Performance Study on dqRNEA – A Novel Dual Quaternion based Recursive Newton-Euler Inverse Dynamics Algorithms

C.M. Farias, L.F.C. Figueredo, J.Y. Ishihara

***Abstract*— In this paper, the well known recursive Newton-Euler inverse dynamics algorithm for serial manipulators is reformulated into the context of the algebra of Dual Quaternions. Here we structure the forward kinematic description with screws and line displacements rather than the well established Denavit-Hartemberg parameters, thus accounting better efficiency, compactness and simpler dynamical models. Furthermore, the backwards iteration uses the previously calculated values for estimating the joint space torques. In addition, a cost analysis of the main Dual Quaternions operations and of the Newton-Euler inverse dynamics algorithm as a whole is made and compared with other results in the literature.**

I. Introduction

Algorithms for rigid-body dynamics computation play a crucial role for simulation of motion, analysis of forces, torques and for the design of control techniques in robotics [1]–[3]. Indeed, rigid-body dynamics are at the core of many recent robotic applications in different fields, such as legged robot stabilization, forceful Human-Robot Interaction (HRI), computer animation and even biomechanics [4].

In robotics literature, given articulated system's motion, two major algorithms for computing their dynamics stand out: the recursive Newton-Euler algorithm (RNEA) and the Lagrange-Euler (LE) formulation. The LE equations and the RNEA are both used to describe the relations between the joint torques, forces at the end effector and kinematic variables, however, they differ in many other aspects. Concerning only computational complexity, the Newton–Euler formulation is considerably more efficient due to its inherently recursive formulation. In this work, the recursive Newton-Euler inverse dynamics algorithm for serial manipulators is reformulated into the context of the algebra of Dual Quaternions and, furthermore, the resulting modeling in terms of computational complexity is addressed.

Computing the rigid body kinematics and dynamics largely depends on the choice of rigid body motion representation. The most widely employed technique to compute the robot forward kinematics is based on the Denavit and Hartenberg (D-H) notation combined with rigid body transformations based on the non-minimal representation of homogeneous transformation matrices (HTM) [5], [6]. Nonetheless, it was argued in [2], [7]–[9] that the unit dual quaternion (UDQ) in addition to being non-minimal and free of singularities, it is also a more compact, efficient and less computationally demanding representation for rigid-body displacements compared to HTM. In addition, dual quaternions have strong algebraic properties and can be used to represent rigid motions, twists, wrenches and several geometrical primitives—e.g., Plücker lines, planes—in a very straightforward way [10]. The benefits of using dual quaternion algebra have been discussed also in many works with different applications comprising rigid body motion stabilization, tracking, multiple body coordination [11], [12], kinematic control of manipulators with single and multiple arms and human-robot interaction [13], [14].

Particularly, in [6], Ozgur has shown that the computational advantages of using UDQ are even more relevant when following the screw theory approach based on line transformation [6], [15] for the forward kinematic modeling. Thus, in this paper, we extend the concepts presented in [6] to address acceleration transformations from joint space to dual quaternion task space. Furthermore, using the unified framework of dual quaternions we also design a novel dual quaternion based formulation for computing the forces and moments acting on a n-link manipulator. We take a recursive approach to accomplish this task, resulting in a novel formulation of the RNEA algorithm based on dual quaternion algebra (dqRNEA). The proposed solution also takes as inspiration the rigid body dynamics analysis from Tsiotras [16], [17] valid for a single-rigid body link analysis.

In this paper, to the best of authors' knowledge, a rigid-body dynamics algorithm is designed for the first time using the algebra of dual quaternions following the Screw theory approach. The computational complexity of the proposed algorithm is also addressed and compared to other methods found in the literature.

II. Mathematical Background

This section provides the reader background regarding quaternion and dual quaternion algebra as well as algebraic properties and operators required throughout this work. Furthermore, novel linear operators are introduced for performance and algorithmic efficiency.

A. Quaternion Algebra

Let $\hat{\imath}, \hat{\jmath}, \hat{k}$ be the three quaternionic units such that $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1$. The algebra of quaternions as defined in [18], [19] is generated by the basis elements $1, \hat{\imath}, \hat{\jmath}$, and $\hat{k}$, which yields the set $\mathbb{H} \triangleq \left\{\eta + \boldsymbol{\mu} \;:\; \boldsymbol{\mu}{=}\mu_1\hat{\imath} + \mu_2\hat{\jmath} + \mu_3\hat{k},\ \eta, \mu_1, \mu_2, \mu_3{\in}\mathbb{R}\right\}$.An element $\boldsymbol{q} \in \mathbb{H}$ may be decomposed into a real component and an imaginary component $\mathrm{Re}\left(\boldsymbol{q}\right) \triangleq \eta$ and $\mathrm{Im}\left(\boldsymbol{q}\right) \triangleq \mu_1\hat{\imath} + \mu_2\hat{\jmath} + \mu_3\hat{k}$, such that $\boldsymbol{q} = \mathrm{Re}\left(\boldsymbol{q}\right) + \mathrm{Im}\left(\boldsymbol{q}\right)$. For each quaternion element $\boldsymbol{q}$, there is one correspondent

This work was partially supported by CNPq, CAPES. C.M. Farias and J.Y Ishihara are with LARA-UnB, Brasilia-DF, Brazil. L.F.C. Figueredo is with University of Leeds, Leeds.

quaternion conjugate given by $\boldsymbol{q}^* \triangleq \mathrm{Re}(\boldsymbol{q}) - \mathrm{Im}(\boldsymbol{q})$, which in turn defines the quaternion norm $\|\boldsymbol{q}\| \triangleq \sqrt{\boldsymbol{q}\boldsymbol{q}^*}$. The norm of quaternion elements coincides with the Euclidean norm of a vector $\mathrm{vec}\,\boldsymbol{q} \triangleq [\eta \;\; \mu_1 \;\; \mu_2 \;\; \mu_3]^T \in \mathbb{R}^4$, that is, $\|\boldsymbol{q}\| = \|\mathrm{vec}\,\boldsymbol{q}\|$. Quaternion elements with real part equal to zero belong to the set of *pure quaternions* $\mathbb{H}_0 \triangleq \{\boldsymbol{q}_0 : \boldsymbol{q}_0 \in \mathbb{H}, \mathrm{Re}(\boldsymbol{q}_0) = 0\}$ whereby $\boldsymbol{q}_0^* = -\boldsymbol{q}_0$. Such elements are isomorphic to three-dimensional vectors and, similarly, they can be used to represent translation, angular and linear velocities, accelerations, momentum and wrenches. Hence, kinematics and dynamics can be compactly represented in a unified framework. Within the linear space of pure quaternions, we can define the inner and cross product operations similarly to their vectorial counterpart [7]. Given two pure quaternions $\boldsymbol{a} = a_1\hat{\imath} + a_2\hat{\jmath} + a_3\hat{k}$ and $\boldsymbol{b} = b_1\hat{\imath} + b_2\hat{\jmath} + b_3\hat{k}$ we have the inner product[1]

$$\boldsymbol{a} \cdot \boldsymbol{b} \triangleq -\frac{\boldsymbol{a}\boldsymbol{b} + \boldsymbol{b}\boldsymbol{a}}{2} = a_1b_1 + a_2b_2 + a_3b_3, \tag{1}$$

and the cross product

$$\boldsymbol{a} \times \boldsymbol{b} \triangleq \frac{\boldsymbol{a}\boldsymbol{b} - \boldsymbol{b}\boldsymbol{a}}{2}. \tag{2}$$

In addition to three-dimensional vectors, a quaternion element can also represent arbitrary rotations when constrained to the set of unit quaternions $\mathcal{S}^3 \triangleq \{\boldsymbol{q} \in \mathbb{H} : \|\boldsymbol{q}\| = 1\}$. The set $\mathcal{S}^3$ together with the multiplication operation forms the Lie group of unit quaternions, Spin(3) [21]. An arbitrary rotation angle $\phi \in \mathbb{R}$ around the rotation axis $\boldsymbol{n} \in \mathbb{H}_p \cap \mathcal{S}^3$, with $\boldsymbol{n} = n_x\hat{\imath} + n_y\hat{\jmath} + n_z\hat{k}$, is represented by the unit quaternion $\boldsymbol{r} = \cos(\phi/2) + \sin(\phi/2)\boldsymbol{n}$ [22].

B. Dual Quaternions

Dual Quaternions are an extension of quaternions introduced to describe the complete and coupled rigid body motion [21]. The dual quaternion algebra is constituted by the set $\mathbb{H} \otimes \mathbb{D} \triangleq \{\underline{\boldsymbol{q}} = \boldsymbol{q_P} + \varepsilon\boldsymbol{q_D} \mid \boldsymbol{q_P}, \boldsymbol{q_D} \in \mathbb{H}\}$,where ε is called dual unit with $\varepsilon^2 = 0$, $\varepsilon \neq 0$. A dual quaternion element can also be decomposed in primary and dual parts $\mathcal{P}(\underline{\boldsymbol{q}}) = \boldsymbol{q_P}$ and $\mathcal{D}(\underline{\boldsymbol{q}}) = \boldsymbol{q_D}$, respectively. And, for each dual quaternion element $\underline{\boldsymbol{q}}$, there exist one correspondent conjugate $\underline{\boldsymbol{q}}^* \triangleq \boldsymbol{q}^* + \varepsilon\boldsymbol{q}'^*$ composed with the quaternion conjugate of the primary and dual parts. Under multiplication, the subset of *unit* dual quaternions $\underline{\boldsymbol{S}} \triangleq \{\underline{\boldsymbol{q}} \in \mathbb{H} \otimes \mathbb{D} : \|\underline{\boldsymbol{q}}\| = 1\}$, forms the Lie group $\mathrm{Spin}(3) \ltimes \mathbb{R}^3$, whose identity element is 1 and the group inverse of $\underline{\boldsymbol{x}} \in \underline{\boldsymbol{S}}$ is $\underline{\boldsymbol{x}}^*$ [21].

The subset of dual quaternions constituted by pure quaternions belong to the set of *pure dual quaternions* $\mathbb{H}_0 \otimes \mathbb{D} \triangleq \left\{\underline{\boldsymbol{q}}_0 = \boldsymbol{q_P} + \varepsilon\boldsymbol{q_D} \mid \boldsymbol{q_P}, \boldsymbol{q_D} \in \mathbb{H}_0\right\}$ where $\underline{\boldsymbol{q}}_0^* = -\underline{\boldsymbol{q}}_0$. Similar to pure quaternions, elements in $\mathbb{H}_0 \otimes \mathbb{D}$ can be deployed into the kinematics and dynamics analysis to compactly express the coupled angular and linear generalized rigid body twist, accelerations, momentum and wrenches. In this context, the primary usually represent the angular vector whilst the dual part denote the generalized linear velocity and acceleration with the proper attitude coupling. Moreover, in a similar fashion, the cross product may also be defined for pure dual quaternions

$$\underline{\boldsymbol{a}} \times \underline{\boldsymbol{b}} \triangleq \frac{\underline{\boldsymbol{a}}\underline{\boldsymbol{b}} - \underline{\boldsymbol{b}}\underline{\boldsymbol{a}}}{2}. \tag{3}$$

Also, we can also define the inner product for $\mathbb{H}_0 \otimes \mathbb{D}$ by taking the combined real value of the inner product from the primary part and dual parts, that is,

$$\underline{\boldsymbol{a}} \odot \underline{\boldsymbol{b}} \triangleq \boldsymbol{a}_P \cdot \boldsymbol{b}_P + \boldsymbol{a}_D \cdot \boldsymbol{b}_D, \tag{4}$$

which is similar to the Euclidean norm of corresponding vectors and is similar to the double-geodesic metric provided in Bullo and Murray [23] for the rigid body transformations using the group of Euclidean displacements.[2]

C. Hamilton Operators

As an associative division algebra over $\mathbb{R}$, the quaternion algebra is endowed with classic operations such as addition, scalar multiplication, quaternion multiplication. Also, dual quaternions are endowed with similar operators—although not being a division ring.[3] Nonetheless, it is also well-known that $\mathbb{H}$ is a non-commutative group, and the same is valid for the dual quaternion group—that is, if $\underline{\boldsymbol{a}}$ and $\underline{\boldsymbol{b}}$ are dual quaternions, then $\underline{\boldsymbol{a}\boldsymbol{b}} \neq \underline{\boldsymbol{b}\boldsymbol{a}}$. In order to perform multiplicative commutativity, a common strategy is to take the matrix algebra representation of quaternions or dual quaternions [7], [10]. In [26], the commutative property is obtained by changes in the sign of the resulting matrix representation in $SU(2)$—isomorphic to the quaternion group. The resulting orthogonal matrix is known as the Hamilton operator $\overset{\pm}{\boldsymbol{H}}(\boldsymbol{x}) \in \mathbb{R}^{4\times4}$ of the quaternion element $\boldsymbol{x}$ [27]. Thus, two quaternions $\boldsymbol{a}$ and $\boldsymbol{b}$ can commute as follows

$$\boldsymbol{y} = \boldsymbol{a}\boldsymbol{b} = \underline{\mathrm{vec}}\left(\overset{+}{\boldsymbol{H}}(\boldsymbol{a})\,\mathrm{vec}\,\boldsymbol{b}\right) = \underline{\mathrm{vec}}\left(\overset{-}{\boldsymbol{H}}(\boldsymbol{b})\,\mathrm{vec}\,\boldsymbol{a}\right), \tag{5}$$

where $\mathrm{vec} : \mathbb{H} \to \mathbb{R}^4$ is the mapping from quaternion to $\mathbb{R}^4$ and $\underline{\mathrm{vec}}$ is the inverse mapping. The Hamilton operators for the right and left multiplications can be computed from the matrix representation of orthogonal vectors from $\boldsymbol{x}$, that is,

$$\begin{aligned}\overset{+}{\boldsymbol{H}}(\boldsymbol{x}) &= [\mathrm{vec}\,\boldsymbol{x} \;\; \mathrm{vec}(\boldsymbol{x}\hat{\imath}) \;\; \mathrm{vec}(\boldsymbol{x}\hat{\jmath}) \;\; \mathrm{vec}\left(\boldsymbol{x}\hat{k}\right)];\\ \overset{-}{\boldsymbol{H}}(\boldsymbol{x}) &= [\mathrm{vec}\,\boldsymbol{x} \;\; \mathrm{vec}(\hat{\imath}\boldsymbol{x}) \;\; \mathrm{vec}(\hat{\jmath}\boldsymbol{x}) \;\; \mathrm{vec}\left(\hat{k}\boldsymbol{x}\right)].\end{aligned} \tag{6}$$

Note the same commutative property can also be obtained in different ways, e.g., by extending (2) to $\mathbb{H}$, that is,

[1] The operators $(\cdot\,, \times)$ can similarly be defined for $\mathbb{H}$, see, e.g., [10], [20].

[2] It is important to highlight that there is no well-defined Riemannian metric for the group Euclidean transformations nor for the (pure) dual quaternions, but the double-geodesic approach used in [23] and herein ensures positiveness and equal actions in the attitude and translation geodesics. The study on metrics for Euclidean displacements and the correspondent topological obstruction lies out of the scope of the current manuscript, but readers are refereed to excel works of [23], [24] for further details.

[3] Additional quaternion and dual quaternion operators can be found in the treatise on octonions by McAulay [25].

$\boldsymbol{a}\boldsymbol{b} = \boldsymbol{b}\boldsymbol{a} - 2\,\mathrm{Im}\,(\boldsymbol{b}) \times \mathrm{Im}\,(\boldsymbol{a})$. Moreover, the even the matrix-vector multiplication can be simplified by the following linear operation over $\mathbb{H}$,

$$\begin{aligned} \boldsymbol{y} = \underline{\mathrm{vec}}(\overset{+}{\boldsymbol{H}}(\boldsymbol{a})\,\mathrm{vec}\,\boldsymbol{b}) &= \boldsymbol{a}b_0 + \boldsymbol{a}\hat{\imath}b_1 + \boldsymbol{a}\hat{\jmath}b_2 + \boldsymbol{a}\hat{k}b_3, \\ = \underline{\mathrm{vec}}(\overset{-}{\boldsymbol{H}}(\boldsymbol{b})\,\mathrm{vec}\,\boldsymbol{a}) &= \boldsymbol{b}a_0 + \hat{\imath}\boldsymbol{b}a_1 + \hat{\jmath}\boldsymbol{b}a_2 + \hat{k}\boldsymbol{b}a_3, \end{aligned} \tag{7}$$

where $\boldsymbol{a} = a_1 + a_2\hat{\imath} + a_3\hat{\jmath} + a_4\hat{k} \in \mathbb{H}$ and $\boldsymbol{b} = b_1 + b_2\hat{\imath} + b_3\hat{\jmath} + b_4\hat{k} \in \mathbb{H}$. The resulting operation has the advantage of being defined only over quaternion multiplication by scalars whilst avoiding the tedious and costly operation of mapping quaternions to $\mathbb{R}^4$ and $\mathbb{R}^{4\times4}$, and back to $\mathbb{H}$.

For dual quaternions, the same matrix algebra operator can be obtained based on a $\mathbb{R}^{8\times8}$ multiplication by the vector representation of a dual quaternion element—and, then mapped back to $\mathbb{H}\otimes\mathbb{D}$—as in [10], [20]. In contrast, [28] introduces the concept of orthogonal dual matrix combining the Hamilton operator matrix for quaternions with dual unit ε with $\varepsilon^2 = 0$, $\varepsilon \neq 0$. Herein, to commute between dual quaternions we take a similar approach based on the direct product of dual numbers with quaternion Hamilton operators (5)-(7), that is, the multiplication between dual quaternions $\underline{\boldsymbol{y}} = \underline{\boldsymbol{a}\boldsymbol{b}}$ can be rewritten as

$$\begin{aligned} \underline{\boldsymbol{y}} &= \underline{\overset{+}{\boldsymbol{H}}}(\underline{\boldsymbol{a}})\,\underline{\boldsymbol{b}} = \overset{+}{\boldsymbol{H}}(\boldsymbol{a})\,\boldsymbol{b} + \varepsilon(\overset{+}{\boldsymbol{H}}(\boldsymbol{a})\,\boldsymbol{b}' + \overset{+}{\boldsymbol{H}}(\boldsymbol{a}')\,\boldsymbol{b}), \\ &= \underline{\overset{-}{\boldsymbol{H}}}(\underline{\boldsymbol{b}})\,\underline{\boldsymbol{a}} = \overset{-}{\boldsymbol{H}}(\boldsymbol{b})\,\boldsymbol{a} + \varepsilon(\overset{-}{\boldsymbol{H}}(\boldsymbol{b})\,\boldsymbol{a}' + \overset{-}{\boldsymbol{H}}(\boldsymbol{b}')\,\boldsymbol{a}), \end{aligned} \tag{8}$$

where $\underline{\boldsymbol{a}} = \boldsymbol{a} + \varepsilon\boldsymbol{a}'$ and $\underline{\boldsymbol{b}} = \boldsymbol{b} + \varepsilon\boldsymbol{b}'$, and $\underline{\overset{+}{\boldsymbol{H}}}$ and $\underline{\overset{-}{\boldsymbol{H}}}$ yields a orthogonal dual matrix as described in [28].

D. *Operators computational complexity*

In addition to describe the group and algebraic properties and operators within quaternion and dual quaternion algebra, we are particularly interested in evaluating algorithm performance and computational costs associated with the rigid body kinematics and dynamics that follow. Such an analysis is not a novelty and has been done in excel works of [6], [9], [29], [30]. Particularly, [6] and [30] advocates over the performance of the UDQ $(8f, 48\times, 40+)$ over the HTM representation $(12f, 64\times, 48+)$, where $f, \times, +$ stands for the storage, scalar addition and multiplications costs related to a single group operation—which in turn relates to a single rigid-body transformation [6]. Herein, we adopt a similar quantitative cost analysis approach combining the cost analysis from [6], [9], [29], [30] with additional operations paramount for the kinematic and dynamics analysis. Table I lists the storage and computational costs related to quaternion algebra, whilst Table II shows a detailed list of dual quaternion operations with respective storage and costs. Note that, although we reckon the same computational complexity could be directly extracted from basic operations, to explicitly state the compound operations is valid for future reference and computational cost analysis within the dual quaternion algebra.

TABLE I: Cost requirem. of quaternion algebra operations

Operation	Storage	$+/-$	$\times/\div$
Addition $(\boldsymbol{x}_1 + \boldsymbol{x}_2)$	$8f$	4+	0×
Multiplication by Scalar $(\alpha\boldsymbol{x})$	$5f$	0+	4×
Quat. Multiplication $\boldsymbol{x}_1\boldsymbol{x}_2$	$8f$	12+	16×
Hamilton Operator $\overset{+}{\boldsymbol{H}}(\cdot)\,(\overset{-}{\boldsymbol{H}})$ (7)	$8f$	12+	16×

TABLE II: Cost requirements of dual quaternion operations

Operation	Storage	$+/-$	$\times/\div$
Addition $(\underline{\boldsymbol{x}}_1 + \underline{\boldsymbol{x}}_2)$	$16f$	8+	0×
Multiplication by Scalar $(\alpha\underline{\boldsymbol{x}}_1)$	$9f$	0+	8×
Dual Quat. Multiplication $(\underline{\boldsymbol{x}}_1\underline{\boldsymbol{x}}_2)$	$16f$	40+	48×
Cross Product (3)	$12f$	12+	18×
Inner Product (4)	$12f$	5+	6×
Dual Quat. Hamilton Operator (8)	$28f$	48+	48×
Adjoint Transformation (16)	$14f$	70+	84×
Adjoint Matrix Representation (18)	$16f$	54+	72×
Adjoint Transf. w/ Dual Matrix (17)	$26f$	21+	27×

* where $\alpha \in \mathbb{R}$ and $\underline{\boldsymbol{x}}_1, \underline{\boldsymbol{x}}_2 \in \mathbb{H}\otimes\mathbb{D}$

III. Unconstrained rigid body motion: kinematics and dynamics

The advantages of using dual quaternion algebra for rigid body pose representation and kinematics kinematic motion description are well justified in the literature [6], [10], [29], [30]. In this section, we will exploit these advantages to address both modeling, kinematics and dynamics related to the motion of the unconstrained rigid body. The results herein will then built the basis for the analysis and model design over serial manipulators that follows.

In the scope of operations with dual quaternions used for three dimensional motions, we aim to present a geometric representation of the space. This geometric notion is particularly interesting when dealing with the representations of screws, as a screw axis is nothing but a tridimensional line in space, and therefore can be represented with a Plucker line through a set of three parameters: a unit direction vector ($\boldsymbol{l}$), a point that passes through the line ($\boldsymbol{p}$) and the line's moment $\boldsymbol{m} = \boldsymbol{p} \times \boldsymbol{l}$. This set of parameters are denoted Plucker coordinates, and although this notation is many times used in other algebras (as a pair of free vectors) it fits particularly with the dual quaternion notation. Indeed, a Plucker line can be defined by the dual quaternion, $\underline{\boldsymbol{l}} = \boldsymbol{l} + \varepsilon\boldsymbol{m}$, with $\boldsymbol{m} = \boldsymbol{p} \times \boldsymbol{l}$ being the moment around the origin and $\boldsymbol{m}$,$\boldsymbol{p}$,$\boldsymbol{l} \in \mathbb{H}_0$ being pure quaternions [6], [10].

More so, when attached to a fixed point the line may rotate and perform a motion with an arc-like trajectory. Much like with imaginary numbers, the dual quaternions exponential is suited to represent a curve in space. Thus, as shown in [31], for the dual quaternion $\underline{\boldsymbol{q}}$, the exponential can be defined as

$$exp(\underline{\boldsymbol{q}}) = \mathcal{P}\left(exp(\underline{\boldsymbol{q}})\right) + \varepsilon\mathcal{D}\left(\underline{\boldsymbol{q}}\right)\mathcal{P}\left(exp(\underline{\boldsymbol{q}})\right), \tag{9}$$

with $\mathcal{P}\left(exp(\underline{\boldsymbol{q}})\right)$ being defined as

$$\begin{cases} cos\left\|\mathcal{P}\left(\underline{\boldsymbol{q}}\right)\right\| + \frac{sin\left\|\mathcal{P}(\underline{\boldsymbol{q}})\right\|}{\left\|\mathcal{P}(\underline{\boldsymbol{q}})\right\|}\mathcal{P}\left(\underline{\boldsymbol{q}}\right) & \text{if } \left\|\mathcal{P}\left(\underline{\boldsymbol{q}}\right)\right\| \neq 0 \\ 1 & \text{o.w.} \end{cases} \tag{10}$$

We highlight here the manner in which the exponential and Plucker line relate to representing a screw axis displacement. That is, given the Plucker line $\underline{\boldsymbol{s}} = \boldsymbol{l} + \varepsilon\boldsymbol{m}$ and the dual angle $\hat{\theta} = \theta + \varepsilon d$, with $\theta, d \in \mathbb{R}$, we can represent the screw axis' position as the UDQ $\underline{\boldsymbol{x}} \in \text{Spin}(3) \ltimes \mathbb{R}^3$, that is,

$$\underline{\boldsymbol{x}} = exp\left(\frac{\hat{\theta}}{2}\underline{\boldsymbol{s}}\right) = cos\left(\frac{\hat{\theta}}{2}\right) + \underline{\boldsymbol{s}}sin\left(\frac{\hat{\theta}}{2}\right). \tag{11}$$

The above expression is a generalization which exploits the dual propriety to couple the rotation of the screw (θ) and its translation (d). However, for brevity and to simplify the proposed method description we will only focus on revolute joints manipulators—which in turn yields $d = 0$ and $\hat{\theta} = \theta$ to the remainder of this text. Note, nonetheless, that extension to prismatic joints should be trivial as stressed in [6].

Furthermore, the UDQ described in (11) can also be rewritten in terms of a quaternion pair as follows

$$\underline{\boldsymbol{x}} = \boldsymbol{r} + \varepsilon(1/2)\boldsymbol{r}\boldsymbol{p}^b. \tag{12}$$

In this case, $\boldsymbol{r} \in \text{Spin}(3)$ is a unit quaternion representation of the rigid-body attitude with respect to a inertia frame, whilst $\boldsymbol{p}^b = p_x\hat{\imath} + p_y\hat{\jmath} + p_z\hat{k}$, is the rigid-body position expressed in the rigid-body frame. The expression (12) is also useful to express a rigid displacement through a rotation $\boldsymbol{r}$ followed by a translation $\boldsymbol{p}^b$.

We can then derive the kinematic equation to obtain the velocity, $\underline{\dot{\boldsymbol{x}}}$, as is done in [10][4]

$$\underline{\dot{\boldsymbol{x}}} = \tfrac{1}{2}\underline{\boldsymbol{x}}\underline{\boldsymbol{\omega}}^b, \tag{13}$$

with $\underline{\boldsymbol{\omega}}^b$ being the generalized twist in body frame with $\underline{\boldsymbol{\omega}}^b = \boldsymbol{\omega}^b + \varepsilon\boldsymbol{v}^b$, where $\boldsymbol{\omega}^b \in \mathbb{H}_0$ is the angular velocity and $\boldsymbol{v}^b \in \mathbb{H}_0$ is the linear velocity .

Although not much has been published on the formulation of the rigid body dynamics with dual quaternion, there are some interesting results in the literature. In the formulation first proposed by Dooley in [33], UDQs are directly applied to solve the general dynamics problem of an unconstrained rigid body (e.g., a holonomic flying robot). However, as argued in [32] the equations of motion are overly complicated, lack an intuitive meaning and are hard to implement. In this context, the proposition in [32] then describes a single rigid body dynamics from wrench/twist formulations, but instead of taking full advantage of the UDQ representation, the operations are made element-wise rather than an operation of a dual object. A more thorough solution is presented in [17], and in this work, we take advantage of such a formulation. To describe the system dynamics, the authors explicitly describe the angular and linear momentums, as well as the resulting kinematic wrenches, stemming from the body motion by introducing an augmented linear matrix representation of the body mass and inertia tensor, the dual inertia matrix (a real 8×8 symmetric matrix). In other words, in [17], the body mass $m \in \mathbb{R}$ is described as a diagonal matrix within a block diagonal matrix structure together the inertia tensor of the body about its center of mass, that is, $\boldsymbol{I}^b \in \mathbb{R}^{3\times3}$, which in turn relied on a manifold mapping from $\mathbb{H}_0 \otimes \mathbb{D}$ to $\mathbb{R}^8$ and its inverse mapping. The result proposed in [17] also required a complicated switch operator to properly describe the resulting angular and linear momentums (and wrenches) in the dual quaternion primary and dual parts.

In contrast, herein, we take advantage of the pure dual quaternion representation for velocities and accelerations to derive a more direct description of angular and linear momentums and wrenches. Since the linear part of the generalized twist and body acceleration are described by a pure quaternion, as seen in (13), the resulting momentum and force is obtained by a simple scalar multiplication. Whilst the angular momentum and torque can be extracted directly from the orthonormal vectors of the inertia tensor about its center of mass w.r.t. the body axis multiplied by the angular velocities and accelerations. The resulting dual quaternion momentum and wrenches stemming, respectively, from body velocities and accelerations are therefore given by a *Dual Quaternion Inertia Transformation*, $G : \mathbb{H}_0 \otimes \mathbb{D} \to \mathbb{H}_0 \otimes \mathbb{D}$, as

$$\begin{aligned} G(\underline{\boldsymbol{\omega}}_b) &\triangleq m\mathcal{D}(\underline{\boldsymbol{\omega}}_b) + \varepsilon\left(\boldsymbol{I}_x^b\omega_x + \boldsymbol{I}_y^b\omega_y + \boldsymbol{I}_z^b\omega_z\right), \\ G(\underline{\dot{\boldsymbol{\omega}}}_b) &\triangleq m\mathcal{D}(\underline{\dot{\boldsymbol{\omega}}}_b) + \varepsilon\left(\boldsymbol{I}_x^b\dot{\omega}_x + \boldsymbol{I}_y^b\dot{\omega}_y + \boldsymbol{I}_z^b\dot{\omega}_z\right) \end{aligned} \tag{14}$$

where $\underline{\boldsymbol{\omega}}_b$ and $\underline{\dot{\boldsymbol{\omega}}}_b$ are the body generalized twist and accelerations, w.r.t. center of mass, with angular component being given by $\mathcal{P}(\underline{\boldsymbol{\omega}}_b) = \omega_x\hat{\imath} + \omega_y\hat{\jmath} + \omega_z\hat{k}$ and $\mathcal{P}(\underline{\dot{\boldsymbol{\omega}}}_b) = \dot{\omega}_x\hat{\imath} + \dot{\omega}_y\hat{\jmath} + \dot{\omega}_z\hat{k}$. And, the inertia tensor of the rigid body characterized by $\boldsymbol{I}_x^b = I_{xx}\hat{\imath} + I_{xy}\hat{\jmath} + I_{xz}\hat{k}$, $\boldsymbol{I}_y^b = I_{xy}\hat{\imath} + I_{yy}\hat{\jmath} + I_{yz}\hat{k}$, and $\boldsymbol{I}_z^b = I_{xz}\hat{\imath} + I_{yz}\hat{\jmath} + I_{yz}\hat{k}$,w.r.t to the coordinate axis of the center of mass—note they can be viewed as pure quaternion representations of column vectors of the tensor matrix described in [17]. Finally, from (14), we can compute the rigid body dynamics as

$$G(\underline{\dot{\boldsymbol{\omega}}}_b) + \underline{\boldsymbol{\omega}}_b \times G(\underline{\boldsymbol{\omega}}_b) - \underline{\boldsymbol{f}}_b = 0, \tag{15}$$

in which $\underline{\boldsymbol{f}}_b = \boldsymbol{f_b} + \boldsymbol{\varepsilon m_b}$ is the body wrench ($\boldsymbol{f_b}$ being the forces and $\boldsymbol{m}_b$ the moments applied about the body's center of mass).

Finally, it is also useful to define the operation for changing a frame of reference useful to describe the congruence transformation that takes velocities, accelerations and wrenches described in pure dual quaternions from one frame to the other, that is an adjoint mapping [10]. Given the dual quaternion frame transformation $\underline{\boldsymbol{x}} = \boldsymbol{x} + \varepsilon\boldsymbol{x}' \in \text{Spin}(3) \ltimes \mathbb{R}^3$ from $\mathcal{F}_0$ to $\mathcal{F}_1$, the adjoint operation that takes the pure dual quaternion $\underline{\boldsymbol{\mu}} = \boldsymbol{\mu} + \varepsilon\boldsymbol{\mu}' \in \mathbb{H}_0 \otimes \mathbb{D}$ from frame $\mathcal{F}_0$ to be

[4]For further information on dual quaternion based kinematics please refer to [10], [20], whilst the readers are refereed to [17], [32] for the dynamics based on dual quaternion—limited to unconstrained rigid bodies.

expressed in $\mathcal{F}_1$ is

$$\underline{\boldsymbol{\mu}}^1 = Ad(\underline{\boldsymbol{x}})\underline{\boldsymbol{\mu}} = \underline{\boldsymbol{x}}\underline{\boldsymbol{\mu}}\underline{\boldsymbol{x}}^*, \tag{16}$$

which can also be expressed in terms of Hamilton operators as described by the linear operations (5)-(8),

$$\begin{aligned}
Ad(\underline{\boldsymbol{x}})\underline{\boldsymbol{\mu}} &= \overset{+}{\underline{\boldsymbol{H}}}(\underline{\boldsymbol{x}})\,\overset{-}{\underline{\boldsymbol{H}}}(\underline{\boldsymbol{x}}^*)\,\underline{\boldsymbol{\mu}} \\
&= \overset{+}{\underline{\boldsymbol{H}}}(\underline{\boldsymbol{x}})\left(\overset{-}{\boldsymbol{H}}(\boldsymbol{x}^*)\,\boldsymbol{\mu}+\varepsilon\left(\overset{-}{\boldsymbol{H}}(\boldsymbol{x}^*)\,\boldsymbol{\mu}'+\overset{-}{\boldsymbol{H}}(\boldsymbol{x}'^*)\,\boldsymbol{\mu}\right)\right) \\
&= \overset{+}{\boldsymbol{H}}(\boldsymbol{x})\,\overset{-}{\boldsymbol{H}}(\boldsymbol{x}^*)\,\boldsymbol{\mu}+\varepsilon\left(\overset{+}{\boldsymbol{H}}(\boldsymbol{x})\,\overset{-}{\boldsymbol{H}}(\boldsymbol{x}^*)\,\boldsymbol{\mu}'\right. \\
&\quad \left.+\overset{+}{\boldsymbol{H}}(\boldsymbol{x})\,\overset{-}{\boldsymbol{H}}(\boldsymbol{x}'^*)\,\boldsymbol{\mu}+\overset{+}{\boldsymbol{H}}(\boldsymbol{x}')\,\overset{-}{\boldsymbol{H}}(\boldsymbol{x}^*)\,\boldsymbol{\mu}\right),
\end{aligned}$$

which can in turn be simplified to

$$Ad(\underline{\boldsymbol{x}})\underline{\boldsymbol{\mu}} = \mathcal{A}_P(\underline{\boldsymbol{x}})\boldsymbol{\mu} + \varepsilon\left(\mathcal{A}_P(\underline{\boldsymbol{x}})\boldsymbol{\mu}' + \mathcal{A}_D(\underline{\boldsymbol{x}})\boldsymbol{\mu}\right), \tag{17}$$

where (17) is a matrix linear transformation to be solved as described in (5)-(8) with $\mathcal{A}_P(\underline{\boldsymbol{x}})$ and $\mathcal{A}_D(\underline{\boldsymbol{x}})$ given by

$$\mathcal{A}_P(\underline{\boldsymbol{x}}) = \overset{+}{\boldsymbol{H}}(\boldsymbol{x})\overset{-}{\boldsymbol{H}}(\boldsymbol{x}^*),\;\; \mathcal{A}_D(\underline{\boldsymbol{x}}) = 2\overset{+}{\boldsymbol{H}}(\boldsymbol{x})\overset{-}{\boldsymbol{H}}(\boldsymbol{x}'^*). \tag{18}$$

IV. Inverse dynamics with unit dual quaternions

In this section, we propose a novel dual-quaternion based formalism for the recursive Newton-Euler inverse dynamics algorithm. The dqRNEA combines the advantages of screw theory formalized over the dual quaternion algebra, with an unified framework representation, to design the manipulator dynamic model based on the balance of forces acting on the links. As can be seen summarized in Algorithm 1, given known joint positions, velocities and accelerations, we compute and propagate, using compact dual quaternion algebra, the pose, velocities and accelerations of the links based on Newton's law of motion with the same algebraic structure without the need of D-H parameters computations—similarly to [6]. Then, based on Newton–Euler equations we compute the wrenches (forces and moments) acting on each link in a recursive fashion, starting from the wrenches applied to the end-effector. Both forward and backward iteration are described herein with the same dual quaternion framework exploiting the manifold capabilities to represent rigid body pose, velocities, accelerations, momentum, and wrenches. Furthermore, we also present the cost analysis in terms of storage, sums and multiplications at each step of the dual-quaternion based algorithm comparing the method with similar HTM formulation.

A. Forward Pose, velocity and acceleration kinematics

The forward iteration of the algorithm, much like the well established forward kinematics methods for open chains, aims at obtaining the pose and velocity for each of the robot's links. As an important addition for this case, we also calculate the twist's first derivative: the acceleration that results in the forces our system is subject to. More so, we propagate these calculations recursively, from the base of the kinematic chain throughout each link up to the end-effector by using a description based on screw theory and line transformations.

Algorithm 1 Dual-quaternion based recursive Newton-Euler Inverse Dynamics for n DoF Manipulator

Initialization:
At the home config., set the frames $\mathcal{F}_0$ to the base, $\mathcal{F}_1$ to $\mathcal{F}_n$ to the n-links' center of mass, and $\mathcal{F}_{n+1}$ to the end-pose. Set $\underline{\boldsymbol{\delta}}_{i-1}^{i}$, $\underline{\boldsymbol{a}}_i$ as the pose of $\mathcal{F}_{i-1}$ and the screw axis of joint i expressed in $\mathcal{F}_i$, with null vel. $\underline{\boldsymbol{\omega}}_0$=0 and gravity accel. $\dot{\underline{\boldsymbol{\omega}}}_0$=$-\varepsilon\boldsymbol{g}$ at base and end-pose wrench $\underline{\boldsymbol{f}}_{EF}$=$\boldsymbol{f}_{EF}$+$\varepsilon\boldsymbol{m}_{EF}$.

Forward iteration

- $\underline{\boldsymbol{x}}_i^{i-1}(\theta_i(t)) = \underline{\boldsymbol{\delta}}_{i-1}^{i} exp\left(\underline{\boldsymbol{a}}_i\frac{\Delta\theta_i}{2}\right)$
- Compute $\mathcal{A}_P(\underline{\boldsymbol{x}}_i^{i-1})$ and $\mathcal{A}_D(\underline{\boldsymbol{x}}_i^{i-1})$ from (18)
- $\underline{\boldsymbol{\omega}}_i = \underline{\boldsymbol{a}}_i\dot{\theta}_i(t) + Ad(\underline{\boldsymbol{x}}_i^{i-1})\underline{\boldsymbol{\omega}}_{i-1}$
- $\dot{\underline{\boldsymbol{\omega}}}_i = \underline{\boldsymbol{a}}_i\ddot{\theta}_i(t) + Ad(\underline{\boldsymbol{x}}_i^{i-1})\dot{\underline{\boldsymbol{\omega}}}_{i-1} + \left(\underline{\boldsymbol{\omega}}_i \times \underline{\boldsymbol{a}}_i\right)\dot{\theta}_i(t)$

Backward iteration

- $\underline{\boldsymbol{f}}_{R,i} = \underline{\boldsymbol{\omega}}_i \times \left(G_i\left(\underline{\boldsymbol{\omega}}_i\right)\right) + G_i\left(\dot{\underline{\boldsymbol{\omega}}}_i\right)$
- $\underline{\boldsymbol{f}}_i = \underline{\boldsymbol{f}}_{R,i} + Ad(\underline{\boldsymbol{x}}_i^{i+1})\underline{\boldsymbol{f}}_{i+1}$
- $\tau_i = \underline{\boldsymbol{f}}_i \odot \underline{\boldsymbol{a}}_i$

It is notable that Algorithm 1 exploits the screw theory formulation from Section III for calculating the forward kinematics, rather than the more traditional alternative, the Denavit Hartemberg (D-H) notation. In [6], [10], [34], we see different recursive techniques being presented to solve the forward kinematics model (FKM) of a serial manipulator based on dual quaternions. In particular, [6] makes a compelling case for the screw theory alternative. As it is argued, this approach based on line transformations [15] is considerably more efficient then the other techniques such as the computation of the FKM based on D-H approach. More so, the screw representation is much more intuitive and the position of the frames can be chosen without the many restrictions that D-H parameters impose, thus each link's frame of reference can be at the center of mass of the body, which simplifies the dynamical equations.

From dual-quaternion algebra, we can easily represent the end-effector pose of an n-joint serial manipulator by means of successive rigid transformations between its links—note that a sequence of rigid motions can be represented by a sequence of UDQ multiplications [10], [20]. Hence, taking an initial joint configuration, we can calculate the end-effector's pose by means of the relative displacements of each links from such initial configuration. To simplify the description, let us define $\boldsymbol{\theta}^h \in \mathbb{R}^n$ as initial joint configuration, or home configuration, where $\boldsymbol{\theta} = [\theta_1 \;\; \theta_2 \; ... \; \theta_i \; ... \; \theta_n]^T \in \mathbb{R}^n$ is the vector of joint positions. At the home configuration, let $\underline{\boldsymbol{\delta}}_{i-1}^{i}$ be the dual quaternion representing the configuration of $\mathcal{F}_{i-1}$ expressed in the frame $\mathcal{F}_i$, where $\mathcal{F}_0$ is the base frame coinciding with the robot's base, frames $\mathcal{F}_1$ to $\mathcal{F}_n$ should be at the center of mass of each link and $\mathcal{F}_{n+1}$ is the reference frame attached to the end-effector. We then represent the transformation from one frame to another as $\underline{\boldsymbol{\delta}}_i^{i-1}$=$\left(\underline{\boldsymbol{\delta}}_{i-1}^{0}\right)^*\underline{\boldsymbol{\delta}}_i^{0}$ and $\left(\underline{\boldsymbol{\delta}}_i^{i-1}\right)^*$=$\underline{\boldsymbol{\delta}}_{i-1}^{i}$=$\left(\underline{\boldsymbol{\delta}}_i^{0}\right)^*\underline{\boldsymbol{\delta}}_{i-1}^{0}$.

Hence, the rigid transformation from robot's base to pose of the arm's end-effector is given by

$$\underline{\boldsymbol{\delta}}_{EF} = \underline{\boldsymbol{\delta}}_1^0 \underline{\boldsymbol{\delta}}_2^1 ... \underline{\boldsymbol{\delta}}_{n+1}^n. \tag{19}$$

We then define the screw axis of the link at home position as the Plucker line $\underline{\boldsymbol{s}}_i = \boldsymbol{\omega} + \varepsilon \boldsymbol{v}$, where $\boldsymbol{\omega}$ and $\boldsymbol{v}$ are the pure quaternions of the angular and linear velocity, respectively and, given a easily defined point $\boldsymbol{p}$ in the screw and we have $\boldsymbol{v} = \boldsymbol{p} \times \boldsymbol{\omega}$. However this representation is in frame $\mathcal{F}_0$, so we also need transform it to the link's frame of reference. That is, for link i, we would apply a transformation $\underline{\boldsymbol{a}}_i = Ad\left(\underline{\boldsymbol{\delta}}_0^i\right) \underline{\boldsymbol{s}}_i$ from frame $\mathcal{F}_0$ to frame $\mathcal{F}_i$. Note the importance of the multiplication order, which defines a sequence of UDQ multiplications from base to end-pose.

Now, from the current joint position $\theta_i(t)$, the pose transformation of the link i expressed in frame $\mathcal{F}_{i-1}$ is given by

$$\underline{\boldsymbol{x}}_i^{i-1}(\theta_i(t)) = \underline{\boldsymbol{\delta}}_{i-1}^i exp\left(\underline{\boldsymbol{a}}_i \frac{\Delta\theta_i}{2}\right), \tag{20}$$

with $\Delta\theta_i = \theta_i(t) - \theta^h$ being the position deviation from the home configuration. Then, the end-pose can be calculated by the sequence of link transformations stemming from the joint displacements, that is,

$$\underline{\boldsymbol{x}}_{EF} = \underline{\boldsymbol{x}}_1^0 \underline{\boldsymbol{x}}_2^1 ... \underline{\boldsymbol{x}}_{n+1}^n. \tag{21}$$

Considering the twist of a body defined as in (13), and that for serial chains the twist of a particular link is the sum of the twist at previous links (propagated from base to end-effector) added to the twist generated by the velocities of that link's joint, we can write the twist equation of joint i as

$$\underline{\boldsymbol{\omega}}_i = \underline{\boldsymbol{a}}_i \Delta\dot{\theta}_i(t) + Ad(\underline{\boldsymbol{x}}_i^{i-1}) \underline{\boldsymbol{\omega}}_{i-1}, \tag{22}$$

where the adjoint transformation is required to properly propagate velocities to the new frame of reference. Note similar strategy to this point has been proposed in [6] showing considerable computational complexity costs improvements over the use of D-H parameters. Herein, we extend such result taking individual links and end-effector pose acceleration into account. Taking the derivative of (22), we have

$$\begin{aligned}\dot{\underline{\boldsymbol{\omega}}}_i =& \underline{\boldsymbol{a}}_i \Delta\ddot{\theta}_i(t) + \frac{d}{dt}\left(\underline{\boldsymbol{x}}_i^{i-1} \underline{\boldsymbol{\omega}}_{i-1} \underline{\boldsymbol{x}}_{i-1}^i\right) \\ =& \underline{\boldsymbol{a}}_i \Delta\ddot{\theta}_i(t) + \underline{\boldsymbol{x}}_i^{i-1} \dot{\underline{\boldsymbol{\omega}}}_{i-1} \underline{\boldsymbol{x}}_{i-1}^i \\ &+ \dot{\underline{\boldsymbol{x}}}_i^{i-1} \underline{\boldsymbol{\omega}}_{i-1} \underline{\boldsymbol{x}}_{i-1}^i + \underline{\boldsymbol{x}}_i^{i-1} \underline{\boldsymbol{\omega}}_{i-1} \dot{\underline{\boldsymbol{x}}}_{i-1}^i\end{aligned} \tag{23}$$

whilst the the derivative of (20) give us

$$\dot{\underline{\boldsymbol{x}}}_{i-1}^i(\theta_i(t)) = \frac{1}{2} \underline{\boldsymbol{x}}_{i-1}^i \underline{\boldsymbol{a}}_i \Delta\dot{\theta}_i(t). \tag{24}$$

Now, combining 23 and (24), we have

$$\begin{aligned}\dot{\underline{\boldsymbol{\omega}}}_i =& \underline{\boldsymbol{a}}_i \Delta\ddot{\theta}_i(t) + Ad(\underline{\boldsymbol{x}}_i^{i-1}) \dot{\underline{\boldsymbol{\omega}}}_{i-1} + \\ & \left(\underline{\boldsymbol{x}}_i^{i-1} \underline{\boldsymbol{\omega}}_{i-1} \underline{\boldsymbol{x}}_{i-1}^i \underline{\boldsymbol{a}}_i - \underline{\boldsymbol{a}}_i \underline{\boldsymbol{x}}_i^{i-1} \underline{\boldsymbol{\omega}}_{i-1} \underline{\boldsymbol{x}}_{i-1}^i\right) \frac{\Delta\dot{\theta}_i(t)}{2},\end{aligned} \tag{25}$$

which can further be simplified by taking the cross product definition (2) as

$$\dot{\underline{\boldsymbol{\omega}}}_i = \underline{\boldsymbol{a}}_i \Delta\ddot{\theta}_i(t) + Ad(\underline{\boldsymbol{x}}_i^{i-1}) \dot{\underline{\boldsymbol{\omega}}}_{i-1} + (\underline{\boldsymbol{x}}_i^{i-1} \underline{\boldsymbol{\omega}}_{i-1} \underline{\boldsymbol{x}}_{i-1}^i \times \underline{\boldsymbol{a}}_i) \Delta\dot{\theta}_i(t)$$

From (22), we obtain a final expression for the recursive forward model for the accelerations derived solely from screw theory based on dual quaternion algebra,

$$\dot{\underline{\boldsymbol{\omega}}}_i = \underline{\boldsymbol{a}}_i \Delta\ddot{\theta}_i(t) + Ad(\underline{\boldsymbol{x}}_i^{i-1}) \dot{\underline{\boldsymbol{\omega}}}_{i-1} + (\underline{\boldsymbol{\omega}}_i \times \underline{\boldsymbol{a}}_i) \Delta\dot{\theta}_i(t). \tag{26}$$

B. Backward Iteration

In the backwards iteration, given provided dual quaternion wrenches acting at the end-effector and the resulting velocities and accelerations of the link's center of mass from the forward kinematics, we can compute the required torques to be applied at the joints to obtain the prescribed motion. Given the end-pose wrenches, we start our analysis from the end-effector down to the base of the robot.

From individual rigid body dynamics stemming from each link's velocities and accelerations (15), the resulting dual quaternion wrench acting on the link i can be computed as

$$\underline{\boldsymbol{f}}_{R,i} = \underline{\boldsymbol{\omega}}_i \times \left(G_i\left(\underline{\boldsymbol{\omega}}_i\right)\right) + G_i(\dot{\underline{\boldsymbol{\omega}}}_i), \tag{27}$$

where $G_i(*)$ is the *Dual Quaternion Inertia Transformation* as defined in (14) for the *ith* link. Now, since the resulting forces in a rigid-body must be equal to the sum of the forces being applied, and given the serial kinematic chain propagation between rigid links (also assuming the only forces being applied on the link stems from attached joints), the resulting wrench $\underline{\boldsymbol{f}}_{R,i}$ must also be given by the addition of the wrenches at the two adjacent joints (all expressed in the current link's frame), that is,

$$\underline{\boldsymbol{f}}_{R,i} = \underline{\boldsymbol{f}}_i + Ad\left(\underline{\boldsymbol{x}}_i^{i+1}\right) \underline{\boldsymbol{f}}_{i+1}. \tag{28}$$

By combining (27) and (28), we obtain the full dynamic equation of the required dual quaternion wrench at link i,that is,

$$\underline{\boldsymbol{f}}_i = \underline{\boldsymbol{\omega}}_i \times \left(G_i\left(\underline{\boldsymbol{\omega}}_i\right)\right) + G_i(\dot{\underline{\boldsymbol{\omega}}}_i) - Ad\left(\underline{\boldsymbol{x}}_i^{i+1}\right) \underline{\boldsymbol{f}}_{i+1}, \tag{29}$$

which, thereafter, can be used to find the torque at the joint. Hence, the torque in the direction of the axis,

$$\tau_i = \underline{\boldsymbol{f}}_i \odot \underline{\boldsymbol{a}}_i. \tag{30}$$

Both the dual quaternion forward iteration that provides us with the link's pose, velocities and accelerations given a prescribed joint space dynamics, as well as the recursive propagation of the wrenches in dual quaternion space required to compute the resulting joint torques are summarized in Algorithm 1.

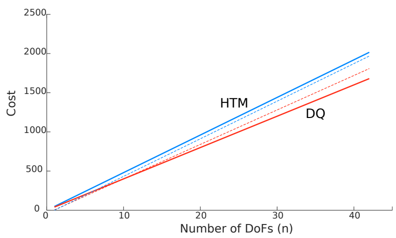
C. Computational Complexity Cost Analysis

From the data of cost operations within the unit dual quaternion framework, as presented in Tables I and II, we can compute the total cost for the forward and backward iteration of the proposed dual quaternion based Newton-Euler inverse dynamics algorithm for a n-joints serial manipulator. Such results are presented on Table III.

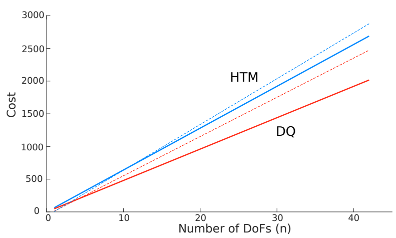
More so, it is interesting to make a comparison of the costs of the algorithm presented in this paper with similar ones using different representations.In particular, only taking into account only the forward kinematics transformation,

TABLE III: Cost of operations for UDQ

Oper.	DQ		
	Storage(f)	$+/-$	$\times/\div$
Eq. (21)	$8n$	$40n$	$48n$
Eq. (22)	$33n$	$27n$	$33n$
Eq. (26)	$46n$	$45n$	$57n$
Eq. (27)	$38n$	$30n$	$42n$
Eq. (28)	$32n$	$27n$	$27n$
Eq. (30)	$12n$	$5n$	$6n$
TOTAL	$169n$	$174n$	$213n$



(a) Costs for Addition



(b) Costs for Multiplication

Fig. 1: Costs for Forward Kinematics computation using different representations. In blue we have HTM and in red we have DQ representation. With the dotted line is the cost for D-H based parametrization and the straight line is the screw theory parametrization.

it is shown in [7], [6] and in Figure 1 that a DQ based implementation performs better than an HTM in terms of computational cost, and also, the use of the D-H parameters also make costs increase. Thus, regarding the modeling of the manipulator's position, we have used a Screw Theory based formulation. We reiterate here that there are many reasons for opting for the Screw formulation, the cost being only one of them. For the forward position kinematics transformation in (19) we can see that our method has the cost of $n(8f, 40+, 48\times)$ and the cost for the same transformation, but in D-H parameters, is at least $3n(8f, 40+, 48\times)$ [6].

Similarly, we can compare our dqRNEA algorithm with a formulation based on HTM. By taking into account the equations analyzed in Table III, the total cost of the dqRNEA is $n(169f, 174+, 213\times)$. The total cost for a HTM formulation, calculated with no optimization, is $n(157f, 347+, 321\times)$. Herein, we highlight that although the storage cost is slightly lower in the HTM formulation , the costs for multiplication and addition are significantly higher.

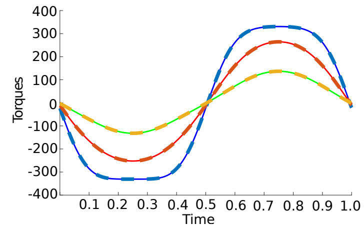


Fig. 2: Joint torques obtained by the dqRNEA with screw theory approach based on UDQ, Algorithm 1 (solid line), and classic RNEA based on HTM with D-H parameters kinematics description (dotted line).

V. Quantitative Analysis

In this section, we present a simple example to validate the proposed algorithm implementation yet, most importantly, we provide a quantitative analysis of the proposed algorithm given different conditions and compare the result with classic HTM based solution.

For the numerical simulation, we modeled a three-link robot arm and implemented Algorithm 1 using the DQ Robotics toolbox[5]. For simplicity, we assumed similar links with length of 0.5 m, weight of 1 kg and centre of mass given in the link's centroid. Figure 2 illustrates the resulting joint-torque required to execute the sinusoidal trajectory in 1s—described by the manipulator pose at given points shown at the top.

Now, to illustrate the quantitative aspects in terms of computational complexity of the proposed algorithm, a cost analysis for different conditions in terms of serial kinematic chain links has been performed as shown in Figure 3. The result was compared to a classic HTM algorithm using D-H parameters as in the Robotics Toolbox [35]. Also, to maintain fairness of comparison, we also implemented a RNEA algorithm with HTM but using similar screw theory approach [6] to avoid additional steps of the D-H parameter. From Figure 3, it is clear that the proposed dual quaternion formulation for the recursive Newton-Euler algorithm provides improved results in terms of computational complexity compared to classic HTM solution, as expected. The benefits are even more relevant when applied to a system with higher DOFs which encourages the use of the proposed algorithm for multiple robots as in [36], [37].

VI. Conclusion

The main focus of this paper was to provide a novel formalization for the RNEA with dual quaternion algebra and thus extend results found in the literature of robotic manipulators. In addition, we have also addressed the computational complexity analysis of our method and compared it to other models found in the literature. Furthermore, to achieve improved performance, we have opted for a less traditional method of describing a serial manipulator. In order to avoid the extra costs of the D-H parameters and in order to

[5]http://dqrobotics.sourceforge.net

[6]This algorithm can be found on chapter 8 of [1].

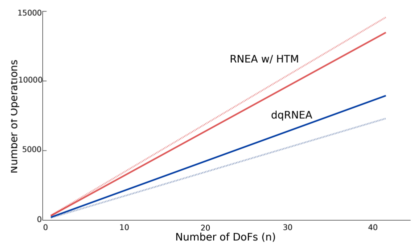


Fig. 3: Total costs for RNEA of both HTM and DQ. In blue we see the number of operation's growth of the DQ algorithm for multiplication (continuous line) and addition (dotted line). In red, we have the number of operation's growth of the HTM. DoFs range from 1-42.

design simpler dynamical equations, we have chosen to base our formulation on screw theory. The resulting algorithm, in addition to provide suitable results within the algebra of dual quaternions, has also shown improved performance over the classic HTM solution. For future works we want to extend our algorithm to cooperative manipulation taking advantage of the cooperative kinematic description within DQ algebra with the efficient computation of the manipulator dynamics based on screw theory with UDQ.

References

[1] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*, 1st ed. New York, NY: Cambridge Uni. Press, 2017.

[2] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. London: Springer Verlag, 2009.

[3] R. Featherstone, *Rigid Body Dynamics Algorithms*. Berlin, Heidelberg: Springer-Verlag, 2007.

[4] J. Carpentier and N. Mansard, "Analytical derivatives of rigid body dynamics algorithms," in *Robotics: Science and Systems (RSS 2018)*, Pittsburgh, United States, Jun. 2018.

[5] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, Jun. 1997.

[6] E. Özgür and Y. Mezouar, "Kinematic modeling and control of a robot arm using unit dual quaternions," *Robot. Auton. Syst.*, vol. 77, no. C, pp. 66–73, Mar. 2016.

[7] B. V. Adorno, "Two-arm manipulation: from manipulators to enhanced human-robot collaboration," PhD thesis, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) - Université Montpellier 2, Montpellier, France, 2011.

[8] X. Wang and H. Zhu, "On the comparisons of unit dual quaternion and homogeneous transformation matrix," *Advances in Applied Clifford Algebras*, vol. 24, pp. 213–229, 2014.

[9] N. A. Aspragathos and J. K. Dimitros, "A comparative study of three methods for robot kinematics," *IEEE Transactions on Syst., Man, and Cybernetics, Part B: Cybernetics*, vol. 28, no. 2, pp. 135–145, 1998.

[10] B. V. Adorno, "Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra — Part I: Fundamentals." Feb. 2017, preprint book. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01478225

[11] D. Han, Q. Wei, and Z. Li, "A dual-quaternion method for control of spatial rigid body," in *IEEE International Conference on Networking, Sensing and Control*, no. 5, Sanya, China, 2008.

[12] X. Wang, D. Han, C. Yu, and Z. Zheng, "The geometric structure of unit dual quaternion with application in kinematic control," *Journal of Mathematical Analysis and Applications*, vol. 389, no. 2, pp. 1352–1364, May 2012.

[13] B. V. Adorno, P. Fraisse, and S. Druon, "Dual position control strategies using the cooperative dual task-space framework," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Oct. 2010, pp. 3955–3960.

[14] L. F. C. Figueredo, B. V. Adorno, J. Y. Ishihara, and G. A. Borges, "Robust kinematic control of manipulator robots using dual quaternion representation," in *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013, pp. 1949–1955.

[15] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.

[16] N. Filipe and P. Tsiotras, "Adaptive model-independent tracking of rigid body position and attitude motion with mass and inertia matrix identification using dual quaternions," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, January 2013, pp. 1–15.

[17] ——, "Simultaneous position and attitude control without linear and angular velocity feedback using dual quaternions," in *2013 American Control Conf.*, Washington, DC, USA, Sept. 2013, pp. 4808–4813.

[18] M. D. Shuster, "A survey of attitude representations," *The Journal of the Astronautical Sciences*, vol. 41, no. 4, pp. 439–517, 1993.

[19] J. C. K. Chou, "Quaternion kinematic and dynamic differential equations," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 53–64, Feb 1992.

[20] L. F. C. Figueredo, "Kinematic control based on dual quaternion algebra and its application to robot manipulators," Ph.D. dissertation, University of Brasilia, Brazil, 2016.

[21] J. M. Selig, *Geometric Fundamentals of Robotics*, 2nd ed. New York: Springer Publishing Company Inc., 2005.

[22] J. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton, NJ: Princeton University Press, 1999.

[23] F. Bullo and R. Murray, "Proportional derivative (PD) control on the Euclidean group," Division of Eng. and Applied Science, California Institute of Technology, Technical Report Caltech/CDS 95–010, 1995.

[24] F. C. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," *ASME. Journal of Mechanical Design*, vol. 117, no. 1, pp. 48–54, 1995.

[25] A. McAulay, *Octonions: A Development of Clifford's Bi-quaternions*, Oxford, UK, 1898.

[26] J. C. K. Chou, "Quaternion kinematic and dynamic differential equations," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 53–64, Feb. 1992.

[27] O. P. Agrawal, "Hamilton operators and dual-number-quaternions in spatial kinematics," *Mechanism and Machine Theory*, vol. 22, no. 6, pp. 569–575, 1987.

[28] B. Akyar, "Dual quaternions in spatial kinematics in an algebraic sense," *Turk. J. Math*, vol. 32, no. 4, pp. 373–391, 2008.

[29] J. Funda and R. Paul, "A computational analysis of screw transformations in robotics," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 348–356, 1990.

[30] A.-N. Sharkawy and N. Aspragathos, "A comparative study of two methods for forward kinematics and Jacobian matrix determination," in *2017 International Conference on Mechanical, System and Control Engineering, ICMSC 2017*, St. Petersburg, Russia, 2017, pp. 179–183.

[31] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "A compact differential formula for the first derivative of a unit quaternion curve," *The Journal of Visualization and Computer Animation*, vol. 7, no. 1, pp. 43–57, 1996.

[32] X. Wang and C. Yu, "Feedback linearization regulator with coupled attitude and translation dynamics based on unit dual quaternion," in *IEEE Int. Symp. Intelligent Control*, Japan, 2010, pp. 2380–2384.

[33] J. R. Dooley and J. M. McCarthy, "Spatial rigid body dynamics using dual quaternion components," in *1991 IEEE Int. Conf. on Robotics and Automation*, 1991, pp. 90–95.

[34] A. S. de Oliveira, E. R. De Pieri, and U. F. Moreno, "A new method of applying differential kinematics through dual quaternions," *Robotica*, vol. 35, no. 4, pp. 907–921, 2017.

[35] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Berlin, Heidelberg: Springer Int. Publishing, 2017.

[36] L. F. C. Figueredo, B. V. Adorno, J. Y. Ishihara, and G. A. Borges, "Switching strategy for flexible task execution using the cooperative dual task-space framework," in *2014 IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, Chicago, IL, 2014, pp. 1703–1709.

[37] C. M. de Farias, Y. G. Rocha, L. F. C. Figueredo, and M. C. Bernardes, "Design of singularity-robust and task-priority primitive controllers for cooperative manipulation using dual quaternion representation," in *IEEE Conf. on Control Technology and Applications (CCTA)*, Mauna Lani, HI, 2017, pp. 740–745.