USING LIE DERIVATIVES WITH DUAL QUATERNIONS FOR PARALLEL ROBOTS

STEPHEN MONTGOMERY-SMITH

Department of Mathematics, University of Missouri, Columbia, MO 65211.
stephen@missouri.edu
https://stephenmontgomerysmith.github.io

CECIL SHY

Johnson Space Center, 2101 E. NASA Parkway, Houston, TX 77058.
cecil.shy-1@nasa.gov

ABSTRACT. We introduce the notion of the Lie derivative in the context of dual quaternions that represent rigid motions and twists. First we define the wrench in terms of dual quaternions. Then we show how the Lie derivative helps understand how actuators affect an end effector in parallel robots, and make it explicit in the two cases case of Stewart Platforms, and cable-driven parallel robots. We also show how to use Lie derivatives with the Newton-Raphson Method to solve the forward kinematic problem for over constrained parallel actuators. Finally, we derive the equations of motion of the end effector in dual quaternion form, which include the effect of inertia from the actuators.

1. INTRODUCTION

This paper is broadly about *poses* and/or *rigid motions*, and how they can be represented by *unit dual quaternions*. A pose is a description of a frame of reference with respect to a fixed frame of reference, and consists of an *orientation*, and a *position*. A rigid motion consists of a *rotation* followed by a *translation*. From a mathematical point of

view, these can be considered to be the same thing, and so we use these terms interchangeably (but see [7] for a different point of view).

Along with poses and/or rigid motions, we have the notion of *screws*. First we have rates of change of poses/rigid bodies, which are angular and translational velocities, and are described by *twists*. Second we have descriptions of how to change the inertia of the rigid body, the *wrench*, that is, the torque and the force applied to the body. For more reading on rigid body kinematics and dynamics, including screw theory, we refer the reader to [5, 6, 13, 37].

The notion of the dual quaternion, and its use to represent poses and rigid motions, seems to go back to McAulay [25], inspired by the earlier work by Clifford [8]. The notion of using dual quaternions to represent twists may be found in [1, 32]. A basic introduction to dual quaternions may be found in [22, 29], the latter also covering twists. Many authors have used dual quaternions to represent hierarchies of poses, that is, chains of manipulators [22, 32, 35, 36, 39]. Papers on representing kinematics or dynamics via dual quaternions include [1, 2, 10, 16, 23, 40, 41]. Dual quaternions have also found great use in computer graphics [19, 20].

(The reader should be aware that [1, 16] have incorrect formulas for the logarithm and exponential of dual quaternions — the correct formulas may be found in [28], and [38] for the exponential.)

The purpose of this paper is to introduce the notion of using Lie derivatives for dual quaternions. We show that these can be used to essentially automate the creation of rather complex formulas, which are required for forward kinematics, and for dynamic equations of motion.

The authors have successfully used these formulas, combine with the algorithm described in [27], to create software for controlling a cable-driven parallel robot, which was built by the Dynamic Systems Test Branch of the Software, Robotics, and Simulation Division (ER5) at the NASA Johnson Space Center. Because of Export Administration Regulations we are unable to provide many more details.

The paper is quite heavy with mathematical formulas. For this reason, the proofs of most of the statements, and many of the comments of a mathematical nature, relegated to the Appendix, Section 12.

2. Notation

Rotations can be represented by unit quaternions [3, 34], 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 $A^* = w - x\boldsymbol{i} - y\boldsymbol{j} - y\boldsymbol{k}$, its *norm*

is $|A| = (w^2 + x^2 + y^2 + z^2)^{1/2} = \sqrt{AA^*} = \sqrt{A^*A}$, its *normalization* is $\widehat{A} = A/|A|$, its *real part* is $\mathrm{Re}(A) = w = \frac{1}{2}(A + A^*)$, and its *imaginary part* is $\mathrm{Im}(A) = \boldsymbol{i}x + \boldsymbol{j}y + \boldsymbol{k}z = \frac{1}{2}(A - A^*)$. It is called a *unit* quaternion if $|A| = 1$, a *real* quaternion if $\mathrm{Im}(A) = 0$, and a *vector* quaternion if $\mathrm{Re}(A) = 0$. If $A \neq 0$, the multiplicative inverse is given by $A^{-1} = A^*/|A|^2$. (Note that many sources use the word "pure" instead of "vector" in this context.)

We identify three dimensional vectors with vector quaternions, by identifying $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ with the three standard unit vectors. A unit quaternion Q represents the rotation which takes the direction $\boldsymbol{r}$ to $Q\boldsymbol{r}Q^*$. A rotation by angle a about an axis $\boldsymbol{s}$, where $|\boldsymbol{s}| = 1$, has two unit quaternion representations: $\pm(\cos(\frac{1}{2}a) + \boldsymbol{s}\sin(\frac{1}{2}a)) = \pm\exp(\frac{1}{2}a\boldsymbol{s})$. Composition of rotations corresponds to multiplication of unit quaternions.

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

$$A \cdot B = \mathrm{Re}(AB^*) = \mathrm{Re}(A^*B). \tag{1}$$

A *dual quaternion* is a pair of quaternions, written as $\eta = A + \epsilon B$, with the extra algebraic operation $\epsilon^2 = 0$. We call $A = \mathcal{P}(\eta)$ the *primary part* of η, and $B = \mathcal{D}(\eta)$ the *dual part* of η.

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

$$(\eta_1\eta_2)^* = \eta_2^*\eta_1^*. \tag{2}$$

There is another conjugation for dual quaternions: $\overline{A + \epsilon B} = A - \epsilon B$, but we have no cause to use it in this paper, except in equation (7) below.

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

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

$$\eta^{-1} = A^{-1} - \epsilon A^{-1}BA^{-1}. \tag{3}$$

If η is a unit dual quaternion, then there is a computationally much faster formula (see [1]):

$$\eta^{-1} = \eta^*. \tag{4}$$

We set $\mathbb{DH}$ for the set of invertible dual quaternions (that is, $A + \epsilon B$ where $A \neq 0$), $\mathfrak{dh}$ for the set of dual quaternions, $\mathbb{SDH}$ for the set of unit dual quaternions, and $\mathfrak{sdh}$ for the set of vector dual quaternions.

A rigid motion of the form

$$\boldsymbol{r} \mapsto Q\boldsymbol{r}Q^* + \boldsymbol{t}, \tag{5}$$

where here $\boldsymbol{t}$ is a translation, can be represented by the unit dual quaternion

$$\eta = Q + \tfrac{1}{2}\epsilon \boldsymbol{t}Q. \tag{6}$$

Composition of rigid motions corresponds to multiplication of unit dual quaternions, where the notation $\eta_1 \eta_2$ means to apply first the rigid motion represented by η_2, and then by η_1, that is, the dual quaternion acts by left multiplication. If $\boldsymbol{r}$ is a 3-vector, and $\boldsymbol{s}$ is the image of $\boldsymbol{r}$ under the action of the rigid motion $\eta = Q + \epsilon B$, then

$$1 + \epsilon \boldsymbol{s} = \eta(1 + \epsilon \boldsymbol{r})\overline{\eta}^*, \tag{7}$$

but generally it is easier to use the formula

$$\boldsymbol{s} = Q\boldsymbol{r}Q^* + 2BQ^* = (Q\boldsymbol{r} + 2B)Q^*. \tag{8}$$

For a dual quaternion, it is not really possible to mix its primary and dual parts additively. For example, for a unit dual quaternion that represents a rigid motion, the primary part is unitless, whereas the dual has units of length. For this reason, when measuring how large such a dual quaternion is, everything must be with respect to a characteristic length scale l. (For example, for a parallel robot, the characteristic length might be the width of the workspace of the end effector.) The *size* of a dual quaternion is defined to be

$$\mathrm{size}_l(\eta) = \left(|\mathcal{P}(\eta)|^2 + l^{-2}|\mathcal{D}(B)|^2 \right)^{1/2}. \tag{9}$$

A *twist* is the pair of vectors $(\boldsymbol{w}, \boldsymbol{v})$ that describes the rate of change of pose or rigid motion, where $\boldsymbol{w}$ is the angular velocity, and $\boldsymbol{v}$ is the translational velocity. One can think of the twist $(\boldsymbol{w}, \boldsymbol{v})$ as a rigid motion function of time t:

$$\boldsymbol{r} \mapsto \boldsymbol{r} + t(\boldsymbol{w} \times \boldsymbol{r} + \boldsymbol{v}) + O(t^2) \quad \text{as } t \to 0. \tag{10}$$

It has two possible meanings, depending upon whether the twist is understood to be with respect to the fixed frame, or with respect to the moving frame. If it is understood to be with respect to the moving frame, we have the formula

$$\frac{d}{dt}(Q\boldsymbol{r}Q^* + \boldsymbol{t}) = Q(\boldsymbol{w} \times \boldsymbol{r} + \boldsymbol{v})Q^* + \boldsymbol{t}, \tag{11}$$

and if it is understood to be with respect to the fixed frame

$$\frac{d}{dt}(Q\boldsymbol{r}Q^* + \boldsymbol{t}) = \boldsymbol{w} \times (Q\boldsymbol{r}Q^* + \boldsymbol{t}) + \boldsymbol{v}. \tag{12}$$

Then this twist can be represented by a vector dual quaternion [1, 2]

$$\varphi = \tfrac{1}{2}\boldsymbol{w} + \tfrac{1}{2}\epsilon\boldsymbol{v}, \tag{13}$$

where if we understand it to be with respect to the moving frame, we have the formula

$$\varphi = \eta^{-1}\dot{\eta}, \quad \text{or} \quad \dot{\eta} = \eta\varphi, \tag{14}$$

and if we understand it to be with respect to the fixed frame

$$\varphi = \dot{\eta}\eta^{-1}, \quad \text{or} \quad \dot{\eta} = \varphi\eta. \tag{15}$$

In this paper, unless otherwise stated, we always understand the twist to be with respect to the moving frame.

We make the identification

$$\mathfrak{dh} \cong \mathbb{R}^8, \tag{16}$$

using the basis

$$(\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8) = (\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}, \epsilon\boldsymbol{i}, \epsilon\boldsymbol{j}, \epsilon\boldsymbol{k}, 1, \epsilon), \tag{17}$$

and similarly, we make the identification

$$\mathfrak{sdh} \cong \mathbb{R}^6, \tag{18}$$

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

$$(A + \epsilon B) \cdot (C + \epsilon D) = A \cdot C + B \cdot D. \tag{19}$$

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

$$\eta = \sum_{i=1}^{8} \eta_i \beta_i, \tag{20}$$

and every vector dual quaternion θ as

$$\theta = \sum_{i=1}^{6} \theta_i \beta_i. \tag{21}$$

Finally, we give a few extra formulas. Let φ_m denote the twist with respect to the moving frame, and φ_f denote the twist with respect to the fixed frame. From equations (14) and (15) we obtain

$$\varphi_f = \eta\varphi_m\eta^{-1}. \tag{22}$$

Since in any algebra we have

$$\frac{d}{dt}\eta^{-1} = -\eta^{-1}\dot{\eta}\eta^{-1}, \tag{23}$$

we obtain the surprisingly simple formula for the change of frame for acceleration:

$$\dot{\varphi}_f = \eta\dot{\varphi}_m\eta^{-1}. \tag{24}$$

Note that this does not generalize to higher derivatives, for example, the formula for change of frame for jerk is

$$\ddot{\varphi}_f = \eta\ddot{\varphi}_m\eta^{-1} + \eta(\varphi\dot{\varphi} - \dot{\varphi}\varphi)\eta^{-1}. \tag{25}$$

3. 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

$$\tau = 2\boldsymbol{q} + 2\epsilon\boldsymbol{p}, \tag{26}$$

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

$$\varphi_0 = \varphi + \tfrac{1}{2}\epsilon\boldsymbol{w} \times \boldsymbol{r}_0, \tag{27}$$

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

$$\tau_0 = \tau + 2\boldsymbol{p} \times \boldsymbol{r}_0. \tag{28}$$

The reason for introducing the factor 2 in definition (26) is so that the rate of change of work done to the end effector is given by

$$\frac{d}{dt}(\text{work done}) = \tau \cdot \varphi = \tau_0 \cdot \varphi_0. \tag{29}$$

(The second equality follows from vector identities.)

See [5] for the origins of the term twist and wrench as pairs of 3-vectors, which are examples of *screws*. The 'work done' formulas are also known as *reciprocal screw relationships*.

4. The normalization of a dual quaternion

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

$$|\eta|^2 = \eta^*\eta = \eta\eta^* = |A|^2 + 2\epsilon(A \cdot B), \tag{30}$$

$$|\eta| = \sqrt{|\eta|^2} = |A| + \epsilon(A \cdot B)/|A|. \tag{31}$$

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

$$|\eta_1 \eta_2| = |\eta_1||\eta_2|. \tag{32}$$

If $\eta = Q + \epsilon B$ is an invertible dual quaternion, then we define its *normalization* to be the unit dual quaternion

$$\widehat{\eta} = |\eta|^{-1}\eta = \eta|\eta|^{-1} = Q/|Q| + \epsilon(B/|Q| - (B \cdot Q)Q/|Q|^3).^{\dagger} \tag{33}$$

(We remark that the normalization of an invertible dual quaternion is used in the computer graphics industry [19, 20].) 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

$$\widehat{\eta_1 \eta_2} = \widehat{\eta}_1 \widehat{\eta}_2. \tag{34}$$

5. Notation for three by three matrices

Let I denote the (3×3) identity matrix, and $\mathsf{0}$ denote the (3×3) zero matrix. If $\boldsymbol{r}$ is a 3-vector, then the *Hodge star operator* of $\boldsymbol{r}$ is

$$\mathsf{S}_{\boldsymbol{r}} = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}. \tag{35}$$

Note that

$$\mathsf{S}_{\boldsymbol{r}}\boldsymbol{s} = \boldsymbol{r} \times \boldsymbol{s}. \tag{36}$$

If $\boldsymbol{u}$ is a unit vector, then the *projection onto the complement* of the unit vector $\boldsymbol{u}$ is defined by

$$\mathsf{P}_{\boldsymbol{u}}\boldsymbol{x} = \boldsymbol{x} - (\boldsymbol{u} \cdot \boldsymbol{x})\boldsymbol{u}. \tag{37}$$

Consistent with the identification (18), if A, B, C, and D are (3×3) matrices, and $\theta = \frac{1}{2}\boldsymbol{a} + \frac{1}{2}\epsilon\boldsymbol{b}$, $\psi = \frac{1}{2}\boldsymbol{c} + \frac{1}{2}\epsilon\boldsymbol{d}$ are vector dual quaternions,

---

†Note this formula is incorrect in the published version [30], and corrected in this paper [31]. The authors are grateful to Brent Koogler for bringing this to our attention. This correction makes only a small difference to the results reported in Subsection 10.3. The corrected results may be found in [31].

we have

$$\begin{bmatrix} \mathsf{A} & \mathsf{B} \end{bmatrix} \theta = \tfrac{1}{2}\mathsf{A}\boldsymbol{a} + \tfrac{1}{2}\mathsf{B}\boldsymbol{b}, \tag{38}$$

$$\begin{bmatrix} \mathsf{A} & \mathsf{B} \\ \mathsf{C} & \mathsf{D} \end{bmatrix} \theta = \tfrac{1}{2}\mathsf{A}\boldsymbol{a} + \tfrac{1}{2}\mathsf{B}\boldsymbol{b} + \tfrac{1}{2}\epsilon\mathsf{C}\boldsymbol{a} + \tfrac{1}{2}\epsilon\mathsf{D}\boldsymbol{b}, \tag{39}$$

$$\theta \cdot \begin{bmatrix} \mathsf{A} & \mathsf{B} \\ \mathsf{C} & \mathsf{D} \end{bmatrix} \psi = \tfrac{1}{4}\boldsymbol{a} \cdot \mathsf{A}\boldsymbol{c} + \tfrac{1}{4}\boldsymbol{a} \cdot \mathsf{B}\boldsymbol{d} + \tfrac{1}{4}\boldsymbol{b} \cdot \mathsf{C}\boldsymbol{c} + \tfrac{1}{4}\boldsymbol{b} \cdot \mathsf{D}\boldsymbol{d}. \tag{40}$$

6. 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 [42]. 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 [24]. 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 [17, 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 vector dual quaternions.

Suppose one has a quantity that is a function of pose $g(\eta)$, and whose range is any vector space. (But for intuition, consider the case when the range is the real numbers, and think of the vector valued case as the formulas below simply being applied component wise.)

Then we usually think of the derivative of $g(\eta)$ 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, $\mathbb{SDH}$, and whose codomain is any vector space over the real numbers, we can extend it arbitrarily to a differentiable function whose domain is an open neighborhood of $\mathbb{SDH}$ in $\mathfrak{dh}$. Given a unit dual quaternion η and a vector dual quaternion θ, we define the *Lie derivative* of $g(\eta)$ in the direction of θ to be

$$\mathcal{L}_\theta g = \lim_{r \to 0} \frac{g(\eta(1 + r\theta)) - g(\eta)}{r} = \frac{d}{dr}g(\eta(1 + r\theta))\Big|_{r=0}. \tag{41}$$

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 $\mathbb{SDH}$, but it is, as is shown in Lemma 2 below.

Given a generic function g whose domain is the invertible dual quaternions, $\mathbb{DH}$, and whose codomain is any vector space over the real numbers, we define its Jacobian to be the dual quaternion

$$\frac{\partial g}{\partial \eta} = \sum_{i=1}^{8} \left[\frac{\partial}{\partial \eta_i} g(\eta) \right] \beta_i, \tag{42}$$

where the partial derivative $\dfrac{\partial}{\partial \eta_i}$ is interpreted using the identification of η_i with the β_i components of η as described in (17).

If the domain of g is the vector dual quaternions, $\mathfrak{sdh}$, we have equation (42) except with 8 replaced by 6.

We have the following formula, which is useful for explicitly calculating the Lie derivative if an extension of g to an open neighborhood of $\mathbb{SDH}$ in $\mathfrak{dh}$ is known:

$$\mathcal{L}_\theta g = \frac{\partial g}{\partial \eta} \cdot (\eta\theta) := \sum_{i=1}^{8} \left(\frac{\partial g}{\partial \eta_i} \right) (\eta\theta)_i, \tag{43}$$

where the notation $(\eta\theta)_i$ means the β_i component of $\eta\theta$, as described in (17).

We define the *partial Lie derivatives* to be

$$\mathcal{L}_i g = \mathcal{L}_{\beta_i} g, \quad (1 \le i \le 6), \tag{44}$$

and its *full Lie derivative* to be the vector dual quaternion (or if the range of g is a vector space, the tensor product of a vector dual quaternion with a vector)

$$\mathcal{L}g = \sum_{i=1}^{6} \mathcal{L}_i g \beta_i, \tag{45}$$

so that for all vector dual quaternions θ it satisfies:

$$\theta \cdot \mathcal{L}g = \mathcal{L}_\theta g. \tag{46}$$

To gain some intuition, write

$$\theta = \tfrac{1}{2}\boldsymbol{a} + \tfrac{1}{2}\epsilon\boldsymbol{b}. \tag{47}$$

Since we have that

$$\frac{\partial}{\partial \theta} = 2\frac{\partial}{\partial \boldsymbol{a}} + 2\epsilon\frac{\partial}{\partial \boldsymbol{b}}, \tag{48}$$

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 $\mathcal{L}g$ is a vector 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 by equation (41), with r replaced by t, we see that

$$\frac{d}{dt}[g(\eta)] = \mathcal{L}_\varphi g. \tag{49}$$

The Lie derivative satisfies various rules, which easily follow from either equations (41) or (43), which are also useful for explicitly calculating the Lie derivative when g is known.

- If $g(\eta)$ is linear in η, then

$$\mathcal{L}_\theta g(\eta) = g(\eta\theta). \tag{50}$$

- The product rule: if $*$ is any binary operator which is bilinear over the real numbers, such as the product of real numbers, the inner product, the cross product, or the dual quaternion product, then

$$\mathcal{L}_\theta(g_1 * g_2) = g_1 * (\mathcal{L}_\theta g_2) + (\mathcal{L}_\theta g_1) * g_2. \tag{51}$$

- The chain rule:

$$\mathcal{L}_\theta(h(g_1, g_2, \ldots, g_m)) = \sum_{i=1}^{m} \frac{\partial}{\partial g_i} h(g_1, g_2, \ldots, g_m) \mathcal{L}_\theta g_i. \tag{52}$$

- Let $\tilde{s}$ be a constant position vector, and $\tilde{n}$ be a constant direction. Let s and n be their corresponding values with respect to the moving frame. Then

$$\mathcal{L}_\theta s = 2 \begin{bmatrix} \mathsf{S}_s & -\mathsf{I} \end{bmatrix} \theta, \tag{53}$$

$$\mathcal{L}_\theta n = 2 \begin{bmatrix} \mathsf{S}_n & 0 \end{bmatrix} \theta. \tag{54}$$

To simplify the writing of application software, Using these rules, one can create a software library in C++ that performs automatic Lie differentiation. Since the domain, and hence range, of the Lie derivative can be any vector space, the sensible way to do this is using templates to allow for a variety of different data types. Note also that the product rule (51) has to be implemented for every product that is used, and similarly the chain rule (52) for every function h that is used.

7. Applications to parallel robots

To aid our description, we show the example of a cable-driven parallel robot in Figures 1 and 2. In Figure 1, we show the end effector in green, controlled by eight cables shown in red, which go around pulleys shown
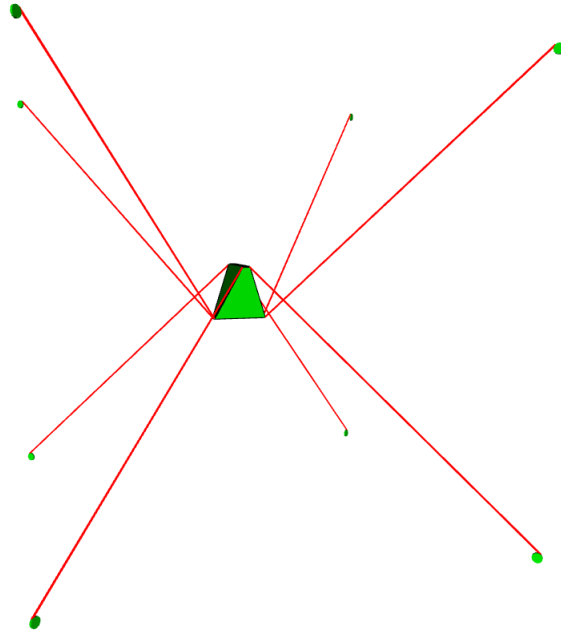
FIGURE 1. Schematic of a cable-driven parallel robot.
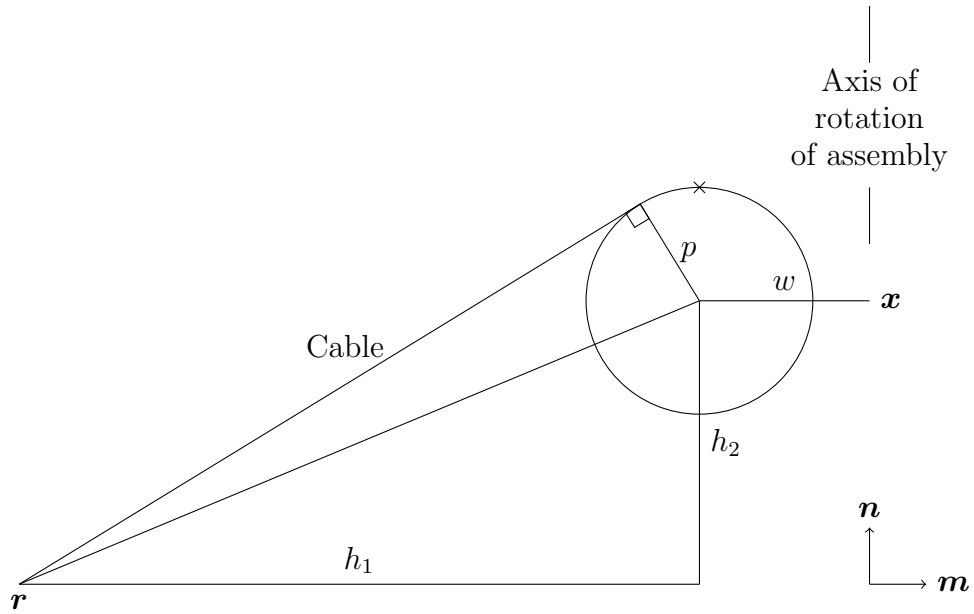

FIGURE 2. The pulley and attached cable

in green. Each cable, and how it is attached to the pulley, is shown in Figure 2.

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

$$\boldsymbol{\ell} = (\ell_j)_{1 \leq j \leq n}. \tag{55}$$

For example, for a cable-driven parallel robot, these represent the lengths of the cables, and typically $n = 8$. The numbers ℓ_j only need to be determined up to a fixed number. Thus, for example, in Figure 2, the number ℓ_j could represent the length of cable from the end effector attachment point, to the small cross marked at the top of the pulley.

For the Gough or Stewart Platforms [13], we have $n = 6$, and ℓ_j is simply the distance between the end effector attachment point and the ground attachment point.

Let us also denote the force exerted by the actuators by

$$\boldsymbol{f} = (f_j)_{1 \leq j \leq n}, \tag{56}$$

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

$$\frac{d}{dt}(\text{work done}) = \boldsymbol{f} \cdot \dot{\boldsymbol{\ell}}. \tag{57}$$

Suppose we have a function $\mathsf{L} : \mathbb{SDH} \to \mathbb{R}^n$, which calculates the required actuator values, $\boldsymbol{\ell}$, from the pose η of the end effector frame. This is the *inverse kinematics* function.

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

$$\Lambda\theta = \mathcal{L}_\theta \mathsf{L} = \mathcal{L}_\theta \boldsymbol{\ell}, \tag{58}$$

or more explicitly, by

$$\Lambda_{i,j} = \mathcal{L}_j \ell_i. \tag{59}$$

From equation (49) we obtain

$$\dot{\boldsymbol{\ell}} = \Lambda\varphi. \tag{60}$$

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

$$\tau = \mathsf{T}\boldsymbol{f}. \tag{61}$$

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:

$$\mathsf{T} = \Lambda^T. \tag{62}$$

8. Second lie derivatives

If g is a function of dual quaternions with codomain any vector space over the real numbers, we define its Hessian to be the (8×8) matrix

$$\left[\frac{\partial^2 g}{\partial \eta^2}\right] = \left[\frac{\partial^2 g}{\partial \eta_i \partial \eta_j}\right]_{1 \leq i,j \leq 8}. \tag{63}$$

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

Second Lie derivatives will be used in Newton's Method, as well as in our statements of the equations of motion, both described below.

We have that

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

$$:= \sum_{i=1}^8 \sum_{j=1}^8 (\eta \psi)_i \left(\frac{\partial^2 g}{\partial \eta_i \partial \eta_j}\right) (\eta \theta)_j + \sum_{i=1}^8 \left(\frac{\partial g}{\partial \eta_i}\right) (\eta \theta \psi)_i. \tag{64}$$

Another way one might try to define the second derivative is to use the formula $\frac{\partial^2}{\partial \theta^2} g(\eta(1+\theta))\big|_{\theta=0}$. 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 $\tilde{g}(\eta) = g(\hat{\eta})$. We have the following formula for the Hessian of $\tilde{g}$:

$$\frac{\partial^2}{\partial \theta^2} \tilde{g}(\eta(1+\theta))\bigg|_{\theta=0} = \left[\tfrac{1}{2}(\mathcal{L}_i \mathcal{L}_j g(\eta) + \mathcal{L}_j \mathcal{L}_i g(\eta))\right]_{1 \leq i,j \leq 6}. \tag{65}$$

9. The examples of a Stewart Platform, and a cable-driven parallel robot

As an example, let us consider parallel robots such as cable-driven robots, or Stewart Platforms. In this case, the end effector has certain 'attachment points' on it, $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_n$ where the cables or legs attach, the cables or legs are attached at the other end to $\boldsymbol{s}_k$, and unit vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$ which are directions the cables or legs come into the end effector, all of these being measured in the end effector's frame of reference. Note that for $1 \leq k \leq n$

$$\boldsymbol{u}_k = \frac{\boldsymbol{r}_k - \boldsymbol{s}_k}{|\boldsymbol{r}_k - \boldsymbol{s}_k|}. \tag{66}$$

Then

$$\mathcal{L}\ell_k = 2\boldsymbol{r}_k \times \boldsymbol{u}_k + 2\epsilon \boldsymbol{u}_k = 2 \begin{bmatrix} \mathsf{S}_{\boldsymbol{r}_k} \\ \mathsf{I} \end{bmatrix} \boldsymbol{u}_k, \tag{67}$$

or representing vectors as columns, we have

$$\Lambda = 2 \begin{bmatrix} (\boldsymbol{r}_1 \times \boldsymbol{u}_1)^T & \boldsymbol{u}_1^T \\ (\boldsymbol{r}_2 \times \boldsymbol{u}_2)^T & \boldsymbol{u}_2^T \\ \vdots & \vdots \\ (\boldsymbol{r}_n \times \boldsymbol{u}_n)^T & \boldsymbol{u}_n^T \end{bmatrix}. \tag{68}$$

For calculating the second Lie derivative of ℓ_k, we need only know the first Lie derivative of $\boldsymbol{u}_k$. Note that in the simple case that the attachment point of the cable to the frame is fixed in the fixed frame (for example, as in a Stewart Platform), we have

$$\mathcal{L}\boldsymbol{u}_k = \frac{1}{|\boldsymbol{r}_k - \boldsymbol{s}_k|} \, \mathsf{P}_{\boldsymbol{u}_k} \begin{bmatrix} -\mathsf{S}_{\boldsymbol{s}_k} & \mathsf{I} \end{bmatrix}, \tag{69}$$

and therefore

$$\mathcal{L}^2 \ell_k = 2 \begin{bmatrix} \mathsf{S}_{\boldsymbol{r}_k} \\ \mathsf{I} \end{bmatrix} \frac{1}{|\boldsymbol{r}_k - \boldsymbol{s}_k|} \, \mathsf{P}_{\boldsymbol{u}_k} \begin{bmatrix} -\mathsf{S}_{\boldsymbol{s}_k} & \mathsf{I} \end{bmatrix}. \tag{70}$$

Let us also describe a more complicated situation, which matches the cable-driven parallel robot the NASA Johnson Space Center described in the introduction. For simplicity of notation, we drop the subscripts k. See Figure 2 for reference. Let us suppose that the cable attaches via an assembly, which is free to rotate about an axis parallel to the unit vector $\boldsymbol{n}$, and passing through the point $\boldsymbol{x}$. Attached to the assembly, at a fixed distance w from the fixed point $\boldsymbol{x}$, by a rod perpendicular to $\boldsymbol{n}$, is the center of a pulley of radius p, which rotates in the plane containing the axis of rotation and the point on the end effector $\boldsymbol{r}$. A cable passes over the pulley in the $\boldsymbol{n}$ direction from the center of the pulley. All of the vectors are expressed in the moving frame of reference.

We work in the $(\boldsymbol{m}, \boldsymbol{n})$ coordinate system in which the origin is $\boldsymbol{r}$. Let (h_1, h_2) be the coordinates of the center of the pulley. Then

$$\boldsymbol{u} = u_1 \boldsymbol{m} + u_2 \boldsymbol{n}, \tag{71}$$

where

$$(u_1, u_2) = -\frac{1}{(h_1^2 + h_2^2)} \begin{bmatrix} h_1 & -h_2 \\ h_2 & h_1 \end{bmatrix} \begin{bmatrix} \sqrt{h_1^2 + h_2^2 - p^2} \\ p \end{bmatrix}. \tag{72}$$

The cable length, up to an additive constant, is

$$\ell = \sqrt{h_1^2 + h_2^2 - p^2} + p \tan^{-1}\left(\frac{u_2}{u_1}\right). \tag{73}$$

To avoid singularities, the optimal way to compute the inverse tangent, at least with the configuration shown in Figure 2, is to use

atan2$(-u_2, -u_1)$, where the commonly available atan2(y, x) function solves for θ where $x = r \cos \theta$, $y = r \sin \theta$, $r > 0$, $-\pi < \theta \leq \pi$.

The various required quantities are

$$\boldsymbol{d} = \mathsf{P}_{\boldsymbol{n}}(\boldsymbol{x} - \boldsymbol{r}), \tag{74}$$

$$\boldsymbol{m} = \widehat{\boldsymbol{d}} = \frac{\boldsymbol{d}}{|\boldsymbol{d}|}, \tag{75}$$

$$h_2 = \boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{r}), \tag{76}$$

$$h_1 = \boldsymbol{m} \cdot (\boldsymbol{x} - \boldsymbol{r}) - w = \sqrt{|\boldsymbol{x} - \boldsymbol{r}|^2 - h_2^2} - w. \tag{77}$$

The Lie derivatives of these quantities can be calculated using the automatic Lie differentiation described at the end of Section 6. The only second Lie derivative required is that of ℓ, and since we have equation (67), no automatic second Lie differentiation is required.

9.1. **Singularity analysis for Stewart Platforms.** When operating a Stewart Platform, a singularity occurs when there are no viable cable forces that can create an arbitrary wrench, or equivalently, when there exists infinitesimal perturbations of the end effector pose that don't require a leg length change. These singularities are often called bifurcations, because after a Stewart Platform encounters a singularity, the end effector is free to move in more than one direction. Encountering a singularity can cause great damage to the Stewart Platform.

From these considerations, it becomes clear that a singularity happens for a Stewart Platform if and only if $\det(\Lambda) = \det(\mathsf{T}) = 0$. This is in agreement with the results obtained by Gosselin and Angeles [15].

Note that when considering more complex parallel robots, that a singularities can happen in other situations as well (see, for example, [26]).

10. Forward kinematics

In robotics, there are several ways to find the pose of the end effector. One method is to use an optical system, but this is not very accurate. Another is to use a proprioceptive sensor, where the pose is found by integrating the acceleration and angular velocity of the end effector, but this is subject to drift. It would be extremely helpful if the end effector could be computed from the numbers ℓ_k. These numbers lengths can be found easily and with high sampling frequency using, for example, encoders.

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

$$\mathsf{Y} : \mathbb{L} \to \mathbb{SDH}, \tag{78}$$

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 $\mathbb{L}$.

The methods described here are essentially the Newton-Raphson Method, and are all iterative methods. Given a guess η_k, we create a new guess η_{k+1}:

$$\eta_{k+1} = \eta_k(\widehat{1 + \theta_{k+1}}), \tag{79}$$

and iterate until some criterion is met. We measure $\text{size}_l(\eta_{k+1} - \eta_k)$, and see when it is smaller than some preset value, like 10^{-16}.

Here we merely describe the method. In Section 12, we explain why they work.

We would also like to mention a different approach to using dual quaternions to solve forward kinematics problems in [43], although we don't think it will cover the more complicated situation described in Section 9.

10.1. **Forward kinematics for Stewart Platforms.** First we describe how to solve the exact-constrained problem, that is, when $n = 6$. This would be the case for a Stewart Platfom. Typically this is solved by writing the pose using Euler angles, which provides a way to represent the pose using a 6-vector. However, in the opinion of the authors, this becomes a rather complicated set of equations, resulting in quite lengthy code.

Our algorithm is

$$\theta_{k+1} = -\Lambda(\eta_k)^{-1}(\mathsf{L}(\eta_k) - \boldsymbol{\ell}). \tag{80}$$

This is easy to code, certainly simpler than methods which use Euler angles.

10.2. **Forward kinematics for over-constrained parallel robots.** Next we focus on the over-constrained problem, that is, when the number of actuators n is greater than 6.

This problem has been solved by many others, for example, [33, 43]. But we feel that this is more easily solved using dual quaternions. For example, using the programming language C++, one can quickly build classes representing dual quaternions, and then these formulas can be applied without any real thought.

We compute

$$\delta_k = \left(\sum_{m=1}^{n} \Lambda_{m,i}(\mathsf{L}_m(\eta_k) - \ell_m) \right)_{1 \le i \le 6}, \tag{81}$$

and

$$\mathsf{H}_k = \left[\sum_{m=1}^{n} \Lambda_{m,i}\Lambda_{m,j} + \tfrac{1}{2}\mathcal{L}_i\Lambda_{m,j}(\mathsf{L}_m(\eta_k)) + \tfrac{1}{2}\mathcal{L}_j\Lambda_{m,i}(\mathsf{L}_m(\eta_k)) \right]_{1 \le i,j \le 6}. \tag{82}$$

The algorithm is

$$\theta_{k+1} = \mathsf{H}_k^{-1}\delta_k. \tag{83}$$

10.3. **Results of simulations for forward kinematics.** The software used to check these algorithms described here is currently proprietary, and so we cannot give too many details. We hope to get permission to release the software at a later date, and make it available to everyone.

To test the simple Stewart Platform algorithm, we created 10,000 random poses. The orientation of each pose had an angle no greater than 30° from the identity pose. For each pose, the cable lengths were calculated. The Newton-Raphson Method was then applied to the cable lengths, with a random initial guess pose.

All runs were successful. The average number of required iterations was about 4.8. The run time for each forwards kinematics calculation was a little under 100 microseconds, using a fairly modern but low-end laptop. Increasing the allowed angle to 45° gave a failure rate of 2 in 10,000.

The Newton-Raphson Method for the over-constrained robot given by the more complicated situation described in Section 9 was more delicate. In particular, since it is minimizing a loss function rather than directly solving the problem, it is possible that it might find local minima of the loss function which didn't correspond to the solution.

For the first test, we created 1,000 random poses, and computed their cable lengths. The Newton-Raphson Method was applied, with a random initial guess pose, and was allowed up to 50 iterations. If the loss function of the final answer was greater than 10^{-16}, the Newton-Raphson Method was applied again.

All poses were found, but the average number of times the Newton-Raphson had to be applied was about 80. The average time to find a pose was about 5 milliseconds.

For the second test, we again created 1,000 random poses and computed their cable lengths. Then the Newton-Raphson was applied, with

an initial guess that was 1% different from the original pose. Again success was measured by computing the loss function. But there were no second chances.

All poses were found, the average number of iterations required was 4.2, and the average time taken was a little under 100 microseconds.

Increasing the allowable difference between the initial guess and the original pose to 5% resulted in only about 88% of the runs being successful.

Note that in all these trials, since the poses were randomly created, it is quite likely that some of the poses were non-feasible for the parallel robots under consideration. (For example, it might be impossible to maintain that pose while keeping the cables under tension, or moving to that pose might require crossing cables.)

When using this algorithm, to find the initial pose, we suggest to use the first method of trying 1,000 different initial guesses, and choosing the pose with the lowest loss function. But thereafter, sample the cable lengths often, and use the previously measured pose as the initial guess. In our numerical simulations, the pose changes by about 0.3% per time step (about 1 millisecond), and the algorithm has never failed to solve the forward kinematics problem.

11. Dynamics of the end effector

The dynamics equations of motion of rigid bodies is well known. But in this section, we also consider the additional effect of overcoming inertia from the actuators. Furthermore, we feel that it is nice to see this stated and derived in the context of dual quaternions.

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*.

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

$$\boldsymbol{f}_0 = \mathsf{M}_0\ddot{\boldsymbol{\ell}}, \tag{85}$$

where M_0 is a positive definite $(n \times n)$ matrix denoting what we shall call the *effective no-load mass of the actuators*. This might be caused by, for example, the reflected moment of inertia of the motor that drives each actuator, in which case M_0 is simply a constant multiple of the identity. Since $\tfrac{1}{2}\dot{\boldsymbol{\ell}} \cdot \mathsf{M}_0\dot{\boldsymbol{\ell}} = \tfrac{1}{2}\varphi \cdot \Lambda^T\mathsf{M}_0\Lambda\varphi$ is part of the kinetic energy, it follows that $\mathsf{M} - \Lambda^T\mathsf{M}_0\Lambda$ is a positive semi-definite matrix.

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

$$e = \tfrac{1}{2}m_e|\boldsymbol{v} + \boldsymbol{w} \times \boldsymbol{r}_0|^2 + \tfrac{1}{2}\boldsymbol{w} \cdot \mathsf{M}_e\boldsymbol{w} + \tfrac{1}{2}\dot{\boldsymbol{\ell}} \cdot \mathsf{M}_0\dot{\boldsymbol{\ell}}, \qquad (86)$$

that is

$$\mathsf{M} = 4\begin{bmatrix} \mathsf{M}_e - m_e\mathsf{S}_{\boldsymbol{r}_0}^2 & m_e\mathsf{S}_{\boldsymbol{r}_0} \\ -m_e\mathsf{S}_{\boldsymbol{r}_0} & m_e\mathsf{I} \end{bmatrix} + \Lambda^T\mathsf{M}_0\Lambda. \qquad (87)$$

Theorem 1. *If the kinetic energy satisfies equation* (84) *with equation* (87) *holding, and the potential energy v is calculated in the usual manner from the mass of the end effector in a constant gravitational field whose value is $\boldsymbol{g}$ measured with respect to the moving frame, then the equation of motion is*

$$\tau = \mu + \mathsf{M}\alpha, \qquad (88)$$

where

$$\begin{aligned} \mu = {}&2\boldsymbol{w} \times (\mathsf{M}_e\boldsymbol{w}) + 2\epsilon m_e\boldsymbol{w} \times \boldsymbol{v} \\ &+ 2m_e((\boldsymbol{w} \cdot \boldsymbol{r}_0)(\boldsymbol{w} \times \boldsymbol{r}_0) + \boldsymbol{r}_0 \times (\boldsymbol{w} \times \boldsymbol{v}) + \epsilon\boldsymbol{w} \times (\boldsymbol{w} \times \boldsymbol{r}_0)) \\ &+ \Lambda^T\mathsf{M}_0(\mathcal{L}_\varphi\Lambda)\varphi - 2m_e(\boldsymbol{r}_0 \times \boldsymbol{g} + \epsilon\boldsymbol{g}), \quad (89) \end{aligned}$$

and

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

The various terms in equation (89) 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, [21]).
-

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

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

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

- $m_e\boldsymbol{g}$ is the force due to gravity.

- 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 (27) and (28).

11.1. **Numerical verification of the dynamics equations.** The way we numerically verified these equations was by running a simulated motion, and then calculating the work done in three ways. The first method was to integrate the inner product of the twist and the wrench on the end effector. The second method was to integrate the sum of the actuator force times the rate of change of the length of the actuator. The third method was to calculate the total kinetic energy using equation (86), and the potential energy using the standard mass times gravity time height formula. The numerical simulations gave the same results for all three methods up to machine precision.

12. Appendix: Proofs

Lemma 1. *If θ is a vector dual quaternion, then*

$$\widehat{(1+\theta)} = 1 + \theta + \tfrac{1}{2}\theta^2 + O(\theta^3). \tag{93}$$

We remark that since the exponential of a unit dual quaternion θ [41] satisfies

$$\exp(\theta) = \sum_{k=0}^{\infty} \frac{\theta^k}{k!}, \tag{94}$$

then by equation (93) we have

$$\exp(\theta) = \widehat{(1+\theta)} + O(\theta^3). \tag{95}$$

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

$$\eta(\theta) = O(\gamma(\theta)) \tag{96}$$

if there exists a constant $c > 0$ such that if $\text{size}_l(\theta)$ is sufficiently small, then

$$\text{size}_l(\eta(\theta)) \leq c\,\text{size}_l(\gamma(\theta)). \tag{97}$$

Let us show that this definition does not depend upon the characteristic length l. In [28], we show how to define the 'functional calculus' of dual quaternions. That is, given a continuously differentiable function $f : \mathbb{C} \to \mathbb{C}$, satisfying $f(\bar{z}) = \overline{f(z)}$, we can make sense of $f(\eta)$ for any dual quaternion $\eta = A + \epsilon B$. First we can define f on quaternions by realizing that any unit vector $\boldsymbol{n}$ satisfies $\boldsymbol{n}^2 = -1$, and hence one merely treats $\boldsymbol{n}$ as an imaginary unit. Next, if B is decomposed as

$B_1 + \boldsymbol{b}_2$, where B_1 commutes with A, and $\boldsymbol{b}_2$ is a vector quaternion that anti-commutes with $\mathrm{Im}(A)$, and setting

$$f_x(z) = \lim_{h \to 0, h \in \mathbb{R}} \frac{f(z+h)}{h}, \tag{98}$$

$$f_{iy}(z) = \lim_{h \to 0, h \in \mathbb{R}} \frac{f(z+ih)}{ih}, \tag{99}$$

$$h_f(z) = \begin{cases} \dfrac{f(z) - f(\bar{z})}{z - \bar{z}} & \text{if } \mathrm{Re}(z) \neq 0 \\ f_{iy}(z) & \text{if } \mathrm{Re}(z) = 0, \end{cases} \tag{100}$$

then we have

$$f(A + \epsilon B) = f(A) + \epsilon f_x(A)\,\mathrm{Re}(B_1) + \epsilon f_{iy}(A)\,\mathrm{Im}(B_1) + \epsilon h_f(A)\boldsymbol{b}_2. \tag{101}$$

This means that for any functions f and g, that $\mathrm{size}_l(f(A + \epsilon B)) \leq \mathrm{size}_l(g(A + \epsilon B))$ for all quaternions B if and only if $|f(A)| \leq |g(A)|$, $|f_x(A)| \leq |g_x(A)|$, $|f_{iy}(A)| \leq |g_{iy}(A)|$, and $|h_f(A)| \leq |h_g(A)|$. In particular, the comparison of size does depend upon which characteristic length l is used.

Proof of Lemma 1. Since $\theta^* = -\theta$, we have

$$|1 + \theta|^2 = 1 - \theta^2. \tag{102}$$

Hence using Taylor's series

$$|1 + \theta|^{-1} = 1 + \tfrac{1}{2}\theta^2 + O(\theta^3) \tag{103}$$

from which it follows that

$$\widehat{(1 + \theta)} = (1 + \theta)(1 + \theta)^{-1} = 1 + \theta + \tfrac{1}{2}\theta^2 + O(\theta^3). \tag{104}$$

$\square$

Lemma 2. *The definition of $\mathcal{L}_\theta g$ in equation (41) does not depend upon the extension of g from $\mathbb{SDH}$ to a neighborhood of $\mathbb{SDH}$ in $\mathbb{DH}$.*

Proof. First note that from equation (93) we have

$$\left. \frac{d}{dr}(\widehat{1 + r\theta}) \right|_{r=0} = \theta. \tag{105}$$

Let g_1 and g_2 be two extensions of g from $\mathbb{SDH}$ to a neighborhood of $\mathbb{SDH}$ in $\mathbb{DH}$. Define

$$\tilde{g}(\eta) = g_1(\widehat{\eta}) = g_2(\widehat{\eta}). \tag{106}$$

Then

$$\left.\frac{d}{dr}\tilde{g}(\eta(1+r\theta))\right|_{r=0} = \left.\frac{d}{dr}g_1(\eta(\widehat{1+r\theta}))\right|_{r=0}$$

$$= \left.\frac{d}{dr}g_1(\eta(1+r\theta))\right|_{r=0},$$

$$(107)$$

where the second equality follows from the chain rule and equation (105). Similarly for g_2. $\qquad\square$

Proof that Definition (41) *implies Equation* (43). Using the chain rule for partial derivatives, we obtain

$$\frac{d}{dr}g(\eta(1+r\theta)) = \sum_{i=1}^{8}\left(\frac{\partial g}{\partial \eta_i}(\eta(1+r\theta))\right)\frac{d}{dr}(\eta(1+r\theta))_i$$

$$= \sum_{i=1}^{8}\left(\frac{\partial g}{\partial \eta_i}(\eta(1+r\theta))\right)(\eta\theta)_i.$$

$$(108)$$

Now set $r = 0$. $\qquad\square$

Proof that Equation(47) *implies Equation* (48). Equation (47) can be written as

$$\boldsymbol{a} = 2(\theta_1\boldsymbol{i} + \theta_2\boldsymbol{j} + \theta_3\boldsymbol{k}), \quad \boldsymbol{b} = 2(\theta_4\boldsymbol{i} + \theta_5\boldsymbol{j} + \theta_6\boldsymbol{k}), \qquad (109)$$

where θ_i is the β_i component of θ as described in (17). If f is any function of the vector dual quaternions, we have

$$\frac{\partial}{\partial\theta_i}f(\theta) = \sum_{j=1}^{3}\frac{\partial\boldsymbol{a}_j}{\partial\theta_i}\frac{\partial}{\partial\boldsymbol{a}_j}f(\theta) + \sum_{j=1}^{3}\frac{\partial\boldsymbol{b}_j}{\partial\theta_i}\frac{\partial}{\partial\boldsymbol{b}_j}f(\theta)$$

$$= \begin{cases} 2\dfrac{\partial}{\partial\boldsymbol{a}_i}f(\theta) & \text{if } 1 \le i \le 3 \\[4mm] 2\dfrac{\partial}{\partial\boldsymbol{b}_{i-3}}f(\theta) & \text{if } 4 \le i \le 6. \end{cases}$$

$$(110)$$

$\qquad\square$

Proof of Equations (53) *and* (54). We have

$$\tilde{\boldsymbol{s}} = Q\boldsymbol{s}Q^* + 2BQ^*, \qquad (111)$$

$$\tilde{\boldsymbol{n}} = Q\boldsymbol{n}Q^*, \qquad (112)$$

or

$$\boldsymbol{s} = Q^*(\tilde{\boldsymbol{s}}Q - 2B), \qquad (113)$$

$$\boldsymbol{n} = Q^*\tilde{\boldsymbol{n}}Q. \qquad (114)$$

Now if $\theta = \frac{1}{2}\boldsymbol{a} + \frac{1}{2}\epsilon\boldsymbol{b}$, then

$$\mathcal{L}_\theta(\eta) = \eta\psi = \tfrac{1}{2}Q\boldsymbol{a} + \tfrac{1}{2}\epsilon(Q\boldsymbol{b} + B\boldsymbol{a}), \tag{115}$$

that is

$$\mathcal{L}_\theta(Q) = \tfrac{1}{2}Q\boldsymbol{a}, \tag{116}$$

$$\mathcal{L}_\theta(B) = \tfrac{1}{2}(Q\boldsymbol{b} + B\boldsymbol{a}). \tag{117}$$

Remembering $\boldsymbol{a}^* = -\boldsymbol{a}$, and that $\boldsymbol{a}\boldsymbol{s} - \boldsymbol{s}\boldsymbol{a} = 2\,\boldsymbol{a} \times \boldsymbol{s}$, we obtain

$$\mathcal{L}_\theta\boldsymbol{s} = -\boldsymbol{a} \times \boldsymbol{s} - \boldsymbol{b}, \tag{118}$$

$$\mathcal{L}_\theta\boldsymbol{n} = -\boldsymbol{a} \times \boldsymbol{n}. \tag{119}$$

$\square$

Proof of Equation (62). The rate of change of work done on the parallel robot can be computed in two different ways, either using equation (29), or (57). Substituting in equations (60) and (61), we obtain

$$\mathsf{T}\boldsymbol{f} \cdot \varphi = \boldsymbol{f} \cdot \mathsf{\Lambda}\varphi = \mathsf{\Lambda}^T\boldsymbol{f} \cdot \varphi, \tag{120}$$

the last equality being a standard formula for the transpose of a matrix. Since this is true for arbitrary actuator forces $\boldsymbol{f}$ and end effector twists φ, the result follows. $\square$

Proof of Equation (64). Applying the rules given in Section 6, we obtain

$$\mathcal{L}_\theta\mathcal{L}_\psi g = \mathcal{L}_\theta \left(\frac{\partial g}{\partial \eta} \cdot (\eta\psi) \right) \tag{121}$$

$$= \sum_{i=1}^{8} \mathcal{L}_\theta \left(\frac{\partial g}{\partial \eta_i} \right) (\eta\psi)_i + \frac{\partial g}{\partial \eta} \cdot \mathcal{L}_\theta(\eta\psi) \tag{122}$$

$$= \sum_{i=1}^{8} \left(\sum_{j=1}^{8} \left(\frac{\partial^2 g}{\partial \eta_j \eta_i} \right) (\eta\theta)_j \right) (\eta\psi)_i + \frac{\partial g}{\partial \eta} \cdot (\eta\psi\theta) \tag{123}$$

$$= (\eta\psi) \cdot \left[\frac{\partial^2 g}{\partial \eta^2} \right] (\eta\theta) + \frac{\partial g}{\partial \eta} \cdot (\eta\theta\psi). \tag{124}$$

$\square$

As a corollary to equation (64), we obtain the well known identity:

$$\mathcal{L}_\theta\mathcal{L}_\psi g - \mathcal{L}_\psi\mathcal{L}_\theta g = \mathcal{L}_{(\theta\psi-\psi\theta)}g, \tag{125}$$

which implies that Lie derivatives do not necessary commute.

Proof of Equation (65). We wish to find the Jacobian δ and the Hessian H of $\tilde{b}$ at the origin, where

$$\tilde{b}(\theta) = b(\eta\widehat{(1+\theta)}). \tag{126}$$

We can find this by considering its Taylor series expansion

$$\tilde{b}(\theta) = \tilde{b}(0) + \sum_{i=1}^{6} \delta_i \theta_i + \tfrac{1}{2} \sum_{i,j=1}^{6} \mathsf{H}_{i,j} \theta_i \theta_j + O(\theta^3), \tag{127}$$

where

$$\theta = \sum_{i=1}^{6} \theta_i \beta_i. \tag{128}$$

Using the Taylor series, and using equation (93), one obtains

$$\begin{aligned}
\tilde{b}(\theta) &= b(\eta\widehat{(1+\theta)}) = b(\eta(1+\theta+\tfrac{1}{2}\theta^2 + O(\theta^3))) \\
&= b(\eta) + \frac{\partial b}{\partial \eta} \cdot (\eta(\theta+\tfrac{1}{2}\theta^2)) \\
&\quad + \tfrac{1}{2}(\eta\theta) \cdot \left[\frac{\partial^2 b}{\partial \eta^2}\right](\eta\theta) + O(\theta^2) \\
&= b(\eta) + \frac{\partial b}{\partial \eta} \cdot (\eta(\theta)) \\
&\quad + \tfrac{1}{2}(\eta\theta) \cdot \left[\frac{\partial^2 b}{\partial \eta^2}\right](\eta\theta) + \frac{\partial b}{\partial \eta} \cdot (\tfrac{1}{2}\eta\theta^2) + O(\theta^3).
\end{aligned} \tag{129}$$

Now by comparing coefficients, and considering equations (43) and (64), we obtain

$$\delta_i = \mathcal{L}_i b(\eta), \quad (1 \le i \le 6), \tag{130}$$

and

$$\begin{aligned}
\mathsf{H}_{i,j} &= (\eta\beta_i) \cdot \left[\frac{\partial^2 b}{\partial \eta^2}\right](\eta\beta_j) + \frac{\partial b}{\partial \eta} \cdot (\tfrac{1}{2}\eta(\beta_i\beta_j + \beta_j\beta_i)) \\
&= \tfrac{1}{2}(\mathcal{L}_i\mathcal{L}_j b(\eta) + \mathcal{L}_j\mathcal{L}_i b(\eta)), \quad (1 \le i,j \le 6).
\end{aligned} \tag{131}$$

$\square$

Proof of Equation (67). Suppose that the end effector is moving with pose $\eta = Q + \epsilon B$, and twist $\phi = \tfrac{1}{2}\boldsymbol{w} + \tfrac{1}{2}\epsilon\boldsymbol{v}$, then with respect to the fixed frame of reference, the velocity of the attachment point $\boldsymbol{r}_k$ is $Q(\boldsymbol{v} + \boldsymbol{w} \times \boldsymbol{r}_k)Q^*$. And if a force f_k is applied along the direction $\boldsymbol{u}_k$, then the force applied to the end-effector is $f_k Q\boldsymbol{u}_k Q^*$ with respect to the fixed frame of reference.

Hence computing the rate of change of virtual work, we obtain

$$f_k \dot{\ell}_k = (\boldsymbol{v} + \boldsymbol{w} \times \boldsymbol{r}_k) \cdot f_k \boldsymbol{u}_k. \tag{132}$$

Therefore

$$\dot{\ell}_k = \boldsymbol{u}_k \cdot \boldsymbol{v} + \boldsymbol{r}_k \times \boldsymbol{u}_k \cdot \boldsymbol{w}. \tag{133}$$

The result follows by equation (60). $\square$

For complex situations, where the cables might pass through pulleys, the simplicity of equation (67) can be a bit surprising. This might best be intuitively understood by seeing that the involute of the curve describing the shape of the pulley is the curve traced by the end effector attachment point when the cable length of that actuator is kept fixed, and that the evolute is the opposite process.

Proof of Equation (69). Note that

$$\mathcal{L}_\psi \boldsymbol{u}_k = -\frac{1}{|\boldsymbol{r}_k - \boldsymbol{s}_k|} \mathsf{P}_{\boldsymbol{u}_k} \mathcal{L}_\psi \boldsymbol{s}_k. \tag{134}$$

Now apply equation (53). $\square$

Next, we justify the Newton-Raphson Methods for forward kinematics. The method as usually states only applies to linear vector spaces, whereas we are working on the non-linear manifold of unit dual quaternions. Thus given η_k, we need to define a map from the vector space of vector dual quaternions to unit dual quaternions close to η_k. See, for example, [18].

Most papers on the Newton-Raphson Method on manifolds construct this map using the so called exp function [9, 11, 12]. So the map is

$$\theta \mapsto \eta_k \exp(\theta). \tag{135}$$

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 11. Another exp map is to follow a one-parameter subgroup, or equivalently, equation (94). We do not take these approaches. Our approach is to normalize:

$$\theta \mapsto \eta_k \widehat{(1 + \theta)}. \tag{136}$$

(However, this is numerically close to the second approach, as is shown by equation (95). Using normalization instead of the exp map slightly reduces the complexity of the calculations as transcendental trigonometric functions are not required. But if one wants to use the exp map, simply replace $\widehat{(1 + \theta)}$ by $\exp(\theta)$ throughout.)

The main substantive difference between these methods, and the standard method that is used on linear vector spaces, is that the map from the linear space of vector dual quaternions to the manifold of unit dual quaternions changes with each iteration, since the map depends

upon η_k. But the theory that the Newton-Raphson Method converges with quadratic order is based upon examination of each iteration separately, so this shouldn't pose a great issue.

Justification of Equation (80). If $\mathsf{F} : \mathbb{R}^6 \to \mathbb{R}^6$ is a function for which we wish to solve for $\mathsf{F}(\boldsymbol{x}) = 0$, the method is to iterate

$$\boldsymbol{x} - \left[\frac{\partial \mathsf{F}}{\partial \boldsymbol{x}}\right]^{-1} F(\boldsymbol{x}). \tag{137}$$

Our approach, then, is to consider the map

$$\theta \mapsto \mathsf{F}(\theta) := \mathsf{L}(\eta_k \widehat{(1 + \theta)}) - \boldsymbol{\ell}. \tag{138}$$

The prior guess is then $\theta_k = 0$. We have that

$$\left.\frac{\partial \mathsf{F}}{\partial \theta}\right|_{\theta=0} = \Lambda(\eta_k). \tag{139}$$

The result follows. □

Justification of Equation (83). We seek to find the pose η so that $\mathsf{L}(\eta)$ is close as possible to $\boldsymbol{\ell}$. This is performed by minimizing the *loss function*

$$b(\eta) = \tfrac{1}{2}|\mathsf{L}(\eta) - \boldsymbol{\ell}|^2. \tag{140}$$

The standard Newton-Raphson Method for optimizing the real valued quantity $F(\boldsymbol{x})$, where $\boldsymbol{x}$ is an element of a vector space, is to iterate

$$\boldsymbol{x} - \left[\frac{\partial^2 F}{\partial \boldsymbol{x}^2}\right]^{-1} \frac{\partial F}{\partial \boldsymbol{x}}. \tag{141}$$

In our case $\boldsymbol{x}$ is θ,

$$F(\theta) = b(\eta_k \widehat{(1 + \theta)}), \tag{142}$$

and our previous iterate is $\theta_k = 0$. The Jacobian is

$$\delta_k = \left.\frac{\partial F(\theta)}{\partial \theta}\right|_{\theta=0} = (\mathcal{L}_i b(\eta_k))_{1 \le i \le 6} = \left(\sum_{m=1}^{n} \Lambda_{m,i}(\mathsf{L}_m(\eta_k) - \ell_m)\right)_{1 \le i \le 6}, \tag{143}$$

and the Hessian is H_k, which by equation (65) is

$$\mathsf{H}_k = \left[\tfrac{1}{2}(\mathcal{L}_i \mathcal{L}_j b(\eta_k) + \mathcal{L}_j \mathcal{L}_i b(\eta_k))\right]_{1 \le i,j \le 6}, \tag{144}$$

noting that

$$\mathcal{L}_i \mathcal{L}_j b(\eta_k) = \sum_{m=1}^{n} \Lambda_{m,i}\Lambda_{m,j} + \mathcal{L}_i \Lambda_{m,j}(\mathsf{L}_m(\eta_k) - \ell_m). \tag{145}$$

□

Now we work on proving Theorem 1. We use the Euler-Lagrange equations. (However, one could also use standard formulas for rotating bodies, and Newtonian physics, to obtain the same result.)

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

$$\alpha \times \beta = \tfrac{1}{2}(\alpha\beta - \beta\alpha) = \boldsymbol{a} \times \boldsymbol{c} + \epsilon(\boldsymbol{a} \times \boldsymbol{d} + \boldsymbol{b} \times \boldsymbol{c}). \qquad (146)$$

Define the *adjoint products* of vector dual quaternions by

$$\alpha \ltimes \beta = \boldsymbol{c} \times \boldsymbol{a} + \boldsymbol{d} \times \boldsymbol{b} + \epsilon(\boldsymbol{c} \times \boldsymbol{b}), \qquad (147)$$

$$\alpha \rtimes \beta = -\beta \ltimes \alpha, \qquad (148)$$

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

$$(\alpha \times \beta) \cdot \gamma = \alpha \cdot (\gamma \ltimes \beta) = \beta \cdot (\alpha \rtimes \gamma). \qquad (149)$$

(Note that in the Lie algebra literature, the map $\beta \mapsto \alpha \times \beta$ is often denoted ad_α. Thus the map $\gamma \mapsto \alpha \rtimes \gamma$ is the formal dual of ad_α.)

Theorem 2. *If the kinetic energy e satisfies equation* (84), *and $v = v(\eta)$ denotes the potential energy, then the equation of motion is*

$$\tau = \mu_1 + \mu_2 + \mathsf{M}\alpha, \qquad (150)$$

where

$$\mu_1 = \mathcal{L}_\varphi \mathsf{M}\varphi - \tfrac{1}{2}\mathcal{L}(\varphi \cdot \mathsf{M}\varphi) + \mathcal{L}v, \qquad (151)$$

and for any constant vector dual quaternion ψ we have

$$\psi \cdot \mu_2 = 2\varphi \cdot \mathsf{M}(\psi \times \varphi) = 2\mathsf{M}\varphi \cdot (\psi \times \varphi), \qquad (152)$$

that is,

$$\mu_2 = 2(\mathsf{M}\varphi) \ltimes \varphi. \qquad (153)$$

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

$$\begin{aligned}
\theta &\mapsto \eta(\theta) \\
&= \eta_0 \widehat{(1 + \theta)} \\
&= \eta_0(1 + \theta + \tfrac{1}{2}\theta^2) + O(\theta^3),
\end{aligned} \qquad (154)$$

where in the last inequality we used equation (93). Then we have

$$\begin{aligned}
\varphi &= \eta^{-1}\dot{\eta} \\
&= \widehat{(1 + \theta)}^{-1}(\dot{\theta} + \tfrac{1}{2}(\dot{\theta}\theta + \theta\dot{\theta})) + O(\theta^2) \\
&= \dot{\theta} + \tfrac{1}{2}(\dot{\theta}\theta - \theta\dot{\theta}) + O(\theta^2),
\end{aligned} \qquad (155)$$

and
$$\alpha = \ddot{\theta} + O(\theta). \tag{156}$$

The Lagrangian of the parallel robot is
$$l = e - v = \tfrac{1}{2}\varphi \cdot \mathsf{M}\varphi - v. \tag{157}$$

We use the local coordinate system θ given by equation (154). The Euler-Lagrange Equation [4, 14] tells us
$$\frac{d}{dt}\left(\frac{\partial l}{\partial \dot{\theta}}\right) - \frac{\partial l}{\partial \theta} = \tau. \tag{158}$$

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
$$\frac{\partial f}{\partial \theta} = \mathcal{L}f. \tag{159}$$

We have
$$\frac{d}{dt}\left(\frac{\partial e}{\partial \dot{\theta}}\right) = \mathsf{M}\alpha + \dot{\mathsf{M}}\varphi$$
$$= \mathsf{M}\alpha + \left(\dot{\theta} \cdot \frac{\partial \mathsf{M}}{\partial \theta}\right)\dot{\theta} \tag{160}$$
$$= \mathsf{M}\alpha + \left(\varphi \cdot \frac{\partial \mathsf{M}}{\partial \theta}\right)\varphi,$$

and
$$\frac{\partial e}{\partial \theta} = \dot{\theta} \cdot \mathsf{M}\frac{\partial}{\partial \theta}(\dot{\theta}\theta - \theta\dot{\theta}) + \tfrac{1}{2}\dot{\theta} \cdot \left(\frac{\partial \mathsf{M}}{\partial \theta}\right)\dot{\theta}$$
$$= \varphi \cdot \mathsf{M}\frac{\partial}{\partial \theta}(\varphi\theta - \theta\varphi) + \tfrac{1}{2}\varphi \cdot \left(\frac{\partial \mathsf{M}}{\partial \theta}\right)\varphi. \tag{161}$$

Note that if f is any linear function whose domain is the vector dual quaternions, then
$$\psi \cdot \left(\frac{\partial}{\partial \theta}f(\theta)\right) = f(\psi). \tag{162}$$

Thus taking the dot product of equation (161) with any constant vector dual quaternion ψ, we obtain the result. $\qquad \square$

Lemma 3. *If v is the potential energy of the end effector in a constant gravity field, whose value with respect to the moving frame is $\boldsymbol{g}$, then*
$$\mathcal{L}v = -2m_0(\boldsymbol{r}_0 \times \boldsymbol{g} + \epsilon\boldsymbol{g}). \tag{163}$$

Proof. Let $\boldsymbol{r}$ be a constant point expressed with respect to the moving frame. We have

$$v = m_0 \boldsymbol{g} \cdot (\boldsymbol{r} - \boldsymbol{r}_0). \tag{164}$$

If $\theta = \frac{1}{2}\boldsymbol{a} + \frac{1}{2}\epsilon\boldsymbol{b}$ is a vector dual quaternion, then by equation (54), it follows that

$$\begin{aligned}
\mathcal{L}_\theta v &= -m_0((\boldsymbol{a} \times \boldsymbol{g}) \cdot (\boldsymbol{r} - \boldsymbol{r}_0) + \boldsymbol{g} \cdot (\boldsymbol{a} \times \boldsymbol{r} + \boldsymbol{b})) \\
&= -m_0(\boldsymbol{a} \cdot \boldsymbol{r}_0 \times \boldsymbol{g} + \boldsymbol{b} \cdot \boldsymbol{g}).
\end{aligned} \tag{165}$$

$\square$

Proof of Theorem 1. The potential energy part is covered by Lemma 3. 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 2, various vector identities, and remembering equation (36).

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 vector dual quaternion ψ, we have

$$\begin{aligned}
&\psi \cdot (\mathcal{L}_\varphi(\mathsf{M}\varphi) - \tfrac{1}{2}\mathcal{L}(\varphi \cdot \mathsf{M}\varphi)) \\
&\quad = (\mathcal{L}_\varphi(\psi \cdot \mathsf{M}\varphi) - \tfrac{1}{2}\mathcal{L}_\psi(\varphi \cdot \mathsf{M}\varphi)) \\
&\quad = \mathcal{L}_\varphi((\Lambda\psi) \cdot \mathsf{M}_0(\Lambda\varphi)) - \tfrac{1}{2}\mathcal{L}_\psi((\Lambda\varphi) \cdot \mathsf{M}_0(\Lambda\varphi)) \\
&\quad = \mathcal{L}_\varphi \mathcal{L}_\varphi \mathsf{L} \cdot \mathsf{M}_0 \mathcal{L}_\psi \mathsf{L} + \mathcal{L}_\varphi \mathcal{L}_\psi \mathsf{L} \cdot \mathsf{M}_0 \mathcal{L}_\varphi \mathsf{L} - \mathcal{L}_\psi \mathcal{L}_\varphi \mathsf{L} \cdot \mathsf{M}_0 \mathcal{L}_\varphi \mathsf{L} \\
&\quad = \mathcal{L}_\varphi(\Lambda\varphi) \cdot \mathsf{M}_0 \Lambda\psi + \mathcal{L}_{(\varphi\psi - \psi\varphi)} \mathsf{L} \cdot \mathsf{M}_0 \Lambda\varphi \\
&\quad = \psi \cdot \Lambda^T \mathsf{M}_0 \mathcal{L}_\varphi(\Lambda\varphi) + \Lambda(\varphi\psi - \psi\varphi) \cdot \mathsf{M}_0 \Lambda\varphi \\
&\quad = \psi \cdot \Lambda^T \mathsf{M}_0 \mathcal{L}_\varphi(\Lambda\varphi) + (\varphi\psi - \psi\varphi) \cdot \mathsf{M}_0\varphi,
\end{aligned} \tag{166}$$

where we used equation (125). Then it is simply a matter of collecting terms. $\square$

13. Conclusion

This paper has given a comprehensive and consistent description of how to use dual quaternions to represent poses, rigid motions, twists, and wrenches. We have introduced the notion of the Lie derivative for dual quaternions. We have shown how these formula are helpful for first producing Newton-Raphson methods for solving the forward kinematics problem for parallel robots, and secondly for a self contained derivation for the dynamic equations of motion of the end effector that includes the inertia of the actuators.

Finally, in equation (93), we give an approximation of the normalization of a vector dual quaternion perturbation of the identity, which shows that it is equal up to the second order to the exponential of the vector dual quaternion. This equation was essential for calculating the Hessian in the forwards kinematics algorithms. We feel that this formula will be of independent interest to other researchers in the field of dual quaternions.

References

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

[2] Om Prakash Agrawal, Hamilton operators and dual-number-quaternions in spatial kinematics, Mechanism and Machine Theory, 22(6), 569-575, Jan 1987.

[3] S.L. Altmann, Hamilton, Rodrigues, and the Quaternion Scandal, Mathematics Magazine, 62(5), 291-308, Dec 1989.

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

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

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

[7] Gregory S Chirikjian, Robert Mahony, Sipu Ruan, and Jochen Trumpf, Pose Changes From a Different Point of View, Journal of Mechanisms and Robotics, 10(2):021008-021008-12, Feb 2018.

[8] M.A. Clifford, Preliminary Sketch of Biquaternions, Proceedings of the London Mathematical Society, Volume s1-4, Issue 1, 81-395, Nov 1871, `https://doi.org/10.1112/plms/s1-4.1.381`.

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

[10] J.R. Dooley and J.M. McCarthy, Spatial rigid body dynamics using dual quaternion components, Proceedings 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, vol.1, 90-95, 1995, DOI: 10.1109/ROBOT.1991.131559.

[11] 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.

[12] 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.

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

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

[15] Clement Gosselin, and Jorge Angeles, Singularity analysis of closed-loop kinematic chains, IEEE Transactions on Robotics and Automation, 6(3), June 1990.

[16] 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), 319-324, July 2008, DOI: 10.1007/s11633-008-0319-1.

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

[18] 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, Pacific Grove, CA, USA, Vol 1, 136-139, 2004, DOI: 10.1109/ACSSC.2004.1399106.

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

[20] L. Kavan, S. Collins, J. Žára, C. O'Sullivan, Geometric Skinning with Approximate Dual Quaternion Blending, ACM Transactions on Graphics, 27(4), Article 105, Oct 2008.

[21] 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`.

[22] 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`.

[23] Hugo T.M. Kussaba, Luis F.C. Figueredo, João Y. Ishihara, and Bruno V. Adorno, Hybrid kinematic control for rigid body pose stabilization using dual quaternions, Journal of the Franklin Institute, 354(7), 2769-2787, May 2017.

[24] John Lee, Introduction to Smooth Manifolds, Graduate Texts in Mathematics, 218, 2nd ed., New York London: Springer-Verlag, ISBN 978-1-4419-9981-8, OCLC 808682771, 2012.

[25] Alexander McAulay, Octonions: a development of Clifford's Biquaternions, Cambridge University Press, 1988.

[26] J.-P. Merlet, Singularity of Cable-Driven Parallel Robot With Sagging Cables: Preliminary Investigation, International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 504-509, 2019, DOI: 10.1109/ICRA.2019.8794218.

[27] Stephen Montgomery-Smith, Control of systems by parallel actuators, WSEAS Transactions on Systems and Control, 17, 207-213, 2022.

[28] Stephen Montgomery-Smith, Functional calculus for dual quaternions, Advances in Applied Clifford Algebras 33, 36, 2023, https://doi.org/10.1007/s00006-023-01282-y.

[29] Stephen Montgomery-Smith and Cecil Shy, An introduction to using dual quaternions to study kinematics, arXiv:2203.13653, `https://arxiv.org/abs/2203.13653`.

[30] Stephen Montgomery-Smith and Cecil Shy, Using Lie Derivatives with Dual Quaternions for Parallel Robots, Machines 2023, 11(12), 1056, `https://doi.org/10.3390/machines11121056`.

[31] Stephen Montgomery-Smith and Cecil Shy, Correction: Montgomery-Smith, S.; Shy, C. Using Lie Derivatives with Dual Quaternions for Parallel Robots. Machines 2023, 11, 1056, Machines 2024, 12(12), 830, `https://doi.org/10.3390/machines12120830`.

[32] A. Perez and J.M. McCarthy, Dual Quaternion Synthesis of Constrained Robotic Systems, ASME. J. Mech. Des. 126(3) 425-435, May 2004.

[33] 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.

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

[35] M. Schilling, Universally manipulable body models—dual quaternion representations in layered and dynamic MMCs, Auton Robot 30, 399, 2011, `https://doi.org/10.1007/s10514-011-9226-3https://link.springer.com/article/10.1007/s10514-011-9226-3`.

[36] M. Schilling, Hierarchical Dual Quaternion-Based Recurrent Neural Network as a Flexible Internal Body Model, 2019 International Joint Conference on Neural Networks (IJCNN), 1-8, 2019, DOI: 10.1109/IJCNN.2019.8852328, `https://ieeexplore.ieee.org/abstract/document/8852328`.

[37] J.M. Selig, Geometric fundamentals of robotics, Springer-Verlag New York Inc., 2nd edition, 2005.

[38] J.M. Selig, Exponential and Cayley Maps for Dual Quaternions, Advances in Applied Clifford Algebras, 20(3-4), 923-936, May 2010.

[39] Frederico F. A. Silva, Juan José Quiroz-Omaña, and Bruno V. Adorno, Dynamics of Mobile Manipulators using Dual Quaternion Algebra, Journal of Mechanisms and Robotics, 1-24, April 2022.

[40] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, Robot Modeling and Control, Wiley, 2006.

[41] 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), 1352-1364, 2012.

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

[43] 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, 27-36, 2017, `http://dx.doi.org/10.1016/j.mechmachtheory.2016.08.003`.