A Dual Quaternion Linear-Quadratic Optimal Controller for Trajectory Tracking

M. M. Marinho, L. F. C. Figueredo, B. V. Adorno

Abstract— **This work addresses the task-space design problem of a linear-quadratic optimal tracking controller for robotic manipulators using the unit dual quaternion formalism. The efficiency, compactness, and lack of singularity of the representation render the unit dual quaternion a suitable framework for simultaneously describing the attitude and the position of the end-effector. Motivated by the advantages of this kinematic description, we propose a new task-space optimal tracking controller in order to find an optimal trajectory for the end-effector, providing a tool to balance more conveniently the end-effector error and its task-space velocity. We show that the kinematic control problem using the dual quaternion transformation invariant error can be reduced to an affine time-varying system. The proposed optimal tracking controller allows the compensation of trajectory induced disturbances, as well as other modeled additive disturbances and known bias. Simulation results with different design parameters provide a performance overview—in comparison with standard kinematic controllers with and without a feed-forward term for tracking a desired reference.**

I. Introduction

The study of modeling and control strategies for robotic manipulators is closely related to the specificities of the desired task. Several tasks can be accomplished by stiff robotic manipulators at relatively low velocities and accelerations, justifying the use of kinematic control strategies. Simplifications that arise from the kinematic description allow designing controllers that do not require inertial parameter specifications of the manipulator. Applications of kinematic controllers branch from the stable proportional gain feedback to several sub-fields of study as, for instance, robustness to singularities and perturbations, workspace optimization, simultaneous execution of multiple tasks, and several forms of optimization [1], [2]. In the present study, we take interest at optimization-based kinematic controllers at task-space.

In practical scenarios, there exists a discrepancy between the task-space, where robot tasks are specified, and joint-space, where the actuation takes place. One of the main drawbacks of joint-space control techniques lies in this discrepancy [3], as they require an external inverse kinematics (IK) solution. Given a task specification at the end-effector, the IK provides a set of joint configurations that match that specification. Such solution is generally not easily derived, and in some cases the IK may even result in impossible or in an infinite number of solutions [1]. Moreover, as modern manipulator robots are often equipped with low level controllers at joint level, task-space techniques have the advantage of defining the control law directly at the end-effector, making direct use of the available low level controllers instead of redesigning them.

Closely intertwined with those fields, efficient representations for rigid body transformations have been a recurring topic in several studies concerning robotics. Considering coupled and singularity-free representations, the use of dual quaternions over homogenous transformations is less computationally demanding, being an efficient and compact form for representing rigid transformations [4]. In addition, control laws are defined directly over a vector field, eliminating the need to extract additional parameters or to design matrix-based controllers (i.e., controllers based on the matrix structure of $SE(3)$). In the light of those advantages, there has been an increasing interest in works related to the design of kinematic controllers in dual quaternion space. Those works comprise rigid motion stabilization and tracking [5], robust control [6], multiple rigid body coordination [7], multiple arm manipulation [8], and human-robot interaction [9].

Nonetheless, results in dual quaternion literature aim solely at stabilizing the trajectory of the rigid body, without much concern on control effort and optimality. For instance, although stable the proportional gain controller [10] shows an intrinsic delay in the presence of time-varying trajectories. Adding a feedforward term may eliminate the delay, but has the disadvantage of requiring a control effort that inevitably grows alongside the trajectory time derivative. In this context, optimal criteria allow the designer to choose a performance in an intermediate ground, with no phase shift and with optimal control effort.

Among optimal and robust control of robot manipulators, earlier works consider mostly the dynamic model of the robot, using the computed torque control scheme. For instance, [11] uses minimum variance control to find optimal controllers considering input constraints, with results applied to a two degree-of-freedom (DOF) planar manipulator. A related work uses fuzzy neural networks to approximately compute the optimal control [12]. Another approach uses a single network adaptive critic controller, resulting in a near optimal solution, which is tested using two DOF of a simulated WAM robotic manipulator [13]. The aforementioned fuzzy controllers are able to show approximate results even without much knowledge of the manipulator dynamics. They are motivated mostly because the dynamic parameters

This work was partially supported by the Brazilian agencies CAPES, CNPq, and FAPEMIG.

M.M. Marinho and L.F.C. Figueredo are with Universidade de Brasília, LARA, Brazil. Email:{murilomarinho, figueredo}@lara.unb.br. B.V. Adorno is with the Graduate Program in Electrical Engineering - Federal University of Minas Gerais (UFMG) - Av. Antônio Carlos 6627, 31270-901, Belo Horizonte-MG, Brazil. Email: adorno@ufmg.br.

are hard to obtain precisely, and vary with different payloads. Despite that advantage, the fuzzy controllers have a degree of arbitrariness in parameter selection. Moreover, works using the computed torque technique optimize in the manipulator joint space, requiring an external tool to obtain the optimal IK. The implementation complexity also reduces most examples to two DOF robots. In a different approach [14], optimal results in the sense of convergence time are presented. In this context, instead of optimizing the end-effector trajectory, the authors are interested in bringing the end-effector to a final location in minimum time. Finally, a linear-quadratic optimization for manipulator control has been found [15] considering the dynamic model of a manipulator. The results mostly concern an efficient and practical trajectory smoothing algorithm, using the computed torque technique. However, most recent manipulator robots can be readily controlled by joint velocity inputs and less so through joint torque inputs. Moreover, as in other cases, the optimization is done in joint space, requiring an external IK planner.

A. *Contributions*

In this work, we extend the results from dual quaternion based controllers with the introduction of an optimal quadratic tracking control law in the task space of the general serial-link manipulator. It compensates for modeled perturbations of the pose error, while considering the control effort. The optimization-based controller is by itself a relevant improvement on the literature as it provides the designer with a more intuitive set of performance indexes and extends the range of applications of the dual quaternion formalism to robotics.

Moreover, in contrast to standard optimal control techniques for manipulators, the proposed criterion yields optimized task-space variables that greatly simplify the control implementation. As a consequence, instead of optimizing the joint-space velocity, we aim at obtaining an optimal result for the end-effector velocity at task-space. The analysis is particularly relevant in advanced manipulation tasks—for instance, manipulation in hazardous or unstructured environments, and multi-arm manipulation—that require precise and safe interaction in contrast to convergence speed. In the view of human-robot interaction, the speed of the end-effector largely influences the human acceptance of the contact [16]. Indeed, even if the robotic system is able to prevent undesired injuries, a human is likely to be in a state of constant stress and discomfort [17]. A controlled speed at task-space also prevents drifts and overshoots caused by large accelerations requirements at the low level controllers.

To derive the optimal control criterion, we show that the dual quaternion kinematic control can be reduced to an affine time-varying system with respect to the dual quaternion transformation invariant error. This allows us to obtain a recursive expression for the optimal control using a linear-quadratic regulator (LQR). By modeling a continuous trajectory as the perturbation of the error, the proposed dual quaternion LQR can be used as a trajectory tracking controller, while also considering other modeled error disturbances. To demonstrate the implementation simplicity and efficiency, the controller is evaluated using a simulated robotic manipulator with six DOF in a trajectory tracking task.

II. Preliminaries

We begin by recalling dual quaternions and their basic algebra when representing rigid transformations to establish the notation used in this work, followed by a brief review of existing work in the kinematic control of robotic manipulators.

A. *Dual Quaternions*

Dual quaternions are the building blocks of the kinematic control theory implemented in this work. We begin with the definition of the three imaginary components $\hat{\imath}$, $\hat{\jmath}$ and $\hat{k}$ of a quaternion such that $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1$. Hence, a general quaternion $\boldsymbol{h}$ is given by $\boldsymbol{h} = h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}$, and its conjugate is defined as $\boldsymbol{h}^* \triangleq h_1 - h_2\hat{\imath} - h_3\hat{\jmath} - h_4\hat{k}$. The norm of a quaternion $\boldsymbol{h}$ is $\parallel \boldsymbol{h} \parallel = \sqrt{\boldsymbol{h}\boldsymbol{h}^*}$.

An arbitrary rotation of a rigid body by an angle θ around an axis $\boldsymbol{n} = n_x\hat{\imath} + n_y\hat{\jmath} + n_z\hat{k}$ is represented by the unit norm quaternion $\boldsymbol{r} = \cos(\theta/2) + \boldsymbol{n}\sin(\theta/2)$. A translation described by $\boldsymbol{t} = t_x\hat{\imath} + t_y\hat{\jmath} + t_z\hat{k}$ can be associated to a rotation $\boldsymbol{r}$ in order to represent the complete rigid motion. This is represented by the unit dual quaternion $\underline{\boldsymbol{h}} = \boldsymbol{r} + \varepsilon\,(1/2)\,\boldsymbol{t}\boldsymbol{r}$, where ε is nilpotent; i.e., $\varepsilon \neq 0$ but $\varepsilon^2 = 0$. Moreover, the conjugate of $\underline{\boldsymbol{h}}$ is defined as $\underline{\boldsymbol{h}}^* \triangleq \boldsymbol{r}^* + \varepsilon(\frac{1}{2}\boldsymbol{t}\boldsymbol{r})^*$.

Since a general dual quaternion $\underline{\boldsymbol{g}}$ is composed of eight elements, that is, $\underline{\boldsymbol{g}} = g_1 + g_2\hat{\imath} + g_3\hat{\jmath} + g_4\hat{k} + \varepsilon(g_5 + g_6\hat{\imath} + g_7\hat{\jmath} + g_8\hat{k})$, the vec operator is used to map it into an eight-dimensional column vector; i.e., $\operatorname{vec}\underline{\boldsymbol{g}} \triangleq \begin{bmatrix} g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7 & g_8 \end{bmatrix}^T$. We also define the matrix $\mathbf{C}_8 \triangleq \operatorname{diag}(1,-1,-1,-1,1,-1,-1,-1)$ such that $\operatorname{vec}\underline{\boldsymbol{h}}^* = \mathbf{C}_8 \operatorname{vec}\underline{\boldsymbol{h}}$. Finally, given dual quaternions $\underline{\boldsymbol{g}}_1$, $\underline{\boldsymbol{g}}_2$, the Hamilton operators $\overset{-}{\mathbf{H}}(\cdot)$, $\overset{+}{\mathbf{H}}(\cdot)$ are transformation matrices satisfying the following relation [18]:

$$\operatorname{vec}\left(\underline{\boldsymbol{g}}_1\underline{\boldsymbol{g}}_2\right) = \overset{+}{\mathbf{H}}\left(\underline{\boldsymbol{g}}_1\right)\operatorname{vec}\underline{\boldsymbol{g}}_2 = \overset{-}{\mathbf{H}}\left(\underline{\boldsymbol{g}}_2\right)\operatorname{vec}\underline{\boldsymbol{g}}_1. \qquad (1)$$

B. *Kinematic Control*

The forward kinematics model (FKM) of a serial manipulator robot (that is, the mapping between the n-dimensional vector of joint positions $\theta \in \mathbb{R}^n$ and the end-effector pose $\underline{\boldsymbol{h}}$) can be obtained directly in dual quaternion space using algebraic manipulations [18]. In addition, the differential FKM (i.e., the mapping between the joint velocities $\dot{\theta} \in \mathbb{R}^n$ and the generalized end-effector velocity $\operatorname{vec}\dot{\underline{\boldsymbol{h}}} \in \mathbb{R}^8$) is given by

$$\operatorname{vec}\dot{\underline{\boldsymbol{h}}} = \mathbf{J}\dot{\theta}, \qquad (2)$$

where $\mathbf{J} \in \mathbb{R}^{8\times n}$ is the manipulator Jacobian (which is also found algebraically [18]) and depends on the current robot posture. As (2) is a simple linear mapping, it is common practice to design closed-loop controllers based on the pseudo-inversion of $\mathbf{J}$ that exponentially reduce the error between the current pose $\underline{\boldsymbol{h}}$ and a constant desired pose $\underline{\boldsymbol{h}}_d$; that is,

$$\dot{\theta} = \mathbf{K}\mathbf{J}^{\dagger}\,\mathrm{vec}(\underline{\boldsymbol{h}}_d - \underline{\boldsymbol{h}}), \tag{3}$$

where $\mathbf{K}$ is a positive definite matrix and $\mathbf{J}^{\dagger}$ is the singular value decomposition (SVD) pseudoinverse of $\mathbf{J}$ [10].

III. Dual Quaternion Kinematic LQR

In this section we describe the novel optimal controller which is the main contribution of this work. We begin with the transformation invariant error definition together with the kinematic equations reviewed in the last section. We show that the kinematic control with a time-varying reference from the point of view of the error reduces to a linear time-varying system with an additive perturbation term. By using a suitable solution, we find the optimal state-feedback controller and its computation is discussed.

First, consider an arbitrary continuous desired trajectory $\underline{\boldsymbol{h}}_d \triangleq \underline{\boldsymbol{h}}_d(t)$ in dual quaternion space and the current end-effector pose $\underline{\boldsymbol{h}}$. The invariant error is given by [6]

$$\underline{\boldsymbol{e}} = 1 - \underline{\boldsymbol{h}}^{*}\underline{\boldsymbol{h}}_d. \tag{4}$$

Since the LQR optimization computation assumes future knowledge of the explicit variables, we need to remove any direct dependency on $\underline{\boldsymbol{h}}$. Hence,

$$\underline{\boldsymbol{e}} = 1 - \underline{\boldsymbol{h}}^{*}\underline{\boldsymbol{h}}_d \implies \underline{\boldsymbol{e}}\underline{\boldsymbol{h}}_d^{*} = \underline{\boldsymbol{h}}_d^{*} - \underline{\boldsymbol{h}}^{*} \implies \underline{\boldsymbol{h}}^{*} = \underline{\boldsymbol{h}}_d^{*} - \underline{\boldsymbol{e}}\underline{\boldsymbol{h}}_d^{*}.$$

With time-varying $\underline{\boldsymbol{h}}$ and $\underline{\boldsymbol{h}}_d$, the derivative of (4) is given by $\dot{\underline{\boldsymbol{e}}} = -\dot{\underline{\boldsymbol{h}}}^{*}\underline{\boldsymbol{h}}_d - \underline{\boldsymbol{h}}^{*}\dot{\underline{\boldsymbol{h}}}_d$. Therefore

$$\dot{\underline{\boldsymbol{e}}} = -\dot{\underline{\boldsymbol{h}}}^{*}\underline{\boldsymbol{h}}_d - (\underline{\boldsymbol{h}}_d^{*} - \underline{\boldsymbol{e}}\underline{\boldsymbol{h}}_d^{*})\dot{\underline{\boldsymbol{h}}}_d = -\dot{\underline{\boldsymbol{h}}}^{*}\underline{\boldsymbol{h}}_d + \underline{\boldsymbol{e}}\underline{\boldsymbol{h}}_d^{*}\dot{\underline{\boldsymbol{h}}}_d - \underline{\boldsymbol{h}}_d^{*}\dot{\underline{\boldsymbol{h}}}_d.$$

Using the vec operator on both sides and defining $e \triangleq \mathrm{vec}\ \underline{\boldsymbol{e}}$ yields

$$\dot{e} = -\overset{-}{\mathbf{H}}(\underline{\boldsymbol{h}}_d)\,\mathbf{C}_8\mathrm{vec}\ \dot{\underline{\boldsymbol{h}}} + \overset{-}{\mathbf{H}}\left(\underline{\boldsymbol{h}}_d^{*}\dot{\underline{\boldsymbol{h}}}_d\right)e - \mathrm{vec}\ \underline{\boldsymbol{h}}_d^{*}\dot{\underline{\boldsymbol{h}}}_d.$$

By defining $\mathbf{A} \triangleq \overset{-}{\mathbf{H}}\left(\underline{\boldsymbol{h}}_d^{*}\dot{\underline{\boldsymbol{h}}}_d\right)$ and $c \triangleq -\mathrm{vec}\ \underline{\boldsymbol{h}}_d^{*}\dot{\underline{\boldsymbol{h}}}_d$, and using (2), we find

$$\dot{e} = -\overset{-}{\mathbf{H}}(\underline{\boldsymbol{h}}_d)\,\mathbf{C}_8\mathbf{J}\dot{\theta} + \mathbf{A}e + c.$$

Finally, with $\mathbf{N} \triangleq \overset{-}{\mathbf{H}}(\underline{\boldsymbol{h}}_d)\,\mathbf{C}_8\mathbf{J}$,

$$\dot{e} = \mathbf{A}e - \mathbf{N}\dot{\theta} + c.$$

Instead of considering $\dot{\theta}$ the input signal for the system, we can consider as input the end effector velocity u using the mapping $u = -\mathbf{N}\dot{\theta}$. This allows the optimization to be done in task-space variables. We focus on finding the optimal controller for the affine time-varying system

$$\dot{e}(t) = \mathbf{A}(t)e(t) + u(t) + c(t). \tag{5}$$

Therefore, from the error point-of-view, we can solve the tracking problem for a continuous trajectory using a finite horizon LQR applied to a disturbed system, as the error disturbance caused by the time-varying trajectory is given by $c(t)$. Other modeled continuous disturbances can also be grouped into $c(t)$ and used in the same solution.

Consider that the manipulator has to track the trajectory during $t \in [0, t_f]$. Therefore, we seek to minimize the following cost function

$$F = \frac{1}{2}e(t_f)^T\mathbf{S}e(t_f) + \frac{1}{2}\int_{t_0}^{t_f}\left(e^T\mathbf{Q}e + u^T\mathbf{R}u\right)dt, \tag{6}$$

given the matrices $\mathbf{S}$, $\mathbf{Q}(t) \geq 0$ and $\mathbf{R}(t) > 0$ with $\mathbf{S}$, $\mathbf{Q}$, $\mathbf{R}$ $\in \mathbb{R}^{8\times 8}$. The matrix $\mathbf{S}$ is the weight of the final error norm, the time-varying matrix $\mathbf{Q}$ weighs the error cost along the trajectory, and the time-varying matrix $\mathbf{R}$ weighs the control effort in terms of end-effector velocity norm. As long as $\mathbf{N}$ is well conditioned, an increase in $\mathbf{R}$ will, *ceteris paribus*, also cause an overall decrease in joint velocities. The optimization of (6) leads to an optimal feedback without excessive expenditure of control energy while keeping the error $e(t)$ near zero [19].

To solve the optimization problem, we introduce the costate variable p, which acts as a Lagrange multiplier for the state equations. Hence, using the equality constraint defined in (5), the function (6) can be rewritten as

$$H = F + \int_{t_0}^{t_f} p^T(\mathbf{A}e + u + c - \dot{e})dt. \tag{7}$$

According to [20] and to distribution theory applied to optimality conditions, we have $\partial H/\partial u = 0$ and $\partial H/\partial e = 0$ as necessary conditions for the optimal trajectory. Hence,

$$\begin{aligned}\frac{\partial H}{\partial u} = 0 &\implies \mathbf{R}u + p = 0\\ &\implies u = -\mathbf{R}^{-1}p.\end{aligned} \tag{8}$$

In addition, we use the Leibniz integral rule in (7), that is,

$$\begin{aligned}-\int_{t_0}^{t_f} p^T\dot{e}dt &= -\int_{t_0}^{t_f}\left(\frac{d}{dt}\left(p^Te\right) - \dot{p}^Te\right)dt\\ &= p^T(t_0)e(t_0) - p^T(t_f)e(t_f) + \int_{t_0}^{t_f}\dot{p}^Tedt,\end{aligned}$$

to find

$$\begin{aligned}\frac{\partial H}{\partial e} = 0 &\implies \mathbf{Q}e + \mathbf{A}^Tp + \dot{p} = 0\\ &\implies \dot{p} = -\left(\mathbf{Q}e + \mathbf{A}^Tp\right).\end{aligned} \tag{9}$$

Please note that $\partial^2H/\partial u^2$ must be positive to minimize (7), which requires $\mathbf{R} > 0$. The system and proposed cost function allow the use of the costate function [20]

$$p(t) = \mathbf{P}e + \xi, \tag{10}$$

where $\mathbf{P}$ is a time-varying proportional gain and ξ is a weighted feedforward term. The derivative of (10) is given by

$$\dot{p}(t) = \dot{\mathbf{P}}e + \mathbf{P}\dot{e} + \dot{\xi}. \tag{11}$$

After using (10) in (8) and substituting the result in (5), and using (10) in (9), we replace the resulting equations in (11) to find

$$\begin{aligned}&-\dot{\xi} - \mathbf{A}^T\xi + \mathbf{P}\mathbf{R}^{-1}\xi - \mathbf{P}c\\ &\qquad = \left(\dot{\mathbf{P}} + \mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P} - \mathbf{P}\mathbf{R}^{-1}\mathbf{P} + \mathbf{Q}\right)e.\end{aligned} \tag{12}$$

Considering that (12) must hold for any choice of initial state e and that both $\mathbf{P}$ and ξ do not depend on the initial error, we need simultaneously

$$\begin{cases} \dot{\mathbf{P}} + \mathbf{PA} + \mathbf{A}^T\mathbf{P} - \mathbf{PR}^{-1}\mathbf{P} + \mathbf{Q} & = \mathbf{0} \\ \dot{\xi} + \mathbf{A}^T\xi - \mathbf{PR}^{-1}\xi + \mathbf{P}c & = 0, \end{cases}$$

which means

$$\begin{cases} \dot{\mathbf{P}} & = -\mathbf{PA} - \mathbf{A}^T\mathbf{P} + \mathbf{PR}^{-1}\mathbf{P} - \mathbf{Q} \\ \dot{\xi} & = -\mathbf{A}^T\xi + \mathbf{PR}^{-1}\xi - \mathbf{P}c. \end{cases} \tag{13}$$

The boundary conditions to solve (13) are obtained using the final time, t_f, of our trajectory. We set the feed-forward term to zero in the terminal time—that is, $\xi(t_f) = 0$—to find the first boundary condition. From (10) with $\xi(t_f) = 0$, we find that $\partial H/\partial e(t_f) = 0$ yields $\mathbf{P}(t_f) = \mathbf{S}$.

Note that we know $\mathbf{A}(t) = \overset{-}{\mathbf{H}}\left(\underline{\boldsymbol{h}}_d^*\dot{\underline{\boldsymbol{h}}}_d\right)$ for all t, as it depends only on the desired trajectory. We can then numerically solve the differential Ricatti equation $\mathbf{P}(t)$ backwards in time. As $c = -\underline{\boldsymbol{h}}_d^*\dot{\underline{\boldsymbol{h}}}_d$ is also known for all t and, with the solution of $\mathbf{P}(t)$, we can find $\xi(t)$ by also solving it backwards in time.

Therefore, from (8) and (10) the optimal control is given by $u(t) = -\mathbf{R}^{-1}\left(\mathbf{P}e + \xi\right)$. Applying as joint velocities, we find

$$\dot{\theta} = \mathbf{N}^{\dagger}\mathbf{R}^{-1}\left(\mathbf{P}e + \xi\right). \tag{14}$$

A. DQ-LQR Computation and Parameter Selection

The system (13) containing the Ricatti equation is nonlinear, and the closed-form solution may not be found for all cases. As a consequence, finite differences approximations are normally used; that is,

$$\frac{d}{dt}\mathbf{P}(t) \approx \frac{\mathbf{P}(t+\tau) - \mathbf{P}(t)}{\tau}$$

for a small sampling time τ that directly affects the approximation precision.

In order to obtain $\mathbf{P}(t)$ for all t, we begin with $\mathbf{P}(t_f)$ and use the derivative approximation backwards in time using the known values of $\mathbf{A}(t)$, which depends only on the desired trajectory, up to $\mathbf{P}(0)$. The same can be applied to find an approximate solution for $\xi(t)$, by also considering $c(t)$. All these calculations can be done before the system starts to operate.

The design parameters of the controller are the matrices $\mathbf{Q}$, $\mathbf{S}$ and $\mathbf{R}$. Matrices $\mathbf{Q}$ and $\mathbf{S}$ are related to the state norm cost. Since the system input is the end-effector velocities, a proper choice of the elements of $\mathbf{R}$ allows the designer to define end effector velocities into which is more costly to move over others.

An interesting characteristic of the LQR design is the ability to define time-varying weights in the optimization process. For instance, we may devise a trajectory that begins far from the end effector initial position. To avoid large joint velocities when the error is expected to be large, we may assign a smaller $\mathbf{Q}$ at the beginning of the trajectory.

IV. Simulated Evaluation & Discussion

In order to demonstrate the controller behavior under different sets of parameters, a simulated task was devised. The controller was implemented using the DQ_robotics[1] in MATLAB. Consider the COMAU Smart Six with six DOF, with which we wish to perform a complex motion to demonstrate the effectiveness and simplicity of the controller.

We begin by choosing a periodic translation in all axes,

$$\begin{aligned} \boldsymbol{t}_d(t) = (0.99707{+}0.1\cos(\pi t))\,\hat{\imath} + (0.1{+}0.1\cos(\pi t))\,\hat{\jmath} \\ + (1.075{+}0.1\sin(\pi t))\,\hat{k}. \end{aligned}$$

In order to generate a motion that modifies all rotational degrees of freedom, we also choose a varying end effector orientation given by

$$\boldsymbol{r}_d(t) = \cos\left(\frac{\phi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\frac{\boldsymbol{v}}{\parallel \boldsymbol{v} \parallel},$$

with $\phi = 0.25\sin(\pi t) + \pi$ and $\boldsymbol{v} = \cos(\pi t)\hat{\imath} + \sin(\pi t)\hat{\jmath} + (\sin(\pi t) - 2)\hat{k}$.

The trajectory is the combined motion given by

$$\underline{\boldsymbol{h}}_d(t) = \boldsymbol{r}_d(t) + \varepsilon\tfrac{1}{2}\boldsymbol{t}_d(t)\boldsymbol{r}_d(t).$$

The robot initial configuration is given by $\theta_0 = \begin{bmatrix}0 & 0 & -\pi/2 & 0 & \pi/2 & 0\end{bmatrix}^T$, with the end effector placed 40 cm away from the desired trajectory.

Three LQR controllers are designed so that their performance can be evaluated. In order to simplify the choice of parameters, we define $s,\ q,\ r \in \mathbb{R}$, such that

$$\mathbf{S} = s\mathbf{I}, \qquad \mathbf{Q} = q\mathbf{I}, \qquad \mathbf{R} = r\mathbf{I},$$

where $\mathbf{I} \in \mathbb{R}^{8\times 8}$ is an identity matrix. The choice of parameters is closely intertwined with the task specificities. For instance, we suppose that the end-time error is not of higher importance than the error in the remainder of the trajectory by setting $s = 0$. By knowing that the end effector starts far from the desired trajectory, we design a trajectory error weight q that begins small and increases over time: $q = 0.001$ for $t \leq 0.5$, linearly increasing to $q = 1$ for $t \in (0.5, 1.0]$, $q = 1$ otherwise. All three controllers share the same s and q.

In order to compare performance, we distinguish the three controllers by choosing a decreasing weight for the control effort parameter, that is $r \in \{0.1, 0.01, 0.001\}$. The overall behavior of the three optimal controllers can be seen in Fig. 1. In all controllers, the choice of q allows a well-behaved initial motion. Moreover, there is no noticeable phase shift after 1.0 s, when q reaches its final value. The effect in the decreasing of r is also clearly shown in Fig. 1, that is, a higher control effort yields smaller trajectory error.

We also compare the performance of the proposed optimal controllers with classic control methods. To this aim, we also simulated a proportional controller (K controller)

$$\begin{cases} \mathbf{K} = k\mathbf{I}, & k \in \mathbb{R}^+ \\ \dot{\theta} = \mathbf{N}^{\dagger}\mathbf{K}e, \end{cases}$$

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

and a proportional controller with a feed-forward term (K + FF), given by

$$\begin{cases}\mathbf{K} = k\mathbf{I}, & k \in \mathbb{R}^{+} \\ \dot{\theta} = \mathbf{N}^{\dagger}(\mathbf{K}e - \operatorname{vec} \underline{\boldsymbol{h}}^{*}\dot{\underline{\boldsymbol{h}}}_{d}). & \end{cases}$$

The performance of the optimal controller with $r = 0.001$, in comparison with the K and K+FF controllers with $k = 10$ is shown in Fig. 2. It is noticeable that whereas the K controller shows the expected phase shift in spite of the high control effort, the propose criterion with $r = 0.001$ closely resembles the K + FF controller, but without the initial peak velocity. This allows the LQR controller to send smoother joint velocities when compared to the abrupt velocities seen in the K and K + FF controllers. It also yields smaller accelerations at the end-effector which, among other benefits, prevents damages to the manipulator and reduces the discomfort during human-robot interaction [17].

The overall results show that by choosing the design parameters accordingly, we are able to define a trade-off between control effort and tracking error along the trajectory. For further information, the integral norm of the end-effector velocity, acceleration, and tracking error for each controller is shown in Table I. From Fig. 2, it is clear that the smaller error from the K+FF controller stems from smaller convergence time which in turn is obtained in exchange of abrupt velocities and accelerations—respectively, 25% and 250% larger than the results from the LQR with $r = 0.001$.

TABLE I

Integral norm of the end-effector velocity (Vel.), acceleration (Acc.), and trajectory error (Err.).

	LQR			K	K+FF
	r=0.1	r=0.05	r=0.001	k=10	k=10
Err.	1.134	0.886	0.450	0.582	0.176
Vel.	3.297	3.833	4.857	5.820	5.954
Acc.	0.028	0.033	0.048	0.176	0.173

V. Conclusions

In this paper, we studied the design of a dual quaternion kinematic finite-time linear-quadratic optimal tracking controller. By deriving a perturbed time-varying linear system from the transformation invariant error definition, we proposed an optimal criterion strategy to control the error in relation to perturbations caused by a time-varying trajectory. The resulting optimal technique can be readily applied to account for other modeled sources of bias and disturbance to the end effector pose error. The controller allows the designer to set optimal gains for a given trajectory in terms of control effort and error regulation, and may be applied in both non-redundant and redundant manipulators. We also provided simulation results using a six DOF manipulator to illustrate the performance and effectiveness of the proposed optimal criterion in comparison with a proportional gain controller and a proportional gain controller with feed-forward.

References

[1] M. W. Spong, S. Hutchingson, and M. Vidyasagar, *Robot Modeling and Control*. New York: John Wiley & Sons.

[2] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control*, M. J. Grimble and M. A. Johnson, Eds. London: Springer, 2008.

[3] O. Khatib, "Manipulator control at kinematic singularities: a dynamically consistent strategy," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 3. IEEE Comput. Soc. Press, 1995, pp. 84–88.

[4] M. Schilling, "Universally manipulable body models dual quaternion representations in layered and dynamic mmcs," *Autonomous Robots*, vol. 30, no. 4, pp. 399–425, 2011.

[5] X. Wang and C. Yu, "Unit dual quaternion-based feedback linearization tracking problem for attitude and position dynamics," *Systems & Control Letters*, vol. 62, no. 3, pp. 225–233, Mar. 2013.

[6] L. Figueredo, B. Adorno, J. Ishihara, and G. Borges, "Robust kinematic control of manipulator robots using dual quaternion representation," in *Proceedings of the IEEE International Conference on Robotics and Automation*. Karlsruhe: Germany, 2013.

[7] X. Wang, C. Yu, and Z. Lin, "A Dual Quaternion Solution to Attitude and Position Control for Rigid-Body Coordination," *IEEE Transactions on Robotics*, vol. To Appear, 2012.

[8] L. Figueredo, B. Adorno, J. Ishihara, and G. Borges, "Switching strategy for flexible task execution using the cooperative dual task-space framework," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1703–1709.

[9] B. V. Adorno, a. P. L. Bó, and P. Fraisse, "Kinematic modeling and control for human-robot cooperation considering different interaction roles," *Robotica*, vol. 33, no. 2, pp. 314–331, Feb. 2015.

[10] H.-L. Pham, V. Perdereau, B. Adorno, and P. Fraisse, "Position and orientation control of robot manipulators using dual quaternion feedback," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei: Taiwan, 2010.

[11] M. Negm, "Application of optimal preview and adaptive controllers for robotics manipulator with control input constraints," in *Proceedings of the IEEE International Symposium on Industrial Electronics*, vol. 2, 1999, pp. 853–860 vol.2.

[12] R.-J. Wai, C.-H. Tu, and K.-Y. Hsieh, "Design of intelligent optimal tracking control for robot manipulator," in *2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, vol. 1, July 2003, pp. 478–483 vol.1.

[13] S. Dutta and L. Behera, "Policy iteration based near-optimal control scheme for robotic manipulator with model uncertainties," in *IEEE International Symposium on Intelligent Control*, Aug 2013, pp. 358–363.

[14] M. Galicki and D. Ucinski, "Time-optimal motions of robotic manipulators," *Robotica*, vol. 18, no. 6, pp. 659–667, 2000.

[15] O. Egeland and E. Lunde, "Trajectory generation for manipulators using linear quadratic optimal tracking," in *IEEE International Conference on Robotics and Automation*, Apr 1988, pp. 376–381 vol.1.

[16] A. Mertens, C. Brandl, I. Blotenberg, M. Ludtke, T. Jacobs, C. Brohl, M. P. Mayer, and C. M. Schlick, "Human-robot interaction: Testing distances that humans will accept between themselves and a robot approaching at different speeds," in *Ambient Assisted Living*, ser. Advanced Technologies and Societal Change, R. Wichert and H. Klausing, Eds. Springer Berlin Heidelberg, 2014, pp. 269–286.

[17] P. A. Lasota, G. F. Rossano, and J. A. Shah, "Toward safe close-proximity human-robot interaction with standard industrial robots," in *IEEE International Conference on Automation Science and Engineering*, 2014.

[18] B. V. Adorno, "Two-arm manipulation: from manipulators to enhanced human-robot collaboration [contribution à la manipulation à deux bras : des manipulateurs à la collaboration homme-robot]," Ph.D. dissertation, Université Montpellier 2, 2011.

[19] A. Locatelli, *Optimal Control: An Introduction*. Birkhäuser Basel, 2001.

[20] V. GARBER, "Optimum intercept laws for accelerating targets." *AIAA Journal*, vol. 6, no. 11, pp. 2586–2591, Nov 1968.

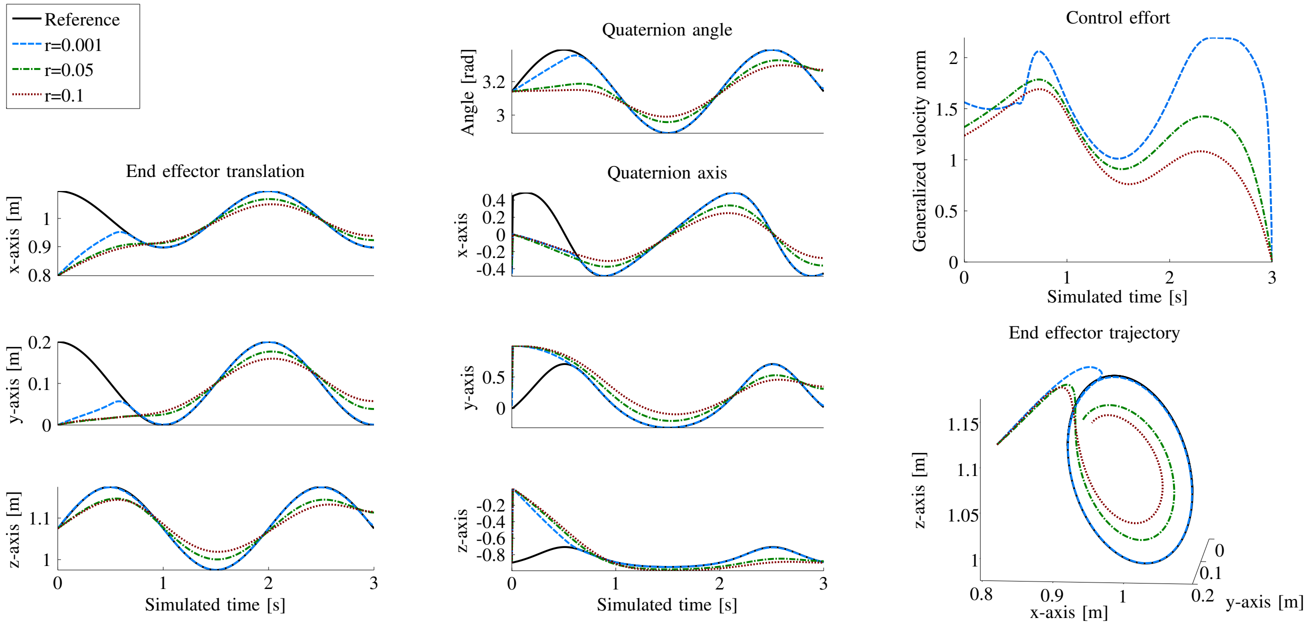

Fig. 1. End effector translation (*left*) and orientation (*center*), the control effort (*top-right*), and end effector trajectory (*bottom-right*); using the linear-quadratic optimal controller with different parameters. The trade-off between error and control effort is noticeable, and the end effector motion shows no steady-state phase shift in relation to the reference.

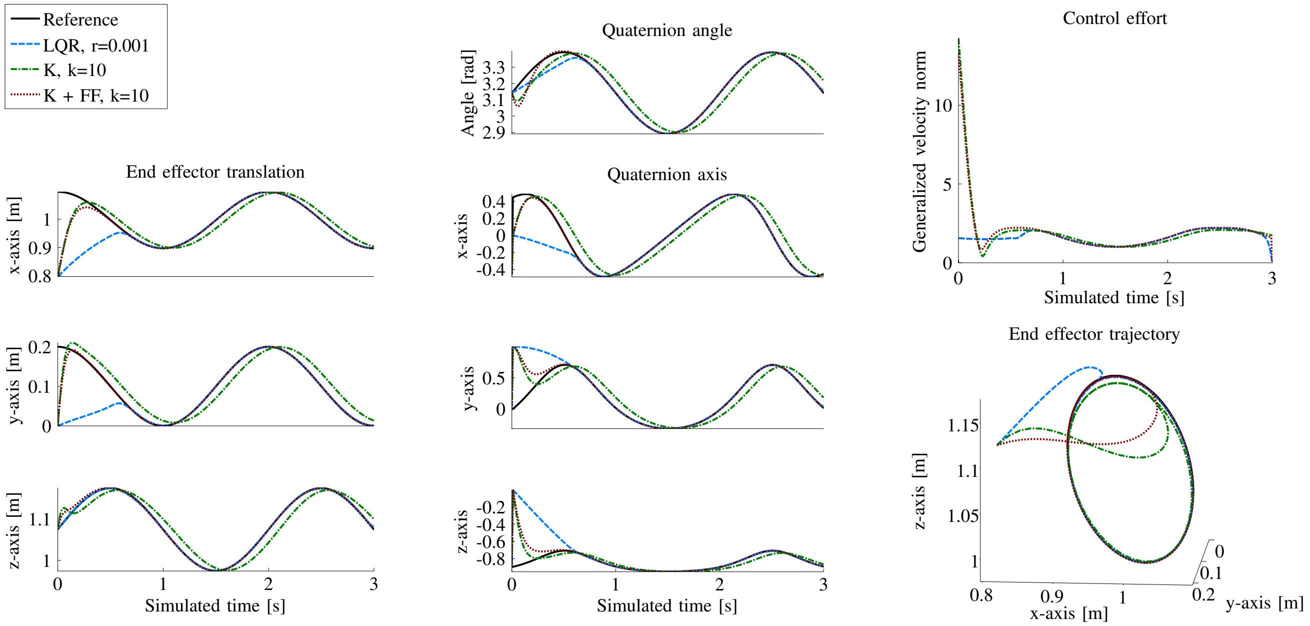

Fig. 2. End effector translation (*left*) and orientation (*center*), the control effort (*top-right*), and end effector trajectory (*bottom-right*); comparing the optimal controller with the proportional gain controllers. The initial peak velocity using the proportional controllers is noticeable, and results from the high initial error. Without the prediction term, the proportional controller has an inherent phase shift in relation to the desired trajectory.