USING DUAL QUATERNIONS IN ROBOTICS

STEPHEN MONTGOMERY-SMITH AND CECIL SHY

ABSTRACT. We advocate for the use of dual quaternions to represent poses and twists for robotics. We show how to represent torques and forces using dual quaternions. We introduce the notion of the Lie derivative, and explain how it can be used to calculate the behavior of actuators. We show how to combine dual quaternions with the Newton-Raphson method to compute forward kinematics for parallel robots. We derive the equations of motion in dual quaternion form.

CONTENTS

1. INTRODUCTION

We assume that the reader is familiar with the notion of *pose* and *rigid motion*, and the use of quaternions to represent rotations. We use the terms pose and rigid motion interchangeably, since a pose can be considered as a rigid motion relative to a fixed reference frame, and the mathematical operations to manipulate them are identical. Thus we also use the terms rotation and orientation interchangeably, and similarly with translation and position. We refer the reader to [5, 10].

The use of dual quaternions to represent poses is well established, especially in the graphic card industry: [1, 12, 6, 16, 17, 19, 26]. However, in the robotics industry, it is more common to represent a pose using a quaternion (for the rotation) and a three dimensional vector (for the translation). In this paper, we advocate for the use of dual quaternions.

Let us start with a general discussion of why people might want to represent rotations by unit quaternions instead of three by three orthogonal matrices with determinant 1. One reason given is that quaternions can be represented with only four numbers, whereas the matrix representation requires nine numbers. Another representation commonly used are Euler angles, and these only require three numbers. However, with Euler angles, the formula for composition of rotations is difficult to calculate. Also, Euler angles suffer from gimbal lock, in which if the rotation takes certain values, the choice of Euler angles isn't unique, and discontinuities and singularities arise. Quaternions suffer from no such problems.

But in modern times, computer memory is very cost effective, and communication is fast, thus the aforementioned advantages might seem trivial at first. This begs the question: what other advantages do quaternions offer over matrices? Quaternions represent a quantity with three degrees of freedom by a four dimensional vector, whereas matrices use a nine dimensional vector. Suppose one obtains an approximation to a matrix or quaternion? For example, this would be common when using Newton-Raphson to find a rotation satisfying some formula, or to interpolate between rotations. So then one has to find a projection which takes the vector to an admissible vector, that is, a vector that actually represents a rotation. And projecting from a nine dimensional quantity to a three dimensional quantity is going to be much harder than the similarly projection from the four dimensional vector. (The projection from

quaternions to unit quaternions is called the *normalization*, and is defined in Section 3. A suggestion of what such a projection would be for (3×3) matrices is given at the end of Section 17.)

For representing poses, the usual representation is by a three by three orthogonal matrix with determinant 1 for the rotation, and a three dimensional vector for the translation, twelve dimensional in all. This has a natural representation as a four by four matrix, as we explain Section 5. A great advantage of this representation is the relationship between pose and twist:

$$\frac{d}{dt}(\text{pose}) = (\text{pose}) \times (\text{twist}). \tag{1}$$

where the multiplication is matrix multiplication, and a twist is a combination of angular velocity and translational velocity measured with respect to the moving frame.

Another natural representation used a lot is to use a quaternion and a three dimensional vector. But this new representation has problems in that it is not clear how the above equation would work in this situation. However, if one introduces the notion of the dual quaternion, which is an eight dimensional vector, equation (1) is still valid if one uses dual quaternion multiplication.

A challenge with dual quaternions is that they are abstract, which can make them harder to understand. Mathematicians find value behind abstract computational ideas, but Engineers usually find the end result, 'black box approach,' more useful.

A useful approach to developing intuition for dual quaternions is to understand the rules with their structure, and the mathematical operations needed to manipulate them, (that is, normalization, basis vector multiplication, pose differentiation). Once one becomes comfortable with their form and function, the advantage of duel quaternions over other methods becomes more apparant. For example, one realizes that computing the projection of an eight dimensional vector to the six degrees of freedom of a pose, is also quite simple. (This projection is also called the normalization, and is defined in Section 4.)

This makes it very useful if one is using the Newton-Raphson Method to find a pose satisfying certain properties. We show how the dual quaternion representation efficiently solves the forward kinematics problem.

Finally we consider the dynamics of poses, that is, the equations of motion of a rigid spinning body. In any representation this can be quite difficult, because of both centripetal and precessional effects. But from a theoretical perspective, using the dual quaternion representation is quite slick.

In summary, while the usual representation using Euler angles and translations is conceptually easy to understand, performing calculations can be difficult. Whereas with dual quaternions, while the are initially conceptually hard to understand, when it comes to calculations they are much easier to work with.

Other papers advocating for the use of dual quaternions in robotics are [1, 12, 26]. But this paper contains formulas we have not seen elsewhere, namely equation (10) which simplifies the comuptation of the inverse of a unit dual quaternion, equation (17) which gives a second order expansion of the normalization of dual quaternions close to 1, Theorem 4 which gives dynamics of the end effector in the language of dual quaternions as well as including the effects of driving the actuators, and Theorems 2 and 3 which show how to evaluate a exponential and logarithm of a dual quaternion. We also haven't seen Lie derivatives discussed in the context of dual quaternions and robotics, nor the Newton-Raphson Method to solve for unit dual quaternions. We also feel that the use of dual quaternions to represent twists, as in equations (31) and (32), hasn't received enough emphasis.

2. Notation

Scalars (or scalar valued functions) are generally represented by lower case Roman letters, the exception being ℓ for actuator values. 3-vectors are represented by bold face Roman letters. Dual quaternions and 6-vectors are represented by lower case Greek letters. Quaternions are generally represented by upper case Roman letters, except since pure quaternions can be identified with 3-vectors, they are often instead denoted by bold face lower case Roman letters.

Matrices are represented by san-serif capital letters, either Roman or Greek. Functions taking vectors to vectors are represented by san-serif capital Roman letters, or by the font corresponding to the type of the output of the function.

The following list gives examples of notation that is commonly used in this paper. We try to restrict ourselves to notation that is used in several places throughout the paper.

- t: time.
- $\dot{g}$: time derivative of g.
- $\ddot{g}$: second time derivative of g.
- $\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$: standard unit 3-vectors/quaternions.
- $\overline{A}$: conjugate of the quaternion A.
- $|A| = \sqrt{A\overline{A}}$: norm of the quaternion A (scalar).

- $\mathrm{Re}(A) = \frac{1}{2}(A + \overline{A})$: real part of the quaternion A (scalar).
- $\mathrm{Im}(A) = \frac{1}{2}(A - \overline{A})$: imaginary part of the quaternion A (3-vector/quaternion).
- $\widehat{A} = A/|A|$: normalization of the quaternion A (unit quaternion).
- $A \cdot B = \mathrm{Re}(A\overline{B})$: dot product of quaternions A and B (scalar).
- R: orthogonal matrix representing rotation ((3×3) matrix).
- $Q = \cos(\frac{1}{2}a) + \boldsymbol{s}\sin(\frac{1}{2}a)$: quaternion representing rotation about axis $\boldsymbol{s}$ by angle a (unit quaternion).
- $\boldsymbol{t}$: translation (3-vector).
- ϵ: dual quaternion such that $\epsilon^2 = 0$.
- l: characteristic length scale (scalar).
- $\|A + \epsilon B\| = \sqrt{|A|^2 + l^{-2}|B|^2}$: size of the dual quaternion $A + \epsilon B$ (scalar).
- $\overline{\eta}$: conjugate of the dual quaternion η.
- $|\eta| = \sqrt{\eta\overline{\eta}}$: norm of the dual quaternion η (dual number).
- $\widehat{\eta} = |\eta|^{-1}\eta$: normalization of the dual quaternion η.
- D: the set of dual quaternions.
- SD: the set of unit dual quaternions.
- $\mathfrak{sd}$: the set of pure dual quaternions.
- $\eta = Q + \frac{1}{2}\epsilon\boldsymbol{t}Q$: pose (unit dual quaternion).
- $\|\eta\|$: size of the dual quaternion η (scalar).
- $\boldsymbol{w}$: angular velocity (3-vector).
- $\boldsymbol{w}^\star$: Hodge star operator of $\boldsymbol{w}$ ((3×3)-matrix).
- $\boldsymbol{v}$: translational velocity (3-vector).
- $\varphi = \frac{1}{2}\boldsymbol{w} + \frac{1}{2}\epsilon\boldsymbol{v}$: twist (pure dual quaternion).
- $\alpha = \dot{\varphi}$: acceleration (pure dual quaternion).
- $\boldsymbol{q}$: torque (3-vector).
- $\boldsymbol{p}$: force (3-vector).
- $\tau = 2\boldsymbol{q} + 2\epsilon\boldsymbol{p}$: wrench (pure) dual quaternion.
- $\boldsymbol{r}_0$: center of mass of end effector (3-vector).
- θ: perturbation of pose (pure dual quaternion).
- $\eta_1 \bigtriangleup \eta_2 = \frac{1}{2}(\overline{\eta}_2\eta_1 - \overline{\eta}_1\eta_2)$: Lie difference of η_1 and η_2 (pure dual quaternion).
- β_i: basis element dual quaternion.
- $\eta_1 \cdot \eta_2$: dot product of dual quaternions as vectors (scalar).
- $\mathcal{L}_\theta g$: directional Lie derivative of g (same type as g).
- $\mathcal{L}_i g$: partial Lie derivative of g (same type as g).
- $\mathcal{L}g$: full Lie derivative of g (pure dual quaternion).
- $\boldsymbol{\ell}$: actuator values (n-vector).
- $\boldsymbol{f}$: actuator forces (n-vector).
- L: function that converts η to $\boldsymbol{\ell}$.

- $\Lambda = \mathcal{L}\mathsf{L}$: matrix that converts φ to $\dot{\boldsymbol{\ell}}$.
- $\mathsf{T} = \Lambda^T$: matrix that converts $\boldsymbol{f}$ to τ.
- $\left[\frac{\partial^2 g}{\partial \eta^2}\right]$: Hessian matrix of g ((8×8) matrix).
- Y: function that converts $\boldsymbol{\ell}$ to η.
- $\boldsymbol{f}_0$: no-load actuator forces (n-vector).
- M_0: effective no-load moment of inertia of actuators ($(n \times n)$ matrix).
- m_e: mass of the end effector (scalar).
- M_e: moment of inertia of end effector ((3×3) matrix).
- $\boldsymbol{g}$: acceleration due to gravity (3-vector).
- $\tilde{\boldsymbol{g}} = \overline{Q}\boldsymbol{g}Q$: acceleration due to gravity in moving frame of reference (3-vector).
- M: effective mass of the parallel robot ((6×6) matrix).
- μ: additive term used in converting acceleration α to wrench τ (pure dual quaternion).

3. Motivation

A *pose*, or *rigid motion*, is a rotation followed by a translation. (Semantically, a pose is the position of the end effector of the robot, and thus different from a rigid motion, but mathematically they are the same if we consider a pose to be a rigid motion with respect to the stationary reference frame.) The rotation is represented by a (3×3) orthogonal matrix R with determinant 1. A translation $\boldsymbol{t}$ is a vector in $\mathbb{R}^3$. These represent the pose $(\mathsf{R}, \boldsymbol{t})$:

$$\boldsymbol{r} \mapsto \mathsf{R}\boldsymbol{r} + \boldsymbol{t}. \tag{2}$$

Composition of poses is written from right to left. Thus

$$(\mathsf{R}_1, \boldsymbol{t}_1) \circ (\mathsf{R}_2, \boldsymbol{t}_2) = (\mathsf{R}_1\mathsf{R}_2, \boldsymbol{t}_1 + \mathsf{R}_1\boldsymbol{t}_2). \tag{3}$$

One way to represent a pose is using a four by four matrix

$$\begin{bmatrix} \mathsf{R} & \boldsymbol{t} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathsf{R}_{11} & \mathsf{R}_{12} & \mathsf{R}_{13} & t_1 \\ \mathsf{R}_{21} & \mathsf{R}_{22} & \mathsf{R}_{23} & t_2 \\ \mathsf{R}_{31} & \mathsf{R}_{32} & \mathsf{R}_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Then composition of poses may be computed simply by multiplying matrices of the form (4). In the literature, the Lie group of rotations is often denoted $SO(3)$. The Lie group of poses is often denoted $SE(3)$.

The problem with this representation is that twelve numbers are required to represent a pose. Other problems, which we discuss below, is that of interpolating a sequence of poses, and of computing forward kinematics.

One common way to solve both of these problems is to represent the rotation by a unit quaternion [2, 23], which we briefly describe here. A *quaternion* is a quadruple of real numbers, written as $A = w + x\boldsymbol{i} +$

$y\boldsymbol{j}+z\boldsymbol{k}$, with the algebraic operations $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$. Its *conjugate* is $\overline{A} = w - x\boldsymbol{i} - y\boldsymbol{j} - y\boldsymbol{k}$, its *norm* is $|A| = (w^2 + x^2 + y^2 + z^2)^{1/2} = \sqrt{A\overline{A}} = \sqrt{\overline{A}A}$, its *normalization* is $\widehat{A} = A/|A|$, its *real part* is $\mathrm{Re}(A) = w = \frac{1}{2}(A\overline{A})$, and its *imaginary part* is $\mathrm{Im}(A) = \boldsymbol{i}x + \boldsymbol{j}y + \boldsymbol{k}z = \frac{1}{2}(A - \overline{A})$. It is called a *unit* quaternion if $|A| = 1$, a *real* quaternion if $\mathrm{Im}(A) = 0$, and a *pure* quaternion if $\mathrm{Re}(A) = 0$. Note the multiplicative inverse is given by $A^{-1} = \overline{A}/|A|^2$.

We identify three dimensional vectors with pure quaternions, by identifying $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ with the three standard unit vectors. A unit quaternion Q represents the rotation $\boldsymbol{r} \mapsto Q\boldsymbol{r}\overline{Q}$. A rotation by angle a about an axis $\boldsymbol{s}$, where $|\boldsymbol{s}| = 1$, has two unit quaternion representations: $\pm(\cos(\frac{1}{2}a) + \boldsymbol{s}\sin(\frac{1}{2}a))$. Composition of rotations corresponds to multiplication of unit quaternions. With some practice, it becomes easier to read a rotation from a quaternion than it does from Euler angles, the standard method of representing rotations.

We can represent quaternions as four dimensional vectors, and give it the inner product

$$A \cdot B = \mathrm{Re}(A\overline{B}) = \mathrm{Re}(\overline{A}B). \tag{5}$$

Then a way to represent poses is as a pair $(Q, \boldsymbol{t})$. This is the way poses are represented internally in the Robotic Operating System [25]. Then the composition rule is

$$(Q_1, \boldsymbol{t}_1) \circ (Q_2, \boldsymbol{t}_2) = (Q_1 Q_2, \boldsymbol{t}_1 + Q_1 \boldsymbol{t}_2 \overline{Q}_1). \tag{6}$$

A difficulty with this is that the composition is not a bilinear operation as it was with the representation (4). (This lack of bilinear representation becomes particularly problematic when representing twists which we describe below.) However there is a way to resolve this as follows. Write $B_1 = \frac{1}{2}\boldsymbol{t}_1 Q_1$ and $B_2 = \frac{1}{2}\boldsymbol{t}_2 Q_2$. Then equation (6) can be rewritten as

$$\begin{aligned}
(Q_1, &2B_1 Q_1^{-1}) \circ (Q_2, 2B_2 Q_2^{-1}) \\
&= (Q_1 Q_2, 2B_1 Q_1^{-1} + 2Q_1 B_2 Q_2^{-1} Q_1^{-1}) \\
&= (Q_1 Q_2, 2(B_1 Q_2 + Q_1 B_2)(Q_1 Q_2)^{-1}).
\end{aligned} \tag{7}$$

This leads to representing the pose $(Q, \boldsymbol{t})$ by a unit dual quaternion, which we now describe.

4. Dual quaternions to represent poses

A *dual quaternion* is a pair of quaternions, written as $\eta = A + \epsilon B$, with the extra algebraic operation $\epsilon^2 = 0$.

The *conjugate* dual quaternion of $\eta = \eta = A + \epsilon B$ is $\overline{\eta} = \overline{A} + \epsilon\overline{B}$. Conjugation reverses the order of multiplication:

$$\overline{\eta_1 \eta_2} = \overline{\eta}_2 \overline{\eta}_1. \tag{8}$$

There is another conjugation for dual quaternions: $(A + \epsilon B)^* = A - \epsilon B$, but we have no cause to use it in this paper, except in equation (22) below.

A *unit* dual quaternion $\eta = Q + \epsilon B$ is a dual quaternion such that $\overline{\eta}\eta = 1$, equivalently, that Q is a unit quaternion and $B \cdot Q = 0$. A *pure* dual quaternion $A + \epsilon B$ is a dual quaternion such that both A and B are pure quaternions.

If $\eta = Q + \epsilon B$ is a dual quaternion with $Q \neq 0$, then its multiplicative inverse can be calculated using the formula

$$\eta^{-1} = Q^{-1} - \epsilon Q^{-1} B Q^{-1}. \tag{9}$$

If η is a unit dual quaternion, then there is a computationally much faster formula:

$$\eta^{-1} = \overline{\eta}. \tag{10}$$

For a dual quaternion $A + \epsilon B$, it is not really possible to mix A and B additively. The quaternion A is unitless, whereas the quaternion B has units of length. For this reason, when measuring how large a dual quaternion is, everything must be with respect to l, the characteristic length scale. The *size* of a dual quaternion is defined to be

$$\|A + \epsilon B\| = \left(|A|^2 + l^{-2}|B|^2\right)^{1/2}. \tag{11}$$

A *dual number* is anything of the form $a + \epsilon b$, where a and b are real numbers. The *norm* of a dual quaternion $\eta = Q + \epsilon B$ is the dual number defined by the two steps:

$$|\eta|^2 = \overline{\eta}\eta = \eta\overline{\eta} = |Q|^2 + 2\epsilon(B \cdot Q), \tag{12}$$

$$|\eta| = \sqrt{|\eta|^2} = |Q| + \epsilon(B \cdot Q)/|Q|. \tag{13}$$

The norm preserves multiplication, that is, if η_1 and η_2 are two dual quaternions, then

$$|\eta_1 \eta_2| = |\eta_1||\eta_2|. \tag{14}$$

If $\eta = Q + \epsilon B$ is any dual quaternion with $Q \neq 0$, then we define its *normalization* to be the unit dual quaternion

$$\begin{aligned}
\widehat{\eta} = |\eta|^{-1}\eta &= \eta|\eta|^{-1} \\
&= Q/|Q| + \epsilon(B - (B \cdot Q)Q/|Q|^2). \tag{15}
\end{aligned}$$

(We remark that the normalization of a dual quaternion is used in the computer graphics industry [16, 17].) While this normalization formula might seem initially quite complicated, after thinking about it one can see that it is the simplest projection that enforces $|Q| = 1$ and $B \cdot Q = 0$.

The normalization also satisfies the following properties.

- If η is a unit dual quaternion, then $\widehat{\eta} = \eta$.

- Normalization preserves multiplication, that is, if η_1 and η_2 are two dual quaternions, then

$$\widehat{\eta_1 \eta_2} = \widehat{\eta}_1 \widehat{\eta}_2. \tag{16}$$

- If θ is pure, then

$$\widehat{(1+\theta)} = 1 + \theta + \tfrac{1}{2}\theta^2 + O(\theta^3). \tag{17}$$

(See Section 16 for the proof of this.)

Note that

$$\eta^{-1} = |\eta|^{-2}\overline{\eta}, \tag{18}$$

and this could have been the definition of the multiplicative inverse of a dual quaternion, except that this definition is circular.

Let us clarify the big-Oh notation. We say that

$$\eta(\theta) = O(\gamma(\theta)) \tag{19}$$

if there exists a constant $c > 0$ such that if $\|\theta\|$ is sufficiently small, then

$$\|\eta(\theta)\| \le c\|\gamma(\theta)\|. \tag{20}$$

We set D for the set of dual quaternions, SD for the set of unit dual quaternions, and $\mathfrak{sd}$ for the set of pure dual quaternions.

If we are given a pose represented by $(Q, \boldsymbol{t})$, then the pose is also represented by the unit dual quaternion

$$\eta = Q + \tfrac{1}{2}\epsilon \boldsymbol{t} Q. \tag{21}$$

As we have shown above, composition of poses corresponds to multiplication of unit dual quaternions. If $\boldsymbol{r}$ is a 3-vector, and $\boldsymbol{s}$ is the image of $\boldsymbol{r}$ under the action of the pose $\eta = Q + \epsilon B$, then

$$1 + \epsilon \boldsymbol{s} = \eta(1 + \epsilon \boldsymbol{r})\overline{\eta}^*, \tag{22}$$

but generally it is easier to use the formula

$$\boldsymbol{s} = Q\boldsymbol{r}\overline{Q} + 2B\overline{Q} = (Q\boldsymbol{r} + 2B)\overline{Q}. \tag{23}$$

The introduction of the factors 2 and $\frac{1}{2}$ is unnecessary, but it slightly improves formulas for the twist, as we see below.

It is difficult for a human to look at the second part of a unit dual quaternion, and from that read what the translational part of the pose should be. So dual quaternions work better for an internal representations of poses rather than human readable representations.

One more notion is the exponential of a dual quaternion [26]:

$$\exp(\theta) = \sum_{k=0}^{\infty} \frac{\theta^k}{k!}, \tag{24}$$

The exponential map takes pure dual quaternions to unit dual quaternions. In particular, if θ is a pure dual quaternion, then by equation (17) we have

$$\exp(\theta) = \widehat{(1+\theta)} + O(\theta^3). \tag{25}$$

We show how to calculate $\exp(\theta)$ in Section 15.

5. KINEMATICS: DUAL QUATERNIONS TO REPRESENT TWISTS

The translational velocity $\boldsymbol{v}$ of a translating reference frame is given by $\boldsymbol{v} = \dot{\boldsymbol{t}}$. The angular velocity $\boldsymbol{w} = (w_1, w_2, w_3)$ of a rotating frame is given by the differential equation

$$\dot{\mathsf{R}} = \mathsf{R}\boldsymbol{w}^{\star}, \tag{26}$$

or $\boldsymbol{w}^{\star} = \mathsf{R}^{-1}\dot{\mathsf{R}}$, where $\boldsymbol{w}^{\star}$ is the *Hodge star operator* of $\boldsymbol{w}$ [27]:

$$\boldsymbol{w}^{\star} = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}. \tag{27}$$

(Equation (26) gives the angular velocity with respect to the rotating reference frame. If one wants the angular velocity with respect to the stationary reference frame, use instead $\dot{\mathsf{R}} = \boldsymbol{w}^{\star}\mathsf{R}$.)

Note that

$$(\boldsymbol{w}^{\star})\boldsymbol{r} = \boldsymbol{w} \times \boldsymbol{r}, \tag{28}$$

where $\times$ denotes the cross product.

A *twist* is the pair of vectors $(\boldsymbol{w}, \boldsymbol{v})$ that describe the change of pose in the moving reference frame, that is:

$$\frac{d}{dt}\begin{bmatrix} \mathsf{R} & \boldsymbol{t} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathsf{R} & \boldsymbol{t} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \boldsymbol{w}^{\star} & \boldsymbol{v} \\ 0 & 0 \end{bmatrix}. \tag{29}$$

(Note in particular that $\dot{\boldsymbol{t}} = \mathsf{R}\boldsymbol{v}$, so that a continual application of a twist can result in a motion following a straight line, an arc of a circle, or a spiral.) The set of angular velocities, and the set of twists are both Lie algebras, often denoted by $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$ in the literature.

The quaternion-translation formulation doesn't have a representation like equation (1):

$$\frac{d}{dt}(Q, \boldsymbol{t}) = \left(\tfrac{1}{2}Q\boldsymbol{w}, Q\boldsymbol{v}\overline{Q}\right). \tag{30}$$

But with the unit dual representation, because the composition of poses is given by a bilinear multiplication, it follows that the twist is represented by the dual quaternion φ satisfying the differential equation.

$$\varphi = \eta^{-1}\dot{\eta}, \quad \text{or} \quad \dot{\eta} = \eta\varphi. \tag{31}$$

It will be shown in Section 16 (see also [12]) that φ is the pure dual quaternion

$$\varphi = \tfrac{1}{2}\boldsymbol{w} + \tfrac{1}{2}\epsilon \boldsymbol{v}. \tag{32}$$

Note that if η is a unit dual quaternion that is a function of time t, then $\eta^{-1}\dot{\eta}$ is always a pure dual quaternion. (Note that $\dot{\eta}\eta^{-1}$ is a pure dual quaternion equal to the twist relative to the fixed frame of reference. Note also that the factor $\frac{1}{2}$ attached to $\boldsymbol{v}$ in

equation (32) is a consequence of the introduction of the factor $\frac{1}{2}$ in equation (21), and thus is essentially arbitrary.)

6. Dual quaternions to represent wrenches

Let the pose η represent the reference frame that moves with the end effector. It is not necessary (although it can simplify things) that the center of mass of the end effector coincides with the origin of the moving frame.

The *wrench dual quaternion* is defined to be

$$\tau = 2\boldsymbol{q} + 2\epsilon\boldsymbol{p}, \tag{33}$$

where $\boldsymbol{q}$ and $\boldsymbol{p}$ are the torque and force, respectively, applied to the end effector at the origin of the moving frame, measured with respect to the moving frame.

If $\boldsymbol{r}_0$ is the center of mass of the end effector in the moving frame, then the twist about the center of mass is given by

$$\varphi_0 = \varphi + \tfrac{1}{2}\epsilon\boldsymbol{w} \times \boldsymbol{r}_0, \tag{34}$$

where $\varphi = \eta^{-1}\dot{\eta}$, and the wrench applied about the center of mass is

$$\tau_0 = \tau + 2\boldsymbol{p} \times \boldsymbol{r}_0. \tag{35}$$

The reason for introducing the factor 2 in definition (33) is so that the rate of change of work done to the end effector is given by

$$\frac{d}{dt}(\text{work done}) = \tau \cdot \varphi = \tau_0 \cdot \varphi_0. \tag{36}$$

See [4] for the origins of the term twist and wrench as pairs of 3-vectors, which are examples of *screws*.

7. Interpolation of poses

Suppose that we are given a sequence of times and rotations, $t_0 < t_1 < \cdots < t_n$ and $\mathsf{R}_0, \mathsf{R}_1, \ldots, \mathsf{R}_n$. We would like to find a function F from $[t_1, t_n]$ to rotations such that the third derivative of F is bounded, and $\mathsf{F}(t_k) = \mathsf{R}_k$. If the quantities $\mathsf{R}_0, \mathsf{R}_1, \ldots, \mathsf{R}_n$ were merely vectors, we could use cubic splines. But if we perform a similar calculation on the matrices, we cannot guarantee that the matrix $\mathsf{F}(t)$ is a rotation matrix.

So what we want is to find a map that projects general matrices onto rotation matrices. But it is not obvious how to find such a map that is computationally efficient. It is here that using quaternions really has great advantages over matrices, because an obvious map from general quaternions to unit quaternions is to normalize.

Let $Q_0, Q_1, \ldots, Q_n$ be the quaternion representation of $R_0, R_1, \ldots, R_n$ respectively. Let $G(t)$ be the cubic spline interpolation such that $G(t_k) = Q_k$. Then

$$F(t) = \frac{1}{|G(t)|}G(t) \tag{37}$$

provides a computationally fast, low jerk, method of interpolating the rotations.

This can be extended to interpolate poses. Given times $t_0 < t_1 < \cdots < t_n$ and unit dual quaternions $\eta_0, \eta_1, \ldots, \eta_n$, first interpolate to get a function γ such that $\gamma(t_k) = \alpha_k$, and then normalize it to get a function $\eta(t) = \widehat{\gamma(t)}$. This gives a computationally fast, low jerk, method for interpolating poses.

It is perhaps better to interpolate the rotation quaternions and the translations separately. This is because the center of gravity typically travels in a straight line rather than the arc of a curve, but naive interpolation of dual quaternions follows arcs of curves if the angular rotation is non-zero. (Interpolation of dual quaternions is more appropriate for 'skinning' [16, 17].)

If one wants to find a function that passes through these points, and has low jerk even at the end points t_0 and t_n, this can be done by adding a 'ramp up' and 'ramp down' at the ends of the trajectory. Create a free cubic spline $\gamma(t)$ on $\frac{1}{2}(t_0 + t_1) < t_1 \cdots < t_{n-1} < \frac{1}{2}(t_{n-1}+t_n)$ and $\eta_0, \eta_1, \ldots, \eta_n$. Then define a function $s : \mathbb{R} \to [\frac{1}{2}(t_0 + t_1), \frac{1}{2}(t_{n-1} + t_n)]$

$$s(t) = \begin{cases} \frac{1}{2}(t_0 + t_1) & \text{if } t < t_0 \\ t_0 + (t_1 - t_0)f\left(\frac{t-t_0}{t_1-t_0}\right) & \text{if } t_0 \le t < t_1 \\ t & \text{if } t_1 \le t < t_{n-1} \\ t_n - (t_n - t_{n-1})f\left(\frac{t_n-t}{t_n-t_{n-1}}\right) & \text{if } t_{n-1} \le t < t_n \\ \frac{1}{2}(t_{n-1} + t_n) & \text{if } t \ge t_n \end{cases} \tag{38}$$

where

$$f(t) = \tfrac{1}{2} + \tfrac{1}{2}(2 - t)t^3. \tag{39}$$

Since $f(0) = \frac{1}{2}$, $f'(0) = f''(0) = f''(1) = 0$, and $f(1) = f'(1) = 1$, it follows that $s(t)$ has bounded third derivative, $s(t_k) = t_k$ for $1 \le k \le n - 1$, $s(t) = \frac{1}{2}(t_0 + t_1)$ for $t \le t_0$, and $s(t) = \frac{1}{2}(t_{n-1} + t_n)$ for $t \ge t_n$. Thus

$$\eta(t) = \widehat{\gamma(s(t))} \tag{40}$$

provides a low jerk function, even at the end points, that passes through the appropriate points. it is important that $t_1 - t_0$ and $t_n - t_{n-1}$ be large enough to allow the trajectory to 'ramp up' and 'ramp down' smoothly.

8. Perturbations of poses

Suppose one has a pose $\eta = \eta(t)$ that varies in time, and a fixed time t_0. Then we would like to find a good representation of the perturbation of $\eta(t)$ from a reference pose $\eta_r = \eta(t_0)$ when t is close to t_0. One way to do this is to choose a new reference frame in which $\eta_r = 1$, the identity pose, or rather, to think about how $\eta_r^{-1}\eta$ behaves.

It can be seen that for t close to t_0 that there is a pure dual quaternion θ such that

$$\eta = \eta_r \widehat{(1+\theta)} \approx \eta_r(1+\theta). \qquad (41)$$

Thus

$$\theta \approx \eta_r^{-1}\eta - 1 = \overline{\eta}_r\eta - 1 \qquad (42)$$

gives a great way to measure the perturbation.

We prefer instead to use what we shall call the *Lie difference*, which we denote using $\triangle$, which is close to (42), but is always a pure dual quaternion:

$$\theta \approx \eta \triangle \eta_r = \tfrac{1}{2}(\overline{\eta}_r\eta - \overline{\eta}\eta_r) = \mathrm{Im}(\overline{\eta}_r\eta). \qquad (43)$$

Note that if θ_1 and θ_2 are pure dual quaternions, and η is an invertible dual quaternion, then

$$\begin{aligned}
& (\eta(1+\theta_1)) \triangle (\eta(1+\theta_2)) \\
&= (1+\theta_1) \triangle (1+\theta_2) \qquad (44) \\
&= \theta_1 - \theta_2.
\end{aligned}$$

Thus these give a great way to map perturbations of non-linear poses to linear pure dual quaternions. These can be very effective in control theory.

9. Dual quaternions as vectors

We make the identifications

$$D \cong \mathbb{R}^8, \qquad (45)$$

using the basis

$$\begin{aligned}
\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8\} \\
= \{\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}, \epsilon\boldsymbol{i}, \epsilon\boldsymbol{j}, \epsilon\boldsymbol{k}, 1, \epsilon\}, \quad (46)
\end{aligned}$$

and similarly, we make the identification

$$\mathfrak{sd} \cong \mathbb{R}^6, \qquad (47)$$

using the basis $\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\}$. With these identifications, we can define the dot product between two dual quaternions by transferring the usual definition of dot product on $\mathbb{R}^8$, that is

$$(A + \epsilon B) \cdot (C + \epsilon D) = A \cdot C + B \cdot D. \qquad (48)$$

In this way, every dual quaternion η can be written in component form as

$$\eta = \sum_{i=1}^8 \eta_i \beta_i, \qquad (49)$$

and every pure dual quaternion θ as

$$\theta = \sum_{i=1}^6 \theta_i \beta_i. \qquad (50)$$

10. Lie derivatives

The notion of the Lie derivative, sometimes in our context called the directional derivative, is a combination of two ideas that may be found in the literature. First is the concept of a Lie derivative with respect to a vector field [28, 30]. Secondly, the definition of the Lie algebra is that it is the vector space of vector fields that are invariant under left multiplication by elements of the Lie group [20, 29]. In this way, we can define the Lie derivative of a function with respect to an element of the Lie algebra. One place in the literature where they are combined is in [13, equation (5), Chapter II].

These standard abstract definitions can be made more concrete in our special case where the Lie group is the set of unit dual quaternions, and the Lie algebra is the set of pure dual quaternions.

If one has a quantity that is a function of pose $g(\eta)$, then we usually think of its derivative as the Jacobian with respect to the components of η. But it really makes more sense to compute the derivative with respect to the components of the perturbation of η. The latter is the Lie derivative.

The definition is this. Given a differentiable function g whose domain is the unit dual quaternions, SD, we can extend it arbitrarily to a differentiable function whose domain is an open neighborhood of SD in D. Given a unit dual quaternion η and a pure dual quaternion θ, we define the *Lie derivative* of $g(\eta)$ in the direction of θ to be

$$\mathcal{L}_\theta g = \lim_{r \to 0} \frac{g(\eta(1+r\theta)) - g(\eta)}{r} = \frac{d}{dr}g(\eta(1+r\theta))\Big|_{r=0}. \qquad (51)$$

Since $\eta(1+r\theta)$ isn't necessarily a unit dual quaternion, it is not obvious that the definition of the directional Lie derivative doesn't depend upon how the domain of g was extended from SD, but it is, as is shown in Lemma 1 below.

Given a generic function g whose domain is the dual quaternions, D, we define its Jacobian to be the dual quaternion

$$\frac{\partial g}{\partial \eta} = \sum_{i=1}^8 \frac{\partial g}{\partial \eta_i} \beta_i. \qquad (52)$$

If its domain is the pure dual quaternions, $\mathfrak{sd}$, we have the same formula except with 8 replaced by 6.

Using the chain rule for partial derivatives, we obtain the following formula, which is useful for explicitly calculating the Lie derivative if the function g is known.

$$\mathcal{L}_\theta g = \frac{\partial g}{\partial \eta} \cdot (\eta\theta). \tag{53}$$

We define the *partial Lie derivatives* to be

$$\mathcal{L}_i g = \mathcal{L}_{\beta_i} g, \quad (1 \le i \le 6), \tag{54}$$

and its *full Lie derivative* to be the pure dual quaternion

$$\mathcal{L}g = \sum_{i=1}^{6} \mathcal{L}_i g \beta_i, \tag{55}$$

so that for all pure dual quaternions θ it satisfies:

$$\theta \cdot \mathcal{L}g = \mathcal{L}_\theta g. \tag{56}$$

To gain some intuition, write

$$\theta = \tfrac{1}{2}\boldsymbol{a} + \tfrac{1}{2}\epsilon\boldsymbol{b}, \tag{57}$$

so that

$$\frac{\partial}{\partial\theta} = 2\frac{\partial}{\partial\boldsymbol{a}} + 2\epsilon\frac{\partial}{\partial\boldsymbol{b}}. \tag{58}$$

Then we see that θ represents a change in pose by an infinitesimal translation $\boldsymbol{b}$ and an infinitesimal rotation $\boldsymbol{a}$, measured in the moving frame of reference. Thus $\mathcal{L}_\theta g$ is a pure dual quaternion giving twice the change in g with respect to an infinitesimal rotation, plus ϵ times twice the change of g with respect to an infinitesimal translation.

One important property of the Lie derivative is that if η represents a pose, with twist φ, then

$$\frac{d}{dt}[g(\eta)] = \mathcal{L}_\varphi g. \tag{59}$$

The Lie derivative satisfies various rules, which are also useful for explicitly calculating the Lie derivative when g is known.

- If $g(\eta)$ is linear in η, then

$$\mathcal{L}_\theta g(\eta) = g(\eta\theta). \tag{60}$$

- The product rule: if $\times$ is any product, such as inner product or dual quaternion product, then

$$\mathcal{L}_\theta(g_1 \times g_2) = g_1 \times (\mathcal{L}_\theta g_2) + (\mathcal{L}_\theta g_1) \times g_2. \tag{61}$$

- The chain rule:

$$\mathcal{L}_\theta(h(g_1, g_2, \ldots, g_m))$$
$$= \sum_{i=1}^{m} \frac{\partial}{\partial g_i} h(g_1, g_2, \ldots, g_m)\mathcal{L}_\theta g_i. \tag{62}$$

11. Applications to parallel robots

Suppose that the position of the end effector of a parallel robot is given by n actuators, described by quantities

$$\boldsymbol{\ell} = (\ell_j)_{1 \le j \le n}. \tag{63}$$

For example, for a cable-driven parallel robot, these represent the lengths of the cables, and typically $n = 8$. For the Gough or Stewart Platforms [10], we have $n = 6$.

Let us also denote the force exerted by the actuators by

$$\boldsymbol{f} = (f_j)_{1 \le j \le n}, \tag{64}$$

defined so that the rate of change of work performed through the actuators is given by

$$\frac{d}{dt}(\text{work done}) = \boldsymbol{f} \cdot \dot{\boldsymbol{\ell}}. \tag{65}$$

Suppose we have a function $\mathsf{L} : SD \to \mathbb{R}^n$, which calculates the required actuator values, $\boldsymbol{\ell}$, from the pose η of the moving frame. This is the *inverse kinematics* function. If it is a simple cable-driven parallel robot, this can be computed using Pythagoras's Theorem, from the positions of the actuators on the fixed frame, and the positions of where the cables are attached to end effector. But this formula could be more complicated in other situations.

We also define the $(n \times 6)$ matrix Λ by

$$\Lambda\theta = \mathcal{L}_\theta\mathsf{L}. \tag{66}$$

From equation (59) we obtain

$$\dot{\boldsymbol{\ell}} = \Lambda\varphi. \tag{67}$$

There is also a $(6 \times n)$ matrix T that maps the actuator forces to the wrench dual quaternion:

$$\tau = \mathsf{T}\boldsymbol{f}. \tag{68}$$

This can be computed by balancing the force and torque exerted upon the end-effector. But it can also be computed with the following important identity:

$$\mathsf{T} = \Lambda^T. \tag{69}$$

This is because the rate of change of work done on the parallel robot can be computed in two different ways, either using equation (36), or (65). Substituting in equations (67) and (68), we obtain

$$\mathsf{T}\boldsymbol{f} \cdot \varphi. = \boldsymbol{f} \cdot \Lambda\varphi. \tag{70}$$

Since this is true for arbitrary actuator forces $\boldsymbol{f}$ and end effector twists φ, the result follows.

12. Second lie derivatives

If g is a function of dual quaternions, we define its Hessian to be the (8×8) matrix

$$\left[\frac{\partial^2 g}{\partial \eta^2}\right] = \left[\frac{\partial^2 g}{\partial \eta_i \partial \eta_j}\right]_{1 \leq i,j \leq 8}. \tag{71}$$

Thus the expression $\left[\frac{\partial^2 g}{\partial \eta^2}\right] \gamma$ should be interpreted with γ treated as an eight dimensional vector.

We also need to compute second Lie derivatives. Calculations show that

$$\mathcal{L}_\theta \mathcal{L}_\psi g = (\eta\psi) \cdot \left[\frac{\partial^2 g}{\partial \eta^2}\right](\eta\theta) + \frac{\partial g}{\partial \eta} \cdot (\eta\theta\psi). \tag{72}$$

As a corollary, we obtain the well known identity:

$$\mathcal{L}_\theta \mathcal{L}_\psi g - \mathcal{L}_\psi \mathcal{L}_\theta g = \mathcal{L}_{(\theta\psi - \psi\theta)}g, \tag{73}$$

which implies that Lie derivatives do not necessary commute.

Another way one might try to define the second derivative is to use the formula $\left.\frac{\partial^2}{\partial\theta^2}g(\eta(1+\theta))\right|_{\theta=0}$. Unfortunately, this definition doesn't work, as it depends upon the choice of how to extend the domain of g to all dual quaternions. The obvious choice of extension is to use the normalization:

$$\tilde{g}(\eta) = g(\hat{\eta}). \tag{74}$$

We have the following formula for the Hessian of $\tilde{g}$:

$$\left.\frac{\partial^2}{\partial\theta^2}\tilde{g}(\eta(1+\theta))\right|_{\theta=0}$$
$$= \left[\tfrac{1}{2}(\mathcal{L}_i\mathcal{L}_j g(\eta) + \mathcal{L}_j\mathcal{L}_i g(\eta))\right]_{1 \leq i,j \leq 6}. \tag{75}$$

This equation is proved in Section 16.

13. Forward kinematics for parallel robots

Let the *set of admissible actuator values*, $\mathbb{L} \subset \mathbb{R}^n$, be the range of the function L. Then the *forward kinematics* function is

$$\mathsf{Y} : \mathbb{L} \to SD, \tag{76}$$

which is a left inverse to L. Because of possible measurement errors, Y should produce decent answers even if the actuator values are merely close to $\mathcal{L}$.

We will focus on the over-constrained problem, that is, when the number of actuators n is greater than 6. If $n = 6$, a much simpler approach is possible by simply applying the Newton-Raphson method directly to L. In that case, only first derivatives of L are required.

This problem has been solved by many others, for example, [22, 31]. But we feel that this is much more easily solved using dual quaternions. The function Y takes actuator values $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_n)$, and seeks to find the pose η so that $\mathsf{L}(\eta)$ is close as possible to $\boldsymbol{\ell}$. We do this by seeking to minimize the *loss function*

$$b(\eta) = \tfrac{1}{2}|\mathsf{L}(\eta) - \boldsymbol{\ell}|^2 \tag{77}$$

We use the Newton-Raphson Method, which given η_k close to a minimum of the loss function, finds η_{k+1} which is much closer to the same minimum of the loss function. Because this method as usually stated works only on linear vector spaces, we first have to define a map $\theta \mapsto \tilde{b}_k(\theta)$, with $\tilde{b}_k(0) = \eta_k$, from the linear vector space of pure dual quaternions to the manifold of unit dual quaternions [15].

Most papers on the Newton-Raphson Method on manifolds construct this map using the so called exp function [7, 8, 9]. So the map is

$$\tilde{b}_k(\theta) = b(\eta_k \exp(\theta)). \tag{78}$$

The exp map in these papers is following the path of a geodesic on the manifold, and this is equivalent to using the equations of motion of the end effector as described in Section 14. Another exp map is to follow a one-parameter subgroup, or equivalently, equation (24). We do not take these approaches, as they can be computationally expensive. Our approach is to normalize:

$$\tilde{b}_k(\theta) = b(\eta_k \widehat{(1+\theta)}). \tag{79}$$

This is numerically close to the second approach, as is shown by equation (25).

In order to apply the Newton-Raphson, we need to compute the Jacobian

$$\delta_k = \left.\frac{\partial \tilde{b}_k}{\partial \theta}\right|_{\theta=0} = (\mathcal{L}_i b(\eta_k))_{1 \leq i \leq 6}, \tag{80}$$

and its Hessian H_k, which by equation (75) is

$$\mathsf{H}_k = \left[\tfrac{1}{2}(\mathcal{L}_i\mathcal{L}_j b(\eta_k) + \mathcal{L}_j\mathcal{L}_i b(\eta_k))\right]_{1 \leq i,j \leq 6}. \tag{81}$$

The Newton-Raphson method we use is to iterate:

$$\eta_{k+1} = \eta_k \widehat{(1 - \mathsf{H}_k^{-1}\delta_k)}. \tag{82}$$

For the criterion of when to terminate the Newton-Raphson Method, we measure $\|\eta_{k+1} - \eta_k\|$.

This method works well in practice, because typically we measure the pose frequently, and thus the current pose won't be much different than the previously measured pose. Thus the previously measured pose is a good starting value for using the Newton-Raphson method for finding the current pose. We found that only about two or three iterations of the Newton-Raphson method were generally required, and that it easily ran at a thousand times per second on a modern computer.

14. Dynamics of the end effector

We define the *no-load forces* to be the actuator forces if there is no end effector present:

$$\boldsymbol{f}_0 = \mathsf{M}_0 \ddot{\boldsymbol{\ell}}, \tag{83}$$

where M_0 is a positive definite $(n \times n)$ matrix denoting what we shall call the *effective no-load mass of the actuators*.

Let us suppose that the kinetic energy of the parallel robot is given by

$$e = \tfrac{1}{2}\varphi \cdot \mathsf{M}\varphi, \tag{84}$$

where M is a (6×6) positive definite matrix, which depends only upon η, and which we call the *effective mass of the parallel robot*. If m_e is the mass of the end effector, M_e is the moment of inertia tensor of the end effector about its center of mass, and $\boldsymbol{r}_0$ is the center of mass of the end effectors, all measured with respect to the moving frame, then

$$e = \tfrac{1}{2}m_e|\boldsymbol{v} + \boldsymbol{w} \times \boldsymbol{r}_0|^2 + \tfrac{1}{2}\boldsymbol{w} \cdot \mathsf{M}_e\boldsymbol{w} + \tfrac{1}{2}\dot{\boldsymbol{\ell}} \cdot \mathsf{M}_0\dot{\boldsymbol{\ell}}, \tag{85}$$

that is

$$\mathsf{M} = 4\begin{bmatrix} \mathsf{M}_e - m_e(\boldsymbol{r}_0^\star)^2 & m_e\boldsymbol{r}_0^\star \\ -m_e(\boldsymbol{r}_0^\star) & m_e\mathsf{I} \end{bmatrix} + \Lambda^T\mathsf{M}_0\Lambda, \tag{86}$$

where I is the (3×3) identity matrix.

The next result is proved in Section 16.

Theorem 1. *If the kinetic energy satisfies equation (84) with equation (86) holding, and the potential energy v is calculated in the usual manner from the mass of the end effector in a constant gravitational field $\boldsymbol{g}$, then the equation of motion is*

$$\tau = \mu + \mathsf{M}\alpha, \tag{87}$$

where

$$\mu = 2\boldsymbol{w} \times (\mathsf{M}_e\boldsymbol{w}) + 2\epsilon m_e\boldsymbol{w} \times \boldsymbol{v}$$
$$+ 2m_e((\boldsymbol{w}\cdot\boldsymbol{r}_0)(\boldsymbol{w}\times\boldsymbol{r}_0) + \boldsymbol{r}_0\times(\boldsymbol{w}\times\boldsymbol{v}) + \epsilon\boldsymbol{w}\times(\boldsymbol{w}\times\boldsymbol{r}_0))$$
$$+ \Lambda^T\mathsf{M}_0(\mathcal{L}_\varphi\Lambda)\varphi - 2m_e(\boldsymbol{r}_0 \times \tilde{\boldsymbol{g}} + \epsilon\tilde{\boldsymbol{g}}), \tag{88}$$

with

$$\tilde{\boldsymbol{g}} = \overline{Q}\boldsymbol{g}Q, \tag{89}$$

and

$$\mathsf{M}\alpha = 2\mathsf{M}_e\dot{\boldsymbol{w}} + 2m_e(\boldsymbol{r}_0\times\dot{\boldsymbol{v}}) + 2\epsilon m_e\dot{\boldsymbol{v}} + \Lambda^T\mathsf{M}_0\Lambda\alpha. \tag{90}$$

The various terms in equation (88) can be interpreted as follows.

- $\mathsf{M}_e\dot{\boldsymbol{w}}$ and $m_e\dot{\boldsymbol{v}}$ are inertial resistance to change of angular and translational velocities.
- $m_e\boldsymbol{w} \times \boldsymbol{v}$ is the centripetal force required to rotate and move at the same time.

- $\boldsymbol{w} \times (\mathsf{M}_e\boldsymbol{w})$ is the precession torque (so that if the moment of inertia is not isotropic, then the body spins in a counter-intuitive manner, see, for example, [18]).

-

$$\Lambda^T\mathsf{M}_0(\mathcal{L}_\varphi\Lambda)\varphi + \Lambda^T\mathsf{M}_0\Lambda\alpha = \mathsf{T}\boldsymbol{f}_0 \tag{91}$$

  is the wrench required to move the actuators, where the no-load forces may be computed using

$$\boldsymbol{f}_0 = \mathsf{M}_0\mathcal{L}_\varphi(\Lambda\varphi) + \mathsf{M}_0\Lambda\alpha. \tag{92}$$

- $m_e\tilde{\boldsymbol{g}}$ is the force due to gravity, translated into the moving frame of reference.
- All terms containing $\boldsymbol{r}_0$ are corrections required since the center of gravity isn't necessarily the same as the origin of the moving frame of reference. They could be derived by first finding the equations of motion when $\boldsymbol{r}_0 = 0$, and then applying equations (34) and (35).

15. Spherical linear interpolation of dual quaternions

Suppose we are given two unit dual quaternions, η_1 and η_2?

The problem is to find the dual quaternion valued function of time, denoted $\text{slerp}(\eta_1, \eta_2, t)$, such that at $t = 0$, it takes the value η_1, that at $t = 1$, it takes the value η_2, and such that its twist, θ, is a constant pure dual quaternion. The name of the function, 'slerp', comes from the equivalent problem for orientations [24].

This requires two functions. The first is the exponential function, defined in equation (24). The function $\exp(t\theta)$ as function of time t is the pose traveled if it is the identity pose at $t = 0$, and maintains a constant twist θ. The second is a logarithm function, which given a unit dual quaternion η, finds θ such that $\exp(\theta) = \eta$. There are infinitely many possibilities for θ, so we will just describe the principal value, that is, the one for which the norm of the angular velocity is minimized. Then

$$\text{slerp}(\eta_1, \eta_2, t) = \eta_1 \exp(t\log(\overline{\eta}_1\eta_2))$$
$$= \exp(t\log(\eta_2\overline{\eta}_1))\eta_1. \tag{93}$$

If one is merely interested in interpolating between the poses represented by these unit dual quaternions, multiply one of them by ± 1 so that non-dual part of $\eta_1\eta_2^{-1}$ has non-negative real part.

The results of this section may be found in [21]. We have seen similar results in other places, but the ones we have found seem to have errors.

Given any dual quaternion

$$\eta = c + \boldsymbol{d} + \epsilon(x + \boldsymbol{y}), \qquad (94)$$

by decomposing $\boldsymbol{y}$ into a vector parallel to $\boldsymbol{d}$ and perpendicular to $\boldsymbol{d}$, we can suppose without loss of generality that there exists orthonormal vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ such that η is a linear combination of 1, $\boldsymbol{a}$, ϵ, $\boldsymbol{a}\epsilon$, and $\boldsymbol{b}\epsilon$.

Theorem 2. *If $\boldsymbol{a}$ and $\boldsymbol{b}$ are orthonormal vectors, then if $w \neq 0$*

$$\exp\left(\tfrac{1}{2}wt\boldsymbol{a} + (\tfrac{1}{2}v_1 t\boldsymbol{a} + \tfrac{1}{2}v_2 t\boldsymbol{b})\epsilon\right)$$
$$= \left(\cos(\tfrac{1}{2}wt) + \sin(\tfrac{1}{2}wt)\boldsymbol{a}\right)\left(1 + \tfrac{1}{2}v_1 t\boldsymbol{a}\epsilon\right) \qquad (95)$$
$$+ \frac{v_2}{w}\sin(\tfrac{1}{2}wt)\boldsymbol{b}\epsilon,$$

and if $w = 0$

$$\exp\left(\tfrac{1}{2}vt\boldsymbol{a}\epsilon\right) = 1 + \tfrac{1}{2}vt\boldsymbol{a}\epsilon. \qquad (96)$$

If $w \neq 0$, this represents a pose that rotates counterclockwise with angular velocity w around $\boldsymbol{a}$, and translates by

$$v_1 t\boldsymbol{a} + \frac{v_2}{w}\left(\sin(wt)\boldsymbol{b} + (1 - \cos(wt))(\boldsymbol{a} \times \boldsymbol{b})\right),$$

that is, the sum of a point on a line in the direction of $\boldsymbol{a}$ moving with speed v_1, and a point on a circle in the plane perpendicular to $\boldsymbol{a}$ of radius v_2/w which is traversed once every $2\pi/w$ time units. If $w = 0$, this represents a pose with the identity rotation and translating in a straight line with velocity v.

Theorem 3. *Suppose $\boldsymbol{a}$ and $\boldsymbol{b}$ are orthonormal vectors, and that*

$$\eta = c + s\boldsymbol{a} + \epsilon(x + y_1\boldsymbol{a} + y_2\boldsymbol{b}) \qquad (97)$$

is a unit dual quaternion, with $c \neq -1$. Let

$$t = \operatorname{atan2}(s, c), \qquad (98)$$

that is, the angle part of the polar coordinates of (c, s), with $|t| < \pi$. Then if $t \neq \pi$ then

$$\log(\eta) = t\boldsymbol{a} + \epsilon(cy_1 - sx)\boldsymbol{a} + \epsilon\frac{ty_2}{s}\boldsymbol{b}, \qquad (99)$$

where if $s = 0$ we assume $y_0 = 0$, and we set $y_2/s = 0$.

It is not possible to assign a canonical principal value for $\log(-1)$, because it could be $\pi\boldsymbol{a}$ for any unit vector $\boldsymbol{a}$.

Proof of Equation (17). Since $\bar{\theta} = -\theta$, we have

$$|1 + \theta|^2 = 1 - \theta^2. \qquad (100)$$

Hence using Taylor's series

$$|1 + \theta|^{-1} = 1 + \tfrac{1}{2}\theta^2 + O(\theta^3) \qquad (101)$$

from which it follows that

$$\widehat{(1 + \theta)} = (1+\theta)(1+\theta)^{-1} = 1+\theta+\tfrac{1}{2}\theta^2+O(\theta^3). \qquad (102)$$
$$\square$$

Proof of Equation (32). First, if we differentiate $\bar{\eta}\eta = 1$, we obtain $\varphi + \bar{\varphi} = 0$, that is, φ is pure.

Suppose the position $\boldsymbol{s}$ is the image of the constant position $\boldsymbol{r}$ under the pose represented by $(\mathsf{R}, \boldsymbol{t})$ and also represented by $\eta = Q + \epsilon B$. Differentiating equation (2) we obtain

$$\dot{\boldsymbol{s}} = \mathsf{R}\boldsymbol{w}^\star\boldsymbol{r} + \mathsf{R}\boldsymbol{v} = \mathsf{R}(\boldsymbol{w} \times \boldsymbol{r} + \boldsymbol{v}). \qquad (103)$$

Differentiating equation (22) we obtain

$$\epsilon\dot{\boldsymbol{s}} = \frac{d}{dt}\left(\eta(1 + \epsilon\boldsymbol{r})\bar{\eta}^*\right)$$
$$= \eta\left(\varphi(1 + \epsilon\boldsymbol{r}) + (1 + \epsilon\boldsymbol{r})\bar{\varphi}^*\right)\bar{\eta}^*. \qquad (104)$$

After some manipulation, and setting $\varphi = \tfrac{1}{2}\boldsymbol{a} + \tfrac{1}{2}\epsilon\boldsymbol{b}$, we obtain

$$\dot{\boldsymbol{s}} = Q(\boldsymbol{a} \times \boldsymbol{r} + \boldsymbol{b})\bar{Q}. \qquad (105)$$

Comparing equations (103) and (105), we obtain

$$\boldsymbol{w} \times \boldsymbol{r} + \boldsymbol{v} = \boldsymbol{a} \times \boldsymbol{r} + \boldsymbol{b}, \qquad (106)$$

and since this holds for all $\boldsymbol{r}$, the result follows. $\square$

Lemma 1. *The definition of $\mathcal{L}_\theta g$ in equation (51) does not depend upon the extension of g from SD to a neighborhood of SD in D.*

Proof. Let g_1 and g_2 be two extensions of g from SD to a neighborhood of SD in D. Define

$$\tilde{g}(\eta) = g_1(\hat{\eta}) = g_2(\hat{\eta}). \qquad (107)$$

Then

$$\frac{d}{dr}\tilde{g}(\eta(1 + r\theta))\Big|_{r=0} = \frac{d}{dr}g_1(\widehat{\eta(1 + r\theta)})\Big|_{r=0}$$
$$= \frac{d}{dr}g_1(\eta(1 + r\theta))\Big|_{r=0}, \qquad (108)$$

where the second equality follows from equation (17). Similarly for g_2. $\square$

Proof of Equation (75). We wish to find the Jacobian δ and the Hessian H of $\tilde{b}$ at the origin, where

$$\tilde{b}(\theta) = b(\eta\widehat{(1 + \theta)}). \qquad (109)$$

We can find this by considering its Taylor series expansion

$$\tilde{b}(\theta) = \tilde{b}(0) + \sum_{i=1}^{6} \delta_i \theta_i + \frac{1}{2} \sum_{i,j=1}^{6} \mathsf{H}_{i,j} \theta_i \theta_j + O(\theta^3), \quad (110)$$

where

$$\theta = \sum_{i=1}^{6} \theta_i \beta_i. \quad (111)$$

Using the Taylor series, and using equation (17), one obtains

$$\tilde{b}(\theta) = b(\eta \widehat{(1+\theta)}) = b(\eta(1 + \theta + \tfrac{1}{2}\theta^2 + O(\theta^3)))$$

$$= b(\eta) + \frac{\partial b}{\partial \eta} \cdot (\eta(\theta + \tfrac{1}{2}\theta^2))$$

$$+ \tfrac{1}{2}(\eta\theta) \cdot \left[\frac{\partial^2 b}{\partial \eta^2}\right](\eta\theta) + O(\theta^2)$$

$$= b(\eta) + \frac{\partial b}{\partial \eta} \cdot (\eta(\theta))$$

$$+ \tfrac{1}{2}(\eta\theta) \cdot \left[\frac{\partial^2 b}{\partial \eta^2}\right](\eta\theta) + \frac{\partial b}{\partial \eta} \cdot (\tfrac{1}{2}\eta\theta^2) + O(\theta^3). \quad (112)$$

Now by comparing coefficients, and considering equations (53) and (72), we obtain

$$\delta_i = \mathcal{L}_i b(\eta), \quad (1 \le i \le 6), \quad (113)$$

and

$$\mathsf{H}_{i,j} = (\eta\beta_i) \cdot \left[\frac{\partial^2 b}{\partial \eta^2}\right](\eta\beta_j) + \frac{\partial b}{\partial \eta} \cdot (\tfrac{1}{2}\eta(\beta_i\beta_j + \beta_j\beta_i))$$

$$= \tfrac{1}{2}(\mathcal{L}_i\mathcal{L}_j b(\eta) + \mathcal{L}_j\mathcal{L}_i b(\eta)), \quad (1 \le i,j \le 6). \quad (114)$$

$\square$

Now we work on proving Theorem 1. We use the Euler-Lagrange equations. However, some may consider it easier to use standard formulas for rotating bodies, and Newtonian physics, to obtain the same result.

Define the *cross product* of two pure dual quaternions $\alpha = \boldsymbol{a} + \epsilon\boldsymbol{b}$ and $\beta = \boldsymbol{c} + \epsilon\boldsymbol{d}$ by

$$\alpha \times \beta = \tfrac{1}{2}(\alpha\beta - \beta\alpha) = \boldsymbol{a} \times \boldsymbol{c} + \epsilon(\boldsymbol{a} \times \boldsymbol{d} + \boldsymbol{b} \times \boldsymbol{c}). \quad (115)$$

Define the *adjoint products* of pure dual quaternions by

$$\alpha \rtimes \beta = \boldsymbol{c} \times \boldsymbol{a} + \boldsymbol{d} \times \boldsymbol{b} + \epsilon(\boldsymbol{c} \times \boldsymbol{b}), \quad (116)$$

$$\alpha \ltimes \beta = -\beta \rtimes \alpha, \quad (117)$$

which can also be defined by the property that for all pure dual quaternions α, β, and γ we have

$$(\alpha \times \beta) \cdot \gamma = \alpha \cdot (\gamma \rtimes \beta) = \beta \cdot (\alpha \ltimes \gamma). \quad (118)$$

Theorem 4. *If the kinetic energy e satisfies equation (84), and $v = v(\eta)$ denotes the potential energy, then the equation of motion is*

$$\tau = \mu_1 + \mu_2 + \mathsf{M}\alpha, \quad (119)$$

where

$$\mu_1 = \mathcal{L}_\varphi \mathsf{M}\varphi - \tfrac{1}{2}\mathcal{L}(\varphi \cdot \mathsf{M}\varphi) + \mathcal{L}v, \quad (120)$$

and for any constant pure dual quaternion ψ we have

$$\psi \cdot \mu_2 = 2\varphi \cdot \mathsf{M}(\psi \times \varphi), \quad (121)$$

that is,

$$\mu_2 = 2(\mathsf{M}\varphi) \rtimes \varphi. \quad (122)$$

Proof. In preparation to apply the Euler-Lagrange Equation, given $\eta_0 \in SD$, we define a map from an open neighborhood of the origin in $\mathbb{R}^6$ to an open neighborhood of η_0 in SD

$$\theta \mapsto \eta(\theta)$$
$$= \eta_0 \widehat{(1+\theta)} \quad (123)$$
$$= \eta_0(1 + \theta + \tfrac{1}{2}\theta^2) + O(\theta^3),$$

where in the last inequality we used equation (17). Then we have

$$\varphi = \eta^{-1}\dot{\eta}$$
$$= \widehat{(1+\theta)}^{-1}(\dot{\theta} + \tfrac{1}{2}(\dot{\theta}\theta + \theta\dot{\theta})) + O(\theta^2) \quad (124)$$
$$= \dot{\theta} + \tfrac{1}{2}(\dot{\theta}\theta - \theta\dot{\theta}) + O(\theta^2),$$

and

$$\alpha = \ddot{\theta} + O(\theta). \quad (125)$$

The Lagrangian of the parallel robot is

$$l = e - v = \tfrac{1}{2}\varphi \cdot \mathsf{M}\varphi - v. \quad (126)$$

We use the local coordinate system θ given by equation (123). The Euler-Lagrange Equation [3, 11] tells us

$$\frac{d}{dt}\left(\frac{\partial l}{\partial \dot{\theta}}\right) - \frac{\partial l}{\partial \theta} = \tau. \quad (127)$$

We suppose that $\eta_0 = \eta(t_0)$, and $\theta(t_0) = 0$, and from now on in this proof, all equations are stated assuming the condition $t = t_0$. Thus we only prove our results when $t = t_0$. But since t_0 is arbitrary, this is not a limitation. However, it is important that derivatives are calculated before setting $t = t_0$. In particular, this means that for any function f of η that

$$\frac{\partial f}{\partial \theta} = \mathcal{L}f. \quad (128)$$

12

We have

$$\frac{d}{dt}\left(\frac{\partial e}{\partial \dot\theta}\right) = \mathsf{M}\alpha + \dot{\mathsf{M}}\varphi$$

$$= \mathsf{M}\alpha + \left(\dot\theta \cdot \frac{\partial \mathsf{M}}{\partial \theta}\right)\dot\theta \qquad (129)$$

$$= \mathsf{M}\alpha + \left(\varphi \cdot \frac{\partial \mathsf{M}}{\partial \theta}\right)\varphi,$$

and

$$\frac{\partial e}{\partial \theta} = \dot\theta \cdot \mathsf{M}\frac{\partial}{\partial \theta}(\dot\theta\theta - \theta\dot\theta) + \tfrac{1}{2}\dot\theta \cdot \left(\frac{\partial \mathsf{M}}{\partial \theta}\right)\dot\theta$$

$$= \varphi \cdot \mathsf{M}\frac{\partial}{\partial \theta}(\varphi\theta - \theta\varphi) + \tfrac{1}{2}\varphi \cdot \left(\frac{\partial \mathsf{M}}{\partial \theta}\right)\varphi. \qquad (130)$$

Note that if f is any linear function whose domain is the pure dual quaternions, then

$$\psi \cdot \left(\frac{\partial}{\partial \theta} f(\theta)\right) = f(\psi). \qquad (131)$$

Thus taking the dot product of equation (130) with any constant pure dual quaternion ψ, we obtain the result. $\qquad\square$

Lemma 2. *If v is the potential energy of the end effector in a constant gravity field $\boldsymbol{g}$, then*

$$\mathcal{L}v = -2m_0(\boldsymbol{r}_0 \times \tilde{\boldsymbol{g}} + \epsilon\tilde{\boldsymbol{g}}), \qquad (132)$$

where

$$\tilde{\boldsymbol{g}} = \overline{Q}\boldsymbol{g}Q. \qquad (133)$$

Proof. By equation (23), the position of the end effector's center of mass in the stationary frame of reference is

$$\boldsymbol{h} = Q\boldsymbol{r}_0\overline{Q} + 2B\overline{Q}, \qquad (134)$$

where the pose is $\eta = Q + \epsilon B$. Hence the potential energy is

$$v(\eta) = -m_0\boldsymbol{g} \cdot \boldsymbol{h}. \qquad (135)$$

Thus the result follows if for any constant pure dual quaternion $\psi = \tfrac{1}{2}\boldsymbol{a} + \tfrac{1}{2}\epsilon\boldsymbol{b}$, we have

$$\mathcal{L}_\psi(\boldsymbol{g} \cdot \boldsymbol{h}) = 2\psi \cdot (\boldsymbol{r}_0 \times \tilde{\boldsymbol{g}} + \epsilon\tilde{\boldsymbol{g}}). \qquad (136)$$

To see this, first note that

$$\eta\psi = \tfrac{1}{2}Q\boldsymbol{a} + \tfrac{1}{2}(B\boldsymbol{a} + Q\boldsymbol{b}), \qquad (137)$$

from which it follows by equation (60) that

$$\mathcal{L}_\psi Q = \tfrac{1}{2}Q\boldsymbol{a} \qquad (138)$$

$$\mathcal{L}_\psi B = \tfrac{1}{2}(B\boldsymbol{a} + Q\boldsymbol{b}). \qquad (139)$$

Since $\overline{\boldsymbol{a}} = -\boldsymbol{a}$, and $\overline{\boldsymbol{b}} = -\boldsymbol{b}$, we have

$$\mathcal{L}_\psi \boldsymbol{h}$$
$$= \tfrac{1}{2}(Q\boldsymbol{a}\boldsymbol{r}_0\overline{Q} - Q\boldsymbol{r}_0\boldsymbol{a}\overline{Q}$$
$$\quad + 2B\boldsymbol{a}\overline{Q} + 2Q\boldsymbol{b}\overline{Q} - 2B\boldsymbol{a}\overline{Q}) \qquad (140)$$
$$= Q(\boldsymbol{a} \times \boldsymbol{r}_0)\overline{Q} + Q\boldsymbol{b}\overline{Q}.$$

Noting that for any three quaternions A, B, and C, we have

$$(AB)\cdot C = \mathrm{Re}(C(\overline{AB}) = A\cdot(C\overline{B}) = B\cdot(\overline{A}C), \quad (141)$$

we obtain that

$$\mathcal{L}_\psi(\boldsymbol{g} \cdot \boldsymbol{h}) = \tilde{\boldsymbol{g}} \cdot (\boldsymbol{a} \times \boldsymbol{r}_0 + \boldsymbol{b})$$
$$= 2\psi \cdot (\boldsymbol{r}_0 \times \tilde{\boldsymbol{g}} + \epsilon\tilde{\boldsymbol{g}}). \qquad (142)$$

$\qquad\square$

Proof of Theorem 1. The potential energy part is covered by Lemma 2. For the parts coming from the kinetic energy, using linearity, it is sufficient to prove it for the additive parts of M. The part not involving m_0 is proved using Theorem 4, various vector identities, and remembering equation (28).

So we only need to prove the kinetic energy portion in the case $\mathsf{M} = \Lambda^T \mathsf{M}_0 \Lambda$. The easiest way to show this is to simply differentiate $\mathsf{M}_0\dot\ell = \mathsf{M}_0\Lambda\varphi$ with respect to time. To do it directly from the formulas is more complicated, as we now show. For any constant pure dual quaternion ψ, we have

$$\psi \cdot (\mathcal{L}_\varphi(\mathsf{M}\varphi) - \tfrac{1}{2}\mathcal{L}(\varphi \cdot \mathsf{M}\varphi))$$
$$= (\mathcal{L}_\varphi(\psi \cdot \mathsf{M}\varphi) - \tfrac{1}{2}\mathcal{L}(\varphi \cdot \mathsf{M}\varphi))$$
$$= \mathcal{L}_\varphi((\Lambda\psi) \cdot \mathsf{M}_0(\Lambda\varphi)) - \tfrac{1}{2}\mathcal{L}_\psi((\Lambda\varphi) \cdot \mathsf{M}_0(\Lambda\varphi))$$
$$= \mathcal{L}_\varphi\mathcal{L}_\varphi\mathsf{L} \cdot \mathsf{M}_0\mathcal{L}_\psi\mathsf{L} + \mathcal{L}_\varphi\mathcal{L}_\psi\mathsf{L} \cdot \mathsf{M}_0\mathcal{L}_\varphi\mathsf{L}$$
$$\quad - \mathcal{L}_\psi\mathcal{L}_\varphi\mathsf{L} \cdot \mathsf{M}_0\mathcal{L}_\varphi\mathsf{L}$$
$$= \mathcal{L}_\varphi(\Lambda\varphi) \cdot \mathsf{M}_0\Lambda\psi + \mathcal{L}_{(\varphi\psi-\psi\varphi)}\mathsf{L} \cdot \mathsf{M}_0\Lambda\varphi$$
$$= \psi \cdot \Lambda^T\mathsf{M}_0\mathcal{L}_\varphi(\Lambda\varphi) + \Lambda(\varphi\psi - \psi\varphi) \cdot \mathsf{M}_0\Lambda\varphi$$
$$= \psi \cdot \Lambda^T\mathsf{M}_0\mathcal{L}_\varphi(\Lambda\varphi) + (\varphi\psi - \psi\varphi) \cdot \mathsf{M}_0\varphi,$$
$$\qquad (143)$$

where we used equation (73). Then it is simply a matter of collecting terms. $\qquad\square$

17. Projection from matrices to orthogonal matrices

What would be a good projection of (3×3) matrices onto rotation matrices? One suggestion is $\mathsf{A} \mapsto \widehat{\mathsf{A}} = (\mathsf{A}\mathsf{A}^T)^{-1/2}\mathsf{A} = \mathsf{A}(\mathsf{A}^T\mathsf{A})^{-1/2}$, which works if we have $\det(\mathsf{A}) > 0$. One way to compute it is via the singular value decomposition $\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^T$, when $\widehat{\mathsf{A}} = \mathsf{U}\mathsf{V}^T$. However this is computationally expensive. See [14].

References

[1] Bruno Vilhena Adorno, Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra – Part I: Fundamentals, 2017, hal-01478225.

[2] S.L. Altmann, Hamilton, Rodrigues, and the Quaternion Scandal, Mathematics Magazine, Vol. 62, No. 5. (Dec., 1989), pp. 291-308.

[3] V.I. Arnold, Mathematical Methods of Classical Mechanics, 2nd Ed, Springer-Verlag, 1989.

[4] R.S. Ball, The theory of screws: A study in the dynamics of a rigid body. Hodges, Foster and Co., Dublin, 1876.

[5] O. Bottema, B. Roth, Theoretical Kinematics, North-Holland, Amsterdam, 1979.

[6] M.A. Clifford, Preliminary Sketch of Biquaternions, Proceedings of the London Mathematical Society, Volume s1-4, Issue 1, November 1871, Pages 381-395, `https://doi.org/10.1112/plms/s1-4.1.381`.

[7] Jean-Pierre Dedieu, Pierre Priouret, Gregorio Malajovich, Newton's Method on Riemannian Manifolds: Covariant Alpha-Theory, 395-419, IMA Journal of Numerical Analysis Volume: 23, Issue: 3, July 2003 DOI: 10.1093/imanum/23.3.395.

[8] Teles A. Fernandes, Orizon P. Ferreira and Jinyun Yuan, On the Superlinear Convergence of Newton's Method on Riemannian Manifolds, Journal of Optimization Theory and Applications, 173, 828-843 (2017).

[9] O.P. Ferreira and B.F. Svaiter, Kantorovich's Theorem on Newton's Method in Riemannian Manifolds, Journal of complexity 18, 304-329 (2002), doi:10.1006-jcom.2001.0582.

[10] Jaime Gallardo-Alvarado, Kinematic Analysis of Parallel Manipulators by Algebraic Screw Theory, Springer, Switzerland, 2016.

[11] Herbert Goldstein, C.P. Poole, J.L. Safko, (2001), Classical Mechanics (3rd ed.). Addison-Wesley.

[12] Da-Peng Han, Qing Wei, and Ze-Xiang Li, Kinematic Control of Free Rigid Bodies Using Dual Quaternions, International Journal of Automation and Computing 05(3), July 2008, 319-324, DOI: 10.1007/s11633-008-0319-1.

[13] Sigurdur Helgason, Differential Geometry, Lie Groups and Symmetric Spaces, 656 pp. Academic Press, New York, 1978.

[14] N.J. Higham, Matrix nearness problems and applications, in M.J.C. Gover and S. Barnett, editors, Applications of Matrix Theory, pages 1-27. Oxford University Press, 1989.

[15] K. Huper and J. Trumpf, Newton-like methods for numerical optimization on manifolds, Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004., Pacific Grove, CA, USA, 2004, 136-139 Vol 1, doi: 10.1109/ACSSC.2004.1399106.

[16] L. Kavan, S. Collins, J. Žára, C. O'Sullivan, Skinning with Dual Quaternions, `https://dl.acm.org/doi/pdf/10.1145/1230100.1230107`.

[17] L. Kavan, S. Collins, J. Žára, C. O'Sullivan, Geometric Skinning with Approximate Dual Quaternion Blending, ACM Transactions on Graphics, Vol. 27, No. 4, Article 105, Publication date: October 2008.

[18] Daniel T. Kawano, Alyssa Novelia and Oliver M. O'Reilly, A tumbling T-handle in space: the Dzhanibekov effect, `https://rotations.berkeley.edu/a-tumbling-t-handle-in-space`.

[19] Ben Kenwright, A Beginners Guide to Dual-Quaternions, What They Are, How They Work, and How to Use Them for 3D Character Hierarchies, `https://cs.gmu.edu/~jmlien/teaching/cs451/uploads/Main/dual-quaternion.pdf`.

[20] John Lee, Introduction to Smooth Manifolds. Graduate Texts in Mathematics. 218 (Second ed.). New York London: Springer-Verlag. ISBN 978-1-4419-9981-8. OCLC 808682771, 2012.

[21] Stephen Montgomery-Smith, Functional Calculus for dual quaternions, arXiv:2202.04681, `https://arxiv.org/abs/2202.04681`.

[22] Andreas Pott and Valentin Schmidt, On the Forward Kinematics of Cable-Driven Parallel Robots, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 2015.

[23] J. Pujol, Hamilton, Rodrigues, Gauss, Quaternions, and Rotations: a Historical Reassessment, Commun. Math. Anal., Volume 13, Number 2 (2012), 1-14.

[24] Ken Shoemake, Animating Rotation with Quaternion Curves, SIGGRAPH 1985, `https://www.cs.cmu.edu/~kiranb/animation/p245-shoemake.pdf`.

[25] Stanford Artificial Intelligence Laboratory et al., Robotic Operating System, `https://www.ros.org`, 2018.

[26] Xiangke Wang, Dapeng Han, Changbin Yu, and Zhiqiang Zheng, The geometric structure of unit dual quaternions with application in kinematic control, Journal of Mathematical Analysis and Applications 389(2), 2012, 1352-1364.

[27] Wikipedia, Hodge star operator, `https://en.wikipedia.org/wiki/Hodge_star_operator#Three_dimensions`.

[28] Wikipedia, Lie derivative, `https://en.wikipedia.org/wiki/Lie_derivative`.

[29] Wikipedia, Lie group-Lie algebra correspondence, `https://en.wikipedia.org/wiki/Lie_group%E2%80%93Lie_algebra_correspondence`.

[30] K. Yano, The Theory of Lie Derivatives and its Applications, North-Holland ISBN 978-0-7204-2104-0, 1957.

[31] XiaoLong Yang, HongTao Wu, Yao Li, Bai Chen, A dual quaternion solution to the forward kinematics of a class of six-DOF parallel robots with full or reductant actuation, Mechanism and Machine Theory 107 (2017) 27-36, `http://dx.doi.org/10.1016/j.mechmachtheory.2016.08.003`.

Department of Mathematics, University of Missouri, Columbia, MO 65211.

Email address: `stephen@missouri.edu`

URL: `https://stephenmontgomerysmith.github.io`

Johnson Space Center, 2101 E. NASA Pkwy, Houston, TX 77058.

Email address: `cecil.shy-1@nasa.gov`