Kinematic screws and dual quaternion based motion controllers ⋆

Hernán Abaunza[a,b,*], Rohit Chandra[a,*], Erol Özgür[a], Juan Antonio Corrales Ramón[a,c], Youcef Mezouar[a]

[a] *Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Institut Pascal, F-63000 Clermont-Ferrand, France.*
[b] *Present address: Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias, Av. General Ramón Corona 2514, Zapopan, 45201, Jalisco México.*
[c] *Present address: Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela 15782, Santiago de Compostela, Spain.*

Abstract

This paper presents a motion control approach with a focus on robotic manipulators based on screw theory and dual quaternions. The stability analysis of a general dual quaternion based controller has been capitalized to design an additional bounded twist controller. This controller is proposed to limit the maximum twist of the end-effector within a desired value, while preserving the accuracy achievable with high-gain feedback controllers. The proposed controllers could be useful for robotic tasks where curved motion is preferred over straight line motion. In that regard, the trajectories taken by the proposed controllers were analyzed for pose-to-pose control and some strategies have been provided for the proposed coupled controller to modify the natural trajectory. These behaviors were verified on a real robot as well as in simulation, comparing the experimental results with conventional decoupled controllers. The proposed controllers achieved smooth coupled motions which can be useful for tasks such as pick and place and assembly operations. Moreover, our coupled controllers based on screw theory need less actuator motion than the conventional decoupled approach for certain situations.

Keywords: Dual Quaternion, Screw Theory, Lyapunov Theory, Kinematic Control, Robotic Manipulator

1. Introduction

Most of the existing work on kinematic control considers rigid body motion as a decoupled sequence of translation and rotation. In these conventional approaches the rotation is represented by Euler angles, angle-axis parameters, or quaternions, while the translational error is usually interpreted as a three-dimensional vector in the euclidean space. There are few works dedicated to the controller design for the motion control of rigid bodies on *SE(3) i.e.*, the special Euclidean group of rigid-body motions which considers a rigid body motion as a translation coupled with the rotational motion. Such coupled controllers can be designed based on two main approaches for representing the complete pose of a rigid body: homogeneous transformation matrices (HTM) as in [1]; and, dual quaternions representing a screw motion [2] or representing a combination of rotation and translation similar to HTM [3, 4, 5].

Dual quaternions can be seen as an extension of unit quaternions, where the complete pose of a rigid body (position and orientation) is represented with one dual hypercomplex number, instead of separately [6, 7, 8]. They provide several advantages compared to HTM for the representation of pose transformation, such as higher computational efficiency, mathematical compactness, lack of singularities, optimal rigid transformation blending, among others (see [9, 10, 11]). These features of dual quaternions have been used for skinning of deformable models in computer graphics [12, 13, 14], for describing rigid body kinematic models [15, 16], and for kinematic and dynamic modeling of systems such as spacecraft and drones [17, 18, 19, 20].

There have been some studies on pose control of rigid bodies based on dual quaternions, such as [9, 10, 21, 22]. Other works have explored applications in pose control for robotic manipulators [4, 5, 2, 23], cooperative frameworks for dual-arm robots [3] and [24], and mobile manipulation systems as in [25]. A few works have adapted different control approaches such as robust control [26, 27], optimal linear quadratic control [28], discontinuous hysteresis [29], self-reconfigurable techniques [30], and sliding-mode control [31]. [32] provided some insights on this problem by applying Lie group theory to dual quaternions and dual vectors that can be involved in controllability and observability analysis for the control of manipulators. Most of these works adapted classical mathematical notions to dual quaternions, considering them as *8-tuples*, thus abandoning the geometrical meaning afforded by such representation.

While there are some works that consider coupled screw-motion with dual quaternion control design [2, 10, 9, 33], most of the mathematical properties of dual numbers and dual vectors are not taken into account for the stability proofs. In this

⋆This work was done in the context of the SOFTMANBOT project, which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 869855.
[c]Juan Antonio Corrales Ramón was funded by the Spanish Ministry of Universities through a 'Beatriz Galindo' fellowship (Ref. BG20/00143)
[*]Equal contribution
Email addresses: habaunza@tec.mx (Hernán Abaunza), rohit.chandra@sigma-clermont.fr (Rohit Chandra), erol.ozgur@sigma-clermont.fr (Erol Özgür), juanantonio.corrales@usc.es (Juan Antonio Corrales Ramón), youcef.mezouar@sigma-clermont.fr (Youcef Mezouar)

work, the stability proofs for such kinematic controllers are directly based on the mathematical properties of dual vectors and screw parameters. This manuscript proposes a generalized dual quaternion kinematic controller, which can be tailored to meet different performance behaviors, as compared to, for instance, [33] where the stability proof presented is very specific for the controllers they proposed. Our proposed controllers are based on screw motion for computing pose errors, thus bringing together the concepts of screw-based manipulator kinematic modeling and screw-based control of rigid bodies. Whereas the dual quaternion controller in [33] only considers rigid bodies.

One of the aspects related to pose-to-pose controllers of manipulators is the natural convergence trajectory. To the best of the authors' knowledge, the trajectory differences between coupled and decoupled controllers in the context of kinematic control for manipulators have never been discussed. The analysis of the natural trajectory taken by different controllers becomes important with regards to the safety of the robot, equipment or humans for situations where trajectory generation is not possible. Such a situation can arise for example during teleoperation tasks where the trajectory from human arm motion is received at an unknown rate, which can be either due to visual occlusion or due to sensor noise.

There are some works in the past that have talked about the natural convergence trajectory of manipulators, for instance [10, 9], and most importantly [33]. Specific controllers were proposed in [33] that resulted in helical trajectories. However, the geometrical meaning of screw displacement is not exploited in the control design, thus the resulting trajectories deviate from a pure screw displacement related to the starting and desired poses. We provide additional insights to [33] and a comparison with our work in Section 4.5.

The main contributions of our work can be summarized as follows:

- We provide a general stability proof which can be useful for synthesizing dual quaternion kinematic controllers. We use this stability analysis for the design of an additional controller that ensures bounded motions for enhancing safety and stability during pose control, for example, transporting liquids without spillage.
- We analyse the trajectory taken by the coupled and the conventional decoupled controller and demonstrate its usefulness for some common tasks and to satisfy additional task constraints like dealing with hinged doors, opening drawers, *etc* [34]. For situations where the curved motion is not possible, which is the case if there is no change in the rotation between starting and goal poses, we provide some strategies to force a curved motion during pose-to-pose control. These strategies have application in tasks such as pick and place or assembly using insertion. We validate these strategies in simulation and on a robotic manipulator, in addition to a detailed comparative analysis with a conventional kinematic controller.

The contents of this paper are organized as follows. Section 2 introduces the main properties of dual numbers, dual vectors, and dual quaternions. The controllers are designed in Section 3, while screw bounds are designed in the subsection 3.2. Experimental results are presented in Section 4. Lastly, some concluding remarks are discussed in Section 5.

2. Dual Quaternion Essentials

2.1. Mathematical Notation

In this work, mathematical symbols will be denoted as in the following table:

Mathematical entity	Notation with examples
Scalar numbers	Italics: $a, b, c \in \mathbb{R}$
Quaternions	Bold italics: $\boldsymbol{q}, \boldsymbol{q}_r, \boldsymbol{q}_t \in \mathbb{H}$
Vectors	Arrow, italics: $\vec{a}, \vec{v}, \vec{g} \in \mathbb{R}^3$
Dual numbers	Hat, italics: $\hat{a}, \hat{b}, \hat{c} \in \mathbb{DR}$
Dual quaternions	Hat, bold italics: $\hat{\boldsymbol{q}}, \hat{\boldsymbol{q}}_d, \hat{\boldsymbol{q}}_e \in \mathbb{DH}$
Dual vectors	Hat, arrow: $\hat{\vec{a}}, \hat{\vec{v}}, \hat{\vec{g}} \in \mathbb{DR}^3$

Table 1: Mathematical notation.

Throughout the theoretical development in this work, if a real number (referred to as scalar) or a vector is referred to as quaternion, it implies that a zero vector or a zero-value real number term has been appended to the variable, respectively. Similarly if a dual vector is then denoted as a dual quaternion, it implies that a zero-value real number has been added as the scalar term to both primary and dual part. For instance, let $\hat{\vec{v}} := \vec{v}_r + \epsilon\vec{v}_t$ denote a dual vector, then $\hat{\mathbf{v}} = 0+\vec{v}_r+\epsilon(0+\vec{v}_t)$. Inversely, from dual quaternion (1), a dual vector can be expressed as $\hat{\vec{q}} = \vec{q}_r + \epsilon\vec{q}_t$, by ignoring the scalar parts of $\boldsymbol{q}_r$ and $\boldsymbol{q}_t$.

Dual Quaternions are a kind of dual numbers composed by two quaternions (for a detailed review of quaternions see [35, 36, 37]), such that

$$\hat{\boldsymbol{q}} \triangleq \boldsymbol{q}_r + \epsilon\, \boldsymbol{q}_t, \quad \epsilon \neq 0, \quad \epsilon^2 = 0, \tag{1}$$

where $\boldsymbol{q}_r \triangleq q_r + \vec{q}_r$ and $\boldsymbol{q}_t \triangleq q_t + \vec{q}_t$ belong to the quaternion set $\mathbb{H}$, and are known as the *primary* and *dual* parts of $\hat{\boldsymbol{q}}$, and ϵ is a *null-potent* term which is sometimes referred to as a dual-unit in the context of dual numbers. In addition to dual quaternions, dual numbers and dual vectors are also used in this paper, which consist of scalars and vectors as *primary* and *dual* parts, respectively. Quaternion and dual quaternion product operator, marked with $\circ$, and other mathematical operations used in this work are defined the same way as in [9].

2.2. Mathematical Operations

Following mathematical operations will be used in this work, which might be new to some of the readers. For a profound review on dual quaternions and dual numbers, refer to [3] and [38, 39, 40].

2.2.1. Product between dual numbers

Following the scalar multiplication rules, and considering $\epsilon^2 = 0$, the product of dual numbers is

$$(a_r + \epsilon\, a_t)(b_r + \epsilon\, b_t) = a_r b_r + \epsilon(a_r b_t + a_t b_r) \tag{2}$$

2.2.2. Product between a dual number and a dual vector

Following the scalar-vector multiplication rules, and considering $\epsilon^2 = 0$, the product of dual numbers is

$$(a_r + \epsilon\, a_t)(\vec{b}_r + \epsilon\, \vec{b}_t) = a_r b_r + \epsilon(a_r \vec{b}_t + a_t \vec{b}_r). \tag{3}$$

2.2.3. Division between dual numbers

The division between two dual numbers is computed as

$$\frac{a_r + \epsilon a_t}{b_r + \epsilon b_t} = \frac{a_r}{b_r} + \epsilon\left(\frac{a_t}{b_r} - \frac{a_r b_t}{b_r^2}\right), \tag{4}$$

where $a_r, a_t, b_r, b_t \in \mathbb{R}$, and $b_r \neq 0$.

2.2.4. Comparison operations

Operators $>, <, \geq, \leq$ involving dual numbers, will be resolved by comparing both their *primary* and *dual* parts (ie. considering $\hat{a} = a_r + \epsilon a_t$ and $\hat{b} = b_r + \epsilon b_t$, then $\hat{a} > \hat{b}$ if $a_r > b_r$ and $a_t > b_t$).

2.2.5. Unit quaternion product

The product between two quaternions is computed as defined in [35], and is symbolized by the operator $\circ$:

$$\boldsymbol{q}_1 \circ \boldsymbol{q}_2 = (q_1 q_2 - \langle \vec{q}_1, \vec{q}_2 \rangle) + (q_2 \vec{q}_1 + q_1 \vec{q}_2 + \vec{q}_1 \times \vec{q}_2)\,, \tag{5}$$

where $\times$ represents the *cross* or *vector* product, and $\langle \cdot, \cdot \rangle$ denotes the *dot* or *inner* product between vectors.

2.2.6. Dual vector norm

For a dual vector $\hat{\vec{a}} = \vec{a}_r + \epsilon \vec{a}_t$ such that $\vec{a}_r \neq [0\ 0\ 0]^T$, its norm is defined as

$$\|\hat{\vec{a}}\| = \|\vec{a}_r\| + \epsilon \frac{\langle \vec{a}_r, \vec{a}_t \rangle}{\|\vec{a}_r\|}, \tag{6}$$

where $\|\cdot\|$ denotes the euclidean norm, notice that the *primary* part of (6) is a positive definite scalar.

2.2.7. Dual vector inner product

The inner product for dual vectors can be computed as

$$\left\langle \hat{\vec{a}}, \hat{\vec{b}} \right\rangle = \langle \vec{a}_r, \vec{b}_r \rangle + \epsilon \left(\langle \vec{a}_r, \vec{b}_t \rangle + \langle \vec{a}_t, \vec{b}_r \rangle \right). \tag{7}$$

Following a geometrical approach, the dual vector inner product can also be computed as

$$\left\langle \hat{\vec{a}}, \hat{\vec{b}} \right\rangle \triangleq \|\hat{\vec{a}}\| \|\hat{\vec{b}}\| \cos \hat{\varphi} = \|\hat{\vec{a}}\| \|\hat{\vec{b}}\| (\cos \alpha + \epsilon\, \delta \sin \alpha), \tag{8}$$

where $\hat{\varphi} = \alpha + \epsilon\, \delta$ represents the *dual displacement*, composed by the angle α between the axes of $\vec{a}_r$ and $\vec{b}_r$, and the distance δ along their common normal, see [40].

2.2.8. Dual product bi-operator

Let $\odot$ represent following operation between two dual numbers or between a dual number and a dual vector:

$$\hat{a} \odot \hat{b} = a_r b_r + \epsilon\, a_t b_t \ , \ \text{or} \ \ \hat{a} \odot \hat{\vec{b}} \triangleq a_r \vec{b}_r + \epsilon\, a_t \vec{b}_t. \tag{9}$$

2.2.9. Dual vector normalization

A normalized vector $\vec{a}\,' = \vec{a}/\|\vec{a}\|$ represents the direction of $\vec{a}$. It can be seen as a 3-dimensional generalization of the scalar sign$(\cdot)$ function.

A normalized dual vector can similarly be computed as

$$\begin{aligned} \hat{\vec{a}}' = \frac{\hat{\vec{a}}}{\|\hat{\vec{a}}\|} &= \frac{\vec{a}_r + \epsilon \vec{a}_t}{\|\vec{a}_r\| + \epsilon \frac{\langle \vec{a}_r, \vec{a}_t \rangle}{\vec{a}_r}} \\ &= \frac{\vec{a}_r}{\|\vec{a}_r\|} + \epsilon \left(\frac{\vec{a}_t \|\vec{a}_r\|^2 - \vec{a}_r \langle \vec{a}_r, \vec{a}_t \rangle}{\|\vec{a}_r\|^3} \right). \end{aligned} \tag{10}$$

For more details on operations and properties of dual numbers and dual vectors see [38].

2.3. Dual quaternions and screw transformations

A very well known approach to express coupled rotational and translational transformations is by employing screw displacement, which we depict in Fig. 1. Similar to the exponential mapping of angle-axis parameters to compute the unit quaternions representing orientation related to a rigid body pose, it is possible to express screw-based coupled transformations with the exponential mapping of screw motion. This characteristic of dual quaternions to inherit the properties related to quaternions has been referred to as the *principle of transference* [41, 20, 9]. Other properties that UDQs inherit from unit quaternions include double cover of $SE(3)$ logarithmic mapping, inverse transformation using conjugate, interpolation, etc. We have exploited these properties in our implementation to deal with controller design, its validation and comparison with existing works.

A representation of pose transformation using screw motion is composed of four displacement parameters: θ, $\vec{\ell}$, d, and $\vec{m}$. An axis-angle rotation is represented by a rotation $\theta \in \mathbb{R}^+$ along an axis $\vec{\ell} \in \mathbb{R}^3$, while $d \in \mathbb{R}$ symbolizes a displacement over the same axis. Here $\vec{m} \in \mathbb{R}^3$ is a moment vector, computed from the screw radius $\vec{c}$ and the rotation axis $\vec{\ell}$ as $\vec{m} = \vec{c} \times \vec{\ell}$, notice that $\vec{c} \perp \vec{\ell}$, $\vec{m} \perp \vec{\ell}$ and $\vec{m} \perp \vec{c}$.

A unit dual quaternion $\hat{\boldsymbol{q}}$ can then be computed from screw parameters with exponential mapping:

$$\begin{aligned} \hat{\boldsymbol{q}} = e^{\hat{\boldsymbol{\vartheta}}/2} &= \cos\left(\frac{\theta}{2}\right) + \vec{\ell} \sin\left(\frac{\theta}{2}\right) \\ &+ \epsilon \left(-\frac{d}{2} \sin\left(\frac{\theta}{2}\right) + \vec{\ell}\, \frac{d}{2} \cos\left(\frac{\theta}{2}\right) + \vec{m} \sin\left(\frac{\theta}{2}\right) \right). \end{aligned} \tag{11}$$

We can compute the screw displacement parameters from a dual quaternion using logarithmic mapping as

$$\hat{\boldsymbol{\vartheta}} = 2 \ln \hat{\boldsymbol{q}} = (0 + \theta \vec{\ell}) + \epsilon(0 + d\vec{\ell} + \theta \vec{m}), \tag{12}$$

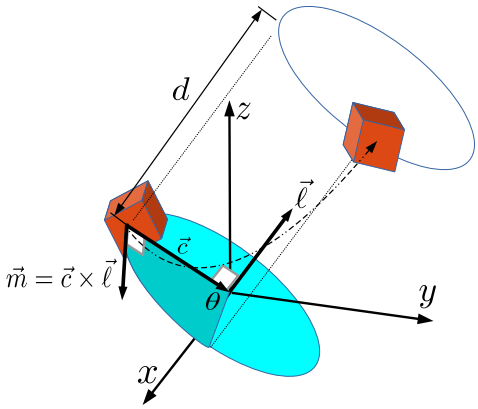

Figure 1: Screw motion with a translation d, and a rotation θ over an axis $\vec{\ell}$, with a moment $\vec{m}$.

The screw parameters are computed by following the strategy proposed in [2]. A discontinuity has been introduced to avoid the multiple representations of the same pose, that compromise stability due to a topological obstruction [42], and to ensure that the shortest path is always followed as $-\pi < \theta < \pi$. This is achieved by inverting the sign of the dual quaternion as:

- Case when $0 < |\cos^{-1}(q_r)| < \pi$:

$$\begin{aligned} \theta &= \cos^{-1}(q_r), \quad d = -2\frac{q_t}{\sin(\theta/2)}, \\ \vec{\ell} &= \frac{\vec{q}_r}{\sin(\theta/2)}, \quad \vec{m} = \left(\vec{q}_t - q_r\frac{d}{2}\vec{\ell}\right)\frac{1}{\sin(\theta/2)}, \end{aligned} \tag{13}$$

- Case when $|\cos^{-1}(q_r)| \geq \pi$:

$$\begin{aligned} \theta &= \cos^{-1}(-q_r), \quad d = 2\frac{q_t}{\sin(\theta/2)}, \\ \vec{\ell} &= \frac{-\vec{q}_r}{\sin(\theta/2)}, \quad \vec{m} = \left(q_r\frac{d}{2}\vec{\ell} - \vec{q}_t\right)\frac{1}{\sin(\theta/2)}, \end{aligned} \tag{14}$$

A special case is also considered to avoid singularities when $\theta \approx 0$ (pure translation) as:

$$d = 2\|\vec{q}_t\| \quad , \quad \vec{\ell} = 2\vec{q}_t/d \quad , \quad \vec{m} = [0,\, 0,\, 0]^T \tag{15}$$

Remark 1. *Considering that* $\hat{\vartheta}$ *from* (12) *denotes a screw transformation, then its norm represents the magnitude of such screw (rotation angle* θ *and translation* d*). Furthermore, applying* (6) *to* (12) *yields:*

$$\hat{\vartheta} = \|2\ln\hat{\boldsymbol{q}}\| = \theta + \epsilon d. \tag{16}$$

Therefore, $\hat{\vartheta}$ *represents the magnitude of a screw transformation as a dual number. Notice from* (6) *that* θ *is positive definite.*

Remark 2. *If a dual vector* $\hat{\vec{\vartheta}}$ *represents a transformation as in* (12)*, then its normalized dual vector* $\hat{\vec{\vartheta}}'$ *represents its screw direction with a unitary magnitude. Furthermore, applying* (10) *to* (12) *results in:*

$$\hat{\vec{\vartheta}}' = \frac{\hat{\vec{\vartheta}}}{\left\|\hat{\vec{\vartheta}}\right\|} = \frac{2\ln\hat{\boldsymbol{q}}}{\|2\ln\hat{\boldsymbol{q}}\|} = \vec{\ell} + \epsilon\vec{m}. \tag{17}$$

Therefore, $\hat{\vec{\vartheta}}'$ *represents a screw transformation over axis* $\vec{\ell}$*, with a moment vector* $\vec{m}$ *and a magnitude* $\left\|\hat{\vec{\vartheta}}'\right\| = 1 + \epsilon\,0$.

2.4. Dual quaternion kinematics

The derivative of a unit dual quaternion is given by

$$\dot{\hat{\boldsymbol{q}}} = \frac{1}{2}\hat{\boldsymbol{q}} \circ \hat{\boldsymbol{\xi}}, \tag{18}$$

where $\hat{\boldsymbol{\xi}}$ is the *twist* (combination of rotational and translational velocities), defined as a dual quaternion such that

$$\hat{\boldsymbol{\xi}} = \frac{d}{dt}2\ln\hat{\boldsymbol{q}} = 0 + \dot{\theta}\vec{\ell} + \theta\dot{\vec{\ell}} + \epsilon(0 + \dot{d}\vec{\ell} + d\dot{\vec{\ell}} + \dot{\theta}\vec{m} + \theta\dot{\vec{m}}) \tag{19}$$

This can be interpreted as a screw velocity represented as a single dual variable. It contains a rotational velocity $\dot{\theta}\vec{\ell} + \theta\dot{\vec{\ell}}$; the translational component composed by a linear velocity $\dot{d}$ along the screw axis $\vec{\ell}$; and a tangential velocity resulting from the rotational velocity $\dot{\theta}$ with the moment vector $\vec{m}$ [10, 33].

2.5. Dual quaternion error kinematics

An error between two dual quaternion poses computed and represented in a common base frame (which has been omitted in the superscript for brevity) can be computed by using the quaternion product between a desired pose $\hat{\boldsymbol{q}}_d$ with a pose $\hat{\boldsymbol{q}}$ as

$$\hat{\boldsymbol{q}}_e = \hat{\boldsymbol{q}}_d \circ \hat{\boldsymbol{q}}^*, \tag{20}$$

where $\hat{\boldsymbol{q}}^* = \boldsymbol{q}_r^* + \epsilon\boldsymbol{q}_t^* = q_r - \vec{q}_r + \epsilon\,(q_t - \vec{q}_t)$ denotes the conjugate of $\hat{\boldsymbol{q}}$ (see [10]). Then the derivative of (20) is given by

$$\dot{\hat{\boldsymbol{q}}}_e = \frac{1}{2}\hat{\boldsymbol{q}}_e \circ \hat{\boldsymbol{\xi}}_e = \dot{\hat{\boldsymbol{q}}}_d \circ \hat{\boldsymbol{q}}^* - \hat{\boldsymbol{q}}_d \circ \hat{\boldsymbol{q}}^* \circ \hat{\boldsymbol{q}}^* \circ \dot{\hat{\boldsymbol{q}}}, \tag{21}$$

where $\hat{\boldsymbol{\xi}}_e$ represents the twist error. Extending the dual quaternion derivatives, and solving for $\hat{\boldsymbol{\xi}}_e$, yields

$$\hat{\boldsymbol{\xi}}_e = \frac{d}{dt}2\ln\hat{\boldsymbol{q}}_e = \hat{\boldsymbol{q}} \circ \hat{\boldsymbol{\xi}}_d \circ \hat{\boldsymbol{q}}^* - \hat{\boldsymbol{\xi}}, \tag{22}$$

where $\hat{\boldsymbol{\xi}}_d$ denotes the desired twist. If the reference transformation is assumed to be quasi-constant, then its twist (composed by the rotational and translational velocities) can be assumed to be almost zero (i.e. $\hat{\boldsymbol{\xi}}_d \approxeq 0 + \vec{0} + (0 + \vec{0})\epsilon$), then it can be considered that $\hat{\boldsymbol{\xi}}_e \approxeq -\hat{\boldsymbol{\xi}}$.

3. Kinematic Screw Control Design

A dual quaternion kinematic control law is now developed for robotic manipulators. Let $\hat{\boldsymbol{q}}_e = \hat{\boldsymbol{q}}_d \circ \hat{\boldsymbol{q}}^*$ represent a pose error between the end effector $\hat{\boldsymbol{q}}$, and a reference $\hat{\boldsymbol{q}}_d$, computed and represented in a common frame.

We define a Lyapunov candidate function as:

$$\hat{V} \coloneqq V_r + \epsilon V_t \coloneqq \frac{1}{2}\|2\ln\hat{\boldsymbol{q}}_e\| \odot \|2\ln\hat{\boldsymbol{q}}_e\| = \frac{1}{2}(\theta_e^2 + \epsilon\, d_e^2)\,, \tag{23}$$

with $V_r,\ V_t \in \mathbb{R}^+$. Such that the time derivative of (23) is also a dual number (following $\dot{\hat{a}} = \dot{a}_r + \epsilon\dot{a}_t$) given by

$$\dot{\hat{V}} = \|2\ln\hat{\boldsymbol{q}}_e\| \odot \left\langle \frac{2\ln\hat{\boldsymbol{q}}_e}{\|2\ln\hat{\boldsymbol{q}}_e\|}, \hat{\vec{\xi}}_e \right\rangle = \theta_e\dot{\theta}_e + \epsilon\, d_e\dot{d}_e. \tag{24}$$

If a controller designed for $\hat{\vec{\xi}}_e$ ensures that the Lyapunov function derivative behaves as

$$\dot{\hat{V}} = \|2\ln\hat{\boldsymbol{q}}_e\| \odot \left\langle \frac{2\ln\hat{\boldsymbol{q}}_e}{\|2\ln\hat{\boldsymbol{q}}_e\|}, \hat{\vec{\xi}}_e \right\rangle < 0 + \epsilon\, 0, \tag{25}$$

then the convergence of $2\ln\hat{\boldsymbol{q}}_e \to \vec{0} + \epsilon\,\vec{0}$ would be guaranteed.

Theorem 1. *A twist controller defined as*

$$\hat{\vec{\xi}}_e := -\hat{\lambda}\frac{2\ln\hat{\boldsymbol{q}}_e}{\|2\ln\hat{\boldsymbol{q}}_e\|} \tag{26}$$

with $\hat{\lambda} := \lambda_r + \epsilon\lambda_t,\ \lambda_r, \lambda_t \in \mathbb{R}$, *such that* $\lambda_r > 0$ *and* $\text{sign}(\lambda_t) = \text{sign}(d_e)$, *ensures convergence of* $2\ln\hat{\boldsymbol{q}}_e \to \vec{0} + \epsilon\,\vec{0}$ *by means of Lyapunov function* (23) *and its derivative* (24). *Thus stabilizing the dual quaternion error kinematics from system* (22) *with a convergence behavior that depends on the design of* $\hat{\lambda}$.

Proof 1. *Introducing* (26) *into* (24), $\dot{\hat{V}}$ *is rewritten as*

$$\dot{\hat{V}} = -\|2\ln\hat{\boldsymbol{q}}_e\| \odot \left\langle \frac{2\ln\hat{\boldsymbol{q}}_e}{\|2\ln\hat{\boldsymbol{q}}_e\|}, \frac{2\ln\hat{\boldsymbol{q}}_e}{\|2\ln\hat{\boldsymbol{q}}_e\|}\hat{\lambda} \right\rangle. \tag{27}$$

Notice that the control action of (26) *consists on a twist motion of magnitude* $\hat{\lambda}$ *in the screw error direction, therefore considering Remark 2,* (27) *becomes*

$$\begin{aligned} \dot{\hat{V}} &= -\|2\ln\hat{\boldsymbol{q}}_e\| \odot \left\langle (\vec{\ell}_e + \epsilon\vec{m}_e), (\vec{\ell}_e + \epsilon\vec{m}_e)\hat{\lambda} \right\rangle \\ &= -\|2\ln\hat{\boldsymbol{q}}_e\| \odot \left\langle (\vec{\ell}_e + \epsilon\vec{m}_e), (\vec{\ell}_e + \epsilon\vec{m}_e)(\lambda_r + \epsilon\lambda_t) \right\rangle. \end{aligned} \tag{28}$$

Considering the dual vector inner product geometrical properties from (8) *yields*

$$\left\langle (\vec{\ell}_e + \epsilon\vec{m}_e), (\vec{\ell}_e + \epsilon\vec{m}_e)\hat{\lambda} \right\rangle = \hat{\lambda}(\cos\alpha_e + \epsilon\,\delta_e \sin\alpha_e), \tag{29}$$

however since $\vec{\ell}_e \parallel \lambda_r\vec{\ell}_e$, *then* $\alpha_e = 0$. *Thus* (28) *becomes*

$$\begin{aligned} \dot{\hat{V}} &= -\|2\ln\hat{\boldsymbol{q}}_e\| \odot \hat{\lambda} \\ &= -(\theta_e\lambda_r + \epsilon\, d_e\lambda_t). \end{aligned} \tag{30}$$

Recalling from (6) *and* (16) *that* θ_e *is a positive definite scalar. Hence,* λ_r *and* λ_t *must be designed such that* $\lambda_r > 0$ *for all* $\theta_e > 0$ *and* $\text{sign}(\lambda_t) = \text{sign}(d_e)$, *then it holds that* $\dot{\hat{V}} < 0 + \epsilon\, 0$ *for all* $\theta_e, d_e \neq 0$. *In the case that* $\theta_e = 0$ *and* $d_e = 0$, *then* $\dot{\hat{V}} = 0 + \epsilon\, 0$ *implying* $\hat{V} \to 0 + \epsilon\, 0$ *and hence* $2\ln\hat{\boldsymbol{q}}_e \to \vec{0} + \epsilon\,\vec{0}$. ■

Theorem 1 implies that several choices can be made to design $\hat{\lambda}$. This fact enables the exploration of different convergence behaviors. In the following subsection, some controllers are proposed as a consequence of this observation.

3.1. Proportional twist controller

The most straightforward option for selecting $\hat{\lambda}$ is to define it as $\hat{\lambda} := \hat{k} \odot \|2\ln\hat{\boldsymbol{q}}_e\|$, with $\hat{k} := k_r + \epsilon\, k_t$, such that

$$\hat{\vec{\xi}}_e = -k_r\theta_e\vec{\ell}_e - \epsilon\left(k_r\theta_e\vec{m}_e + k_t d_e\vec{\ell}_e\right) \tag{31}$$

with $k_r, k_t \in \mathbb{R}^+$. (31) can be seen as a proportional control action in the screw error opposite direction (see [2]).

Corollary 1.1. *Introducing* $\hat{\lambda} := \hat{k} \odot \|2\ln\hat{\boldsymbol{q}}_e\|$ *to the Lyapunov function derivative* (28) *results in*

$$\dot{\hat{V}} = -(k_r\theta_e^2 + \epsilon\, k_t d_e^2). \tag{32}$$

The proportional action of the controller implies that $\dot{\hat{V}} < 0 + \epsilon\, 0$ *for all* $\theta_e, d_e \neq 0$, *and* $\dot{\hat{V}} = 0 + \epsilon\, 0$ *if* $\theta_e, d_e = 0$ *the system will exponentially converge to zero.*

3.2. Bounded proportional twist controller

The inconvenience of controller (31) is that large unbounded pose errors could yield excessive twists. This may result in aggressive movement that could be hazardous for the robot and nearby operators. A solution to this problem is to define $\hat{\lambda}$ in different cases of the screw error, given that a proportional twist is given by $\hat{k} \odot \|2\ln\hat{\boldsymbol{q}}_e\| = k_r\theta_e + \epsilon k_t d_e$, such as

$$\hat{\lambda} := \begin{cases} k_r\theta_e + \epsilon k_t d_e & \text{for } k_r\theta_e < \theta_\mu\,,\ k_t|d_e| < d_\mu, \\ k_r\theta_e + \epsilon d_\mu \text{sign}(d_e) & \text{for } k_r\theta_e < \theta_\mu\,,\ k_t|d_e| \geq d_\mu, \\ \theta_\mu + \epsilon k_t d_e & \text{for } k_r\theta_e \geq \theta_\mu\,,\ k_t|d_e| < d_\mu, \\ \theta_\mu + \epsilon\, d_\mu \text{sign}(d_e) & \text{for } k_r\theta_e \geq \theta_\mu\,,\ k_t|d_e| \geq d_\mu, \end{cases} \tag{33}$$

where $\hat{\vartheta}_\mu := \theta_\mu + \epsilon\, d_\mu \text{sign}(d_e)$ is a constant twist bound given by a maximum angle θ_μ and displacement d_μ.

The twist controller can be formed as

$$\hat{\vec{\xi}}_e := \begin{cases} (k_r\theta_e + \epsilon k_t d_e)\hat{\vec{\vartheta}}' & \text{for } k_r\theta_e < \theta_\mu\,,\ k_t|d_e| < d_\mu, \\ (k_r\theta_e + \epsilon d_\mu \text{sign}(d_e))\hat{\vec{\vartheta}}' & \text{for } k_r\theta_e < \theta_\mu\,,\ k_t|d_e| \geq d_\mu, \\ (\theta_\mu + \epsilon k_t d_e)\hat{\vec{\vartheta}}' & \text{for } k_r\theta_e \geq \theta_\mu\,,\ k_t|d_e| < d_\mu, \\ (\theta_\mu + \epsilon\, d_\mu \text{sign}(d_e))\hat{\vec{\vartheta}}' & \text{for } k_r\theta_e \geq \theta_\mu\,,\ k_t|d_e| \geq d_\mu, \end{cases} \tag{34}$$

Corollary 1.2. *The consequence of introducing* (34) *in* (28) *can be analyzed in four cases. First consider that* (34) *is saturated when* $k_r\theta_e < \theta_\mu\,,\ k_t|d_e| < d_\mu$, *such that it consists on a constant twist in the opposite direction of* $\hat{\boldsymbol{q}}_e$. *Therefore*

$$\dot{\hat{V}} = -\theta_\mu\theta_e - \epsilon\, d_\mu d_e \text{sign}(d_e) \tag{35}$$

is negative definite, and ensures finite time convergence of $2\ln\hat{\boldsymbol{q}}_e \to \vec{0} + \epsilon\vec{0}$, *see [43]. Therefore, a finite time exists when (depending on the initial error magnitude), either the rotation or the translation will arrive the second or third case, such that* $k_r\theta_e < \theta_\mu\,,\ k_t|d_e| \geq d_\mu$, *or* $k_r\theta_e \geq \theta_\mu\,,\ k_t|d_e| < d_\mu$, *then,* (25) *becomes*

$$\dot{\hat{V}} = -k_r\theta_e^2 - \epsilon\, d_\mu d_e \text{sign}(d_e) \tag{36}$$

or

$$\dot{\hat{V}} = -\theta_\mu\theta_e - \epsilon\, k_t d_e^2, \tag{37}$$

which are both negative definite, and will continue to ensure $2\ln\hat{\boldsymbol{q}}_e \to \vec{0} + \epsilon\vec{0}$. *Finally, a time exists such that* $\hat{k} \odot \|2\ln\hat{\boldsymbol{q}}_e\| < \hat{\vartheta}_\mu$, *and* (25) *becomes*

$$\dot{\hat{V}} = -(k_r\theta_e^2 + \epsilon\, k_t d_e^2). \tag{38}$$

Then, it holds that $\dot{\hat{V}} < 0 + \epsilon\, 0$ *for all* $\theta_e, d_e \neq 0$, *and* $\dot{\hat{V}} = 0 + \epsilon\, 0$ *if* $\theta_e, d_e = 0$, *such that* $\|2\ln\hat{\boldsymbol{q}}_e\| < \hat{\vartheta}_\mu$ *after a finite time, and then* $2\ln\hat{\boldsymbol{q}}_e \to \vec{0} + \epsilon\,\vec{0}$ *exponentially.*

Remark: In order maintain the harmony between the convergence of rotational and translational error, *i.e.* to respect the screw motion, both the rotational and translational gains are adjusted when either $k_r\theta_e$ or $k_t d_e$ goes out of bound. We achieve this by the considering the control input $\hat{\vec{\xi}}_e$ as a $6D$ vector ([44]) and scaling it to a value so that the lowest value of the rotational and translational bounds are respected.

4. Experimental Validation

We chose Franka Emika PANDA robot with 7 revolute joints for the validation of the proposed controllers. In the following sections we will provide more detail about the implementation strategy.

4.1. Dual quaternion based coupled Jacobian

Let $\underline{\omega} = [\omega_1 \ \omega_2 \ \cdots \ \omega_7]^T$ denote the current joint angular velocities. The relation between the end-effector twist (or the control action) and $\underline{\omega}$ is given by

$$[\vec{\xi}_r^T \ \vec{\xi}_t^T]^T = \begin{bmatrix} \hat{\vec{s}}_1 & \hat{\vec{s}}_2 & \cdots & \hat{\vec{s}}_7 \end{bmatrix} \underline{\omega} = \mathcal{J}_{DQ}\underline{\omega}\,, \qquad (39)$$

where $\hat{\vec{s}}_i$ is the current joint screw axis for the i_{th} joint represented as a column vector, $\vec{\xi}_r$ and $\vec{\xi}_t$ denote the rotational and translational vector components of $\hat{\vec{\xi}}_e = \vec{\xi}_r + \epsilon\vec{\xi}_t$, computed by controllers (31) or (34). $\mathcal{J}_{DQ}$ is a dual quaternion based Jacobian matrix, as described in [2] and [45], where the scalar components of both primary and dual parts are null, and therefore not considered.

4.2. Decoupled Jacobian

The proposeed controllers were compared with a popular controller that decouples the rotational and translational motions. This decoupled controller is defined as

$$\vec{u}_r = k_r(\vec{\theta}_e)\,, \quad \vec{u}_t = k_t(\vec{p}_d - \vec{p}), \qquad (40)$$

where $k_r, k_t > 0$ denote positive gains. The orientation error, given in angle-axis notation (see [35]), between the current orientation $\boldsymbol{q}$ and the desired orientation quaternion $\boldsymbol{q}_d$ is given as $\vec{\theta}_e = 2\ln(\boldsymbol{q}_d^* \circ \boldsymbol{q})$. The current and desired end-effector positions are represented with $\vec{p}$ and $\vec{p}_d$ respectively.

The relation between controller (40) and joint velocities $\underline{\omega}$ is then given by

$$[\vec{u}_r^T \ \vec{u}_t^T]^T = \mathcal{J}_{KDL}\underline{\omega}, \qquad (41)$$

where $\mathcal{J}_{KDL}$ is obatined from the KDL library [46, 47]. While we only validate our bounded twist controller (Section 3.2) in the experiments, we can devise a similar strategy to bound the control twist in the case of a decoupled controller as well. We can consider independent bounds for the rotational and translational sub-systems. We can either bound the norm of those components (three components of each), or we can compute the scaling factor based on the largest magnitude components and the desired bound.

In order to avoid singularities, a "damped" pseudo-inverse $\mathcal{J}^{-1} \approxeq \mathcal{J}^T * (\mathcal{J}\mathcal{J}^T + z\mathbf{I}_{6\times 6})^{-1}$ was used for both coupled and decoupled controllers, given in (39) and (41), respectively, with a small damping factor $z = 0.001$. $\mathbf{I}_{6\times 6}$ refers to a 6×6 identity matrix.

4.3. Experimental Setup

The controllers were implemented in ROS and on Franka Emika Panda robot. We used the in-built joint velocity control mode for the robot. The algorithms ran on a computer equipped with an *Intel Core i7-8700T CPU* with 12 cores running at 2.40 *GHz*, and 16 *GB* RAM. The frequency of the control loop was ≈ 970 *Hz*.

4.4. Validation Strategy

The convergence behavior and natural trajectory of the proposed controllers, namely, proportional twist and bounded proportional twist controllers, were compared against the conventional decoupled controller for a pose-to-pose control task. Additionally, the natural convergence trajectory of a control law claiming helical trajectories in [33] were compared against ours in simulation.

4.4.1. Twist control validation

In this task, the desired pose $\hat{\boldsymbol{q}}_d$ was defined by manually moving the robot from a randomly selected home pose to another pose by rotating just one joint of the robot and recording the end-effector pose. Then, the robot was set to move from its home position to the recorded pose using both the controllers. We defined two such tasks by: rotating joint 3 (*Task 1*); and by rotating joint 4 (*Task 2*).

The $3D$ position trajectory taken by the end-effector is depicted in Fig. 3. Notice that the DQ controller always took a circular path (following a screw motion) from its home position to the desired pose. In contrast, the decoupled controller moved in a straight line. This circular path taken by the proposed coupled controller can be useful for tasks that involve twisting, screwing, or tasks involving mechanical constraints like opening a hinged door.

Fig. 2 illustrates the absolute traslational and rotational errors, obtained with the euclidean norm of the position and orientation vectors, respectively. Both controllers converged exponentially as expected from proportional feedback control law. The time of convergence was also similar for both controllers, even if the DQ control law took a longer circular path as shown in Fig. 3.

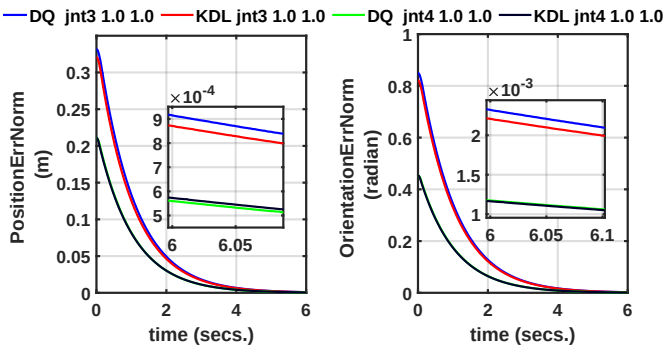

Figure 2: *Twist control validation*: Pose error convergence comparison in terms of norm of the position error and the norm of the orientation error computed using angle-axis parameters of the error quaternion.

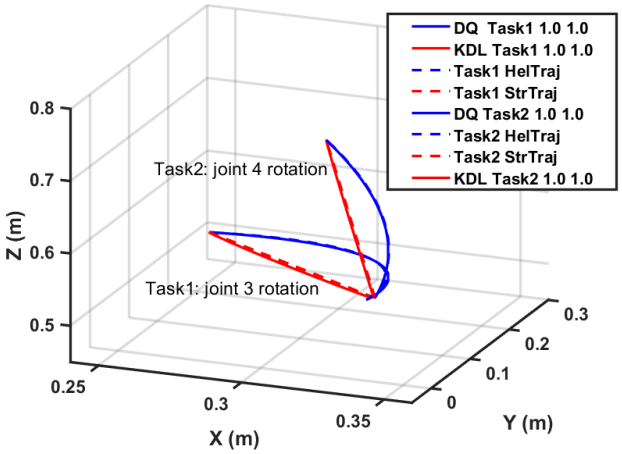

Figure 3: *Twist control validation*: End-effector 3D trajectory performed by the dual quaternion proportional twist (blue) and the decoupled (red) controllers, along with the predicted helical and straight line trajectories during two different tasks. The rotational gain (k_r) and translational proportional gain (k_t) are given in the legend in this same order. Both tasks start at the same initial pose.

The change in the joint positions of PANDA robot for the coupled and decoupled controllers is shown in Fig. 4. We noticed that the decoupled controller motion required more joints (joints 2 and 4 for *Task 1*, and joint 2 for *Task 2*), as compared to the coupled DQ controller taking a screw path to achieve the same tasks.

4.4.2. Arbitrary pose-to-pose control

We also studied the trajectory for pose-to-pose control with randomly generated poses in a realistic simulation environment implemented in Matlab. We generated random screw-axis and random rotation and translation value to compute a goal pose within the workspace of the manipulator. The trajectories taken by the end-effector for coupled, decoupled and bounded coupled controller for one such pair of random home and goal poses have been given in Fig. 6. The expected reference trajectories for the decoupled and coupled controllers, i.e. a straight line path and a screw trajectory, are also shown in the same plot. The deviation of the path taken by the manipulator from the corresponding reference trajectories in terms of root mean square error were computed and are shown in Fig. 7.

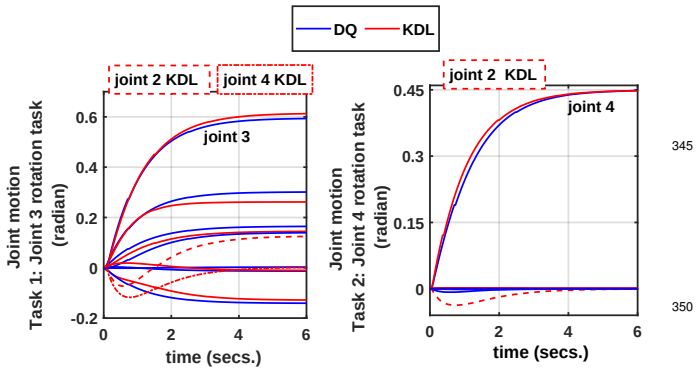

Figure 4: *Twist control validation*: Joint motion plot for the two tasks involving single joint rotation in joint 3 and joint 4, respectively. Notice redundant motions of joints 2 and 4 for *Task 1* and joint 4 for *Task 2*, required for decoupled controller shown in red.

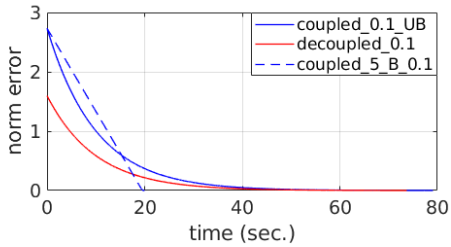

Figure 5: Pose error convergence plot for arbitrary pose control in terms of the norm of 6D screw twist for gains 0.1 for unbounded coupled and decoupled controller (UB), and 5 for the bounded coupled controller (B) with the bound limit 0.1, for both translation and rotation part.

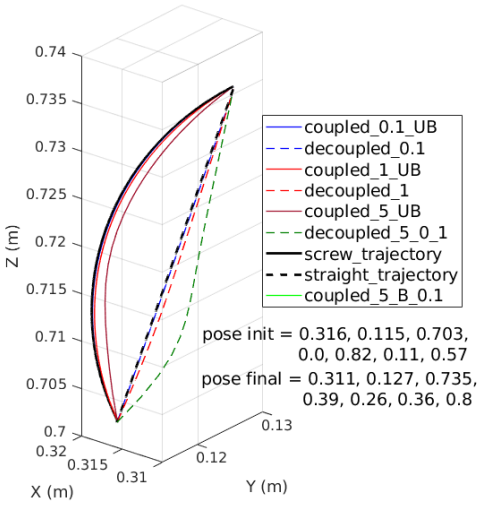

Figure 6: Trajectory taken by coupled unbounded (UB), decoupled, and coupled bounded (B) controllers during arbitrary pose control for gains 0.1 for UB controller, and 5 for the B controller with bound limit 0.1, for both translation and rotation parts. The first three terms for *pose init* and *pose final* are the translation in metres and the last four terms are related to the orientation quaternion.

The coupled twist controller yields a velocity screw which minimizes the pose error expressed as a screw displacement. This velocity screw is then decomposed into the linear superposition of the manipulator's joints' screw velocities, more explicitly into joints' screw axes and speeds. This decomposition is done through the manipulator's Jacobian whose columns are written in terms of the joints' screw axes [48, 49]. This ensures that the end-effector takes a screw path to minimize the pose error. Indeed, an articulated mechanism such as the serial manipulator is a physical representation of a screw system where the links perform screw displacements through the joints. This spontaneously makes screw theory a convenient tool for robotics.

On the other hand, the decoupled controller yields a velocity control law which minimizes the pose error expressed as the difference in Cartesian position coordinates and axis-angle 3-dimensional vectors. This velocity control law is then decomposed into the linear superposition of the manipulator's joints' velocities. This decomposition is done through the manipulator's Jacobian whose columns are written such that the com-

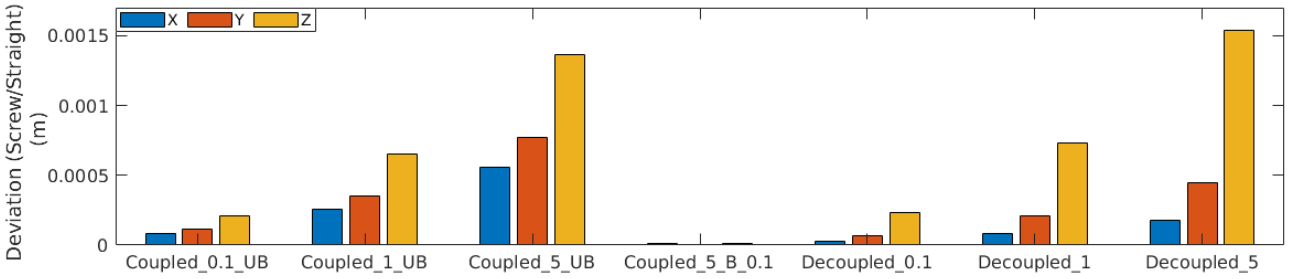

Figure 7: Deviation from screw path given as root mean square error for coupled controller and straight line path for decoupled controller for gains 0.1 for unbounded coupled and decoupled controller (UB), and 5 for the bounded coupled controller (B) with the bound limit 0.1, for both translation and rotation part.

position of the joint speeds decouples the linear and angular velocities of the end-effector [50, 47]. However, this is not sufficient to ensure that the end-effector takes a straight path to minimize the pose error. It also depends on the manipulator's joint configurations to be able to reproduce an adequate screw axis to minimize the orientation errors.

As shown in the two plots (Fig. 7 and Fig. 6), the deviation of the actual trajectories from their expected reference trajectories increases as we increase the gain for both controllers. This behaviour is expected because the generated control command for the manipulator joints are higher for higher gains which leads to larger motions and thus leading to deviation from the natural error minimization trajectory. Note that the deviation is smallest for the bounded control, in fact, even lower compared to unbounded coupled controller with the same gain (0.1) as the bound. This is due to the fact that the bound is applied after computing the control input and not on the screw error. The convergence of the norm of the error twist for all the controllers is given in Fig. 5. While both coupled and decoupled unbounded controller converges exponentially, the bounded controller converges linearly. This is due to the fact that a high initial gain was chosen for the bounded controller to emphasize its effectiveness.

4.4.3. Curved motion without final rotation

Previous experiments demonstrated that the coupled controller takes a screw trajectory whenever there is some rotation required to achieve the goal pose. However, if the starting and the goal pose have the same orientation, the trajectory taken by the coupled and decoupled controller will be exactly the same. In this section we propose some modification to the natural trajectory taken by the coupled controller for situations where we still wanted a curved trajectory in the case where there is no final rotation, for example during a pick and place task.

4.4.3.1 *Rotation at the beginning and end*

One of the simplest strategy can be imparting a pure rotation at the starting or at the end of the task, where this extra rotation is nullified during reduction of the translational error. The trajectories taken by the end-effector using these strategies for pose-to-pose control without rotation is given in Fig. 8. For this task the unbounded controller with the gain 1.0 was used.

As can be seen in the plot for the norm-error of the 6-D error vector shown in Fig. 9 and trajectory plot (Fig. 8), there is a slight difference between the resulting trajectories related to these two strategies. The small circular motion at the end of the task for the *rotation-end* strategy is due to the fact that we defined two successive goal poses for the *rotation-end*: one with the original translation and added rotation; and, other with the original desired pose. The small circle is related to the translation error while achieving the first goal. Whereas, for *rotation-start* strategy there is only an orientation error at the starting of the motion with respect to the home pose.

4.4.3.2 *Smooth transition*

One of the issues with the previous strategy is the requirement of big rotations at the beginning or at the end of the task. This kind of trajectory is undesired for tasks such as picking and placing materials that could spill. To avoid this, we can spread the added rotation by defining two intermediate poses with a strategy shown in Fig. 10, which can be parameterized by a maximum desired tilt (θ).

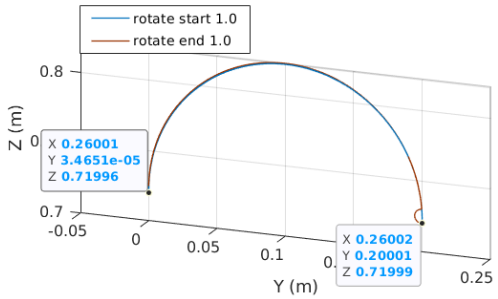

Figure 8: Trajectory taken by unbounded coupled (UB) controller during *rotation-start* and *rotation-end* strategies with gain equal to 1.0.

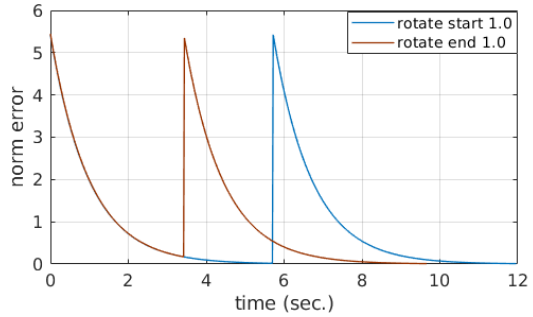

Figure 9: Pose error convergence plot for arbitrary pose control in terms of the norm of 6D screw twist for gains equal to 1.0 with unbounded coupled controller for *rotation-start* and *rotation-end* strategies.

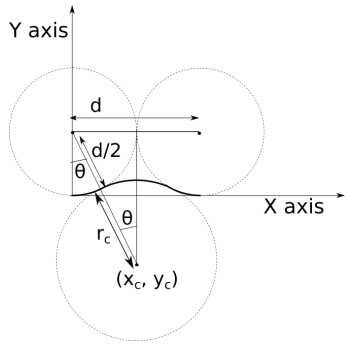

Figure 10: Strategy for smooth transition during pose-to-pose control where there is no rotation.

In the shown strategy, without losing generality, the frame attached to the end-effector goes through a rotation of angle θ along the positive z-axis and then along negative z axis on an arc spanning over 2θ defined by the circle with centre (x_c, y_c), and then finally, again a rotation of θ along the positive z-axis. The center of the circle can be obtained as follows:

$$x_c = d/2 \tag{42}$$
$$(d/2 + r_c)\sin(\theta) = d/2 \tag{43}$$
$$(d/2 + r_c)\cos(\theta) = d/2 - y_c \tag{44}$$
$$y_c = (1 - \arctan(\theta))d/2, \tag{45}$$

where d is the total translation desired.

The trajectory followed by the manipulator's end-effector and the corresponding plot for the norm error of a 6D twist is given in Fig. 11, and Fig. 12 for maximum desired tilt $\theta = \pi/4$ and $\pi/8$. A bounded control strategy was used to ensure smooth trajectory with gain 1.0 for bound 0.1 for both translation and rotation parts. As it is clear from Fig. 11, the end-effector takes the expected curved trajectory.

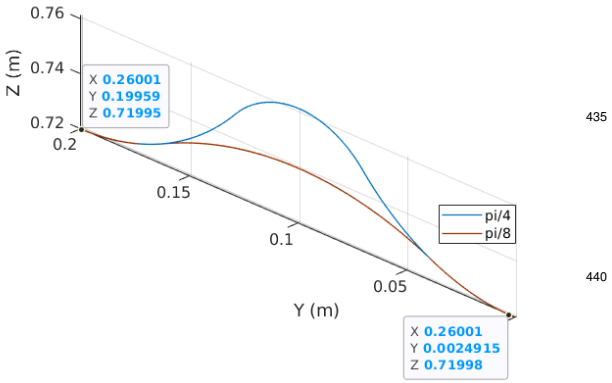

Figure 11: Trajectory taken by bounded coupled controller for smooth transition strategy with maximum desired tilt $\pi/8$ and $\pi/4$ and gains 5.0 and bound limit 0.1.

An animated simulation of these tests can be seen in the following link: `https://youtu.be/8rQxVsea90w`

4.5. Coupled controllers and helical trajectory

We compared our controller (31) to a controller proposed in [33], for which the natural convergence trajectory during pose-to-pose control was also claimed to be helical. The controller, which we will refer to as *Controller 1*, was defined as:

$$\dot{\hat{\xi}}_e = -2k(\eta_e\sigma_e + \varepsilon\eta_e\sigma'_e) \tag{46}$$

In the above equation, η_e, σ_e and σ'_e refer to the scalar component of the primary part, vector component of the primary part and vector component of the dual part, respectively, of the error UDQ. We compared the closeness of the trajectory taken by the controller (46) to ours with regards to an ideal screw trajectory in simulation. A new goal pose was defined using random screw-axis and random rotation and translation value. The initial screw axis related to the pose error and the completed rotation during the control were used to compute the translational part of the motion, which were then combined to get the ideal screw trajectory.

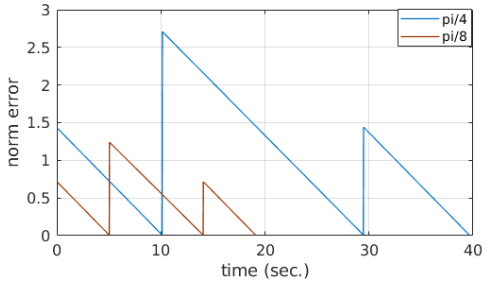

Figure 12: Pose error convergence plot for arbitrary pose control in terms of the norm of 6D screw twist for gains 1.0 with bounded coupled controller for smooth transition strategy with maximum desired tilt $\pi/8$ and $\pi/4$ and gains 5.0 and bound limit 0.1.

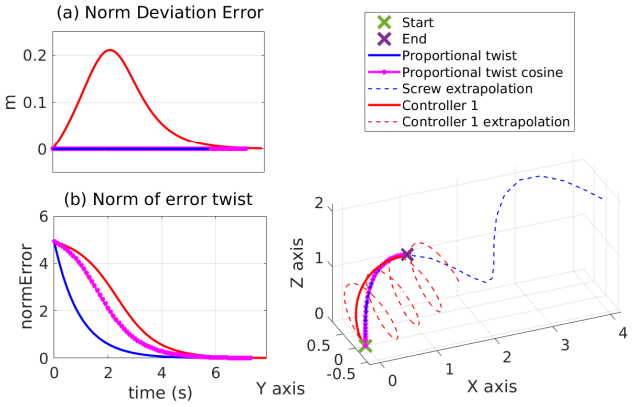

Figure 13: Comparison of proportional twist controller (31) with *Controller 1* (46) in terms of convergence and closeness to ideal screw trajectory (referred to as *Screw extrapolation*).

Fig. 13 shows the deviation from ideal screw trajectory in (a), evolution of the norm of screw-based error in (b) and, the convergence trajectory in (c), during pose-to-pose control for the above-mentioned controllers. The controller proposed in our work is referred to as the *Proportional twist* controller in the given figure. The deviation of the trajectory from the ideal helical trajectory was computed in terms of the norm of the position error. It is evident from Fig. 13 (a) and (c) that *Controller 1* deviated significantly from the ideal screw motion, while our proposed proportional twist controller trajectory almost coin-

cided with it. The reason for this deviation is the pitch of the helical trajectory imparted by the *Controller 1*, which is not equal to pitch related to the screw motion from initial to final pose. Therefore, *Controller 1* might require trajectory planning to achieve certain tasks mentioned in this paper.

One of the attractive properties of the controllers proposed in our work is that it can be easily modified to include the desired property from other controllers. *Controller 1*, for example, has a smooth start and end convergence as can be seen in Fig. 13 (c), owing to $\eta_e = cos(\theta_e/2)$ term in the controller. We tested a new controller, referred to as *Proportional twist cosine* controller in Fig. 13, where we added the cosine term:

$$\hat{\vec{\xi}}_e = -k\eta_e \left(\theta_e \vec{\ell}_e + \epsilon \left(\theta_e \vec{m}_e + d_e \vec{\ell}_e\right)\right) \tag{47}$$

It can be seen in Fig. 13 that the *Screw cosine* controller retains the ideal helical trajectory, while at the same has smooth convergence at the start and end of the motion.

5. Conclusion

This work provided a dual-quaternion-based pose controller for rigid bodies, applied to robotic manipulators. While similar controllers have been proposed before, the novelty lies in the usage of dual number and dual vector algebra. Additionally, the properties of the proposed coupled controllers were compared against a conventional decoupled controller that minimizes the translation and rotation error separately. It was observed that the decoupled controller has a tendency to drive the rigid body, or in our case the end-effector of a manipulator, in a straight line path during pose-to-pose control. On the other hand, the coupled controller takes a circular or screw path. These observations were verified with a real Franka Emika Panda robot, as well as in simulation, with randomly generated start and goal configurations within the workspace of the robot.

We also provided additional strategies to generate curved path along any desired axis during pose-to-pose control for coupled controller where there is no change desired in the goal orientation with regards to the starting configuration. Such strategies can be useful for achieving collision-free pick and place tasks, without having to generate and follow actual trajectory with time parametrization. A bounded controller proposed for the coupled control strategy ensures that the end-effector follows the screw path with bounded end-effector velocity. These strategies were verified in simulation.

References

[1] F. Bullo, R. M. Murray, Proportional derivative (pd) control on the euclidean group, in: European control conference, Vol. 2, 1995, pp. 1091–1097.

[2] E. Özgür, Y. Mezouar, Kinematic modeling and control of a robot arm using unit dual quaternions, Robotics and Autonomous Systems 77 (2016) 66–73.

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

[4] H.-L. Pham, V. Perdereau, B. V. Adorno, P. Fraisse, Position and orientation control of robot manipulators using dual quaternion feedback, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 658–663.

[5] H.-L. Pham, B. V. Adorno, V. Perdereau, P. Fraisse, Set-point control of robot end-effector pose using dual quaternion feedback, Robotics and Computer-Integrated Manufacturing 52 (2018) 100–110.

[6] J. F. Guerrero-Castellanos, N. Marchand, A. Hably, S. Lesecq, J. Delamare, Bounded attitude control of rigid bodies: Real-time experimentation to a quadrotor mini-helicopter, Control Engineering Practice 19 (8) (2011) 790–797.

[7] E. N. Hartley, P. A. Trodden, A. G. Richards, J. M. Maciejowski, Model predictive control system design and implementation for spacecraft rendezvous, Control Engineering Practice 20 (7) (2012) 695–713.

[8] M. H. Shahna, M. Abedi, Design of a finite time passivity based adaptive sliding mode control implementing on a spacecraft attitude dynamic simulator, Control Engineering Practice 114 (2021) 104866.

[9] D.-P. Han, Q. Wei, Z.-X. Li, Kinematic control of free rigid bodies using dual quaternions, International Journal of Automation and Computing 5 (3) (2008) 319–324.

[10] X. Wang, D. Han, C. Yu, Z. Zheng, The geometric structure of unit dual quaternion with application in kinematic control, Journal of Mathematical Analysis and Applications 389 (2) (2012) 1352–1364.

[11] K. Daniilidis, Hand-eye calibration using dual quaternions, The International Journal of Robotics Research 18 (3) (1999) 286–298.

[12] L. Kavan, S. Collins, J. Žára, C. O'Sullivan, Geometric skinning with approximate dual quaternion blending, ACM Transactions on Graphics (TOG) 27 (4) (2008) 1–23.

[13] Y. Kim, J. Han, Bulging-free dual quaternion skinning, Computer Animation and Virtual Worlds 25 (3-4) (2014) 321–329.

[14] B. H. Le, J. K. Hodgins, Real-time skeletal skinning with optimized centers of rotation, ACM Transactions on Graphics (TOG) 35 (4) (2016) 1–10.

[15] J. Dooley, J. M. McCarthy, Spatial rigid body dynamics using dual quaternion components, in: Proceedings. 1991 IEEE International Conference on Robotics and Automation, IEEE Computer Society, 1991, pp. 90–91.

[16] M. Lang, O. Dunkley, S. Hirche, Gaussian process kernels for rotations and 6d rigid body motions, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 5165–5170.

[17] H. Dong, Q. Hu, G. Ma, Dual-quaternion based fault-tolerant control for spacecraft formation flying with finite-time convergence, ISA transactions 61 (2016) 87–94.

[18] D. Seo, Fast adaptive pose tracking control for satellites via dual quaternion upon non-certainty equivalence principle, Acta Astronautica 115 (2015) 32–39.

[19] H. Abaunza, P. Castillo, A. Victorino, R. Lozano, Dual quaternion modeling and control of a quad-rotor aerial manipulator, Journal of Intelligent & Robotic Systems 88 (2-4) (2017) 267–283.

[20] Y. Wu, X. Hu, M. Wu, D. Hu, Error analysis of strapdown inertial navigation using dual quaternion algebra, in: PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No. 04CH37556), IEEE, 2004, pp. 259–267.

[21] D. Han, Q. Wei, Z. Li, W. Sun, Control of oriented mechanical systems: A method based on dual quaternion, IFAC Proceedings Volumes 41 (2) (2008) 3836–3841.

[22] H. Gui, G. Vukovich, Finite-time output-feedback position and attitude tracking of a rigid body, Automatica 74 (2016) 270–278.

[23] B. V. Adorno, M. M. Marinho, Dq robotics: A library for robot modeling and control, IEEE Robotics & Automation Magazine (2020).

[24] F. F. A. Silva, B. V. Adorno, Whole-body control of a mobile manipulator using feedback linearization and dual quaternion algebra, Journal of Intelligent & Robotic Systems 91 (2) (2018) 249–262.

[25] H. J. Savino, L. C. Pimenta, J. A. Shah, B. V. Adorno, Pose consensus based on dual quaternion algebra with application to decentralized formation control of mobile manipulators, Journal of the Franklin Institute 357 (1) (2020) 142–178.

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

[27] L. F. da Cruz Figueredo, B. V. Adorno, J. Y. Ishihara, Robust H−∞ kinematic control of manipulator robots using dual quaternion algebra, Automatica 132 (2021) 109817.

[28] M. M. Marinho, L. Figueredo, B. V. Adorno, A dual quaternion linear-quadratic optimal controller for trajectory tracking, in: 2015 IEEE/RSJ

International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 4047–4052.
[29] H. T. Kussaba, L. F. Figueredo, J. Y. Ishihara, B. V. Adorno, Hybrid kinematic control for rigid body pose stabilization using dual quaternions, Journal of the Franklin Institute 354 (7) (2017) 2769–2787.
[30] W. D. Price, C. Ton, W. MacKunis, S. V. Drakunov, Self-reconfigurable control for dual-quaternion/dual-vector systems, in: 2013 European Control Conference (ECC), IEEE, 2013, pp. 860–865.
[31] J. Wu, K. Liu, D. Han, Adaptive sliding mode control for six-dof relative motion of spacecraft with input constraint, Acta Astronautica 87 (2013) 64–76.
[32] C. D. Mladenova, Robotic problems over a configurational manifold of vector parameters and dual vector parameters, Journal of Intelligent and Robotic Systems 11 (1) (1994) 117–133.
[33] A. M. Sjøberg, O. Egeland, Kinematic feedback control using dual quaternions, in: 2018 26th Mediterranean Conference on Control and Automation (MED), IEEE, 2018, pp. 1–6.
[34] A. Sarker, A. Sinha, N. Chakraborty, On screw linear interpolation for point-to-point path planning, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 9480–9487.
[35] J. B. Kuipers, et al., Quaternions and rotation sequences, Vol. 66, Princeton university press Princeton, 1999.
[36] J. D. Biggs, N. Horri, Optimal geometric motion planning for a spin-stabilized spacecraft, Systems & Control Letters 61 (4) (2012) 609–616.
[37] K. W. Spring, Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: a review, Mechanism and machine theory 21 (5) (1986) 365–373.
[38] G. Veldkamp, On the use of dual numbers, vectors and matrices in instantaneous, spatial kinematics, Mechanism and Machine Theory 11 (2) (1976) 141–156.
[39] B. Jüttler, Visualization of moving objects using dual quaternion curves, Computers & Graphics 18 (3) (1994) 315–326.
[40] C. Huang, Notes on Screw Product Operations in the Formulations of Successive Finite Displacements, Journal of Mechanical Design 119 (4) (1997) 434–439.
[41] J. Selig, Note on the principle of transference, American Society of Mechanical Engineers (Paper) (1986).
[42] S. P. Bhat, D. S. Bernstein, A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon, Systems & control letters 39 (1) (2000) 63–70.
[43] V. Utkin, Variable structure systems with sliding modes, IEEE Transactions on Automatic control 22 (2) (1977) 212–222.
[44] R. Featherstone, Plucker basis vectors, in: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., IEEE, 2006, pp. 1892–1897.
[45] R. Chandra, C. M. Mateo, J. A. Corrales-Ramon, Y. Mezouar, Dual-arm coordination using dual quaternions and virtual mechanisms, in: 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2018, pp. 759–765.
[46] R. Smits, KDL: Kinematics and Dynamics Library, `http://www.orocos.org/kdl` (visited on april 2021).
[47] H. Bruyninckx, Open robot control software: the orocos project, in: Proceedings 2001 ICRA. IEEE international conference on robotics and automation (Cat. No. 01CH37164), Vol. 3, IEEE, 2001, pp. 2523–2528.
[48] K. J. Waldron, S.-L. Wang, S. J. Bolin, A Study of the Jacobian Matrix of Serial Manipulators, Journal of Mechanisms, Transmissions, and Automation in Design 107 (2) (1985) 230–237. `doi:10.1115/1.3258714`. URL `https://doi.org/10.1115/1.3258714`
[49] L.-W. Tsai, Robot analysis: the mechanics of serial and parallel manipulators, John Wiley & Sons, 1999.
[50] H. Bruyninckx, J. De Schutter, Symbolic differentiation of the velocity mapping for a serial kinematic chain, Mechanism and machine theory 31 (2) (1996) 135–148.