Resolved-Acceleration Control of Serial Robotic Manipulators Using Unit Dual Quaternions [★]

Rohit Chandra [*] Juan Antonio Corrales-Ramon [**]
Youcef Mezouar [**]

[*] *Université Clermont Auvergne, SIGMA Clermont, Institut Pascal, (e-mail: ol.chandra@gmail.com)*
[**] *Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000 Clermont–Ferrand, France, (e-mail: juan-antonio.corrales-ramon@sigma-clermont.fr, youcef.mezouar@sigma-clermont.fr)*

Abstract: A new method for resolved-acceleration control of serial chain manipulators has been proposed which uses dual quaternion representation of screw-based motion variables, i.e. pose, velocity and acceleration. The corresponding coupled controller considers both translation and orientation errors simultaneously for trajectory tracking and utilizes spatial acceleration to compute the feedforward compensation term for feedback linearization. The proposed coupled control law was validated on a robotic arm along a pre-defined trajectory. The controller demonstrated an improved trajectory tracking performance as compared to the conventional decoupled resolved-acceleration controller which treats translation and orientation error separately.

Keywords: Robots manipulators, motion control systems, torque control

1. INTRODUCTION

Robotic manipulators are often required to follow the desired trajectory composed of position as well as orientation set points, along with the desired profile of linear and angular velocities for tighter control and smooth motion. Due to intuitive ease of trajectory specification and the control of end-effector interaction with the environment, a task space controller is often preferred over a joint space controller. Operational space control (Khatib (1987), Nakanishi et al. (2008)) and resolved-acceleration control (Caccavale et al. (1998), Luh et al. (1980)) have been the two preferred methods for task space control in previous decades. They differ primarily in their treatment of singularity, apart from the dynamic formulation.

Traditionally the trajectory tracking strategies in the case of robotics manipulation involve the decoupling of position and orientation variables. However, it is desired to strive for a coupled controller to take into account the inherent effect of rotation on the translation of a rigid body as mentioned by Han et al. (2008). The basis of such control was given by Bullo and Murray (1995), along with a strategy to deal with coupled trajectory tracking problem for a rigid body using the homogeneous transformation matrix (HTM) representation of pose. The difference in the convergence behavior of coupled and decoupled control of linear and angular terms were reported in Han et al. (2008) for the motion on $SE(2)$ of a ground omnidirectional robot they concluded that the coupled treatment is a better choice when the synergy of position and orientation is important, i.e. for applications where the position and orientation setpoints are desired to be achieved simultaneously, for instance, robotic welding of a curved surface.

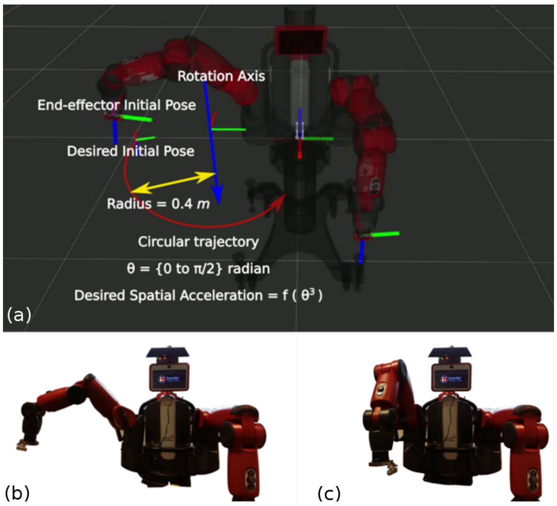

Fig. 1. (a) Desired trajectory description; (b, c) Baxter robot performing trajectory tracking task.

The coupled treatment of position and orientation variables using spatial dynamics concepts given by Feath-

[★] "The authors are supported by the H2020 FoF project SoftManBot. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 869855.

erstone (2010a,b) and compact representation of screw displacements using unit dual quaternions (UDQs) are the motivations for the development of the proposed new approach of UDQ based resolved-acceleration control. The interpretation of screw-based dynamics using spatial dynamics Featherstone (2010a,b, 2001) led to compact and more tractable formulations of kinematic and dynamic equations. The simplicity is evident from the formulations of the Jacobian and the derivative of the Jacobian for a kinematic chain as compared to the approach followed by Bruyninckx and De Schutter (1996).

The UDQ based approach for kinematic modeling of serial manipulator provides storage and computational advantage, as demonstrated by Özgür and Mezouar (2016). We also combine the approach given by Featherstone (2010a,b) for the computation of Jacobian derivative with the UDQ based kinematic formulation developed by Özgür and Mezouar (2016). The control law for task-space trajectory tracking by a robotic serial manipulator is based on the work of Wang and Yu (2013), where UDQ based error dynamics as derived from the UDQ based dynamics for a single rigid body for the attitude and position tracking. In our implementation, we derive the error dynamics of an end-effector in a desired frame attached to the base of a manipulator using a slightly different definition of the UDQ based pose error. Hence it allows for an intuitive representation of trajectory which is more prevalent in robotics literature, compared to the error represented in the body frame in Wang and Yu (2013).

The contributions of this paper can be summarized as:

- Extension of the kinematics developed by Özgür and Mezouar (2016) for the computation of the derivative of Jacobian for a serial manipulator using the theory of spatial dynamics given by Featherstone (2010b). The proposed approach for the computation of the Jacobian derivative is computationally advantageous.
- A screw-based resolved-acceleration controller using UDQ representation to compute the joint torque commands for the end-effector trajectory tracking. The coupled controller designed for a serial manipulator to follow the end-effector trajectory treats both orientation and position set-points jointly, thus addressing the inherent effect of rotational motion on translation motion.

In the following section (section 2) the theoretical background and the design of resolved-acceleration controller in explained. The derivation of UDQ error dynamics based control Law is given in section 3. The experimental validation and related discussions are given in section 4. Section 5 presents the conclusion and discussion about the current work.

2. THEORY AND PROBLEM FORMULATION

2.1 Resolved-Acceleration Control

The dynamic model of a serial manipulator consisting on n joints in joint space is given as:

$$\mathbf{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{G}(\boldsymbol{q}) = \boldsymbol{\Gamma} \tag{1}$$

where $\mathbf{H}(\boldsymbol{q})$ is a symmetric and positive definite inertia matrix. $\mathbf{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ consists of Coriolis and centrifugal terms, whereas $\mathbf{G}(\boldsymbol{q})$ represents gravity vector. $\boldsymbol{\Gamma}$ is a vector of joint driving torques. The joint positions, velocities and acceleration are represented above with $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$, respectively.

Assuming perfect dynamic compensation, the driving torques can be computed with a new control input as:

$$\boldsymbol{\Gamma} = \mathbf{H}(\boldsymbol{q})\hat{\mathbf{J}}^{-1}(\boldsymbol{q})(\hat{\boldsymbol{a}}_{cmd} - \dot{\hat{\mathbf{J}}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}) + \mathbf{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{G}(\boldsymbol{q}) \tag{2}$$

The $(\hat{\boldsymbol{a}}_{cmd} - \dot{\hat{\mathbf{J}}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}})$ terms actually replaces $\ddot{\boldsymbol{q}}$ in (1) using following equation relating the joint velocities to the end-effector velocity, and its derivative:

$$\hat{\boldsymbol{\omega}} = \hat{\mathbf{J}}\dot{\boldsymbol{q}} \tag{3}$$

$$\hat{\boldsymbol{a}} = \hat{\mathbf{J}} \cdot \underline{\ddot{\boldsymbol{q}}} + \dot{\hat{\mathbf{J}}} \cdot \underline{\dot{\boldsymbol{q}}} \tag{4}$$

The goal of resolved acceleration control is to find the acceleration command ($\hat{\boldsymbol{a}}_{cmd}$) for (2) from the desired trajectory consisting of desired acceleration, velocity, and position of end-effector. Note that the dual velocity and acceleration terms, i.e. $\hat{\boldsymbol{\omega}}$ and $\hat{\boldsymbol{a}}$, has different meaning in the context of this paper, and is explained in the next section.

2.2 Mathematical and Theoretical Preliminaries

In this section, the concepts required for subsequent formulation are given. It includes: the meaning of mathematical notations used; relevant background related to the dual-quaternions; the formulation of screw-based forward and inverse kinematics concepts proposed in Özgür and Mezouar (2016); and, concepts and formulations related to the spatial acceleration.

Notations: Dual quaternions and dual vectors have been notated with bold characters with hat, e.g. $\hat{\mathbf{x}}$ and $\hat{\mathbf{s}}$, respectively. Dual numbers have been named with variables with hats (e.g. $\hat{\theta}$), whereas dual number arrays have been notated with underlined bold characters with hats (e.g. $\underline{\hat{\boldsymbol{\theta}}}$). Arrays of real number variables are named using underlined bold characters (e.g. $\underline{\boldsymbol{\theta}}$) and vectors and quaternions are notated with bold characters (e.g. $\boldsymbol{l}$ and $\boldsymbol{q}$ respectively).

Quaternions: A Quaternion is a 4-tuple consisting of a scalar and a vector part and is defined as:

$$\boldsymbol{q} = s + \boldsymbol{v} = s + (v_1\hat{\mathbf{i}} + v_2\hat{\mathbf{j}} + v_3\hat{\mathbf{k}}) \tag{5}$$

A pure quaternion $\boldsymbol{q}$ has zero as the scalar component and is defined as $\boldsymbol{q} = q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k}$. Quaternion product is given as:

$$\boldsymbol{q}_1 \cdot \boldsymbol{q}_2 = (s_1s_2 - \boldsymbol{v_1}^T\boldsymbol{v_2}) + (s_1\boldsymbol{v_2} + s_2\boldsymbol{v_1} + \boldsymbol{v_1} \times \boldsymbol{v_2}) \tag{6}$$

Dual Quaternions: A dual quaternions is an 8-tuple and consists of quaternions as their real and dual parts:

$$\hat{\mathbf{x}} = \boldsymbol{q}_r + \varepsilon\boldsymbol{q}_d = (s_r + \boldsymbol{v}_r) + \varepsilon(s_d + \boldsymbol{v}_d) \tag{7}$$

where ε is the dual number element, such that $\varepsilon^2 = 0$, and $\varepsilon \neq 0$. The *classical* conjugate of dual quaternion, used for transforming poses, is given as:

$$\hat{\mathbf{x}}^* = \boldsymbol{q}_r^* + \varepsilon\boldsymbol{q}_d^* = s_r - \boldsymbol{v}_r + \varepsilon(s_d - \boldsymbol{v}_d) \tag{8}$$

Dual quaternion product is given as:

$$\hat{\mathbf{x}}_a \cdot \hat{\mathbf{x}}_b = (\boldsymbol{q}_{r1} + \varepsilon\boldsymbol{q}_{d1}) \cdot (\boldsymbol{q}_{r2} + \varepsilon\boldsymbol{q}_{d2}) \tag{9}$$

$$= \boldsymbol{q}_{r1} \cdot \boldsymbol{q}_{r2} + \varepsilon(\boldsymbol{q}_{r1} \cdot \boldsymbol{q}_{d2} + \boldsymbol{q}_{d1} \cdot \boldsymbol{q}_{r2}) \tag{10}$$

If the product of a dual quaternion with its *classical* conjugate has unity norm represented with a dual number, $\hat{\mathbf{x}} \cdot \hat{\mathbf{x}}^* = \|\hat{\mathbf{x}}\|^2 = 1 + \varepsilon 0$, it is defined as a unit dual quaternion.

An *adjoint* operation for dual quaternions (Wang and Yu (2013)) is defined as:

$$Ad_{({}^n\hat{\mathbf{x}}_m)}\hat{\mathbf{x}}_a = {}^n\hat{\mathbf{x}}_m \cdot \hat{\mathbf{x}}_a \cdot {}^n\hat{\mathbf{x}}_m^* \tag{11}$$

where ${}^n\hat{\mathbf{x}}_m^*$ is the classic dual quaternion conjugate of ${}^n\hat{\mathbf{x}}_m$. If $\hat{\mathbf{x}}_a$ was a directed line represented using a unit dual vector or a motion screw $\in M^6$ represented with dual vectors, and ${}^n\hat{\mathbf{x}}_m$ represented the transformation from frame m to frame n, the *adjoint* operation given in (11), will transform the corresponding element $\in M^6$ from frame m to n.

A $\widehat{\mathbf{vec}}_{\hat{\mathbf{x}}}$ operation is defined for *pure* dual quaternions (dual quaternions with zero scalar parts in both real and dual components) which extracts the vectors from the dual quaternion product if applied to another *pure* dual quaternion. This operation is similar to the cross product operation for spatial vectors with motion basis (*i.e.* vectors $\in M^6$). For instance, if $\hat{\mathbf{x}} = \boldsymbol{q}_r + \varepsilon \boldsymbol{q}_d$ was a spatial vector representing a dual vector $\in M^6$ (note that we have reduced the dimension from 8 to 6 for $\hat{\mathbf{x}}\times$ operation by eliminating the scalar part, for the sake of explanation), $\hat{\mathbf{x}}\times$ (*cross product* operation) is given by Featherstone (2010a) as:

$$\hat{\mathbf{x}}\times = \begin{pmatrix} \boldsymbol{q}_r\times & \mathbf{0} \\ \boldsymbol{q}_d\times & \boldsymbol{q}_r\times \end{pmatrix} \tag{12}$$

Pose of a frame is represented using UDQ as:

$$\hat{\mathbf{x}} = \exp\left(\frac{\hat{\theta}}{2} \cdot\ \hat{\mathbf{s}}\right) = \cos\left(\frac{\hat{\theta}}{2}\right) + \hat{\mathbf{s}}\ \sin\left(\frac{\hat{\theta}}{2}\right) \tag{13}$$

$$= \left(\cos\left(\frac{\theta}{2}\right) + \boldsymbol{l}\sin\left(\frac{\theta}{2}\right)\right) \tag{14}$$

$$+ \varepsilon\left(-\frac{d}{2}\sin\left(\frac{\theta}{2}\right) + \boldsymbol{l}\frac{d}{2}\cos\left(\frac{\theta}{2}\right) + \boldsymbol{m}\sin\left(\frac{\theta}{2}\right)\right) \tag{15}$$

where $\hat{\theta} \in \mathrm{D}$ is a dual angle and $\hat{\mathbf{s}} \in \mathrm{D}^{3\times 1}$ is a directed line represented using unit dual vector.

$$\hat{\theta} = \theta + \varepsilon\ d, \qquad \hat{\mathbf{s}} = \boldsymbol{l} + \varepsilon\ \mathbf{m} \tag{16}$$

where θ is a rotation around the screw axis, d is translation along the same axis, $\boldsymbol{l}$ is the unit direction vector of the screw axis, and $\mathbf{m}$ $(= \boldsymbol{p} \times \boldsymbol{l})$ is the moment of this screw axis with respect to the origin of the reference frame, and $\boldsymbol{p}$ is any point lying on the screw axis.

Serial Manipulator Kinematics: For a serial kinematic chain, let $\hat{\mathbf{x}}_{c_0}$ be the initial end-effector configuration, and $\underline{\hat{\boldsymbol{\theta}}}_0$ be the initial joint configuration such that:

$$\underline{\hat{\boldsymbol{\theta}}}_0 = \begin{bmatrix}\hat{\theta}_{1_0} & \hat{\theta}_{2_0} & \cdots & \hat{\theta}_{n_0}\end{bmatrix}^T \in \mathrm{D}^{n\times 1} \tag{17}$$

The end-effector pose $\hat{\mathbf{x}}_c$ corresponding to the new joint configuration $\underline{\hat{\boldsymbol{\theta}}}$ is obtained by multiplying the successive screw displacements caused by the intermediate joints between the base frame and the end-effector link, starting from the link to the base and ending at the joint closest to the end-effector.

$$\hat{\mathbf{x}}_c = \hat{\boldsymbol{\delta}}_1 \cdot \hat{\boldsymbol{\delta}}_2 \cdot \cdots \cdot \hat{\boldsymbol{\delta}}_n \cdot \hat{\mathbf{x}}_{c_0} \tag{18}$$

$$\hat{\boldsymbol{\delta}}_i = \exp\left(\frac{\hat{\theta}_i}{2} \cdot \hat{\mathbf{s}}_{i_0}\right)$$

where

$$\hat{\theta}_i = \Delta\theta_i \qquad \text{for revolute joints,}$$

$$\hat{\theta}_i = \varepsilon\ \Delta d_i \qquad \text{for prismatic joints.}$$

In (18), $\hat{\mathbf{s}}_{i_0}$ refers to the unit dual vector representing the initial configuration of the joint screw, and $\hat{\boldsymbol{\delta}}_i$ represents the screw displacement of link i relative to the link $(i-1)$ resulting from the motion of the previous joint.

The end-effector screw-velocity or spatial velocity (Featherstone (2010a)) represented as a pure dual quaternion, is obtained by adding the twist cause by the intermediate joints. The screw-based Jacobian is given as:

$$\hat{\boldsymbol{\omega}} = \begin{bmatrix}\boldsymbol{\omega} \\ \boldsymbol{v}_{\boldsymbol{c}_{/b0}}\end{bmatrix} = \hat{\mathbf{J}}\underline{\dot{\hat{\boldsymbol{\theta}}}} \tag{19}$$

$$= \begin{bmatrix}\hat{\mathbf{s}}_1 & \hat{\mathbf{s}}_2 & \cdots & \hat{\mathbf{s}}_n\end{bmatrix}\underline{\dot{\hat{\boldsymbol{\theta}}}}$$

where

$$\underline{\dot{\hat{\boldsymbol{\theta}}}} = \begin{bmatrix}\dot{\hat{\theta}}_1 & \dot{\hat{\theta}}_2 & \cdots & \dot{\hat{\theta}}_n\end{bmatrix}^T \in \mathrm{D}^{n\times 1}$$

In (19), $\boldsymbol{v}_{\boldsymbol{c}_{/b0}}$ represents the linear velocity of a point attached to the end-effector, which is instantaneously coincident with the origin of the base frame. It will be hereafter notated as $\boldsymbol{v}_{\mathbf{0}}$.

The current unit dual vector of joint screw $(\hat{\mathbf{s}}_i)$ for the i_{th} joint, can be obtained from its initial value $\hat{\mathbf{s}}_{i_0}$, by transforming it using the total displacement caused by the previous $(i-1)$ joints, using the *adjoint* operation.

$$\hat{\mathbf{s}}_i = Ad_{({}^b\hat{\boldsymbol{\delta}}_{T_{(i-1)}})}\hat{\mathbf{s}}_{i_0} \tag{20}$$

${}^b\hat{\boldsymbol{\delta}}_{T_{(i-1)}}$ can be obtained as:

$${}^b\hat{\boldsymbol{\delta}}_{T_{(i-1)}} = \hat{\boldsymbol{\delta}}_1 \cdot \hat{\boldsymbol{\delta}}_2 \cdot \cdots \cdot \hat{\boldsymbol{\delta}}_{i-1} \tag{21}$$

Note that in the case of the first joint, *i.e.* $\hat{\mathbf{s}}_1$, the dual vector of the corresponding joint screw is constant with respect to the reference frame. However, the first link closest to the base frame will move due to the motion of the first joint.

Spatial Acceleration: A brief discussion about spatial acceleration is provided here to relate our control approach to traditional controllers. All the terms are given in the base frame of the robot.

Spatial acceleration is the derivative of screw velocity and defines a helicoidal vector field Featherstone (2001).

$$\hat{\boldsymbol{a}} = \dot{\hat{\boldsymbol{\omega}}} = \begin{bmatrix}\dot{\boldsymbol{\omega}} \\ \dot{\boldsymbol{v}}_0\end{bmatrix} \tag{22}$$

The angular acceleration $\dot{\hat{\boldsymbol{\omega}}}$ of a body in *conventional* sense and spatial dynamics is the same. However, the linear acceleration of a point attached to the body is quite different in both paradigm.

Let $\hat{\boldsymbol{a}}_{c_{/b0}} = \dot{\boldsymbol{\omega}} + \varepsilon\ \boldsymbol{a}_{c_{/b0}}$ represents the *conventional* acceleration of a frame $c_{/b0}$ rigidly attached to the end-effector and instantaneously coincident with the origin of the base frame. The linear acceleration $\boldsymbol{a}_{c_{/b0}}$ is related to the linear part of spatial acceleration $(\dot{\boldsymbol{v}}_0)$ as :

$$\boldsymbol{a}_{c_{/b0}} = \dot{\boldsymbol{v}}_0 + \boldsymbol{\omega} \times \boldsymbol{v}_0 \tag{23}$$

Note that the dual part of $\hat{\boldsymbol{a}}_{c_{/b0}}$, *i.e.* $\boldsymbol{a}_{c_{/b0}}$ is not a derivative of $\boldsymbol{v}_{\boldsymbol{c}_{/\boldsymbol{b0}}}$ (*i.e.* $\boldsymbol{v}_{\boldsymbol{c_0}}$). In fact, $\boldsymbol{a}_{c_{/b0}}$ "refers to the acceleration of an individual body-fixed point at the moment when it happens to be passing through the origin" Featherstone (2001).

The *conventional* dual acceleration with the linear acceleration of a general point *i.e.* origin of a frame on the end-effector, say c, can be obtained from spatial acceleration ($\boldsymbol{a}_{c_{/b0}}$) as follows:

$$\hat{\boldsymbol{a}}_c = \begin{bmatrix} \dot{\boldsymbol{\omega}}_c \\ \boldsymbol{a}_c \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \boldsymbol{a}_{c_{/b0}} + \dot{\boldsymbol{\omega}} \times \overrightarrow{\mathrm{OC}} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \overrightarrow{\mathrm{OC}}) \end{bmatrix} \tag{24}$$

3. DERIVATION OF CONTROL LAW

3.1 UDQ based Pose Error

Given two poses, i.e. current ($\hat{\mathbf{x}}_c$) and desired ($\hat{\mathbf{x}}_c$), both expressed in a static base frame, there are four possible permutations for error computation. The error UDQ used in our formulation is:

$$\hat{\mathbf{x}}_e = \hat{\mathbf{x}}_d \cdot \hat{\mathbf{x}}_c^* = {}^b\hat{\mathbf{x}}_{c \to d} \tag{25}$$

The screw axis related this error choice represents a screw displacement vector directed from the current frame c to the desired frame d (expressed in the base frame b). It means if we transform the error screw axis derived from $\hat{\mathbf{x}}_e$, from the base frame to the current frame using $Ad_{({}^b\hat{\mathbf{x}}_c^*)}$:

$${}^c\hat{\mathbf{s}}_e = Ad_{({}^b\hat{\mathbf{x}}_c^*)}{}^b\hat{\mathbf{s}}_e = Ad_{({}^c\hat{\mathbf{x}}_b)}{}^b\hat{\mathbf{s}}_e \tag{26}$$

and then postmultiply the corresponding UDQ (obtained using (13)) to the current frame pose $\hat{\mathbf{x}}_c$, we will get the desired pose:

$$\hat{\mathbf{x}}_c \cdot \exp\left(\frac{{}^c\hat{\theta}_e}{2} \cdot \ {}^c\hat{\mathbf{s}}_e\right) = \hat{\mathbf{x}}_d \tag{27}$$

The same interpretation can also applied for obtaining orientation error in a desired frame when using unit quaternion representation.

3.2 Control Law Design:

In this section we derive the error dynamics similar to Wang and Yu (2013) for the attitude and position tracking problem of a rigid body using feedback linearization. Taking the derivative of the error UDQ given in (25) we obtain:

$$\dot{\hat{\mathbf{x}}}_e = \dot{\hat{\mathbf{x}}}_d \cdot \hat{\mathbf{x}}_c^* + \hat{\mathbf{x}}_d \cdot \dot{\hat{\mathbf{x}}}_c^* \tag{28}$$

Now, it has been proved in Han et al. (2008) that:

$$\dot{\hat{\mathbf{x}}} = \frac{1}{2}\hat{\boldsymbol{\omega}} \cdot \hat{\mathbf{x}} \tag{29}$$

where $\hat{\boldsymbol{\omega}}$ is the screw velocity of the body, whose pose is represented using $\hat{\mathbf{x}}$. The conjugate is given as:

$$\dot{\hat{\mathbf{x}}}^* = -\frac{1}{2}\hat{\mathbf{x}}^* \cdot \hat{\boldsymbol{\omega}} \tag{30}$$

Hence (28) can be rewritten as:

$$\dot{\hat{\mathbf{x}}}_e = \frac{1}{2}\hat{\mathbf{x}}_e \cdot \hat{\boldsymbol{\omega}}_e \tag{31}$$

where

$$\hat{\boldsymbol{\omega}}_e = Ad_{(\hat{\mathbf{x}}_e^*)}\hat{\boldsymbol{\omega}}_d - \hat{\boldsymbol{\omega}}_c \tag{32}$$

Taking the derivative of (32), we obtain:

$$\dot{\hat{\boldsymbol{\omega}}}_e = \dot{A}d_{(\hat{\mathbf{x}}_e^*)}\hat{\boldsymbol{\omega}}_d + Ad_{(\hat{\mathbf{x}}_e^*)}\dot{\hat{\boldsymbol{\omega}}}_d - \dot{\hat{\boldsymbol{\omega}}}_c \tag{33}$$

After expanding the involved terms and using the property of UDQs, we obtain:

$$\dot{\hat{\boldsymbol{\omega}}}_e = Ad_{(\hat{\mathbf{x}}_e^*)}\dot{\hat{\boldsymbol{\omega}}}_d - \dot{\hat{\boldsymbol{\omega}}}_c + \widehat{\mathbf{vec}}_{\hat{\boldsymbol{\omega}}_c}Ad_{(\hat{\mathbf{x}}_e^*)}\hat{\boldsymbol{\omega}}_d \tag{34}$$

Now $\dot{\hat{\boldsymbol{\omega}}}_c$ can be substituted with the following acceleration command ($\hat{\boldsymbol{a}}_{cmd}$) for feedback linearisation:

$$\begin{aligned} \hat{\boldsymbol{a}}_{cmd} = {} & 2\hat{\boldsymbol{K}}_p \cdot \ln\hat{\mathbf{x}}_e + \hat{\boldsymbol{K}}_v \cdot \hat{\boldsymbol{\omega}}_e + \widehat{\mathbf{vec}}_{\hat{\boldsymbol{\omega}}_c}Ad_{(\hat{\mathbf{x}}_e^*)}\hat{\boldsymbol{\omega}}_d \\ & + Ad_{(\hat{\mathbf{x}}_e^*)}\dot{\hat{\boldsymbol{\omega}}}_d \end{aligned} \tag{35}$$

Substituting $\hat{\boldsymbol{a}}_{cmd}$ in (34), we obtain the following error dynamics:

$$\dot{\hat{\boldsymbol{\omega}}}_e + \hat{\boldsymbol{K}}_v\hat{\boldsymbol{\omega}}_e + 2\,\hat{\boldsymbol{K}}_p \ln\hat{\mathbf{x}}_e = 0 \tag{36}$$

where the expression ($2\ln\hat{\mathbf{x}}_e$) refers to the product of dual angle and unit dual vector pertaining to the axis of the screw (see (13)),

$$\ln\hat{\mathbf{x}}_e = \frac{1}{2}\hat{\theta}_e \cdot \ \hat{\mathbf{s}}_e \tag{37}$$

The details of obtaining the screw parameters from a dual quaternion can be found in Özgür and Mezouar (2016). The asymptotic stability of equilibrium point $(\ln(\hat{\mathbf{x}}_e), \hat{\boldsymbol{\omega}}_e) = (\hat{\mathbf{0}}, \hat{\mathbf{0}})$ for the above system has been proven in Wang and Yu (2013) for an appropriate choice of the gains $\hat{\boldsymbol{K}}_p$ and $\hat{\boldsymbol{K}}_v$. Therefore, the end-effector motion will eventually converge to the desired trajectory for the control command obtained in (35). The *two equilibria problem* for dual quaternion has been discussed in Wang and Yu (2013), where the system (36) has two identical equilibria at $\hat{\mathbf{x}}_e = (\hat{I}.\hat{\mathbf{0}})and(-\hat{I}.\hat{\mathbf{0}})$, which was resolved by multiplying the error UDQ with -1 to make the scalar part of the real quaternion positive.

The $\hat{\boldsymbol{a}}_{cmd}$ obtained in (35) can be used in (2) to compute the joint torque commands for the manipulator. Note that the dual angle $\hat{\theta}_i$ used in (19) is converted to q_i in (4), depending on the type of joint (see (17)).

3.3 Formulation of Jacobian Derivative:

In order to compute the driving torque command in (2), from the control law $\hat{\boldsymbol{a}}_{cmd}$ obtained in (35), the computation of Jacobian derivative ($\dot{\hat{\mathbf{J}}}$) is required. Derivative of the Jacobian of a serial robotic manipulator can be obtained as in Featherstone (2010b), assuming a joint screw axis $\hat{\mathbf{s}}_i$ given as a dual vector is fixed on the child link i, and noting that $\hat{\mathbf{s}}_i \in M^6$:

$$\begin{aligned} \dot{\hat{\mathbf{J}}} &= \begin{bmatrix} \dot{\hat{\mathbf{s}}}_1 & \dot{\hat{\mathbf{s}}}_2 & \cdots & \dot{\hat{\mathbf{s}}}_n \end{bmatrix} \\ &= \begin{bmatrix} \widehat{\mathbf{vec}}_{\hat{\boldsymbol{\omega}}_1}\hat{\mathbf{s}}_1 & \widehat{\mathbf{vec}}_{\hat{\boldsymbol{\omega}}_2}\hat{\mathbf{s}}_2 & \cdots & \widehat{\mathbf{vec}}_{\hat{\boldsymbol{\omega}}_n}\hat{\mathbf{s}}_n \end{bmatrix} \end{aligned} \tag{38}$$

where $\hat{\boldsymbol{\omega}}_{\mathbf{i}}$ represents the screw velocity of an i_{th} link, and can be computed from the twists effected by the joints between the base frame and the i_{th} link (see (19)).

4. EXPERIMENTAL VALIDATION

The controller obtained in section 3.2 was validated in on one of the redundant arms of Baxter dual arm collabo-

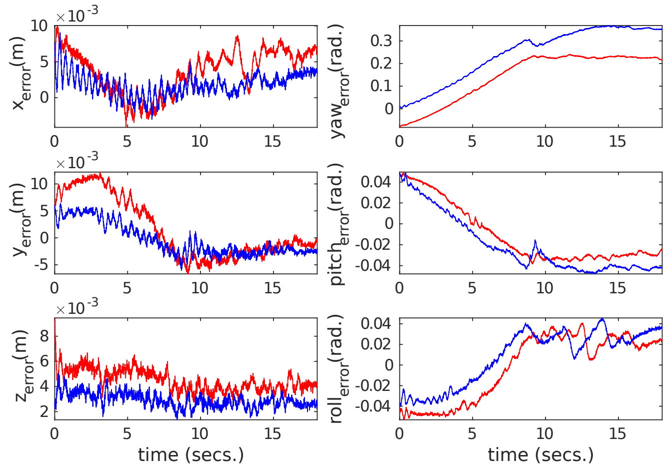

Fig. 2. Pose error (*Desired* – *Current*) plot for the coupled (——) and decoupled (——) controller with optimally tuned gains.

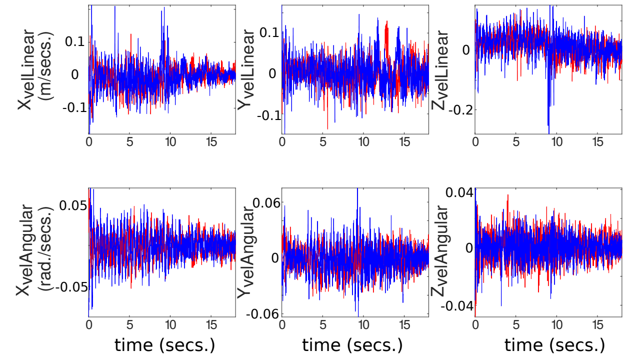

Fig. 3. Velocity error (*Desired* – *Current*) plot for the coupled (——) and decoupled (——) controllers with optimally tuned gains.

rative robot (Robotics (2017)). The joint inertia matrix, Coriolis and centrifugal torques, and the gravity torques were obtained using KDL library Smits et al. (2011), and $\dot{\boldsymbol{q}}$ was provided by the Baxter ROS interface. The presented implementation does not address redundancy resolution for the extra joint in the used robotic arm, and the starting configuration of the manipulator was carefully chosen to avoid the effects of null space motion on the tracking task. The complete implementation along with the instructions to repeat the experiments and reproduce the results discussed in this section can be obtained from the source [1] .

Table 1. Root mean square error (RMSE) and standard deviation (StD) of position (*Pos*) and orientation (*Orient*) errors for the coupled controller with optimal gain, and decoupled controller with the same and optimal gain.

Controller	Coupled	Decoup$_{same}$	Decoup$_{opt}$
RMSE$_{Orient}$ (rad.)	0.1852	0.3436	0.2748
StD$_{Orient}$ (rad.)	0.0718	0.1553	0.1119
RMSE$_{Pos}$ (m)	0.0089	0.0066	0.0049
StD$_{Pos}$ (m)	0.0026	0.0021	0.0013

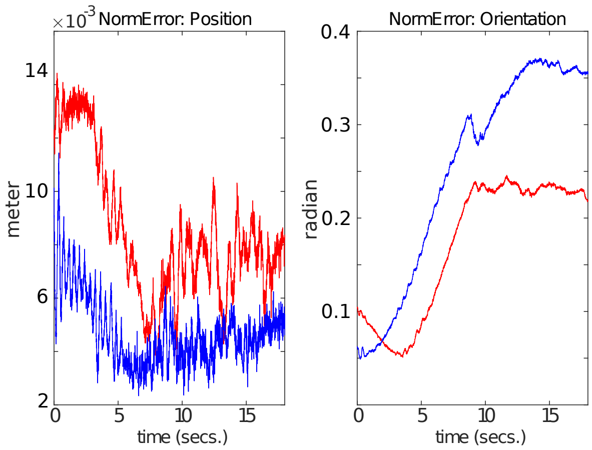

Fig. 4. Position and orientation norm error of for the coupled (——) and decoupled (——) controllers with optimally tuned gains.

The controller performance was compared with the control law used in Caccavale et al. (1998), where the translation and orientation components of the trajectory were treated separately. The Jacobian of the manipulator and the derivative of this Jacobian for the quaternion based controller were obtained from the KDL library since the corresponding terms obtained in (19) and (38) are screw-based.

The Cartesian position and quaternion based control law was used for comparative validation of the proposed controller was given as follows in Caccavale et al. (1998):

$$\mathbf{a}_c = \ddot{\mathbf{p}}_d + \mathbf{K}_{V_p}\dot{\mathbf{p}}_e + \mathbf{K}_{P_p}\mathbf{p}_e$$
$$\dot{\boldsymbol{\omega}}_c = \boldsymbol{\omega}_d + \mathbf{K}_{V_o}\boldsymbol{\omega}_e + \mathbf{K}_{P_o}\mathbf{v}_e \qquad (39)$$

where $\mathbf{p}_e$ refers to the position error of the end-effector frame for a given desired position $\mathbf{p}_d$. $\mathbf{v}_e$ refers to the vector part of the error quaternion computed as $\mathbf{q}_e = \mathbf{q}_d \cdot \mathbf{q}_c^*$, from which the orientation error in the base frame can be derived. $\mathbf{a}_c$ and $\dot{\boldsymbol{\omega}}_c$ are used to obtain the combined acceleration command similar to $\hat{\boldsymbol{a}}_{cmd}$, to be used in (2).

Both the controllers were tuned for a stable profile in velocity and to obtain the best tracking performance. The proportional and derivative gain, 190 and 6 respectively, were given in a diagonal matrix form in (35) for the coupled controller using UDQ. The gains for the decoupled controller for the position and orientation error were $\mathbf{K}_{P_p} = \mathbf{K}_{P_o} = 270$ and for the linear and angular velocity error, $\mathbf{K}_{V_p} = \mathbf{K}_{V_o} = 7.5$ in (39), and the control loop frequency for both the controllers was set 200 *Hz*. Additionally, the conventional decoupled controller was also tested with the same gain as the one chosen for the proposed coupled controller for fair assessment of their performance.

The description of trajectory generation for validation of the controller has been given in Fig. 1. The end-effector was desired to rotate around a pre-selected line, defined the base frame of the robot, while keeping one of its frame axes always pointing towards the line, thus requiring both translation and orientation control. The screw axis corresponding to the line, and taking the rotation ($\theta_{traj}(t)$) as a cubic function of time were used to obtain the

[1] https://github.com/rohitChan/ifacWC2020

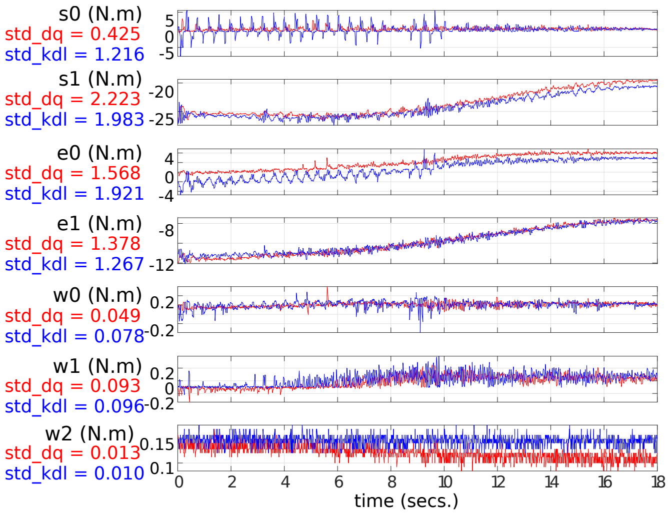

Fig. 5. Joint Effort plot for the coupled (——) and decoupled (——) controllers. *std_dq* and *std_kdl* refers to the standard deviation of the joint efforts for coupled approach and decoupled approach, respectively.

desired trajectory of the end-effector. The coefficients of the polynomial were computed with four boundary condition: $\theta_{trajInit} = 0$, $\theta_{trajFinal} = \pi/2$, $\dot{\theta}_{trajInit} = 0$, $\dot{\theta}_{trajFinal} = \pi/2$. The $\ddot{\theta}_{traj}(t)$, $\dot{\theta}_{traj}(t)$ and $\theta_{traj}(t)$ thus computed, gave spatial acceleration, spatial velocity and pose, when multiplied by the given screw axis. The total duration of the experiment was 18 seconds.

The robot performing the trajectory tracking task can be seen in Fig. 1, and performance of both the controller for pose and velocity tracking have been given in Fig. 2 and 3, respectively. The evolution of position and orientation error norm is given in Fig. 4. The error norm for the position and orientation tracking for both the controllers have been summarized in Table 1. The joint efforts given as an input the joints of the manipulator is given in Fig. 5.

While the coupled controller performance for position tracking is slightly worse than the traditional decoupled controller for both the same and optimal gains, the coupled controller performed better in terms of orientation tracking. The performance is identical in terms of velocity error as is shown in Fig. 3. The commanded joint torques were also close in terms of magnitude, however higher oscillations in the commanded joint torques can be observed in Fig. 5 for decoupled controller for joints *s0*, *e0* and *w0*.

5. CONCLUSION

A new controller for resolved acceleration control of robotic manipulators was proposed for the trajectory tracking of the serial robotic manipulator using screw theory and concepts from spatial dynamics. Representation of the motion variables with dual quaternions allowed coupled treatment of translation and orientation components of trajectory tracking error. A comparison with a decoupled controller reveals better orientation tracking while achieving comparable performance for the translation components. However, a more thorough analysis is required to formulate a better control law that is appropriate to keep the joint torque bounded for this kind of dynamics.

In addition to that, redundancy resolution is needed to utilize the additional degrees of freedom for redundant manipulators, as during the current implementation special attention was taken during the definition of trajectory.

REFERENCES

Bruyninckx, H. and De Schutter, J. (1996). Symbolic differentiation of the velocity mapping for a serial kinematic chain. *Mechanism and machine theory*, 31(2), 135–148.

Bullo, F. and Murray, R.M. (1995). Proportional derivative (pd) control on the euclidean group. In *European Control Conference*, volume 2, 1091–1097.

Caccavale, F., Natale, C., Siciliano, B., and Villani, L. (1998). Resolved-acceleration control of robot manipulators: A critical review with experiments. *Robotica*, 16(5), 565–573.

Featherstone, R. (2001). The acceleration vector of a rigid body. *The International Journal of Robotics Research*, 20(11), 841–846.

Featherstone, R. (2010a). A beginner's guide to 6-d vectors (part 1). *IEEE robotics & automation magazine*, 17(3), 83–94.

Featherstone, R. (2010b). A beginner's guide to 6-d vectors (part 2)[tutorial]. *IEEE robotics & automation magazine*, 17(4), 88–99.

Featherstone, R. (2014). *Rigid body dynamics algorithms.* Springer.

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

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1), 43–53.

Luh, J., Walker, M., and Paul, R. (1980). Resolved-acceleration control of mechanical manipulators. *IEEE Transactions on Automatic Control*, 25(3), 468–474.

Nakanishi, J., Cory, R., Mistry, M., Peters, J., and Schaal, S. (2008). Operational space control: A theoretical and empirical comparison. *The International Journal of Robotics Research*, 27(6), 737–757.

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

Robotics, R. (2017). Baxter collaborative robots for industrial automation.

Smits, R., Bruyninckx, H., and Aertbeliën, E. (2011). Kdl: Kinematics and dynamics library. *Avaliable: http://www. orocos. org/kdl.*

Wang, X. and Yu, C. (2013). Unit dual quaternion-based feedback linearization tracking problem for attitude and position dynamics. *Systems & Control Letters*, 62(3), 225–233.

Wang, X. and Zhu, H. (2014). On the comparisons of unit dual quaternion and homogeneous transformation matrix. *Advances in Applied Clifford Algebras*, 24(1), 213–229.