{j,\text{op}} - \left(\mathbf{T}_{\text{op}}\mathbf{p}_j\right)^{\odot}\boldsymbol{\xi}, \tag{76}$$

where we have made use of $\boldsymbol{\xi}^{\wedge}\mathbf{p} \equiv \mathbf{p}^{\odot}\boldsymbol{\xi}$ [5,6] with

$$\mathbf{p}^{\odot} = \begin{pmatrix} \mathbf{1} & -\mathbf{v}^{\wedge} \\ \mathbf{0}^T & \mathbf{0}^T \end{pmatrix}. \tag{77}$$

Our approximate error expression is now linear and inserting it into the cost function we have

$$J(\boldsymbol{\xi}) \approx \frac{1}{2} \sum_j \left(\mathbf{e}_{j,\text{op}} - \left(\mathbf{T}_{\text{op}}\mathbf{p}_j\right)^{\odot}\boldsymbol{\xi}\right)^T \mathbf{W}_j \left(\mathbf{e}_{j,\text{op}} - \left(\mathbf{T}_{\text{op}}\mathbf{p}_j\right)^{\odot}\boldsymbol{\xi}\right), \tag{78}$$

which is exactly quadratic in $\boldsymbol{\xi}$, and can be minimized by solving the following linear system of equations for $\boldsymbol{\xi}$:

$$\left(\sum_j \left(\mathbf{T}_{\text{op}}\mathbf{p}_j\right)^{\odot^T} \mathbf{W}_j \left(\mathbf{T}_{\text{op}}\mathbf{p}_j\right)^{\odot}\right)\boldsymbol{\xi} = \sum_j \left(\mathbf{T}_{\text{op}}\mathbf{p}_j\right)^{\odot^T} \mathbf{W}_j \mathbf{e}_{j,\text{op}}. \tag{79}$$

After solving for the optimal $\boldsymbol{\xi}$, we can update our initial guess using our pose mapping of choice,

$$\mathbf{T}_{\text{op}} \leftarrow \delta\mathbf{T}(\boldsymbol{\xi})\,\mathbf{T}_{\text{op}}, \tag{80}$$

and iterate to convergence. This approach is discussed in detail by Barfoot [5] for the $\phi(\varphi) = \varphi$ case, while here we show it generalizes to any vectorial mapping of pose.

4.3.3. Alternate Cayley perturbation solution

The previous section relied on using the infinitesimal expression for pose at each iteration. Here, we take a slightly different strategy based on the Cayley transformation that is similar in spirit to that of refs. [22–27]. We will refer to the resulting algorithm as *CayPer*. If we choose the $\phi(\varphi) = 2\tan\frac{\varphi}{2}$ generating function, we can then let our perturbation to the operating point be

$$\delta\mathbf{T}(\boldsymbol{\xi}) = \left(\mathbf{1} - \frac{1}{2}\boldsymbol{\xi}^{\wedge}\right)^{-1}\left(\mathbf{1} + \frac{1}{2}\boldsymbol{\xi}^{\wedge}\right). \tag{81}$$

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

Our point error is then

$$\mathbf{e}_j = \mathbf{q}_j - \left(\mathbf{1} - \frac{1}{2}\boldsymbol{\xi}^\wedge\right)^{-1}\left(\mathbf{1} + \frac{1}{2}\boldsymbol{\xi}^\wedge\right)\mathbf{T}_{\text{op}}\,\mathbf{p}_j. \tag{82}$$

We can then define a new error, $\mathbf{e}'_j$, by premultiplying by $\left(\mathbf{1} - \frac{1}{2}\boldsymbol{\xi}^\wedge\right)$:

$$\mathbf{e}'_j = \left(\mathbf{1} - \frac{1}{2}\boldsymbol{\xi}^\wedge\right)\mathbf{e}_j = \left(\mathbf{1} - \frac{1}{2}\boldsymbol{\xi}^\wedge\right)\mathbf{q}_j - \left(\mathbf{1} + \frac{1}{2}\boldsymbol{\xi}^\wedge\right)\mathbf{T}_{\text{op}}\mathbf{p}_j = \mathbf{e}_{j,\text{op}} - \frac{1}{2}\left(\mathbf{q}_j + \mathbf{T}_{\text{op}}\mathbf{p}_j\right)^\odot\,\boldsymbol{\xi}. \tag{83}$$

Importantly, we have that

$$\lim_{\boldsymbol{\xi}\to\mathbf{0}} \mathbf{e}'_j = \mathbf{e}_j. \tag{84}$$

This suggests that we can define a new cost function,

$$\begin{aligned} J'(\boldsymbol{\xi}) &= \frac{1}{2}\sum_j \mathbf{e}_j'^T \mathbf{W}_j \mathbf{e}'_j \\ &\approx \frac{1}{2}\sum_j \left(\mathbf{e}_{j,\text{op}} - \frac{1}{2}\left(\mathbf{q}_j + \mathbf{T}_{\text{op}}\mathbf{p}_j\right)^\odot\,\boldsymbol{\xi}\right)^T \mathbf{W}_j \left(\mathbf{e}_{j,\text{op}} - \frac{1}{2}\left(\mathbf{q}_j + \mathbf{T}_{\text{op}}\mathbf{p}_j\right)^\odot\,\boldsymbol{\xi}\right), \end{aligned} \tag{85}$$

which we minimize at each iteration instead of (78). The optimal perturbation, $\frac{1}{2}\boldsymbol{\xi}$, is this time the solution to the linear system,

$$\left(\sum_j \left(\mathbf{q}_j + \mathbf{T}_{\text{op}}\mathbf{p}_j\right)^{\odot^T} \mathbf{W}_j \left(\mathbf{q}_j + \mathbf{T}_{\text{op}}\mathbf{p}_j\right)^\odot\right)\frac{1}{2}\boldsymbol{\xi} = \sum_j \left(\mathbf{q}_j + \mathbf{T}_{\text{op}}\mathbf{p}_j\right)^{\odot^T} \mathbf{W}_j \mathbf{e}_{j,\text{op}}. \tag{86}$$

The update to the operating point is then

$$\mathbf{T}_{\text{op}} \leftarrow \left(\mathbf{1} - \frac{1}{2}\boldsymbol{\xi}^\wedge\right)^{-1}\left(\mathbf{1} + \frac{1}{2}\boldsymbol{\xi}^\wedge\right)\mathbf{T}_{\text{op}}, \tag{87}$$

which requires only linear algebraic operations (e.g., no trigonometric functions). Notably, on the first iteration with $\mathbf{T}_{\text{op}} = \mathbf{1}$, the CayPer approach becomes a one-shot weighted optimal linear attitude and translation estimator [22–27] directly on $SE(3)$. However, we can also continue to iterate to convergence of the original problem.

4.3.4. Experiment

We conducted a simulated pointcloud alignment experiment motivated by aligning two sets of matched features from a stereo camera. Our approach is very similar in nature to that of Maimone et al. [28] and Barfoot et al. [21] in that we align matched stereo points in Euclidean space rather than image space.

We generated feature locations in image space, $\mathbf{y}$, by observing a set of static landmarks using a stereo camera model:

$$\mathbf{y} = \begin{pmatrix} u_\ell \\ v_\ell \\ u_r \\ v_r \end{pmatrix} = \underbrace{\begin{pmatrix} f & 0 & c_u & f\frac{b}{2} \\ 0 & f & c_v & 0 \\ f & 0 & c_u & -f\frac{b}{2} \\ 0 & f & c_v & 0 \end{pmatrix}}_{\mathbf{M}} \frac{1}{z}\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} + \mathbf{n}, \tag{88}$$

where $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ and $\mathbf{M}$ is a combined parameter matrix for the stereo rig. We used focal length $f = 200$ (pixels), stereo baseline $b = 0.25$ (m), optical center $(c_u, c_v) = (0, 0)$ (pixels), and measurement covariance $\mathbf{R} = 0.25^2\mathbf{1}$ (pixels^2). This model was used to generate two noisy pointclouds where the second frame is transformed by groundtruth pose, $\mathbf{T}_{\text{gt}}$, from the first. Once the noisy feature locations were generated in the two different stereo image frames, we transformed these back to two Euclidean

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

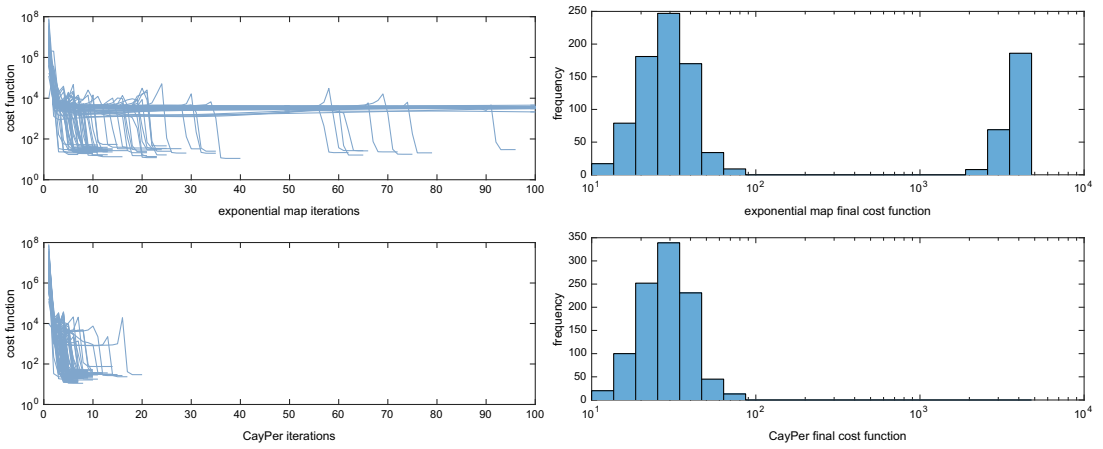


Figure 4. *Results of pointcloud alignment problem for both the exponential map and CayPer algorithm. The left plot shows the convergence history of the cost function J for 100 trials where the initial pose was randomly selected. We see that the exponential map regularly takes a long time to converge and often becomes stuck in a local minimum when the initial pose is far from the groundtruth. CayPer always converged in 20 iterations or less and always ended at the global minimum. The right plot shows a histogram of the costs, J, at the final iteration for 1000 trials where the initial pose was randomly selected. Again, we see that in approximately 20% of the trials the exponential map failed to converge to the global minimum while CayPer succeeded in all trials.*

pointclouds, $\mathbf{p}_j$ and $\mathbf{q}_j$, using the inverse stereo camera model. The alignment problem is then to recover $\mathbf{T}_{\text{gt}}$ using $\mathbf{p}_j$ and $\mathbf{q}_j$. We assume the points have already been correlated correctly and do not treat the outlier rejection problem here.

To generate the weight matrices, $\mathbf{W}_j$, for our cost function, we mapped the isotropic covariance in image space, $\mathbf{R}$, through the inverse stereo camera model to produce an (inverse) covariance of our error in Euclidean space:

$$\mathbf{W}_j^{-1} = \mathbb{E}[\mathbf{e}_j \mathbf{e}_j^T]. \tag{89}$$

The resulting $\mathbf{W}_j^{-1}$ is a long, skinny uncertainty ellipsoid in the depth direction of the camera as established by Matthies and Shafer [29]. This anisotropic measurement covariance makes this problem particularly challenging. As far we know, there is no practical algorithm that can guarantee finding the global minimum of the cost function, J, in this anisotropic situation. Yang et al. [30] show how to certify finding the global minimum in the case that $\mathbf{W}_j^{-1}$ is isotropic (i.e., a scalar times the identity matrix) even with significant outliers, but this does not apply in the chosen situation.

We selected a pyramid distribution of 12 landmarks in the field of view from 5 to 15 m away from the camera. The camera was translated 1 m in the depth, z, direction to generate the two noisy pointclouds. We then ran many trials where we varied (i) the random draw of the measurement noise, and (ii) the random draw of the initial guess for the alignment algorithms. Our goal was to see if choosing different vector parameterizations of pose would result in different paths to get from the initial guess to the groundtruth pose.

4.3.5. Results

We found that using different vector parameterizations of pose in the general perturbation approach of Section 4.3.2 made little difference and so selected the "exponential map" as a representative method. However, the CayPer approach of Section 4.3.3 turned out to be a lot better at not getting stuck in poor local minima.

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

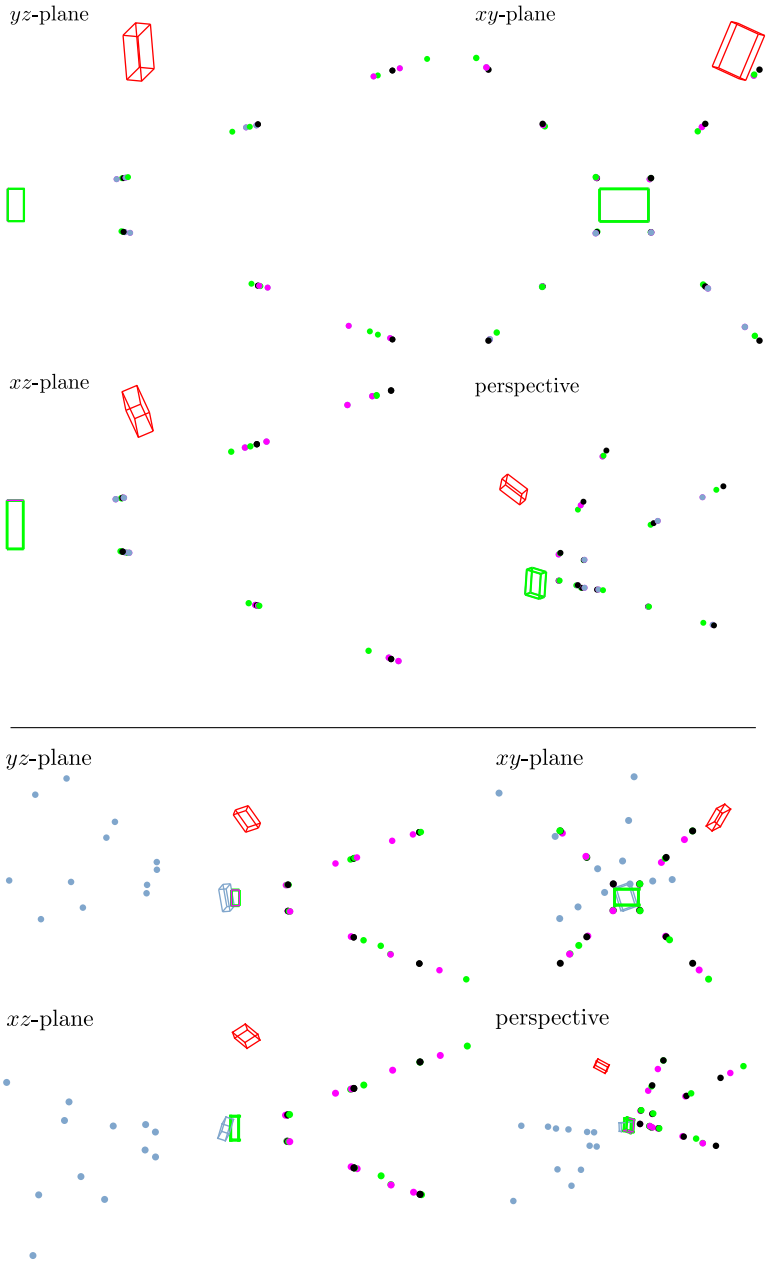


Figure 5. *Two trials of the pointcloud alignment problem. In both cases, CayPer successfully finds the correct minimum while the exponential map succeeds in the top case but not the bottom where it becomes stuck in an undesirable local minimum. Red box is the initial guess, green box is the groundtruth pose, black dots are the groundtruth landmark locations, green dots are the first noisy pointcloud (static), magenta box (hidden by green box) is the CayPer final pose, blue box (when not hidden by green box) is the exponential map final pose, magenta dots are the second noisy pointcloud (transformed) at the final CayPer pose, and blue dots are the second noisy pointcloud (transformed) at the final exponential map pose.*

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

Figure 4 shows quantitative convergence results for several trials. We see that the CayPer approach does not get stuck in local minima as easily as the exponential map approach. In fact, in 1000 trials CayPer always arrived at the global minimum, but we are not prepared to claim this will always be true at this point. The exponential map frequently became trapped in local minima when the initial guess was particularly bad. We also found that even when both algorithms did find the global minimum, CayPer took fewer iterations to get there; both algorithms were iterated until $\boldsymbol{\xi}^T\boldsymbol{\xi} < 10^{-10}$ or hit 100 iterations.

Figure 5 shows 2 of the 1000 trials we ran. The top case shows the final alignment for a trial where both algorithms successfully converged to the global minimum. The bottom shows a trial where CayPer succeeded but the exponential map failed (catastrophically).

5. Conclusion and future work

We presented a family of different vector mappings of pose that may find application in computer vision, robotics, and even graphics. We showed that given a vector parameterization of rotation, there is actually an infinite number of possible vector pose mappings with different couplings between the rotational and translational components of the pose. We explored commutative mappings as well as Cayley and Cayley-type mappings. We discussed inverse mappings as well as provided a general formula for the compounding of two general pose mappings. Finally, we showed several of these pose mappings in action on three different applications: pose interpolation, pose servoing control, and pointcloud alignment estimation. The main conclusion is that while we might think of the exponential map as the *canonical* way to map vectors to poses, there are many other possibilities that might be preferable, depending on the application. In particular, we found that the Cayley transformation was particularly successful in the CayPer pointcloud alignment method we presented.

Moving forward, there are certainly other theoretical results and connections that might be found related to vector pose mappings. Moreover, several of these ideas might extend to analysis of other useful matrix Lie groups that involve $SO(3)$ as a building block. On the practical side, further analysis of the CayPer algorithm is worthwhile. We also believe that using the Cayley transformation to iteratively optimize poses in more involved problems such as pose-graph optimization and simultaneous localization and mapping will be worth pursuing.

Conflicts of interest. None.

Financial support. This work was partially supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Ethical considerations. None.

Authors' contributions. None.

References

[1] G. Mozzi, *Discorso matematico sopra il rotamento momentaneo dei corpi* (Donate Campo, 1763).
[2] M. Chasles, "Note sur les propriétés générales du système de deux corps semblables entr'eux et placés d'une manière quelconque dans l'espace; et sur le déplacement fini ou infiniment petit d'un corps solide libre," *Bulletin des Sciences Mathématiques, Férussac* **14**, 321–26 (1830).
[3] R. S. Ball, *A Treatise on the Theory of Screws* (Cambridge University Press, London, 1900).
[4] R. M. Murray, Z. Li and S. Sastry, *A Mathematical Introduction to Robotic Manipulation* (CRC Press, London, 1994).
[5] T. D. Barfoot, *State Estimation for Robotics* (Cambridge University Press, Cambridge, 2017).
[6] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Trans. Rob.* **30**(3), 679–693 (2014), (code).
[7] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups: Classical Results and Geometric Methods*, vol. 1–2 (Birkhauser, New York, 2009).
[8] P. C. Hughes, *Spacecraft Attitude Dynamics* (John Wiley and Sons, New York, 1986).

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press

[9] O. A. Bauchau and L. Trainelli, "The vectorial parameterization of rotation," *Nonlinear Dyn.* **32**(1), 71–92 (2003).
[10] O. A. Bauchau and J.-Y. Choi, "The vector parameterization of motion," *Nonlinear Dyn.* **33**(2), 165–188 (2003).
[11] O. A. Bauchau and L. Li, "Tensorial parameterization of rotation and motion," *J. Comput. Nonlinear Dyn.* **6**(3), 031007-1–031007-8 (2011).
[12] O. A. Bauchau, *Flexible Multibody Dynamics* (Springer Science & Business Media, London, 2011).
[13] M. Borri, L. Trainelli and C. L. Bottasso, "On representations and parameterizations of motion," *Multibody Syst. Dyn.* **4**(2), 129–193 (2000).
[14] J. M. Selig, "Cayley Maps for SE(3)," *12th International Federation for the Promotion of Mechanism and Machine Science World Congress* (London South Bank University, 2007) p. 6.
[15] A. Cayley, "Sur quelques propriétés des déterminants gauches," *J. für die reine und angewandte Mathematik* 1846(32), 119–123 (**1846**).
[16] G. M. T. D'Eleuterio and T. D. Barfoot, "On the Eigenstructure of Rotations and Poses: Commonalities and Peculiarities" (2021), in preparation for submission.
[17] J. Stuelpnagel, "On the parametrization of the three-dimensional rotation group," *SIAM Rev.* **6**(4), 422–430 (1964).
[18] D. Condurache and I.-A. Ciureanu, "Baker–Campbell–Hausdorff–Dynkin formula for the lie algebra of rigid body displacements," *Mathematics* **8**(7), 1185 (2020).
[19] S. Han and O. A. Bauchau, "Manipulation of motion via dual entities," *Nonlinear Dyn.* **85**(1), 509–524 (2016).
[20] S. Han and O. A. Bauchau, "On the global interpolation of motion," *Comput. Methods Appl. Mech. Eng.* **337**, 352–386 (2018).
[21] T. D. Barfoot, J. R. Forbes and P. T. Furgale, "Pose estimation using linearized rotations and quaternion algebra," *Acta Astronautica* **68**(1–2), 101–112 (2011).
[22] J. L. Junkins, M. Majji, B. Macomber, J. Davis, J. Doebbler and R. Nosterk, "Small body proximity sensing with a Novel HD 3D ladar system," *Adv. Astronaut. Sci.* **141**, 341–354 (2011).
[23] M. Majji, J. Davis, J. Doebbler, J. Junkins, B. Macomber, M. Vavrina and J. Vian, "Terrain Mapping and Landing Operations Using Vision Based Navigation Systems," AIAA Guidance, Navigation, and Control Conference (2011) p. 6581.
[24] D. Mortari, F. L. Markley and P. Singla, "Optimal linear attitude estimator," *J. Guidance Control Dyn.* **30**(6), 1619–1627 (2007).
[25] D. Qian, G. Charland-Arcand and J. R. Forbes, "TWOLATE: Total Registration of Point-Clouds Using a Weighted Optimal Linear Attitude and Translation Estimator," *2020 IEEE Conference on Control Technology and Applications (CCTA)* (IEEE, 2020) pp. 43–48.
[26] X. I. Wong and M. Majji "A structured light system for relative navigation applications," *IEEE Sens. J.* 16(17), 6662–6679 (2016).
[27] X. I. Wong, P. Singla, T. Lee and M. Majji, "Optimal Linear Attitude Estimator for Alignment of Point Clouds," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (IEEE, 2018) pp. 1577–15778.
[28] M. Maimone, Y. Cheng and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *J. Field Rob.* **24**(3), 169–186 (2007).
[29] L. Matthies and S. Shafer, "Error modeling in stereo navigation," *IEEE J. Rob. Autom.* **3**(3), 239–248 (1987).
[30] H. Yang, J. Shi and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Trans. Rob.* **37**(2), 314–333 (2020).

Cite this article: T. D. Barfoot, J. R. Forbes and G. M. T. D'Eleuterio (2022). "Vectorial parameterizations of pose", Robotica **40**, 2409–2427. https://doi.org/10.1017/S0263574721001715

https://doi.org/10.1017/S0263574721001715 Published online by Cambridge University Press