A Smooth Orientation Planner for Trajectories in the Cartesian Space

Andrea Tagliavini and Corrado Guarino Lo Bianco, *Senior Member, IEEE*

***Abstract*—Trajectories in the operational space, when conceived for manipulators with more then three degrees of freedom, impose the adoption of an orientation primitive for the end-effector. The planning complexity increases if smoothness represents one of the motion requirements and the trajectory is obtained through the combination of several basic primitives. In this eventuality, vibrations and mechanical solicitations can be reduced by avoiding motion stops at the end of each segment. Good tracking performances can be conversely achieved by guaranteeing jerk-continuous reference signals for the actuators. The orientation planner proposed in this letter allows the smart generation of smooth trajectories. As experimentally proved in the work, the novel planning primitive is characterized by very short computational times.**

***Index Terms*—Motion and path planning, task and motion planning, orientation planner, jerk continuity.**

I. Introduction

THE orientation planning problem is common to many automation contexts and can be summarized as follows: given a sequence of through points, a trajectory must be planned so that the end-effector could exactly cross all of them with the desired orientation. Typically, generated motions should simultaneously satisfy some additional features, and smoothness is certainly one of them. To this purpose, reduced solicitation can be achieved by guaranteeing that the actuators reference signals are jerk-continuous.

Orientations can be expressed in multiple ways and, consequently, different planning techniques can be conceived. The most commonly used notations were compared in [1] in terms of effectiveness and conciseness. Rotation matrices have an immediate physical meaning but, being based on 9 terms, are certainly redundant. Consequently, they are not immediately suited to planning purposes. Euler angles return a compact orientation description, but they lack of effectiveness in some particular configurations. Angle-axis representation and quaternions are the most effective orientation notations: they both use four terms (one of them is clearly redundant) and are not subject to singular configurations.

One of the first planning strategies specifically conceived for quaternions was proposed by Shoemake in 1985 [2]: the Spherical Linear Interpolation (SLERP). Such strategy is well suited for the management of point-to-point trajectories, but it is not immediately usable in multi-point contexts, which, conversely, are very common in robotics applications. When the motion involves multiple via-points, the SLERP approach causes "orientation corners," which induce discontinuities in the actuators speeds, accelerations and jerks. The "orientation corner" problem is equivalent to the well known "path corner" problem arising in CNC machines: in both cases discontinuities can be avoided by admitting trajectory approximations. In order clarify the concept, let us consider the path generation problem for a CNC machine schematically shown in Fig. 1 (the orientation planning problem poses similar problems). Such figure shows a situation in which a generic curve, represented by the black solid line, should be followed by a milling tool. As known, in CNC applications the original path is preliminarily converted into a sequence of via-points: at run-time the machining tool executes a series of linear segments joining such points (see the blue dotted lines), thus introducing a first approximation with respect to the desired profile. In order to eliminate discontinuity issues caused by the adoption of straight line primitives, corner smoothing methods (see the green dash-dotted lines) have been proposed [3], [4], [5], [6]. However, such strategies, as shown in Fig. 1, induce further approximations w.r.t the nominal path.

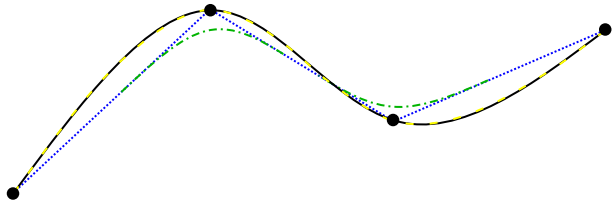

Fig. 1. Comparison between the desired CAD path (black solid line), the GCODE converted path (blue dotted line), the corner smoothed path (green dash-dotted line) and η^{3D}-spline path (yellow dashed line).

An equivalent problem arises for orientations trajectories when using the above mentioned SLERP primitive. In order to partially overcome the continuity issues, an alternative primitive, named SQUAD, was proposed in [7]. However, as shown in [8], such orientation primitive only guarantees the $\mathcal{C}^1$ continuity of the reference signals at the via-points. Previous considerations make it possible to assert that, for an accurate generation of the surface profile, particular care must be dedicated to the

Manuscript received 16 June 2022; accepted 2 March 2023. Date of publication 14 March 2023; date of current version 23 March 2023. This letter was recommended for publication by Associate Editor Tirthankar Bandyopadhyay and Editor Aniket Bera upon evaluation of the reviewers' comments. This work was supported under the National Recovery and Resilience Plan (NRRP), in part by Mission 4 Component 2 Investment 1.5 - Call for Tender 3277 of 30/12/2021 of the Italian Ministry of University and Research funded by the European Union - NextGenerationEU, under Grant ECS00000033, and in part by Concession Decree 1052 of 23/06/2022 adopted by the Italian Ministry of, under Grant CUP D93C22000460001, through "Ecosystem for Sustainable Transition in Emilia-Romagna" (Ecosister). *(Corresponding author: Corrado Guarino Lo Bianco.)*

The authors are with the Dip. di Ingegneria e Architettura, University of Parma, I-43124 Parma, Italy (e-mail: andrea.tagliavini@unipr.it; corrado.guarinolobianco@unipr.it).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3256921, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3256921

generation of both the path and the orientation trajectories. More in details, path approximations can be strongly reduced, for example, by adopting a primitive recently proposed in [9]: a curve which exactly crosses the assigned via-points can be generated by interpolating them through the η^{3D}-splines. As indicatively shown by the yellow dashed lines in Fig. 1, such primitive allows path errors which are smaller than the ones that can be achieved with the corner smoothing strategy. However, in order to implement a fully working planner in the operational space, such path primitive needs to be combined to an orientation planner. This letter is devoted to the design of such planner, so as to associate a proper orientation to each point of the η^{3D}-splines path. The proposed primitive is characterized by a $\mathcal{C}^3$ continuity level.

A. State of the Art

As previously stated, continuity problems can be overcome by means of orientation corner smoothing techniques, like the ones proposed in [3], [4], [5], [6]. Alternatively, new primitives have been designed for the direct smooth interpolation of the quaternions assigned at the via-points. To this purpose, the literature proposes several approaches, which differ each other for the adopted primitive: to obtain the $\mathcal{C}^1$ continuity Shoemake [7] and Dam et al. [8] use a spherical interpolation, while to obtain the $\mathcal{C}^2$ continuity Legnani et al. [10] use a repeated combination of SLERP curves, Kim et al. [11], Ge et al. [12] and Pu et al. [13] use B-splines, Nielson [14] uses ν-splines, Liu et al. [15] use quartic polynomials while Tan et al. [16] use quintic polynomials. It is important to point out that all the aforementioned strategies [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] only guarantee the $\mathcal{C}^1$ or the $\mathcal{C}^2$ continuity of the orientation profiles. Consequently, the corresponding actuators jerks are discontinuous. The sole alternative strategies proposed in the literature, which generates position and orientation trajectories in the operational space and also guarantees jerk-continuous signals, are the ones proposed in [5], [6].

B. The Novel Planner

In this letter, in order to avoid jerk discontinuities and to obtain very smooth trajectories, a novel orientation planning approach, able to guarantee the $\mathcal{C}^3$ continuity of the reference signals, is designed. The new primitive is specifically conceived to be used together with the η^{3D}-splines [9] path planner. When used together, the two primitives allow the generation of jerk-continuous trajectories. A common denominator of both primitives is represented by their low computational burden. The orientation planner proposed in this work could also be combined with alternative $\mathcal{G}^3$ path primitives, like the one given in [17] or, if jerk continuity is not mandatory, with $\mathcal{G}^2$ primitives [18], [19].

The novel planner differs from the ones proposed in [5], [6] for a relevant aspect. While all the three approaches guarantee the $\mathcal{C}^3$ continuity – actually [5] allows the $\mathcal{C}^4$ continuity – the planners designed in [5], [6] are explicitly conceived to handle corner smoothing problems, so that trajectories are made of linear segments properly joined at the via-points by means of blending curves. As seen earlier, this implies that the assigned positions and orientations at the via-points are satisfied with a given tolerance. Conversely, the trajectory planner proposed in this letter allows a strict fulfillment of the assigned positions and orientations. This is possible because the η^{3D}-splines can generate a wide variety of curves – like, for example, linear segments, circular arcs, clothoids, conic spirals, etc. – directly passing through the assigned via-points (see [9] for more details), so that corner smoothing techniques, which necessarily cause tolerances, are not required at all.

The letter is organized as follows. Section II introduces some preliminary considerations on the continuity characteristics which must be owned by the orientation planner in order to achieve continuous jerk signals. In the same section, some basic concepts on the quaternion theory are recalled. Section III proposes the orientation planner. Tests on the new primitive are proposed in Section IV. First, a simulated case is considered for a CNC milling problem, in order to prove that the orientation planner can be used in multiple contexts. Then, experimental tests have been executed with a Comau Smart SiX 6.14 manipulator and the corresponding outcomes are discussed in the same section and are visually shown by means of two multimedia attachments. Final conclusions are drawn in Section V.

II. Preliminary Considerations on the Generation of Jerk Continuous Signals

The main purpose of this work is to devise a new orientation planning primitive for trajectories in the operational space, so as to guarantee very smooth movements of the actuators. More precisely, the generated Cartesian trajectories must guarantee that the third order derivatives of the corresponding joint trajectories, i.e., $\dddot{\mathbf{q}}$, are continuous.

Given a manipulator with n degrees of freedom, a direct relationship exists between a point in the configuration space, i.e., $\mathbf{q} \in \mathbb{R}^n$, and the corresponding point in the operational space. In this work, points in the operational space are represented through a position $\mathbf{p} \in \mathbb{R}^3$ and an orientation $\boldsymbol{\varepsilon} \in \mathbb{R}^4$, the latter expressed through the unit quaternion representation. As a consequence, the continuity on $\mathbf{q}$ implies, in turn, the continuity on both $\mathbf{p}$ and $\boldsymbol{\varepsilon}$.

Analogously, velocities in the two spaces are correlated through the Jacobian matrix $\mathbf{J}(\mathbf{q})$, according to the following expression

$$\mathbf{v}_N = \mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}}, \tag{1}$$

where $\mathbf{v}_N := [\dot{\mathbf{p}}^T \boldsymbol{\omega}^T]^T \in \mathbb{R}^6$ is the generalized velocity vector, which contains the linear and the angular velocities of the end-effector. The applications considered in this work involve non-redundant systems, so that their Jacobian matrices, far from singularities, are invertible. Consequently, (1) can be rewritten as follows

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\,\mathbf{v}_N. \tag{2}$$

Equation (2) makes it possible to assert that the continuity of $\dot{\mathbf{q}}$ is achieved by guaranteeing the continuity on $\mathbf{v}_N$.

Similar considerations hold for the higher order derivatives. $\mathbf{J}(\mathbf{q})$ is continuously differentiable, so that some algebraic manipulations on the derivatives of (1) w.r.t. the time, lead to the following expressions

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\left(\mathbf{a}_N - \dot{\mathbf{J}}\,\dot{\mathbf{q}}\right)$$

$$\dddot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\left(\mathbf{j}_N - \ddot{\mathbf{J}}(\mathbf{q})\,\dot{\mathbf{q}} - 2\dot{\mathbf{J}}(\mathbf{q})\,\ddot{\mathbf{q}}\right)$$

where $\mathbf{a}_N := [\ddot{\mathbf{p}}^T \boldsymbol{\alpha}^T]^T$ and $\mathbf{j}_N := [\dddot{\mathbf{p}}^T \boldsymbol{\iota}^T]^T$ are the first and second time derivatives of $\mathbf{v}_N$ which, necessarily, need to be continuous in order to guarantee the continuity of $\ddot{\mathbf{q}}$ and of $\dddot{\mathbf{q}}$.

The aforementioned considerations indicate that $\mathcal{C}^3$ trajectories in the joint space can be achieved by means of trajectories in the Cartesian space which fulfill some specific path and orientation properties. A complete analysis on the path characteristics can be found in [9] and it is here omitted for conciseness. Conversely, in this letter the attention is focused on the orientation planning problem: specific conditions for the generation of smooth profiles are provided in the following.

The end-effector orientation can be expressed through the Euler parameters $\boldsymbol{\varepsilon} := [\varepsilon_0\, \varepsilon_1\, \varepsilon_2\, \varepsilon_3]^T$. As known, the following condition applies

$$||\boldsymbol{\varepsilon}|| = 1.$$

A direct relationship between $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\varepsilon}}$ can be found according to the following reasoning. The end-effector orientation can be represented through a rotation matrix $\mathbf{R}$ or, alternatively, through $\boldsymbol{\varepsilon}$. The two representations are each other correlated by the following expression

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \left[\begin{array}{c|c|c} 1-2\varepsilon_2^2-2\varepsilon_3^2 & 2(\varepsilon_1\varepsilon_2-\varepsilon_3\varepsilon_0) & 2(\varepsilon_1\varepsilon_3+\varepsilon_2\varepsilon_0) \\ 2(\varepsilon_1\varepsilon_2+\varepsilon_3\varepsilon_0) & 1-2\varepsilon_1^2-2\varepsilon_3^2 & 2(\varepsilon_2\varepsilon_3-\varepsilon_1\varepsilon_0) \\ 2(\varepsilon_1\varepsilon_3-\varepsilon_2\varepsilon_0) & 2(\varepsilon_2\varepsilon_3+\varepsilon_1\varepsilon_0) & 1-2\varepsilon_1^2-2\varepsilon_2^2 \end{array}\right]. \tag{3}$$

As known, the following relation applies

$$\mathbf{S}(\boldsymbol{\omega}) = \dot{\mathbf{R}}\,\mathbf{R}^T = \begin{bmatrix} \dot{r}_{11} & \dot{r}_{12} & \dot{r}_{13} \\ \dot{r}_{21} & \dot{r}_{22} & \dot{r}_{23} \\ \dot{r}_{31} & \dot{r}_{32} & \dot{r}_{33} \end{bmatrix} \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \tag{4}$$

where

$$\mathbf{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

is a skew-symmetric matrix which depends on the angular velocity of the end-effector $\boldsymbol{\omega} := [\omega_x\, \omega_y\, \omega_z]^T$.

The components of $\boldsymbol{\omega}$ can be extracted from (4). In particular, a few algebraic manipulations make it possible to write

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} r_{21}\,\dot{r}_{31} + r_{22}\,\dot{r}_{32} + r_{23}\,\dot{r}_{33} \\ r_{31}\,\dot{r}_{11} + r_{32}\,\dot{r}_{12} + r_{33}\,\dot{r}_{13} \\ r_{11}\,\dot{r}_{21} + r_{12}\,\dot{r}_{22} + r_{13}\,\dot{r}_{23} \end{bmatrix}. \tag{5}$$

By substituting the terms of (3) into (5), after a few algebraic manipulations the following relationship between $\dot{\boldsymbol{\varepsilon}}$ and $\boldsymbol{\omega}$ is finally obtained

$$\boldsymbol{\omega} = 2\mathbf{M}(\boldsymbol{\varepsilon})\dot{\boldsymbol{\varepsilon}}, \tag{6}$$

where

$$\mathbf{M}(\boldsymbol{\varepsilon}) := \begin{bmatrix} -\varepsilon_1 & \varepsilon_0 & -\varepsilon_3 & \varepsilon_2 \\ -\varepsilon_2 & \varepsilon_3 & \varepsilon_0 & -\varepsilon_1 \\ -\varepsilon_3 & -\varepsilon_2 & \varepsilon_1 & \varepsilon_0 \end{bmatrix}. \tag{7}$$

The inverse relationship between $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\varepsilon}}$ can be obtained according to the following reasoning. Let us add an additional row to (7), so as to obtain the following matrix

$$\mathbf{M}^*(\boldsymbol{\varepsilon}) := \begin{bmatrix} \varepsilon_0 & \varepsilon_1 & \varepsilon_2 & \varepsilon_3 \\ -\varepsilon_1 & \varepsilon_0 & -\varepsilon_3 & \varepsilon_2 \\ -\varepsilon_2 & \varepsilon_3 & \varepsilon_0 & -\varepsilon_1 \\ -\varepsilon_3 & -\varepsilon_2 & \varepsilon_1 & \varepsilon_0 \end{bmatrix}.$$

It can be easily proved that $\boldsymbol{\varepsilon}$ and $\dot{\boldsymbol{\varepsilon}}$ are each other orthogonal, so that $\boldsymbol{\varepsilon}^T\dot{\boldsymbol{\varepsilon}} = 0$. Consequently, (6) can be rewritten as follows

$$\begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = 2\mathbf{M}^*(\boldsymbol{\varepsilon})\,\dot{\boldsymbol{\varepsilon}}. \tag{8}$$

Matrix $\mathbf{M}^*(\boldsymbol{\varepsilon})$ is non-singular since it can be verified that

$$det(\mathbf{M}^*(\boldsymbol{\varepsilon})) = \left(\varepsilon_0^2+\varepsilon_1^2+\varepsilon_2^2+\varepsilon_3^2\right)^2 = ||\boldsymbol{\varepsilon}||^4 = 1,$$

so that $\mathbf{M}^*(\boldsymbol{\varepsilon})^{-1}$ always exists. Additionally, $\mathbf{M}^*(\boldsymbol{\varepsilon})$ is orthonormal, i.e. $\mathbf{M}^*(\boldsymbol{\varepsilon})^{-1} = \mathbf{M}^*(\boldsymbol{\varepsilon})^T$. Indeed, the following condition applies

$$\mathbf{M}^*(\boldsymbol{\varepsilon})^T\mathbf{M}^*(\boldsymbol{\varepsilon}) = \begin{bmatrix} ||\boldsymbol{\varepsilon}||^2 & 0 & 0 & 0 \\ 0 & ||\boldsymbol{\varepsilon}||^2 & 0 & 0 \\ 0 & 0 & ||\boldsymbol{\varepsilon}||^2 & 0 \\ 0 & 0 & 0 & ||\boldsymbol{\varepsilon}||^2 \end{bmatrix} = I^{4\times 4}.$$

As a consequence, (8) can be inverted as follows

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\mathbf{M}^*(\boldsymbol{\varepsilon})^T \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix},$$

or, more compactly, as follows

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\mathbf{M}(\boldsymbol{\varepsilon})^T\,\boldsymbol{\omega}, \tag{9}$$

where $\mathbf{M}(\boldsymbol{\varepsilon})$ is given by (7).

Due to (6) and (9), it can be asserted that the continuity of $\boldsymbol{\omega}$ is achieved through the continuity of $\dot{\boldsymbol{\varepsilon}}$.

By differentiating (6) the following equations are further obtained

$$\boldsymbol{\alpha} = 2\left[\mathbf{M}(\boldsymbol{\varepsilon})\ddot{\boldsymbol{\varepsilon}} + \mathbf{M}(\dot{\boldsymbol{\varepsilon}})\dot{\boldsymbol{\varepsilon}}\right],$$
$$\boldsymbol{\iota} = 2\left[\mathbf{M}(\boldsymbol{\varepsilon})\dddot{\boldsymbol{\varepsilon}} + 2\mathbf{M}(\dot{\boldsymbol{\varepsilon}})\ddot{\boldsymbol{\varepsilon}} + \mathbf{M}(\ddot{\boldsymbol{\varepsilon}})\dot{\boldsymbol{\varepsilon}}\right],$$

where

$$\mathbf{M}(\dot{\boldsymbol{\varepsilon}}) := \begin{bmatrix} -\dot{\varepsilon}_1 & \dot{\varepsilon}_0 & -\dot{\varepsilon}_3 & \dot{\varepsilon}_2 \\ -\dot{\varepsilon}_2 & \dot{\varepsilon}_3 & \dot{\varepsilon}_0 & -\dot{\varepsilon}_1 \\ -\dot{\varepsilon}_3 & -\dot{\varepsilon}_2 & \dot{\varepsilon}_1 & \dot{\varepsilon}_0 \end{bmatrix},$$

$$\mathbf{M}(\ddot{\boldsymbol{\varepsilon}}) := \begin{bmatrix} -\ddot{\varepsilon}_1 & \ddot{\varepsilon}_0 & -\ddot{\varepsilon}_3 & \ddot{\varepsilon}_2 \\ -\ddot{\varepsilon}_2 & \ddot{\varepsilon}_3 & \ddot{\varepsilon}_0 & -\ddot{\varepsilon}_1 \\ -\ddot{\varepsilon}_3 & -\ddot{\varepsilon}_2 & \ddot{\varepsilon}_1 & \ddot{\varepsilon}_0 \end{bmatrix},$$

i.e., the continuity on $\boldsymbol{\alpha}$ and $\boldsymbol{\iota}$ is mapped into equivalent continuity conditions on $\ddot{\boldsymbol{\varepsilon}}$ and $\dddot{\boldsymbol{\varepsilon}}$.

III. The $\mathcal{C}^3$ Continuous Orientation Planner

According to the discussion in Section II, the $\mathcal{C}^3$ continuity of the trajectory in the configuration space imposes, in turn, that also the planning primitive adopted for $\boldsymbol{\varepsilon}$ must be $\mathcal{C}^3$. In order to guarantee a strict relationship between orientation and curvilinear coordinate, $\boldsymbol{\varepsilon}$ is planned as a function of the curvilinear coordinate s. Consequently, a trajectory for $\boldsymbol{\varepsilon}$ is obtained by

combining $\boldsymbol{\varepsilon}(s)$ with a timing law $s(t)$. The time derivatives of the orientation trajectory can be expressed as follows

$$\dot{\boldsymbol{\varepsilon}} = \frac{d\boldsymbol{\varepsilon}}{dt} = \frac{d\boldsymbol{\varepsilon}}{ds}\frac{ds}{dt} = \dot{s}\boldsymbol{\varepsilon}',$$
$$\ddot{\boldsymbol{\varepsilon}} = \dot{s}^2\boldsymbol{\varepsilon}'' + \ddot{s}\boldsymbol{\varepsilon}',$$
$$\dddot{\boldsymbol{\varepsilon}} = \dot{s}^3\boldsymbol{\varepsilon}''' + 3\ddot{s}\dot{s}\boldsymbol{\varepsilon}'' + \dddot{s}\boldsymbol{\varepsilon}',$$

where $\boldsymbol{\varepsilon}' := (d\boldsymbol{\varepsilon})/(ds)$, $\boldsymbol{\varepsilon}'' := (d^2\boldsymbol{\varepsilon})/(ds^2)$, and $\boldsymbol{\varepsilon}''' := (d^3\boldsymbol{\varepsilon})/(ds^3)$. Clearly, the required $\mathcal{C}^3$ continuity on $\dddot{\boldsymbol{\varepsilon}}$ is achieved if $\boldsymbol{\varepsilon}(s)$ and $s(t)$ are both $\mathcal{C}^3$ continuous. For conciseness, the dependency of $\boldsymbol{\varepsilon}$ on s is dropped in the following.

The orientation primitive could be potentially planned by only considering components ε_1, ε_2, and ε_3. The fourth component, i.e. ε_0 could be subsequently obtained by imposing condition $\|\boldsymbol{\varepsilon}\| = 1$. However, such planning strategy would lead to complex expressions for $\boldsymbol{\varepsilon}'$, $\boldsymbol{\varepsilon}''$, and $\boldsymbol{\varepsilon}'''$. For such reason, the four components of the quaternion are independently planned. The resulting vector, i.e., $\bar{\boldsymbol{\varepsilon}}$, is later normalized in order to fulfill condition $\|\boldsymbol{\varepsilon}\| = 1$ by means of the following equation

$$\boldsymbol{\varepsilon} = \frac{\bar{\boldsymbol{\varepsilon}}}{\|\bar{\boldsymbol{\varepsilon}}\|}.$$

The derivatives of $\boldsymbol{\varepsilon}$ w.r.t. s can be easily obtained through some algebraic manipulations. More precisely, it can be easily proved that they can be represented as follows

$$\boldsymbol{\varepsilon}' = \frac{\bar{\boldsymbol{\varepsilon}}'}{\|\bar{\boldsymbol{\varepsilon}}\|} + \alpha\bar{\boldsymbol{\varepsilon}}, \tag{10}$$

$$\boldsymbol{\varepsilon}'' = \frac{\bar{\boldsymbol{\varepsilon}}''}{\|\bar{\boldsymbol{\varepsilon}}\|} + 2\alpha\bar{\boldsymbol{\varepsilon}}' + \beta\bar{\boldsymbol{\varepsilon}}, \tag{11}$$

$$\boldsymbol{\varepsilon}''' = \frac{\bar{\boldsymbol{\varepsilon}}'''}{\|\bar{\boldsymbol{\varepsilon}}\|} + 3\alpha\bar{\boldsymbol{\varepsilon}}'' + 3\beta\bar{\boldsymbol{\varepsilon}}' + \gamma\bar{\boldsymbol{\varepsilon}}, \tag{12}$$

where

$$\alpha = \frac{d}{ds}\left(\frac{1}{\|\bar{\boldsymbol{\varepsilon}}\|}\right) = -\frac{\bar{\boldsymbol{\varepsilon}}^T\bar{\boldsymbol{\varepsilon}}'}{\|\bar{\boldsymbol{\varepsilon}}\|^3}, \tag{13}$$

$$\beta = \frac{d^2}{ds^2}\left(\frac{1}{\|\bar{\boldsymbol{\varepsilon}}\|}\right) = -\frac{\|\bar{\boldsymbol{\varepsilon}}'\|^2}{\|\bar{\boldsymbol{\varepsilon}}\|^3} - \frac{\bar{\boldsymbol{\varepsilon}}^T\bar{\boldsymbol{\varepsilon}}''}{\|\bar{\boldsymbol{\varepsilon}}\|^3} + 3\frac{(\bar{\boldsymbol{\varepsilon}}^T\bar{\boldsymbol{\varepsilon}}')^2}{\|\bar{\boldsymbol{\varepsilon}}\|^5}, \tag{14}$$

$$\gamma = \frac{d^3}{ds^3}\left(\frac{1}{\|\bar{\boldsymbol{\varepsilon}}\|}\right) = -3\frac{\bar{\boldsymbol{\varepsilon}}'^T\bar{\boldsymbol{\varepsilon}}''}{\|\bar{\boldsymbol{\varepsilon}}\|^3} - \frac{\bar{\boldsymbol{\varepsilon}}^T\bar{\boldsymbol{\varepsilon}}'''}{\|\bar{\boldsymbol{\varepsilon}}\|^3} + 9\frac{(\bar{\boldsymbol{\varepsilon}}^T\bar{\boldsymbol{\varepsilon}}'' + \|\bar{\boldsymbol{\varepsilon}}'\|^2)(\bar{\boldsymbol{\varepsilon}}^T\bar{\boldsymbol{\varepsilon}}')}{\|\bar{\boldsymbol{\varepsilon}}\|^5} - 15\frac{(\bar{\boldsymbol{\varepsilon}}^T\bar{\boldsymbol{\varepsilon}}')^3}{\|\bar{\boldsymbol{\varepsilon}}\|^7}. \tag{15}$$

Under the hypothesis that $\|\bar{\boldsymbol{\varepsilon}}\| \neq 0$, and since (10)–(15) only contains sums, products, and ratios of continuous functions, according to [20] (see p. 59) one can assert that continuity properties of $\bar{\boldsymbol{\varepsilon}}$ are inherited by $\boldsymbol{\varepsilon}$, i.e., if $\bar{\boldsymbol{\varepsilon}}$ is $\mathcal{C}^3$, then $\boldsymbol{\varepsilon}$ is $\mathcal{C}^3$ as well.

According to the premises, the components of $\bar{\boldsymbol{\varepsilon}}$ are planned independently. Thus, a scalar primitive is adopted for each of them: in the reminder of this section it is indicated as $f(s)$.

Given a set of $n+1$ via-points points, whose displacement along the curve is given through the corresponding curvilinear coordinates $\mathbf{s} = \{0, s_1, \ldots, s_n\}$, function $f(s)$ can be partitioned into a set of n sub-functions defined as follows

$$f(s) := \begin{cases} f_1(s) & \text{if } 0 \leq s \leq s_1 \\ f_2(s - s_1) & \text{if } s_1 < s \leq s_2 \\ \ldots & \\ f_n(s - s_{n-1}) & \text{if } s_{n-1} < s \leq s_n \end{cases}. \tag{16}$$

By defining $\hat{s} := s - s_{k-1}$, functions $f_k(\cdot)$ in (16) can be rewritten as follows

$$f_k(\hat{s}), \quad 0 < \hat{s} \leq \hat{s}_k,$$

where $\hat{s}_k := s_k - s_{k-1}$.

Many alternative primitives can be adopted for $f_k(\hat{s})$, $k = 1, 2, \ldots, n$. A nice survey is proposed in [21] for the achievement of composite $\mathcal{C}^2$ function. In this letter, a possible strategy is devised in order to achieve the required $\mathcal{C}^3$ continuity. Function $f_k(\hat{s})$ is defined as follows

$$f_k(\hat{s}) = a_{0\,k} + a_{1\,k}\hat{s} + a_{2\,k}\hat{s}^2 + a_{3\,k}\hat{s}^3 + a_{4\,k}\hat{s}^4 + a_{5\,k}\hat{s}^5.$$

The polynomials coefficients can be evaluated by solving a linear system which can be set up by considering a proper set of interpolating conditions. For example, since the orientations at the via-points are assigned, $\hat{f}_0 := f_1(0)$, $\hat{f}_k := f_k(\hat{s}_k), k = 1, 2, \ldots, n$ are known and must be satisfied by the polynomials. In the strategy next proposed, jerks at the beginning and at the end of each segment of the composite trajectory are posed equal to zero in order to limit the signal variability. Furthermore, two additional free-displacement points are added in the middle of the first and of the last intervals of the composite function (see also [21]) in order to balance the number of equations and unknowns. This implies that $\bar{n} = n + 2$ polynomial functions must be generated.

The following conditions must be satisfied in order to generate a $\mathcal{C}^3$ composite function

- imposition of the initial and final values for each segment of the composite curve ($2\bar{n} - 4$ conditions)

$$f_1(0) = \hat{f}_0$$
$$f_k(\hat{s}_k) = f_{k+1}(0) = \hat{f}_k, \quad k = 2, 3, \ldots, \bar{n} - 2,$$
$$f_n(\hat{s}_{\bar{n}}) = \hat{f}_{\bar{n}}.$$

  It is worth noticing that the positions of the free-displacement points have not been assigned;
- imposition of the composite function continuity at the free-displacement points (2 conditions)

$$f_1(\hat{s}_1) = f_2(0), \quad f_{\bar{n}-1}(\hat{s}_{\bar{n}-1}) = f_{\bar{n}}(0);$$

- imposition of the speed and the acceleration continuity [$2(\bar{n} - 1)$ conditions]

$$f_k'(\hat{s}_k) = f_{k+1}'(0), \quad k = 1, 2, \ldots, \bar{n} - 1,$$
$$f_k''(\hat{s}_k) = f_{k+1}''(0), \quad k = 1, 2, \ldots, \bar{n} - 1;$$

- imposition of null jerks at the beginning and at the end of each segment ($2\bar{n}$ conditions)

$$f_k'''(0) = f_k'''(\hat{s}_k) = 0, \quad k = 1, 2, \ldots, \bar{n};$$

- imposition of the speeds and the accelerations at the beginning and end of the composite curve (4 conditions)

$$f_1'(0) = \hat{f}_0', \qquad f_1''(0) = \hat{f}_0'',$$
$$f_{\bar{n}}'(\hat{s}_{\bar{n}}) = \hat{f}_{\bar{n}}', \qquad f_{\bar{n}}''(\hat{s}_{\bar{n}}) = \hat{f}_{\bar{n}}''.$$

Summarizing, the planning problem requires solving the following linear system made of $6\bar{n}$ equations in $6\bar{n}$ unknowns

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{17}$$

where $\mathbf{x} := [a_{01}\, a_{11}\, a_{21}\, a_{31}\, a_{02}\, a_{12}\, \ldots\, a_{\bar{n}\bar{n}}]^T \in \mathbb{R}^{6\bar{n}}$ is the vector of the unknowns, $\mathbf{A} \in \mathbb{R}^{6\bar{n}\times 6\bar{n}}$, and $\mathbf{b} \in \mathbb{R}^{6\bar{n}}$. Some algebraic manipulations make it possible to reduce the order of (17), so that $\mathbf{A}$ is converted into the following three-diagonal matrix

$$\mathbf{A} = \begin{bmatrix} 20\hat{s}_1 & 3\hat{s}_1 & & & \\ 20\hat{s}_1 & 10\hat{s}_1 + 7\hat{s}_3 & 3\hat{s}_3 & & \\ \ddots & \ddots & \ddots & \ddots & \\ & 3\hat{s}_k & 7(\hat{s}_k + \hat{s}_{k+1}) & 3\hat{s}_{k+1} & \\ & \ddots & \ddots & \ddots & \\ & & 3\hat{s}_{\bar{n}-2} & 7\hat{s}_{\bar{n}-2} + 10\hat{s}_{\bar{n}} & 20\hat{s}_{\bar{n}} \\ & & & 3\hat{s}_{\bar{n}} & 20\hat{s}_{\bar{n}} \end{bmatrix}$$

with $\mathbf{A} \in \mathbb{R}^{(\bar{n}-1)\times(\bar{n}-1)}$. Analogously, the structure of $\mathbf{b} \in \mathbb{R}^{\bar{n}-1}$ changes as follows

$$\mathbf{b} = \begin{bmatrix} 10\frac{\hat{f}_2 - \hat{f}_0}{\hat{s}_1} - 20\hat{f}_0' - \frac{17}{2}\hat{f}_0''\hat{s}_1 \\ 10\frac{\hat{f}_3 - \hat{f}_2}{\hat{s}_3} - 10\hat{f}_0' - 5\hat{f}_0''\hat{s}_1 \\ \vdots \\ 10\frac{\hat{f}_{k+1} - \hat{f}_k}{\hat{s}_{k+1}} - 10\frac{\hat{f}_k - \hat{f}_{k-1}}{\hat{s}_k} \\ \vdots \\ -10\frac{\hat{f}_{\bar{n}-2} - \hat{f}_{\bar{n}-3}}{\hat{s}_{\bar{n}-2}} + 10\hat{f}_{\bar{n}}' - 5\hat{f}_{\bar{n}}''\hat{s}_{\bar{n}} \\ -10\frac{\hat{f}_{\bar{n}} - \hat{f}_{\bar{n}-2}}{\hat{s}_{\bar{n}}} + 20\hat{f}_{\bar{n}}' - \frac{17}{2}\hat{f}_{\bar{n}}''\hat{s}_{\bar{n}} \end{bmatrix} \in \mathbb{R}^{\bar{n}-1}.$$

and $\mathbf{x} = [a_{22}\, a_{23}\, \ldots\, a_{2\bar{n}}]^T \in \mathbb{R}^{\bar{n}-1}$ becomes the vector of the unknowns.

Efficient algorithms exist for the inversion of three-diagonal matrices [22], so that large values of $\bar{n}$ can be easily handled. Once the reduced order system has been solved, the remaining coefficients of the polynomials can be obtained through the following closed form equations

$$a_{01} = \hat{f}_0, \tag{18}$$

$$a_{02} = \frac{6a_{22} + 7\hat{f}_0''}{20}\hat{s}_1^2 + \hat{f}_0'\hat{s}_1 + \hat{f}_0, \tag{19}$$

$$a_{0j} = \hat{f}_{j-1}, \quad j = 3, 4, \ldots, \bar{n}-1, \tag{20}$$

$$a_{0\bar{n}} = \frac{6a_{2\bar{n}} + 7\hat{f}_{\bar{n}}''}{20}\hat{s}_{\bar{n}}^2 + \hat{f}_{\bar{n}}'\hat{s}_{\bar{n}} + \hat{f}_{\bar{n}}, \tag{21}$$

$$a_{11} = \hat{f}_0', \tag{22}$$

$$a_{12} = -\frac{20a_{22} + 6a_{23} + 7\hat{f}_0''}{20}\hat{s}_1 - \hat{f}_0' - \frac{\hat{f}_0 - \hat{f}_2}{\hat{s}_1}, \tag{23}$$

$$a_{1j} = -\frac{7a_{2j} + 3a_{2(j+1)}}{10}\hat{s}_j + \frac{\hat{f}_j - \hat{f}_{j-1}}{\hat{s}_j},$$
$$j = 3, 4, \ldots, \bar{n}-2, \tag{24}$$

$$a_{1(\bar{n}-1)} = -7\frac{2a_{2(\bar{n}-1)} - \hat{f}_{\bar{n}}''}{20}\hat{s}_{\bar{n}} - \hat{f}_{\bar{n}}' + \frac{\hat{f}_{\bar{n}} - \hat{f}_{\bar{n}-2}}{\hat{s}_{\bar{n}}}, \tag{25}$$

$$a_{1\bar{n}} = -\frac{2a_{2\bar{n}} + \hat{f}_{\bar{n}}''}{2}\hat{s}_{\bar{n}} + \hat{f}_{\bar{n}}', \tag{26}$$

$$a_{21} = \frac{1}{2}\hat{f}_0'', \tag{27}$$

$$a_{3j} = 0, \quad j = 1, 2, \ldots, \bar{n}, \tag{28}$$

$$a_{41} = \frac{2a_{22} - \hat{f}_0''}{4\hat{s}_1^2}, \tag{29}$$

$$a_{4j} = \frac{a_{2(j+1)} - a_{2j}}{2\hat{s}_j^2}, \quad j = 2, 3, \ldots, \bar{n}-1, \tag{30}$$

$$a_{4\bar{n}} = \frac{\hat{f}_{\bar{n}}'' - 2a_{2\bar{n}}}{4\hat{s}_{\bar{n}}^2}, \tag{31}$$

$$a_{51} = \frac{\hat{f}_0'' - 2a_{22}}{10\hat{s}_1^3}, \tag{32}$$

$$a_{5j} = \frac{a_{2j} - a_{2j+1}}{5\hat{s}_j^3}, \quad j = 2, 3, \ldots, \bar{n}-1, \tag{33}$$

$$a_{5\hat{n}} = \frac{2a_{2\hat{n}} - \hat{f}_{\bar{n}}''}{10\hat{s}_{\bar{n}}^3}. \tag{34}$$

The aforementioned solution applies for $\bar{n} > 4$, i.e., when the problem admits more than 3 planning points. If $\bar{n} = 4$, matrices $\mathbf{A}$ and $\mathbf{b}$ of (17) change as follows

$$\mathbf{b} = \begin{bmatrix} 10\frac{\hat{f}_2 - \hat{f}_0}{\hat{s}_1} - 20\hat{f}_0' - \frac{17}{2}\hat{f}_0''\hat{s}_1 \\ 2(\hat{f}_4' - \hat{f}_0') - \hat{f}_0''\hat{s}_1 - \hat{f}_4''\hat{s}_4 \\ -10\frac{\hat{f}_4 - \hat{f}_2}{\hat{s}_4} + 20\hat{f}_4' - \frac{17}{2}\hat{f}_4''\hat{s}_4 \end{bmatrix},$$

$$\mathbf{A} = \begin{bmatrix} 20\hat{s}_1 & 3\hat{s}_1 & 0 \\ 4\hat{s}_1 & 2(\hat{s}_1 + \hat{s}_4) & 4\hat{s}_4 \\ 0 & 3\hat{s}_4 & 20\hat{s}_4 \end{bmatrix},$$

while $\mathbf{x} = [a_{22}\, a_{23}\, a_{24}]^T \in \mathbb{R}^3$. The remaining coefficients are still obtained through (18)–(34) by assuming $\bar{n} = 4$.

Remark 1: As previously stated, condition $\|\bar{\varepsilon}\| \neq 0$ must be strictly satisfied so as to guarantee that normalized ε is $\mathcal{C}^3$. Possible violations could be caused by the independent planning of the 4 components of $\bar{\varepsilon}$. However, in applications like the ones considered in this work, adjacent via-points are each other very close and their orientation changes are minimal: along each segment of the composite path the components of $\bar{\varepsilon}$ change minimally and almost linearly, so that $\|\bar{\varepsilon}\| \simeq 1$ along the whole trajectory (see also Fig. 7 in Section IV).

IV. Simulations and Experimental Results

In order to obtain a $\mathcal{C}^3$ timing law, a jerk-continuous $s(t)$ profile is needed. Typically, Cartesian trajectories are used for applications which require a constant speed of the tool frame

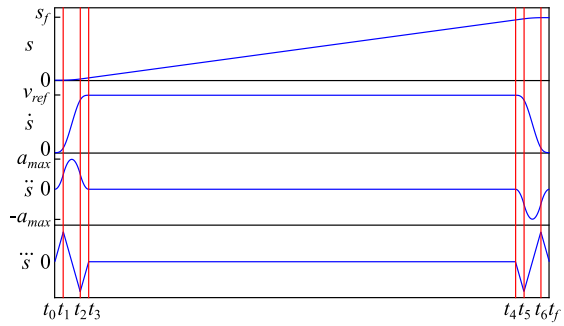

Fig. 2. The timing law used for the experiments.

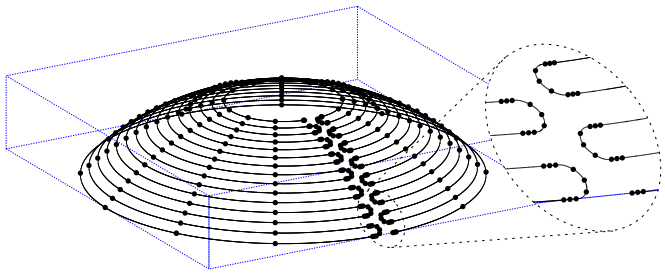

Fig. 3. The trajectory adopted for the CNC example. Black dots indicate the assigned via-points, the end-effector longitudinal axis (which is not shown in the figure) is aligned to the sphere radius.

(machining, gluing, soldering, laser cutting, etc.). For this reason, a simple timing law, like the one shown in Fig. 2, was adopted for the experiments. More complex shapes can be assumed for alternative tasks, provided that the resulting function is $\mathcal{C}^3$: for example, in order to satisfy a set of given velocity or acceleration limits, a trajectory scaling strategy like the one proposed in [23] can be used. The timing law planning problem is not considered in this letter, so that no further details are given.

The orientation planner was tested by means of simulations concerning a CNC application and by executing a real trajectory with a Comau Smart SiX 6.14 manipulator, a six link anthropomorphic robot. More precisely, the simulated test concerned a sphere milling problem and was conceived so as to prove the effectiveness of the planner for machining purposes. The experiment on the real manipulator consisted in the generation and the execution of two composite trajectories. The end-effector path was generated by means of the η^{3D}-splines planner proposed in [9], while the orientation was provided through the primitive proposed in this letter.

In the simulated test case, a milling trajectory was planned for a spherical surface. Fig. 3 shows its shape. When finishing an object, the perpendicularity between the milling cutter and the surface – or the preservation of a given angle between them – is critically important in order to guarantee both efficiency (in terms of chip rate, bit temperature, etc.) and surface quality. The proposed example assumes that the tool tip axis is always aligned with the radius of the sphere, so that the planner effectiveness is checked by considering the error angle between the axis of the milling tool and the normal vector of the surface. It was verified that the largest errors occur when the path direction suddenly changes: at the turning points, the milling precision can be improved by increasing the density of via-points, as shown by the detail in Fig. 3. A maximum error equal to $6.5 \cdot 10^{-4}$ rad was

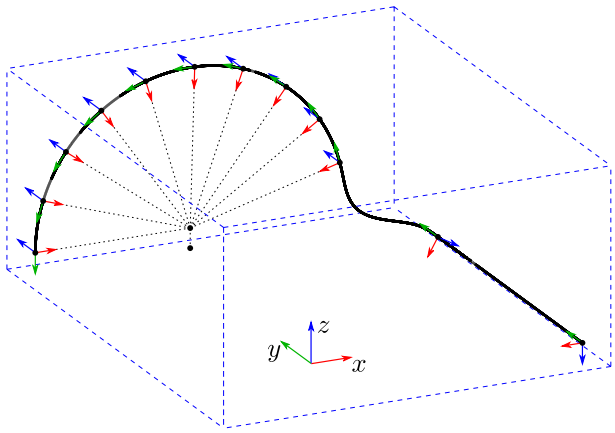

Fig. 4. The trajectory adopted for the first experiment on the manipulator. Black dots indicate the assigned via points, while the reference frames highlight the desired end-effector orientations. The end-effector longitudinal axis is aligned to the $\hat{\mathbf{z}}$ unit vector.

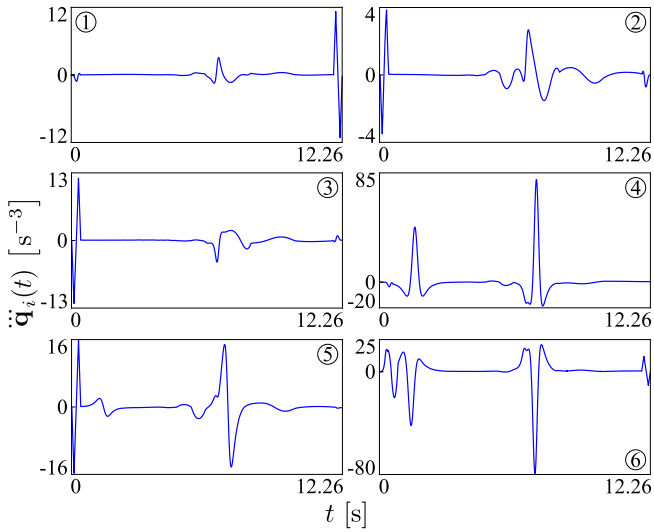

Fig. 5. First experiment: time profiles of the jerks associated to the six joints. The continuity condition is satisfied for all of them.

achieved along the whole path by assuming the via-points shown in Fig. 3: apart from the changes of direction, a very limited number of via-points are required to mill the entire surface.

As noted earlier, the second set of experiments involved a real industrial manipulator. Two different trajectories were considered. As shown in Fig. 4, the first trajectory is composed by a circular arc immediately followed by a straight segment. The same figure also shows the assigned orientations of the end-effector frame at the via-points. The two paths are joined through a Cartesian curve still obtained by means of an η^{3D}-spline curve. The whole path is $\mathcal{G}^3$. The jerk reference signals for the six joints, obtained by means of the proposed planner, are shown in Fig. 5: as expected, they are continuous.

The second manipulator experiment exploits a composite path made of circular arcs, linear segments, conic spirals, and helical curves generated by means of the η^{3D}-splines. As shown in Fig. 6, proper orientations have been assigned in each via-point of the path. In order to show the flexibility of the proposed planner, a constant orientation has been assumed for the first

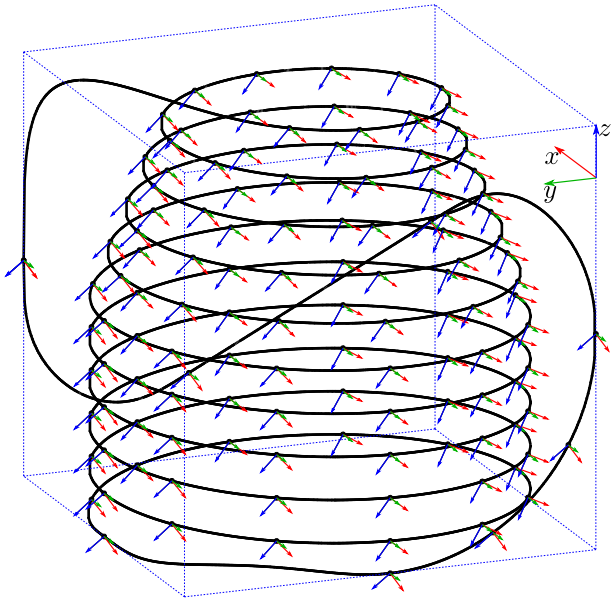

Fig. 6. The trajectory adopted for the second experiment on the manipulator. Black dots indicate the assigned via-points, while the reference frames highlight the desired end-effector orientations. The end-effector longitudinal axis is aligned to the $\hat{\mathbf{z}}$ unit vector.

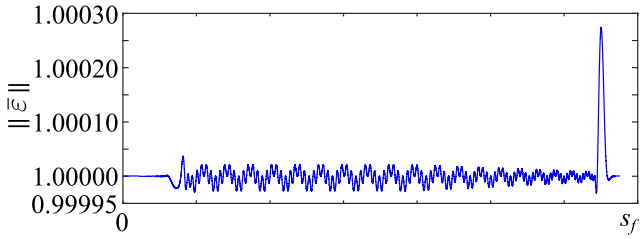

Fig. 7. Second experiment: $\|\bar{\varepsilon}\|$ is everywhere close to 1.

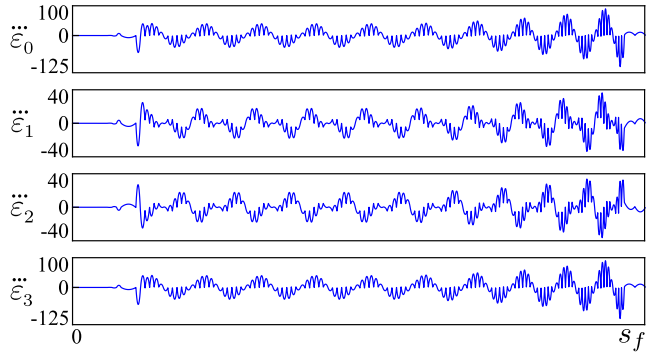

Fig. 8. Second experiment: the four components of $\dddot{\varepsilon}$. The continuity condition is satisfied for all of them.

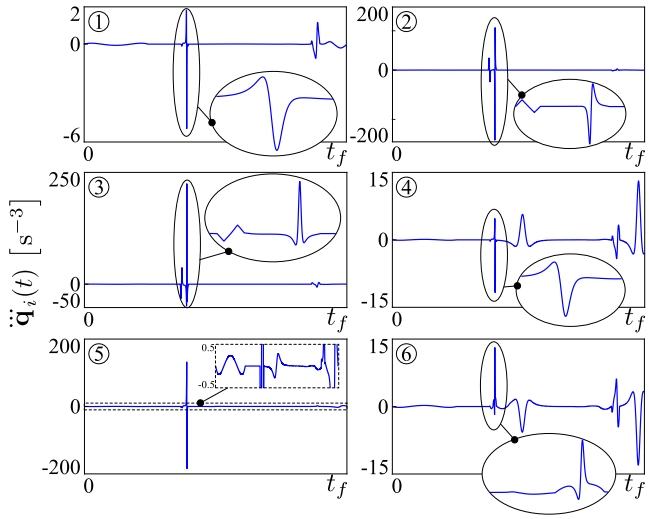

Fig. 9. Second experiment: jerks profiles associated to the six joints. The continuity condition is satisfied for all of them.

segments of the composite trajectory, while the tool orientation changes during the execution of the conic spirals and of the helical curves. More precisely, the tool-frame rotates around its own $\hat{\mathbf{y}}$ axis, so that its $\hat{\mathbf{x}}$ and $\hat{\mathbf{z}}$ axes remain confined inside a vertical plane.

Fig. 7 shows that, as expected, $\|\bar{\varepsilon}\|$ is everywhere very close to 1, even considering the final peak which occurs in a segment whose via-points are very far each other. Consequently, as shown in Fig. 8, ε is a continuous function and, it turn, joint jerks are continuous as well (see Fig. 9).

The acquired orientation errors, i.e. the angular differences between the unit vectors of the trajectory frame and the ones of the tool frame, these latter obtained from the encoders readings, are shown in Fig. 10. Thanks to the jerk continuity (the corresponding figure has been omitted for conciseness), the absolute values of the errors do not show evident peaks along the whole path but, conversely, they remain confined within a constant strip whose amplitude is equal to $4.2 \cdot 10^{-3}$ rad.

The orientation planner was conceived to be used in rapidly changing scenarios. Consequently, its computing times have been checked in order to quantify the computational burden. The planning algorithm was run by considering 998 test sets. More in detail, the nth set, with $n = 3, 4, \ldots, 1000$, was made of n randomly chosen via-points. Tests were executed on one single

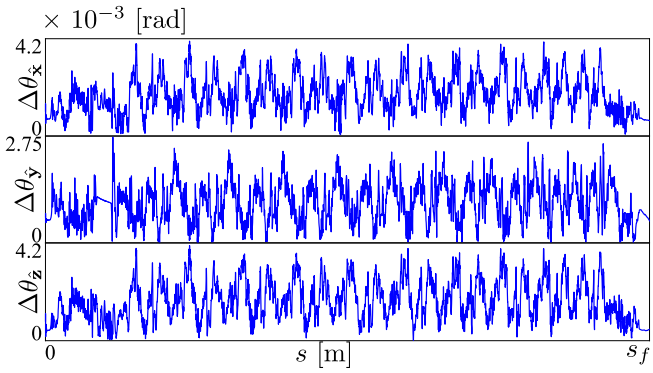

Fig. 10. Orientation errors of the tool-frame.

core of an Intel i7-1165G7 processor running at @2.80 GHz. The investigation made it possible to establish that the computational burden depends linearly on the number of via points. More precisely, the computational time is equal to $n \times 2.19 \cdot 10^{-7}$ s, so that the trajectory shown in Fig. 4, which is composed by 127 segments, is planned in $2.781 \cdot 10^{-5}$ s. Consequently, a trajectory made of 4500 points could be processed within 1 ms.

The attached multimedia files visually show the execution of the manipulator trajectories. A constant longitudinal speed equal to 0.15 ms^{-1} was assumed for the first experiment, while it was raised to 0.3 ms^{-1} for the second one. As shown by both videos, trajectories are fluently executed and no stops are required at the via-points.

V. Conclusion

The orientation planner proposed in this work makes it possible to generate, in a straightforward way, composite trajectories in the operational space. Its combination with the η^{3D}-splines, a recently devised path planning primitive, guarantees that the corresponding joint trajectories are $\mathcal{C}^3$. Consequently, smooth joint movements can be achieved and mechanical solicitations can be reduced. The computational burden of the novel primitive is particularly light, so that trajectories with many via-points can be easily processed even in real-time contexts. As shown in the letter, despite the orientation planner was initially conceived for applications involving industrial manipulators, it could also be profitably used for the management of CNC machines with more than 3 axes. The approach is evidently scalable, so that a future extension of the work could consider the generation of continuous-snap signals obtained by combining $\mathcal{G}^4$ path curves with $\mathcal{C}^4$ orientation primitives. Another potential future improvement of this work could concern the achievement of trajectories with a specified degree of smoothness, obtained by bounding joint velocities, accelerations, and jerks. Such result can be achieved, for example, by means of scaling techniques acting on the timing law [23].

References

[1] R. Campa and H. de la Torre, "Pose control of robot manipulators using different orientation representations: A comparative review," in *Proc. IEEE Amer. Control Conf.*, 2009, pp. 2855–2860.

[2] K. Shoemake, "Animating rotation with quaternion curves," in *Proc. 12th Conf. Comput. Graph. Interactive Techn.*, 1985, pp. 245–254.

[3] M.-C. Ho, Y.-R. Hwang, and C.-H. Hu, "Five-axis tool orientation smoothing using quaternion interpolation algorithm," *Int. J. Mach. Tools Manufacture*, vol. 43, pp. 1259–1267, Sep. 2003.

[4] R. Xu, X. Cheng, G. Zheng, and Z. Chen, "A tool orientation smoothing method based on machine rotary axes for five-axis machining with ball end cutters," *Int. J. Adv. Manuf. Technol.*, vol. 92, pp. 3615–3625, Oct. 2017.

[5] R. M. Grassmann and J. Burgner-Kahrs, "Quaternion-based smooth trajectory generator for via poses in $\boldsymbol{SE(3)}$ considering kinematic limits in cartesian space," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4192–4199, Oct. 2019.

[6] Y. Jixiang, L. Dingwei, Y. Congcong, and D. Han, "An analytical C^3 continuous tool path corner smoothing algorithm for 6R robot manipulator," *Robot. Comput.-Integr. Manuf.*, vol. 64, 2020, Art. no. 101947. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584519305976

[7] K. Shoemake, "Quaternion calculus and fast animation, computer animation: 3-D motion specification and control," in *Proc. Int. Conf. Comput. Graph. Interactive Techn.*, 1987, pp. 101–121.

[8] E. B. Dam, M. Koch, and M. Lillholm, "Quaternions, interpolation and animation," Datalogisk Institut, Københavns Universitet, Copenhagen, Denmark, DIKU-TR-98/5, 1998.

[9] A. Tagliavini and C. G. Lo Bianco, "η^{3D}-splines for the generation of 3D Cartesian paths with third order geometric continuity," *Robot. Comput.-Integr. Manuf.*, vol. 72, pp. 1–11, 2021.

[10] G. Legnani, I. Fassi, A. Tasora, and D. Fusai, "A practical algorithm for smooth interpolation between different angular positions," *Mechanism Mach. Theory*, vol. 162, 2021, Art. no. 104341. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094114X21000999

[11] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "A C^2-continous B-spline quaternion curve interpolating a given sequence of solid orientations," in *Proc. Comput. Animation*, 1995, pp. 72–81.

[12] W. Ge, Z. Huang, and G. Wang, "Interpolating solid orientations with a C^2-Continuous B-spline quaternion curve," in *Proc. Int. Conf. Technol. E- Learn. Digit. Entertainment*, 2007, pp. 606–615.

[13] Y. Pu, Y. Shi, X. Lin, Y. Hu, and Z. Li, "C^2-Continuous orientation planning for robot end-effector with B-spline curve based on logarithmic quaternion," *Math. Problems Eng.*, vol. 2020, Jul. 2020, Art. no. 2543824.

[14] G. Nielson, "υ–Quaternion splines for the smooth interpolation of orientations," *IEEE Trans. Vis. Comput. Graph.*, vol. 10, no. 2, pp. 224–229, Mar./Apr. 2004.

[15] Y. Liu, Z. Xie, Y. Gu, C. Fan, X. Zhao, and H. Liu, "Trajectory planning of robot manipulators based on unit quaternion," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatron.*, 2017, pp. 1249–1254.

[16] J. Tan, Y. Xing, W. Fan, and P. Hong, "Smooth orientation interpolation using parametric quintic-polynomial-based quaternion spline curve," *J. Comput. Appl. Math.*, vol. 329, pp. 256–267, 2018.

[17] Q.-B. Xiao, M. Wan, Y. Liu, X.-B. Qin, and W.-H. Zhang, "Space corner smoothing of CNC machine tools through developing 3D general clothoid," *Robot. Comput.-Integr. Manuf.*, vol. 64, 2020, Art. no. 101949.

[18] S. He, C. Yan, Y. Deng, C.-H. Lee, and X. Zhou, "A tolerance constrained G^2 continuous path smoothing and interpolation method for industrial SCARA robots," *Robot. Comput.-Integr. Manuf.*, vol. 63, 2020, Art. no. 101907. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584519303916

[19] X. Huang, F. Zhao, T. Tao, and X. Mei, "A newly developed corner smoothing methodology based on clothoid splines for high speed machine tools," *Robot. Comput.-Integr. Manuf.*, vol. 70, 2021, Art. no. 102106.

[20] I. N. Bronshtein, K. A. Semendyayev, G. Musiol, and H. Muehlig, *Handbook of Mathematics*, 5th ed. Berlin, Germany: Springer, 2007.

[21] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Berlin, Germany:Springer, 2008.

[22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed.Cambridge, U.K.: Cambridge Univ. Press, 1992.

[23] C. Guarino Lo Bianco, M. Faroni, M. Beschi, and A. Visioli, "A predictive technique for the real-time trajectory scaling under high-order constraints," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 1, pp. 315–326, Feb. 2022.