

Position and Orientation Control of Robot Manipulators Using Dual Quaternion Feedback

Hoang-Lan Pham, Véronique Perdereau, Bruno Vilhena Adorno, Philippe
Fraisie

► To cite this version:

Hoang-Lan Pham, Véronique Perdereau, Bruno Vilhena Adorno, Philippe Fraisie. Position and Orientation Control of Robot Manipulators Using Dual Quaternion Feedback. IROS: Intelligent Robots and Systems, Oct 2010, Taipei, Taiwan. IEEE/RSJ, pp.658-663, 2010, <<http://www.iros2010.org.tw/>>. <10.1109/IROS.2010.5651097>. <lirmm-00535893>

HAL Id: lirmm-00535893

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00535893>

Submitted on 13 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Position and Orientation Control of Robot Manipulators Using Dual Quaternion Feedback

Hoang-Lan Pham[†], Véronique Perdereau[†], Bruno Vilhena Adorno[‡] and Philippe Fraisse[‡]

[†] UPMC Univ Paris 06, UMR 7222, F-75005,
Paris, FRANCE

{hoang-lan.pham, veronique.perdereau}@upmc.fr

[‡] Université Montpellier II, UMR 5506, LIRMM,
Montpellier, FRANCE

{adorno, fraisse}@lirmm.fr

Abstract— We propose in this paper a new concept of unified position/orientation control of robot manipulator by describing the end-effector motion as a dual quaternion involving both translation and rotation. The development of the forward kinematic model and Jacobian matrix in dual quaternion space is detailed as well as the stability of the controller. At last, simulation and experimental results highlight the efficiency and performance of this controller.

I. INTRODUCTION

Many robotic applications involve not only pick and place operations but also grasping and fine manipulation of any kind of objects. The task is then described by means of end-effector motion which should be efficiently controlled. In the case of a robot manipulator, the gripper or possibly the hand attached to the end-effector can be viewed as an actuator manoeuvring the position and the orientation of the object. As a result, the desired task can be described by a desired position and orientation of the coordinate frame attached to the robot end-effector with respect to the base frame (i.e. the task-space variables). Control of the end-effector motion is commonly performed using feedback of either the joint variables or the task-space variables. The joint-based control requires the solution of the inverse kinematics to convert the desired task-space trajectory into the desired joint trajectory, but the inverse model is often difficult to obtain. In contrast, task-space control does not need the inverse kinematics. However, the precise and stable control of the end-effector position and orientation must be ensured.

The position and orientation can be parametrized in different ways, such as with the well-known 4×4 homogeneous transformation matrix, Euler angles [6], unit quaternion [9], [10] or dual quaternion [3], [8]. In an homogeneous transformation matrix, twelve parameters are used to represent the position and orientation of a body. When the unit quaternion is adopted for the rotation, a seven-dimensional vector must be defined. Only the Euler angles and any kind of three-element orientation vectors can provide a six-dimensional vector as a unified representation of position and orientation. But a 4×4 matrix is still to be used for deriving the forward kinematics and extracting the unit quaternion or the Euler angles.

Moreover, all types of Euler angles have a "rotation-in-sequence" nature. Thus, the Euler angle method is suitable for representing a single frame orientation but not for rep-

resenting orientation paths in the case of trajectory tracking. Another critical issue for the Euler angles is that they suffer from representation singularities. The unit quaternion represents the end-effector orientation without singularities but complicates the controller design since the position and orientation errors are separately calculated [9], [10].

Indeed, position control schemes are usually made of two control loops, for controlling the rotation and the translation, respectively [7]. The angular velocity is typically obtained through the approximate differentiation of some orientation representation, provided that it is differentiable.

Despite the eight-dimension of the dual quaternion, some authors [1], [4] stated that the dual quaternion is the most compact and efficient way to express the screw motion, that is, both translational and rotational transformations in a robot kinematic chain. The dual quaternion turns out to be an elegant and useful tool for kinematic analysis in many researches such as in inertial navigation [8] and computer vision [3].

In this paper, we use the dual quaternion as the basis for a position control scheme of a manipulator robot. The error between desired and actual position/orientation is simply expressed as a difference between both corresponding dual quaternions, which widely facilitates the error computation in comparison with classical methods. Considering main position control strategies [7], we have built two different control schemes - with pseudo-inverse Jacobian and Jacobian transpose, respectively - allowing position and orientation control in only one control loop, hereafter called dual position control strategy. The global asymptotic stability is proven for each.

In order to test the validity of the computation methods and to evaluate the efficiency of the control strategies, a 6-axis Adept Viper robot is used in simulated and real conditions.

This paper is organized as follows: Section II presents the mathematical background that will be useful for the remaining of the paper. Two methods for dual position control are proposed in Section III. Section IV describe a systematic way for finding the forward kinematic model and the Jacobian matrix in the dual quaternion space. Then, to validate our technique based on dual quaternions, a variety of experiments with computer simulations and a real robot are carried out and reported in Section V. Finally, Section VI presents the conclusions and proposes future works.

II. MATHEMATICAL BACKGROUND

To begin with, we recall in this section some necessary definitions on quaternion and dual quaternion. More detailed treatments on these mathematical tools can be found in previous research works [8]. First, quaternion is explained, followed by a short description of dual number. Finally, the dual quaternion and its relevant properties are introduced.

A. Quaternion

The quaternion, introduced by Hamilton in 1843 [5], is an extension of the complex numbers to a four dimensional manifold. There are two formalisms to define a quaternion. The classical one is:

$$\mathbf{q} = a_0 + a_1i + a_2j + a_3k \quad (1)$$

where $a_0, a_1, a_2, a_3 \in \mathbb{R}$ and the three imaginary components i, j, k are defined as:

$$i^2 = j^2 = k^2 = -1 \quad (2)$$

and have the following properties:

$$ij = k, jk = i, ki = j, ji = -k, kj = -i, ik = -j \quad (3)$$

The other definition of a quaternion is as a pair (s, \vec{v}) , where $s \in \mathbb{R}$ is a scalar and $\vec{v} \in \mathbb{R}^3$. By definition, a quaternion satisfies the following property:

$$\lambda(s, \vec{v}) = (\lambda s, \lambda \vec{v}) \quad (4)$$

where λ is a scalar. and the addition and multiplication between two quaternions \mathbf{q}_1 and \mathbf{q}_2 are respectively defined as:

$$\mathbf{q}_1 + \mathbf{q}_2 = (s_1 + s_2, \vec{v}_1 + \vec{v}_2) \quad (5)$$

$$\mathbf{q}_1 \mathbf{q}_2 = (s_1 s_2 - \vec{v}_1 \vec{v}_2, s_1 \vec{v}_2 + s_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2) \quad (6)$$

Conjugation and norm of a quaternion are given by:

$$\mathbf{q}^* = (s, -\vec{v}) \quad (7)$$

$$\|\mathbf{q}\| = \mathbf{q} \mathbf{q}^* \quad (8)$$

If $\|\mathbf{q}\| = 1$, \mathbf{q} is a unit quaternion. The unit quaternion can be used to represent a rotation of an angle θ about a unit vector \vec{n} in the three-dimensional Euclidean space as [5]:

$$\mathbf{q} = (\cos(\theta/2), \vec{n} \sin(\theta/2)) \quad (9)$$

B. Dual number

The Dual numbers were introduced by Clifford [2] and are defined as:

$$\hat{z} = a + \varepsilon a' \text{ with } \varepsilon^2 = 0 \text{ but } \varepsilon \neq 0 \quad (10)$$

where a is the real part and a' the dual part.

The operator ε can be used to represent any dual quantity like dual vector, dual quaternion, etc. A useful concept concerns dual vectors with orthogonal real and dual parts. Such particular dual vectors represent lines in \mathbb{R}^3 known as Plücker coordinates (or Plücker line). The real part \vec{l} is the direction of the line and the dual part $\vec{m} = \vec{p} \times \vec{l}$ is its moment, as illustrated in Fig. 1. The inner product between

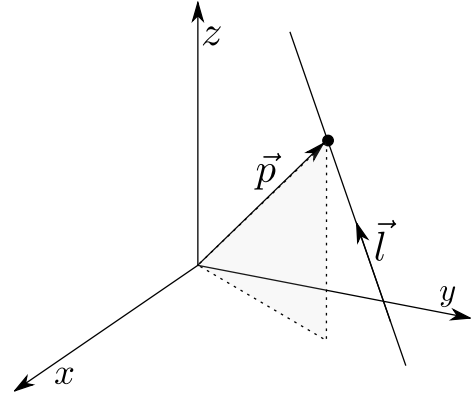


Fig. 1: Geometrical representation of the Plücker line.

two such dual vectors is equal to the cosine of a dual angle $\hat{\theta} = \theta + \varepsilon d$, which has a nice geometric interpretation: θ is the crossing angle between the two space lines and d their common perpendicular distance.

C. Dual quaternion

The dual quaternion is defined as a dual number with quaternion components:

$$\underline{\mathbf{q}} = \mathbf{q} + \varepsilon \mathbf{q}' \quad (11)$$

where \mathbf{q} and \mathbf{q}' are quaternions. Alternatively, the dual quaternions can be mapped in a vector space as

$$\underline{\mathbf{q}} = [q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8]^T \quad (12)$$

The usual algebraic quaternion operations, with the addition of the operator ε , can be applied to the dual quaternions, that is:

$$\lambda \underline{\mathbf{q}} = \lambda \mathbf{q} + \varepsilon \lambda \mathbf{q}' \quad (13)$$

$$\underline{\mathbf{q}}_1 + \underline{\mathbf{q}}_2 = \mathbf{q}_1 + \mathbf{q}_2 + \varepsilon (\mathbf{q}'_1 + \mathbf{q}'_2) \quad (14)$$

$$\underline{\mathbf{q}}_1 \underline{\mathbf{q}}_2 = \mathbf{q}_1 \mathbf{q}_2 + \varepsilon (\mathbf{q}_1 \mathbf{q}'_2 + \mathbf{q}'_1 \mathbf{q}_2) \quad (15)$$

The norm of a dual quaternion is given by $\|\underline{\mathbf{q}}\|^2 = \underline{\mathbf{q}} \underline{\mathbf{q}}^*$ with $\underline{\mathbf{q}}^* = \underline{\mathbf{q}}^* + \varepsilon \underline{\mathbf{q}}'^*$ and the unity condition becomes:

$$\mathbf{q} \mathbf{q}^* = 1 \text{ and } \mathbf{q}^* \mathbf{q}' + \mathbf{q}'^* \mathbf{q} = 0 \quad (16)$$

Unit dual quaternions can be used to represent any rigid transformation including translation and rotation, similarly to the way unit quaternions can be used to represent rotations. It can be shown [3], [8] that the rigid transformation of a line through the point \vec{p} , represented by its direction \vec{l} and moment $\vec{m} = \vec{p} \times \vec{l}$, is given by:

$$\underline{\mathbf{q}}(1 + \varepsilon \mathbf{m})\underline{\mathbf{q}}^* \quad (17)$$

where the dual quaternion $\underline{\mathbf{q}} = \mathbf{q} + \varepsilon \mathbf{q}'$ with $\mathbf{q}' = \frac{1}{2} \mathbf{t} \mathbf{q}$, \mathbf{q} is the unit quaternion describing the rotation and $\mathbf{t} = (0, \vec{t})$ is the quaternion describing the translation represented by the vector \vec{t} , and \mathbf{n} and \mathbf{m} are quaternions defined as $\mathbf{l} = (0, \vec{l})$ and $\mathbf{m} = (0, \vec{m})$, respectively.

Therefore, the unit dual quaternion is a useful tool to represent the transformation between two frames \mathcal{F}_1 and \mathcal{F}_2 resulting from either a rotation \mathbf{q} succeeded by a translation \mathbf{t}_2 with respect to \mathcal{F}_2 , that is,

$$\underline{\mathbf{q}} = \mathbf{q} + \varepsilon \frac{1}{2} \mathbf{q} \mathbf{t}_2 \quad (18)$$

or a translation \mathbf{t}_1 with respect to \mathcal{F}_1 succeeded by a rotation \mathbf{q} , *i.e.*,

$$\underline{\mathbf{q}} = \mathbf{q} + \varepsilon \frac{1}{2} \mathbf{t}_1 \mathbf{q} \quad (19)$$

III. DUAL POSITION CONTROL SCHEME

Since a dual quaternion is a compact representation for the position and the orientation, we can easily define the desired position and orientation of the robot end-effector with respect to the base frame by a unit dual quaternion as follows:

$$\underline{\mathbf{q}}_d = (\underline{\mathbf{q}}_d \quad \underline{\mathbf{q}}_d')^T \quad (20)$$

where the unit quaternion $\underline{\mathbf{q}}_d$ represents the desired orientation and the quaternion $\underline{\mathbf{q}}_d'$ representing the desired translation is calculated according to (19).

When using joint based position controllers, a natural metric for the error calculation arises as being the vector difference between the desired input vector $\underline{\theta}_d$ and the measured vector $\underline{\theta}$. And the aim of the position control loop is to make this error converge to zero. For this to happen, two different position control schemes are commonly used, based on the pseudo-inverse Jacobian matrix in one hand and on the Jacobian transpose matrix in another hand [7].

Let us consider the dual quaternion error $\underline{\mathbf{q}}_e$ as being the difference between the desired and the actual dual quaternions:

$$\underline{\mathbf{q}}_e = \underline{\mathbf{q}}_d - \underline{\mathbf{q}} \quad (21)$$

The aim of the dual position control loop presented in this paper is to make this dual quaternion error converge to zero.

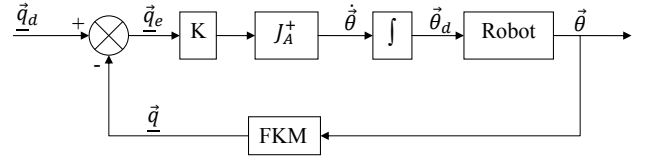
This dual quaternion error $\underline{\mathbf{q}}_e$ turns out to be an arbitrary vector in a \mathbb{R}^8 manifold. Although this vector does not satisfy the constraints of (16), it can be used as a metric for the error in a joint based position scheme provided that an appropriate analytical Jacobian matrix $\mathbf{J}_A(\underline{\mathbf{q}})$ is derived in the dual quaternion space of the manipulator, relating the joint velocity vector $\dot{\underline{\theta}}$ and the time derivative of the dual quaternion as:

$$\dot{\underline{\mathbf{q}}} = \mathbf{J}_A(\underline{\mathbf{q}}) \dot{\underline{\theta}} \quad (22)$$

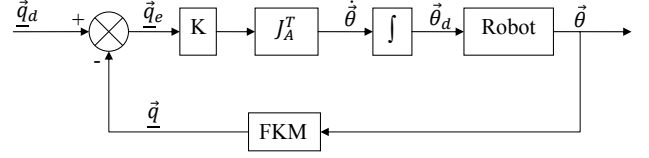
The usual position control schemes are modified accordingly to (22) and are represented in Fig. 2a and Fig. 2b, respectively.

Let us prove in the following that both solutions ensure the convergence of the dual quaternion error to zero. The time derivation of (21)

$$\dot{\underline{\mathbf{q}}}_e = \dot{\underline{\mathbf{q}}}_d - \dot{\underline{\mathbf{q}}} \quad (23)$$



(a) Dual position control scheme using the Jacobian pseudo-inverse



(b) Dual position control scheme using the Jacobian transpose

Fig. 2: Dual position control strategies.

leads to

$$\dot{\underline{\mathbf{q}}}_e = -\mathbf{J}_A(\underline{\mathbf{q}}) \dot{\underline{\theta}} \quad (24)$$

when the reference $\underline{\mathbf{q}}_d$ is constant. In the case of the pseudo-inverse Jacobian scheme (Fig. 2a), the choice $\dot{\underline{\theta}} = \mathbf{J}_A^+ \mathbf{K} \underline{\mathbf{q}}_e$ leads to the equivalent linear system

$$\dot{\underline{\mathbf{q}}}_e + \mathbf{K} \underline{\mathbf{q}}_e = 0 \quad (25)$$

If \mathbf{K} is a positive definite (usually diagonal) matrix, the error tends to zero with a convergence rate that depends on the eigenvalues of matrix \mathbf{K} ; the larger are the eigenvalues, the faster is the convergence. The system is asymptotically stable.

In the case of the Jacobian transpose scheme (Fig. 2b), let us choose as Lyapunov candidate the positive definite quadratic function:

$$V(\underline{\mathbf{q}}_e) = \frac{1}{2} \underline{\mathbf{q}}_e^T \mathbf{K} \underline{\mathbf{q}}_e \quad (26)$$

where \mathbf{K} is assumed to be a symmetric matrix.

Differentiating (26) with respect to time gives:

$$\dot{V} = \frac{1}{2} \dot{\underline{\mathbf{q}}}_e^T \mathbf{K} \underline{\mathbf{q}}_e + \frac{1}{2} \underline{\mathbf{q}}_e^T \mathbf{K} \dot{\underline{\mathbf{q}}}_e \quad (27)$$

$$\stackrel{(24)}{=} -\frac{1}{2} \dot{\underline{\theta}}^T \mathbf{J}_A^T \mathbf{K} \underline{\mathbf{q}}_e - \frac{1}{2} \underline{\mathbf{q}}_e^T \mathbf{K} \mathbf{J}_A \dot{\underline{\theta}} \quad (28)$$

Since \mathbf{K} is a symmetric matrix, the choice $\dot{\underline{\theta}} = \mathbf{J}_A^T \mathbf{K} \underline{\mathbf{q}}_e$ leads to

$$\dot{V} = -\underline{\mathbf{q}}_e^T \mathbf{K} \mathbf{J}_A \mathbf{J}_A^T \mathbf{K} \underline{\mathbf{q}}_e \quad (29)$$

If \mathbf{K} is positive definite, the derivative of the Lyapunov function obtained in (29) is negative. Hence, the system is asymptotically stable.

Both control schemes ensure stable control of the end-effector motion proving that the task can efficiently be described in terms of dual quaternions, which greatly simplifies the implementation of the controller since position and orientation are no more separately controlled. The position/orientation error is expressed in a unified and compact vector which avoids singularities related to the representation of the orientation.

IV. ROBOT KINEMATICS IN DUAL QUATERNION SPACE

This section details the computation of the forward kinematic model (FKM) and of the Jacobian matrix in the dual quaternion space that are required in both control schemes.

A. Forward kinematic model in dual quaternion space

To derive the forward kinematic model (FKM) of a serial link manipulator, the standard Denavit-Hartenberg convention [6] has widely been used. Having assigned coordinate frames to each link, the transformation between two successive frames \mathcal{F}_{n-1} and \mathcal{F}_n is described with the following rotations and translations:

- 1) Rotation about axis z_{n-1} of an angle θ_n : $Rot(z_{n-1}, \theta_n)$
- 2) Translation along axis z_{n-1} of a distance d_n : $Trans(z_{n-1}, d_n)$
- 3) Translation along axis x_n of a distance a_n : $Trans(x_n, a_n)$
- 4) Rotation about axis x_n of an angle α_n : $Rot(x_n, \alpha_n)$

Similarly, in the dual quaternion space, the transformation between these frames is obtained by multiplication of the 4 dual quaternions corresponding to each transformation:

$$\underline{\mathbf{q}}_{n-1,n} = \underline{\mathbf{q}}_{rot}(z_{n-1}, \theta_n) \underline{\mathbf{q}}_{trans}(z_{n-1}, d_n) \underline{\mathbf{q}}_{trans}(x_n, a_n) \underline{\mathbf{q}}_{rot}(x_n, \alpha_n) \quad (30)$$

The subscripts *rot* and *trans* indicate whether the transformation is a pure rotation or a pure translation, respectively. In these cases, the dual quaternions are simplified. For a pure rotation, the translation vector is $\vec{0}$ and thus $\underline{\mathbf{q}}_{rot} = \mathbf{q}_{rot}$. For a pure translation, the rotation quaternion is the identity and hence $\underline{\mathbf{q}}_{trans} = 1 + \varepsilon \frac{\mathbf{t}}{2}$.

The FKM can be then calculated for an n -link robot as:

$$\underline{\mathbf{q}} = \underline{\mathbf{q}}_{01} \underline{\mathbf{q}}_{12} \cdots \underline{\mathbf{q}}_{n-1,n} \quad (31)$$

B. Jacobian matrix in the dual quaternion space

For any robot manipulator with n joints, the kinematic expression that relates the end-effector velocity vector $\dot{\mathbf{X}}$ to the joint velocity vector $\dot{\boldsymbol{\theta}}$ is given by the well-known relationship [6]:

$$\dot{\mathbf{X}} = \mathbf{J}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} \quad (32)$$

where the Jacobian matrix $\mathbf{J}(\boldsymbol{\theta})$ is given analytically by

$$\mathbf{J}(\boldsymbol{\theta}) = \frac{d\mathbf{X}}{d\boldsymbol{\theta}} \quad (33)$$

In the dual quaternion space, (32) becomes

$$\dot{\underline{\mathbf{q}}} = J(\underline{\mathbf{q}}) \dot{\boldsymbol{\theta}} \quad (34)$$

Now the Jacobian matrix relating the joint velocity vector $\dot{\boldsymbol{\theta}}$ and the time derivative of the dual quaternion $\dot{\underline{\mathbf{q}}}$ is given analytically by

$$J(\underline{\mathbf{q}}) = \frac{d\underline{\mathbf{q}}}{d\boldsymbol{\theta}} = \frac{d}{d\boldsymbol{\theta}} \begin{pmatrix} q_1 \\ \vdots \\ q_4 \\ q_5 \\ \vdots \\ q_8 \end{pmatrix} = \begin{pmatrix} \frac{\partial q_1}{\partial \theta_1} & \cdots & \frac{\partial q_1}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial q_8}{\partial \theta_1} & \cdots & \frac{\partial q_8}{\partial \theta_n} \end{pmatrix} \quad (35)$$

where $\underline{\mathbf{q}} = [q_1, q_2, q_3, q_4]^T$ and $\underline{\mathbf{q}}' = [q_5, q_6, q_7, q_8]^T$, with \mathbf{q} and \mathbf{q}' being the components of the dual quaternion $\underline{\mathbf{q}} = \mathbf{q} + \varepsilon \mathbf{q}'$.

V. EXPERIMENTS

To validate the dual position control approach proposed and presented in Section III, we have implemented both control schemes (Fig. 2a and Fig. 2b) on a six-link manipulator. To this aim, we have derived for this Adept Viper robot (Fig. 3) the forward kinematic model in the dual quaternion space and the corresponding Jacobian matrix using (31) and (35).



Fig. 3: Adept Viper s850 robot

A. Chosen task

We have conducted a variety of experiments in simulated and real conditions, and in all of them the proposed approaches achieved an efficient performance.

As initial configuration, we consider all joint positions equal to $\theta_{init} = (20 \ -110 \ 190 \ 25 \ 45 \ 20)^T$. This configuration corresponds to the following vector in the dual quaternion space:

$$\underline{\mathbf{q}}_{init} = (0.3530 \ 0.0651 \ 0.8660 \ 0.3482 \ -0.2192 \ -0.2526 \ -0.0098 \ 0.2938)^T \quad (36)$$

The task consists of moving the robot's end-effector so that the tool frame coincides with the desired frame, as illustrated in Fig. 4. This movement combines a translation

and a rotation of a 60° angle about the global axis y leading to the final end-effector position and orientation:

$$\vec{q}_d = \begin{pmatrix} 0.8660 & 0 & 0.5 & 0 \\ -0.0160 & 0.2197 & 0.0277 & 0.3706 \end{pmatrix}^T \quad (37)$$

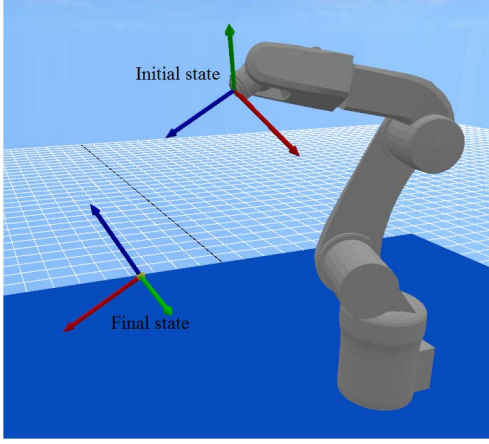


Fig. 4: Description of the task

In terms of homogeneous transformation matrix, these initial and final configurations are written as following:

$$\mathbf{T}_{init} = \begin{pmatrix} -0.7423 & -0.1331 & 0.6567 & 0.3658 \\ 0.3585 & 0.7491 & 0.5570 & 0.1586 \\ -0.5661 & 0.6489 & -0.5083 & 0.7963 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{T}_d = \begin{pmatrix} 0.5 & 0 & 0.866 & 0.7511 \\ 0 & 1 & 0 & 0.0641 \\ -0.8660 & 0 & 0.5 & 0.4223 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Motion control between both configurations will be assumed by the control schemes described in Section III.

B. Results

First results are obtained with the control solution involving the pseudo-inverse Jacobian matrix (Fig. 2a). In order to ensure a same dynamic behaviour for each component of the dual quaternion vector, the positive definite gain matrix \mathbf{K} is chosen as $k * \text{diag}(1, 1, 1, 1, 1, 1, 1, 1)$ where the scalar k is tuned to minimize the time response and the overshoot. Fig. 5 shows the time response of each component of the dual quaternion vector. Each component asymptotically converges to its desired value. The resulting trajectory of the end-effector is illustrated in Fig. 6. The end-effector position and the angle of rotation are shown in Fig. 7a and Fig. 7b, which demonstrate the efficiency of the method. Tuning the gain k to a higher value leads to a shorter time response as shown in Fig. 8.

Next results aim to compare the performance of the pseudo-inverse based position control loop to the performance of the transpose based position control loop. The positive definite gain matrix \mathbf{K} in the second solution (Fig. 2b) is also chosen as $k * \text{diag}(1, 1, 1, 1, 1, 1, 1, 1)$ where k imposes

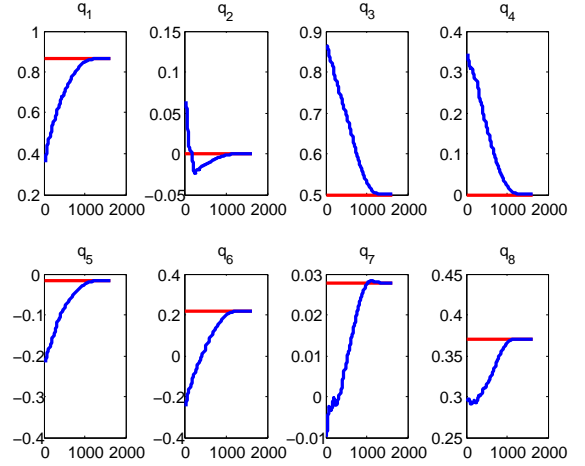


Fig. 5: Time response of the dual quaternion components

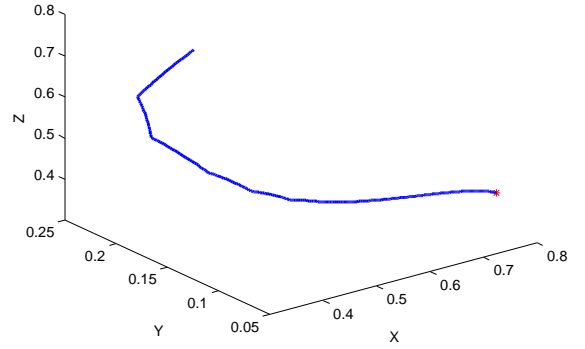


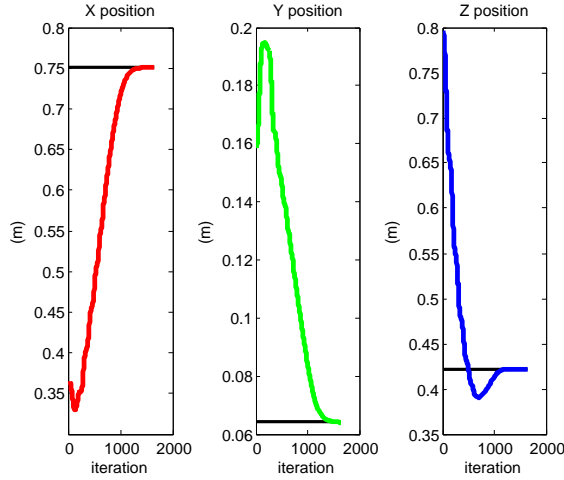
Fig. 6: Trajectory of the end-effector

the dynamic behaviour of each dual quaternion component. This symmetric form also ensures that the system is asymptotically stable as stated in Section III.

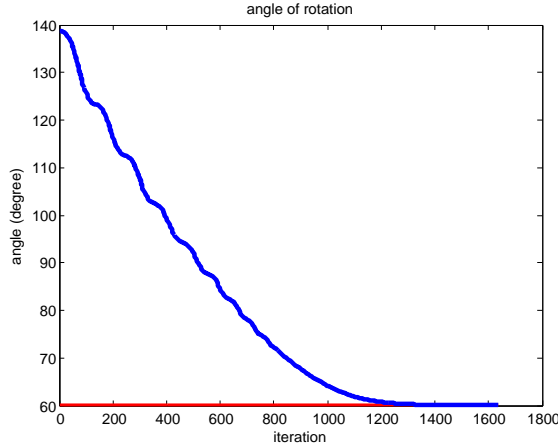
Fig. 9 shows the time response of the rotation angle. It looks similar to the previous one. The control scheme with the Jacobian transpose is as efficient as the one with the pseudo-inverse. But this second solution avoids the computation of the inverse matrix.

VI. CONCLUSIONS

We have presented in this paper the concept of dual position control in dual quaternion space, exploiting the compactness of this mathematical tool. Position and orientation are defined in a common vector which simplifies the design of the controller and avoids the computation of complex transformations to obtain the orientation error. Two dual position control schemes have been presented as an adaptation of classical ones and their convergence has been proven. The necessary forward kinematic model and Jacobian matrix have been derived in dual quaternion space. Several experimental results have demonstrated the efficiency of the proposed solutions. This new formulation avoids representation singularities. Future works will study the kinematic singularities involved by the new Jacobian matrix and intend to use these control schemes for redundant robots.



(a) Time response of the end-effector position components



(b) Time response of the angle of rotation

Fig. 7: Dual position control using the pseudo-inverse Jacobian matrix

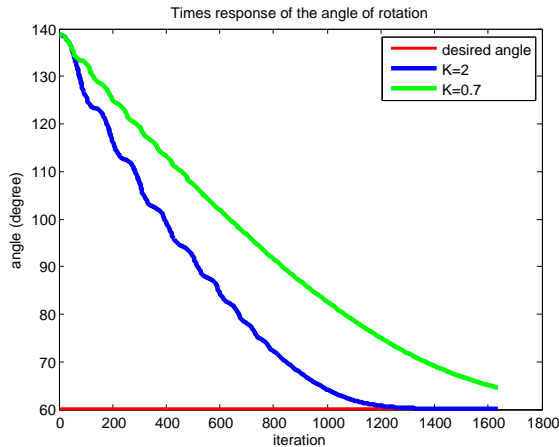


Fig. 8: Time response of the angle of rotation with different values of K when using the pseudo-inverse Jacobian matrix

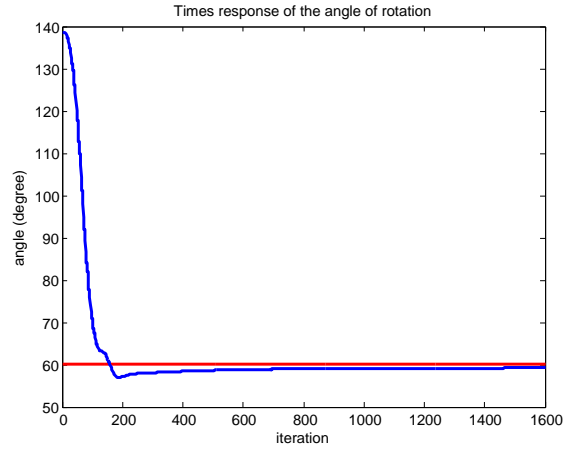


Fig. 9: Time response of the angle of rotation for the dual position control using the transpose Jacobian matrix

VII. ACKNOWLEDGMENT

This work is supported by Agence Nationale de la Recherche (ANR) under grant ANR 07-ROBO-011 in the context of the ASSIST project.

REFERENCES

- [1] N.A. Aspragathos and J.K. Dimitros. A comparative study of three methods for robot kinematics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 28(2):135–145, Apr 1998.
- [2] Clifford. Preliminary sketch of biquaternions. In Oxford Journals, editor, *Proceedings London Mathematical Society*, 1-4, pages 381–395, 1873.
- [3] Konstantinos Daniilidis. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research*, 18:286–298, 1999.
- [4] J. Funda, R.H. Taylor, and R.P. Paul. On homogeneous transforms, quaternions, and computational efficiency. *Robotics and Automation, IEEE Transactions on*, 6(3):382–388, Jun 1990.
- [5] Jack B. Kuipers. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. 1999.
- [6] Richard P. Paul. *Robot Manipulators: Mathematics, Programming, and Control*. MIT Press, Cambridge, MA, USA, 1982.
- [7] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control (Advanced Textbooks in Control and Signal Processing)*. Springer, 2008.
- [8] Yuanwin Wu, Xiaoping Hu, Dewen Hu, Tao Li, and Junwang Lian. Strapdown inertial navigation system algorithms based on dual quaternions. *IEEE transactions on aerospace and electronic systems*, 41(1):110–132, 2005.
- [9] B. Xian, M.S. de Queiroz, D. Dawson, and I. Walker. Task-space tracking control of robot manipulators via quaternion feedback. *Robotics and Automation, IEEE Transactions on*, 20(1):160–167, 2004.
- [10] J.S. Yuan. Closed-loop manipulator control using quaternion feedback. *IEEE Journal of Robotics and Automation*, 4(4):434–440, Aug 1988.