2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

Smooth Orientation Path Planning with Quaternions Using B-Splines*

Matthias Neubauer and Andreas Müller

Abstract— **Many robotics applications require smooth orientation planning, i.e. interpolation or approximation of a frame orientation through prescribed configurations such that the angular velocity and its time derivatives are smooth. This for instance ensures the continuity of the motor torques of robotic manipulator. Yet no satisfactory solution to this problem has been presented. This paper presents a solution using a B-spline parameterization of rotations. The method allows for a continuous interpolation and approximation through a given set of quaternions up to a specified order. The algorithm resembles the well-known B-spline interpolation in the sense that it boils down to solving a system of linear equations. The method is demonstrated for a polishing application (car fender) with an industrial robot.**

I. INTRODUCTION

Path planning is a topic in robotics that has been treated exhaustively but is still an important issue documented by recent publications such as [1], [2] where different planning strategies are discussed. However, there are various problems that have not been solved adequate, and prominently, orientation path planning is one of them.

This is addressed in this paper, and an orientation path planning algorithm is proposed that allows for smooth following through a set of prescribed orientations. Its relevance for robotics is apparent as general object guidance involves its orientation. In robotics applications in particular the continuity of angular velocities and accelerations is important.

Spatial rotations are represented by certain rotation parameters. The interpolation of rotations between two given orientations hence requires tailored interpolation of the respective parameters. Clearly, the effect of interpolating the parameters on the actual rotation depends on the particular parameterizations. In [11] and [3] advantages and disadvantages of interpolating Euler angels, axis-angle, quaternions, and rotation matrices are discussed. Some remarks regarding the parametrization w.r.t. orientation interpolation can be found in [6]. The most apparent effect of using different orientation interpolations, is the 'length' of the rotation. It is know that the 'shortest' rotation, i.e. the geodesic on the rotation group, is given by the rotation about the constant axis and angle corresponding to the relative rotation of two given rotations. This led Shoemake to introduce the so-called spherical linear interpolation (SLERP) [5] using unit quaternions (Euler parameters). The basic idea is to replace the non-linear interpolation problem (quaternions) by a linear one (in terms of scaled rotation vector). The crucial relation is the exponential mapping that maps vector quaternions to unit quaternions. In other words, it gives an element in the Lie group of unit quaternions in terms of a vector in its Lie algebra, where the interpolation is carried out. This concept is clearly applicable to any other Lie group in particular to the rotation group $SO(3)$.

SLERP describes a path between two quaternions along the shortest connection with a constant angular velocity. Therefore, and because of its simplicity, it is the standard for the path generation between two orientations. But the path planning for a set of quaternions has only been addressed in a few publications. In [8] and [9] optimization techniques are presented that allow generating smooth trajectories through desired orientations. In [12] a spline algorithm for generating orientation trajectories is presented that approximately minimizes the angular acceleration. In [13] also curvature interpolation algorithms are introduced. Their computational complexity and robustness properties put their application in perspective, however. A different method is presented in [10] using SLERP in conjunction with blending functions, e.g. B-splines or Bézier curves, where the initial and terminal orientation (quaternion) is exactly achieved. In [11] an incremental path planning method with SLERP is presented where intermediate points are calculated to get continuous tangents, but only a C^1 continuous curve is obtained, which is not sufficient for robotic applications.

The method presented in this paper is based on the exponential mapping of quaternions where its argument is evaluated by B-splines. Each B-spline is computed due to desired orientations which are quaternions or also quaternion derivatives.

The paper starts by summarizing relevant relations of quaternions in section II, followed by a review of SLERP in section III. Section IV treats the path generation with splines and the evaluation of the control points for different scenarios, like passing the desired orientations exactly or not. In order to emphasize the potential of our procedure in section V a path planning based on measurement data of a car fender is presented. The proposed method is compared to the standard SLERP method, and its benefits are highlighted.

II. QUATERNIONS

Unit quaternions uniquely describe the orientation of an object. A quaternion is represented by

$$\mathbf{Q} = q_0 + q_1\,\mathfrak{i} + q_2\,\mathfrak{j} + q_3\,\mathfrak{k} \qquad (1)$$

with the 4 real numbers q_r for $r \in \{0, \dots, 3\}$, see [4]. The complex units satisfy $\mathfrak{i}$, $\mathfrak{j}$ and $\mathfrak{k}$ with $\mathfrak{i}^2 = \mathfrak{j}^2 = \mathfrak{k}^2 = \mathfrak{i}\mathfrak{j}\mathfrak{k} = -1$.

*Authors are with Institute of Robotics, Johannes Kepler University Linz, 4040 Linz, Austria matthias.neubauer_1@jku.at

This work has been supported by the Austrian COMET-K2 program of the Linz Center of Mechatronics (LCM), and was funded by the Austrian federal government and the federal state of Upper Austria.

To simplify the notation the vector representation with the real part q_0 and the (imaginary) vector part $\mathbf{q}$ is used, leading to

$$\mathbf{Q} = (q_0, \mathbf{q}),\ \ \mathbf{q} = (q_1, q_2, q_3) \in \mathbb{R}^3. \tag{2}$$

Denote $q_0 = \mathrm{Re}\,(\mathbf{Q})$ and $\mathbf{q} = \mathrm{Im}\,(\mathbf{Q})$. The multiplication is given by

$$\begin{aligned}\mathbf{Q} &= \mathbf{Q}_1\,\mathbf{Q}_2 = (q_{10}, \mathbf{q}_1)\,(q_{20}, \mathbf{q}_2)\\ \mathbf{Q} &= (q_{10}q_{20} - \mathbf{q}_1^T\mathbf{q}_2, q_{10}\mathbf{q}_2 + q_{20}\mathbf{q}_1 + \mathbf{q}_1 \times \mathbf{q}_2).\end{aligned} \tag{3}$$

which is not commutative. The conjugate of $\mathbf{Q}$ is defined to be $\mathbf{Q}^* = (q_0, -\mathbf{q})$. Rotations are represented by quaternions that satisfy $||\mathbf{Q}||_2 = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$. This subset of quaternions forms the group of unit quaternions.

III. ORIENTATION PATH PLANNING

Using unit quaternions the path planning become more difficult because an element-wise planning is not possible. As a consequence, the spherical linear interpolation (SLERP), first mentioned in [5], was introduced. SLERP is up to now, the standard method to generate a path between two quaternions. It interpolates along the geodesic of two quaternions which is the shortest connection between them. It is implemented by

$$\mathbf{Q}(\sigma) = \mathbf{Q}_1 \exp\left(\log\left(\mathbf{Q}_1^*\,\mathbf{Q}_2\right)\sigma\right) \text{ with } \sigma \in [0,1] \tag{4}$$

and $\mathbf{Q}_1$ and $\mathbf{Q}_2$ the two boundary points of the interpolation. In this context the parametrization by an angle $\Theta \in \mathbb{R}$ and an axis $\mathbf{n} \in \mathbb{R}^3$ with $||\mathbf{n}||_2 = 1$ is useful leading to the quaternion representation

$$\mathbf{Q} = \begin{pmatrix} \cos\left(\frac{\Theta}{2}\right) \\ \sin\left(\frac{\Theta}{2}\right)\mathbf{n} \end{pmatrix} = \exp \mathbf{\Psi} \text{ with } \mathbf{\Psi} = \begin{pmatrix} 0 \\ \frac{\Theta}{2}\,\mathbf{n} \end{pmatrix} \tag{5}$$

where the exponential mapping gives the unit quaternion, i.e. a rotation, corresponding to the vector quaternion $\mathbf{\Psi}$. In the following the scaled rotation axis is denoted as $\boldsymbol{\psi} = \frac{\Theta}{2}\mathbf{n}$. Using the angle axis representation, the logarithm function of a quaternion is defined to

$$\mathbf{\Psi} = \begin{pmatrix} 0 \\ \boldsymbol{\psi} \end{pmatrix} = \log \mathbf{Q} = \log \begin{pmatrix} \cos\left(\frac{\Theta}{2}\right) \\ \sin\left(\frac{\Theta}{2}\right)\mathbf{n} \end{pmatrix}. \tag{6}$$

The vector part $\boldsymbol{\psi}$ is referred to as scaled rotation axis. Necessarily, not only the path but also its derivatives are of prime interest. Especially for robotics, the angular velocity and acceleration are important. Within this paper, only path planning and not the trajectory generation is discussed, nevertheless ${}_I\boldsymbol{\omega}$ and ${}_I\boldsymbol{\alpha}$ are denoted as angular velocity and angular acceleration although they depend on the path parameter σ rather than on time t. Then the angular rate (velocity if σ replaced by t) is computed by

$$\begin{pmatrix} 0 \\ {}_I\boldsymbol{\omega} \end{pmatrix} = 2\,\mathbf{Q}'\,\mathbf{Q}^* \text{ with } \mathbf{Q}' = \frac{d\mathbf{Q}}{d\sigma}. \tag{7}$$

The derivative of the SLERP parameterization (4) is

$$\mathbf{Q}'(\sigma) = \mathbf{Q}_1 \exp\left(\log\left(\mathbf{Q}_1^*\,\mathbf{Q}_2\right)\sigma\right)\,\log\left(\mathbf{Q}_1^*\,\mathbf{Q}_2\right) \tag{8}$$

and by inserting (4) and (8) into (7) it yields to

$$\begin{pmatrix} 0 \\ {}_I\boldsymbol{\omega}(\sigma) \end{pmatrix} = 2\,\mathbf{Q}(\sigma)\,\log\left(\mathbf{Q}_1^*\,\mathbf{Q}_2\right)\mathbf{Q}^*(\sigma) \tag{9}$$

or

$$\begin{pmatrix} 0 \\ {}_B\boldsymbol{\omega}(\sigma) \end{pmatrix} = 2\,\log\left(\mathbf{Q}_1^*\,\mathbf{Q}_2\right) \tag{10}$$

where the subscripts B and I indicate a representation in body or inertial fixed frame, respectively. Note, SLERP performs a rotation with a constant angular velocity and thus with zero angular acceleration. This is not applicable for path generation through a set of quaternions, or for concatenation of several paths. The resultant path is only continuous but not smooth with a discontinuous change in the angular velocity. (see Fig. 3)

IV. SMOOTH ORIENTATION PATH PLANNING

This section introduces the smooth quaternion path planning for a set of desired orientations. In contrast to SLERP a global method is used considering all desired points at once in the path planning strategy.

A. Arc length parametrization of rotation

To guaranty that always unit quaternions are calculated, the map

$$\begin{aligned}\mathbf{Q}(\sigma) &= \mathbf{Q}_1 \exp\left(\begin{pmatrix} 0 \\ \frac{\Theta(\sigma)}{2}\,\mathbf{n}(\sigma) \end{pmatrix}\right) && (11)\\ &= \mathbf{Q}_1 \begin{pmatrix} \cos\left(\frac{\Theta(\sigma)}{2}\right) \\ \sin\left(\frac{\Theta(\sigma)}{2}\right)\mathbf{n}(\sigma) \end{pmatrix}, && (12)\end{aligned}$$

which is similar to (4) with the difference that now Θ and $\mathbf{n}$ are functions of σ, is used. Note, $\mathbf{Q}_1$ represents the start orientation of our path. By changing $\mathbf{n}(\sigma)$ and $\Theta(\sigma)$ properly, the desired path in quaternions is obtained. Because, $\mathbf{n}$ is also constrained to length 1, the local rotation vector $\mathbf{\Psi}(\sigma) = (0, \frac{\Theta(\sigma)}{2}\,\mathbf{n}^T(\sigma))^T = (0, \boldsymbol{\psi}^T(\sigma))^T$ is introduced and used for the path planning. Therewith (11) reads

$$\mathbf{Q}(\sigma) = \mathbf{Q}_1 \exp\left(\mathbf{\Psi}(\sigma)\right) \tag{13}$$

with the local rotation vector $\mathbf{\Psi}(\sigma)$ as imaginary quaternion. Within the imaginary part of $\mathbf{\Psi}(\sigma)$ a path can be designed meeting the demands of the desired orientation path with B-splines, see [7].

B. B-spline fitting of the scaled rotation axis

In order to get a smooth path, B-splines in each element of $\boldsymbol{\psi}(\sigma) = \mathrm{Im}\left(\mathbf{\Psi}(\sigma)\right)$ are used. The B-spline $\psi_i(\sigma)$ for $i = \{1, 2, 3\}$ is calculated by

$$\psi_i(\sigma) = \sum_{c=1}^{n_c} N_c^m(\sigma)\,d_{ci} \tag{14}$$

with the blending function $N_c^m(\sigma)$ of degree m and the associated control point d_{ci}. The blending function is defined by the recursive formula

$$N_c^m(\sigma) = \frac{\sigma - \sigma_c}{\sigma_{c+m} - \sigma_c} N_c^{m-1}(\sigma) + \frac{\sigma_{c+d+1} - \sigma}{\sigma_{c+m+1} - \sigma_{c+1}} N_{c+1}^{m-1}(\sigma) \quad (15)$$

with

$$N_c^0 = \begin{cases} 1 & \sigma \in [\sigma_c, \sigma_{c+1}[\\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

Note, σ_c are the knots of the B-spline and have to be chosen according to the conditions of the desired path. The condition matrix $\mathbf{R}$ for the path planning problem, arise by the desired values of the scaled rotation axis $\boldsymbol{\psi}(\sigma_l) = \mathrm{Im}\left(\boldsymbol{\Psi}(\sigma_l)\right)$ which are stacked and lead to

$$\mathbf{R} = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_{n_\sigma} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\psi}^T(\sigma_1) \\ \boldsymbol{\psi}^T(\sigma_2) \\ \vdots \\ \boldsymbol{\psi}^T(\sigma_{n_\sigma}) \end{pmatrix} \quad (17)$$

with σ_l for $l = \{1, \ldots, n_\sigma\}$ the dedicated positions on the path. Now, the control points d_{ci} for the ith spline with $i = \{1, 2, 3\}$ are calculated by solving the linear equation

$$\mathbf{R} = \mathbf{N} \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ \vdots & \vdots & \vdots \\ d_{n_c 1} & d_{n_c 2} & d_{n_c 3} \end{pmatrix} \quad (18)$$

$$= \mathbf{N} \begin{pmatrix} \mathbf{d}_1 & \mathbf{d}_2 & \mathbf{d}_3 \end{pmatrix} = \mathbf{N}\,\mathbf{D} \quad (19)$$

with $\mathbf{R} \in \mathbb{R}^{p \times 3}$, $\mathbf{D} \in \mathbb{R}^{n_c \times 3}$ and p the number of conditions in $\mathbf{R}$ (see next section). Note, if only desired orientations are prescribed and not any desired velocity or acceleration $p = n_\sigma$ and the matrix $\mathbf{N} \in \mathbb{R}^{p \times n_c}$ consists of the evaluated values of the blending functions:

$$\mathbf{N} = \begin{pmatrix} N_1^m(\sigma_1) & N_2^m(\sigma_1) & \ldots & N_{n_c}^m(\sigma_1) \\ N_1^m(\sigma_2) & N_2^m(\sigma_2) & \ldots & N_{n_c}^m(\sigma_2) \\ \vdots & \vdots & \ddots & \vdots \\ N_1^m(\sigma_{n_\sigma}) & N_2^m(\sigma_{n_\sigma}) & \ldots & N_{n_c}^m(\sigma_{n_\sigma}) \end{pmatrix}. \quad (20)$$

Depending on how many orientations are desired and if they must be met exactly, different number of control points and different methods for the calculation of the control points arise.

C. Interpolation through desired orientations

In order to pass through all desired orientations an interpolation of the quaternions is necessary. Therefore, first the matrix $\mathbf{R}$ has to be defined. For n_σ desired orientations at their desired positions σ_l for $l = \{1, \ldots, n_\sigma\}$ the condition for the local rotation vector is given by

$$\boldsymbol{\Psi}(\sigma_l) = \log\left(\mathbf{Q}_1^* \, \mathbf{Q}_l\right) \quad (21)$$

with $\mathbf{Q}_l$ the desired orientation at σ_l. Calculating that for all quaternions and inserting the scaled rotation vector $\boldsymbol{\psi}(\sigma_l) = \mathrm{Im}(\boldsymbol{\Psi}(\sigma_l))$ into (17) the condition matrix in order to pass the desired orientations is defined. Because not only $\boldsymbol{\Psi}(\sigma)$ also derivatives of $\boldsymbol{\Psi}(\sigma)$ can be affected, e.g. also values for the angular velocities and acceleration can be prescribed. In order to obtain a desired velocity ${}_I\boldsymbol{\omega}(\sigma_l)$, a condition for $\boldsymbol{\Psi}'(\sigma_l)$ have to be added to $\mathbf{R}$. Therefore, first the quaternion derivative according to the desired angular velocity is computed by

$$\mathbf{Q}'(\sigma_l) = \frac{1}{2} \begin{pmatrix} 0 \\ {}_I\boldsymbol{\omega}(\sigma_l) \end{pmatrix} \mathbf{Q}(\sigma_l). \quad (22)$$

Secondly, values for $\Theta(\sigma_l)$, $\Theta'(\sigma_l)$, $\mathbf{n}(\sigma_l)$ and $\mathbf{n}'(\sigma_l)$ have to be evaluated. $\Theta(\sigma_l)$ and $\mathbf{n}(\sigma_l)$ follow due to the norm $||\mathbf{n}||_2 = 1$ leading to

$$\Theta(\sigma_l) = 2\,||\boldsymbol{\Psi}(\sigma_l)||_2$$
$$\mathbf{n}(\sigma_l) = \frac{2\,\mathrm{Im}\big(\boldsymbol{\Psi}(\sigma_l)\big)}{\Theta(\sigma_l)}. \quad (23)$$

For the special case, that $||\boldsymbol{\Psi}(\sigma_l)||_2 = 0$, it follows

$$\Theta(\sigma_l) = 0 \quad (24)$$
$$\mathbf{n}(\sigma_l) = \text{not defined.} \quad (25)$$

$\Theta'(\sigma_l)$ and $\mathbf{n}'(\sigma_l)$ reveal from the derivative of $\mathbf{Q}(\sigma)$, whereby $\mathbf{Q}'(\sigma)$ is obtained by derivation of (12) which leads to

$$\mathbf{Q}'(\sigma_l) = \mathbf{Q}_1 \begin{pmatrix} -\sin(\frac{\Theta_{\sigma_l}}{2})\,\frac{\Theta'_{\sigma_l}}{2} \\ \cos(\frac{\Theta_{\sigma_l}}{2})\,\frac{\Theta'_{\sigma_l}}{2}\,\mathbf{n}_{\sigma_l} + \sin(\frac{\Theta_{\sigma_l}}{2})\,\mathbf{n}'_{\sigma_l} \end{pmatrix}. \quad (26)$$

For sake of simplicity $\Theta(\sigma_l)$ and $\mathbf{n}(\sigma_l)$ are abbreviated with Θ_{σ_l} and $\mathbf{n}_{\sigma_l}$ respectively. Rearranging (26) we get

$$\Theta'_{\sigma_l} = 2\,\frac{-\mathrm{Re}\big(\mathbf{Q}_1^*\,\mathbf{Q}'(\sigma_l)\big)}{\sin(\frac{\Theta_{\sigma_l}}{2})}$$
$$\mathbf{n}'_{\sigma_l} = \frac{\mathrm{Im}\big(\mathbf{Q}_1^*\,\mathbf{Q}'(\sigma_l)\big) - \cos(\frac{\Theta_{\sigma_l}}{2})\,\frac{\Theta'_{\sigma_l}}{2}\,\mathbf{n}_{\sigma_l}}{\sin(\frac{\Theta_{\sigma_l}}{2})}. \quad (27)$$

Now $\boldsymbol{\Psi}'(\sigma_l)$ for a desired angular velocity can be calculated by

$$\boldsymbol{\Psi}'(\sigma_l) = \begin{pmatrix} 0 \\ \frac{\Theta'_{\sigma_l}}{2}\,\mathbf{n}_{\sigma_l} + \frac{\Theta_{\sigma_l}}{2}\,\mathbf{n}'_{\sigma_l} \end{pmatrix}. \quad (28)$$

Here too, the case $\Theta_{\sigma_l} = 0$ has to be considered separately. However, by comparing (28) and (26) for $\Theta_{\sigma_l} = 0$ the relation

$$\boldsymbol{\Psi}'(\sigma_l) = \mathbf{Q}_1^*\,\mathbf{Q}'(\sigma_l) \quad (29)$$

is obtained. Certainly, also the angular acceleration can be prescribed by calculating a restriction for $\boldsymbol{\Psi}''(\sigma)$. Here only the methodology, without any explicit details is explained because the derivation is very similar to the previous one. First $\mathbf{Q}''(\sigma_l)$ has to be calculated from ${}_I\boldsymbol{\alpha}(\sigma_l)$ leading to Θ''_{σ_l} and $\mathbf{n}''_{\sigma_l}$. Using the 2nd derivation of (12) we get $\boldsymbol{\Psi}''(\sigma_l)$ for the path generation. Not only the 2nd, also conditions for higher order derivatives can be similarly included in the path planning.

Finally, all additional conditions for $\boldsymbol{\Psi}$ have to be added to the condition matrix $\mathbf{R}$. If rows in $\mathbf{R}$ are added, also $\mathbf{N}$ has to be extended. For example, if a condition for $\boldsymbol{\Psi}'(\sigma_*)$ is added, also the values of $N_c^{m\prime}(\sigma_*)$ for all control points n_c have to be added to $\mathbf{N}$. With $\mathbf{N}$ and $\mathbf{R}$ evaluated the control points can be calculated. For fulfilling all conditions the matrix $\mathbf{N}$ has to be quadratic and thus the number of control points n_c has to be equal to the number of conditions in $\mathbf{R}$. Finally, the control points are calculated by

$$\mathbf{D} = \mathbf{N}^{-1}\mathbf{R}. \tag{30}$$

With $\mathbf{D}$ calculated the local rotation vector is defined and yields

$$\boldsymbol{\Psi}(\sigma) = \begin{pmatrix} 0 & \boldsymbol{\psi}(\sigma) \end{pmatrix}^T \tag{31}$$

with ψ_i calculated by (14). The quaternion path is then obtained by the evaluation of (13) using (5) and (23).

D. Approximation of desired orientations

If the desired path is given by a large set of quaternions and especially when measurement data is used, oscillations due to measurement errors can appear and thus an approximation of the desired quaternions is maybe the better choice. Therefore the number of control points n_c has to be selected lower than the number of conditions p, so that the system (19) is overdetermined. To calculate the control points, the error matrix $\mathbf{E} = (\mathbf{e}_1\ \mathbf{e}_2\ \mathbf{e}_3)$ is introduced which is defined to be

$$\mathbf{E} = \mathbf{R} - \mathbf{N}\,\mathbf{D}. \tag{32}$$

In order to get an approximated solution of the desired path the control points are calculated by minimizing $\mathbf{E}$. Therefore the cost functionals $\mathcal{L}_i = \frac{1}{2}\mathbf{e}_i^T\,\mathbf{e}_i$ for $i = \{1, 2, 3\}$ are defined. The least square solution of (32) simultaneously minimizing the functionals is

$$\mathbf{D} = (\mathbf{N}^T\,\mathbf{N})^{-1}\,\mathbf{N}^T\,\mathbf{R}. \tag{33}$$

With the control points $\mathbf{D}$ the specified conditions $\mathbf{R}$ are not fulfilled exactly, but the error defined in (32) is minimized. If desired, also a weighting of $\mathbf{E}$ ($\mathbf{E}_W = \mathbf{W}\,\mathbf{E}$) is possible, thus some conditions are approximated closer than others, and leading to solution

$$\mathbf{D} = (\mathbf{N}^T\,\mathbf{W}^T\,\mathbf{W}\mathbf{N})^{-1}\,\mathbf{N}^T\,\mathbf{W}^T\,\mathbf{W}\,\mathbf{R} \tag{34}$$

with $\mathbf{W} \in \mathbb{R}^{p \times p}$ the weighting matrix for the minimization of the weighted error $\mathbf{E}_W$.

E. Combined approximation and interpolation of desired orientations

Frequently in applications a path is required where certain orientations are approximated while some others must be passed exactly, e.g. if two paths are connected so that the start and endpoint should be equal with a certain continuity. Therefore, a new method based on the approximation method from before is introduced. For that, the rows in the matrix $\mathbf{R}$ and $\mathbf{N}$ have to be separated in two parts. In the first part the conditions which should be fulfilled are combined, denoted as $\overline{\mathbf{R}}$ and $\overline{\mathbf{N}}$. In the second part the points for the approximation are stored, denoted by $\hat{\mathbf{R}}$ and $\hat{\mathbf{N}}$. Now, a quadratic optimization problem with equality constraints can be formulated which yields

$$\begin{gathered}\min_{\mathbf{d}_i}\left(\hat{\mathbf{r}}_i - \hat{\mathbf{N}}\,\mathbf{d}_i\right)^T\left(\hat{\mathbf{r}}_i - \hat{\mathbf{N}}\,\mathbf{d}_i\right)\\ \text{subject to}:\\ \overline{\mathbf{r}}_i = \overline{\mathbf{N}}\,\mathbf{d}_i\end{gathered} \tag{35}$$

Note, the optimization is solved separately for each B-spline $i = \{1, 2, 3\}$. The vectors $\hat{\mathbf{r}}_i$ and $\overline{\mathbf{r}}_i$ represent the ith row of matrix $\hat{\mathbf{R}}$ and $\overline{\mathbf{R}}$, respectively. This type of problem can be solved with the method of Lagrange multipliers, whereby first a new cost functional is calculated which in our case leads to

$$\mathcal{L}_i = \left(\hat{\mathbf{r}}_i - \hat{\mathbf{N}}\,\mathbf{d}_i\right)^T\left(\hat{\mathbf{r}}_i - \hat{\mathbf{N}}\,\mathbf{d}_i\right) + \boldsymbol{\lambda}^T\,\left(\overline{\mathbf{r}}_i - \overline{\mathbf{N}}\,\mathbf{d}_i\right) \tag{36}$$

with $\boldsymbol{\lambda}$ the Lagrangian multiplier and second the optimal solution is obtained by solving the two conditions

$$\frac{d\mathcal{L}_i}{d\mathbf{d}_i^T} = 0 \tag{37}$$

$$\frac{d\mathcal{L}_i}{d\boldsymbol{\lambda}^T} = 0. \tag{38}$$

Thus finally the solution is obtained and the calculation of the control points can be implemented by

$$\begin{aligned}\mathbf{d}_i &= \mathbf{Z}\overline{\mathbf{N}}^T\left(\overline{\mathbf{N}}\mathbf{Z}\overline{\mathbf{N}}^T\right)^{-1}\overline{\mathbf{r}}_i + \\ &\quad \mathbf{Z}\left(\mathbf{I} - \overline{\mathbf{N}}^T\left(\overline{\mathbf{N}}\mathbf{Z}\overline{\mathbf{N}}^T\right)^{-1}\overline{\mathbf{N}}\mathbf{Z}\right)\hat{\mathbf{N}}^T\hat{\mathbf{r}}_i \qquad (39)\\ \mathbf{Z} &= \left(\hat{\mathbf{N}}^T\hat{\mathbf{N}}\right)^{-1}. \qquad (40)\end{aligned}$$

V. EXAMPLE: SMOOTH QUATERNION PATH PLANNING FOR POLISHING A CAR FENDER

In this section an example of smooth orientation path planning is discussed. Particularly, the path planning for polishing a car fender with an industrial robot is presented. As desired orientations measurements from the surface of a car fender are used, see Fig. 1. In order to calculate a smooth path the conditions for the path planning have to be defined. We start by defining the desired orientation values by inserting $\boldsymbol{\Psi}(\sigma_l)$ for each desired orientation l according to (21) into the condition matrix $\mathbf{R}$. Note, because the orientation measurements are equally spaced along the path in the Cartesian space also σ_l for the orientation path planning is equidistant, leading to $\sigma_l = (l-1)\,\Delta\sigma$ with $\Delta\sigma = 1/(n_\sigma - 1)$. Note, the knots σ_c for the calculation of the blending functions in (15) are chosen equidistant. Additionally, the starting and final velocity and acceleration of our path are prescribed. Thus conditions as described in IV-C are calculated yielding to values for $\boldsymbol{\Psi}'(0)$, $\boldsymbol{\Psi}''(0)$, $\boldsymbol{\Psi}'(1)$ and $\boldsymbol{\Psi}''(1)$. Once again, also the corresponding derivatives of the blending functions have to be added to $\mathbf{N}$. As starting velocity for our smooth path, the angular velocity of the SLERP path resulting from the interpolation between the first two desired orientations is chosen. The desired values for the angular velocity at the end as well as the

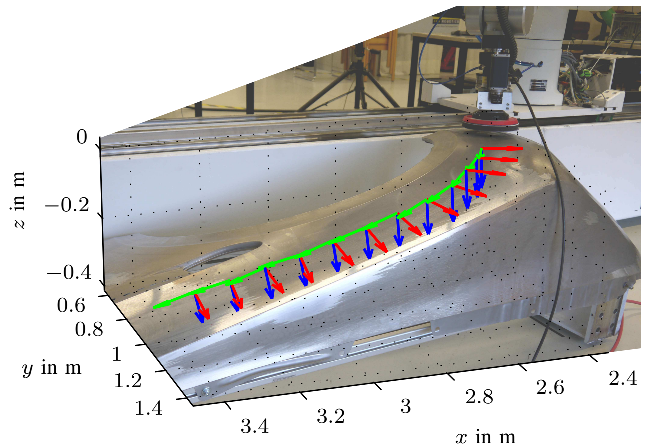

Fig. 1. Waypoints for the path generation of the car fender with their orientation represented by the vector x →, y → and z →.

accelerations at the start and end are set to zero. Overall 16 conditions are defined. 12 due to desired orientations and 4 due to the angular velocity and acceleration conditions of the start and endpoint. In order to fulfill all conditions the interpolation method from IV-C is used, hence (30) is used for the calculation of the control points. In order to compare our method with SLERP, both quaternion paths are depicted in Fig. 2. More important, also the corresponding angular velocities are shown in Fig. 3, whereby the velocities of SLERP are plotted with dashed lines. Because SLERP has a dicontinuous change in the angular velocity it is clearly not applicable for a robotic system due to its infinite angular acceleration. Whereas our smooth path has a continuous angular acceleration, see Fig. 4. The angular velocity of the smooth path is evaluated by the values of the scaled rotation axis $\boldsymbol{\psi}$ and is calculated by

$$
\begin{aligned}
{}_I\boldsymbol{\omega}(\sigma) &= \mathbf{T}(\boldsymbol{\psi})\,\boldsymbol{\psi}' \qquad (41)\\
\mathbf{T}(\boldsymbol{\psi}) &= \mathbf{I} + \left(\frac{\cos(||\boldsymbol{\psi}||)-1}{||\boldsymbol{\psi}||^2}\right)\tilde{\boldsymbol{\psi}} \\
&\quad + \left(1 - \frac{\sin(||\boldsymbol{\psi}||)-1}{||\boldsymbol{\psi}||}\right)\frac{\tilde{\boldsymbol{\psi}}\tilde{\boldsymbol{\psi}}}{||\boldsymbol{\psi}||^2}.
\end{aligned}
$$

The angular acceleration is calculated by the derivative of (41) and leads to

$$
{}_I\boldsymbol{\alpha}(\sigma) = \mathbf{T}(\boldsymbol{\psi})\boldsymbol{\psi}'' + \mathbf{T}'(\boldsymbol{\psi})\,\boldsymbol{\psi}'. \qquad (42)
$$

Nevertheless, due to measurement errors and due to the interpolation method oscillations in the calculated path appear which are unsuitable for our polishing application. As a consequence the interpolation and approximation method IV-E is applied, because anyhow only the start and endpoint with their derivatives (angular velocity and acceleration) are important for our experiment.

Thus the orientations in between are approximated. The lines of $\mathbf{R}$ and $\mathbf{N}$ describing the orientation, the angular velocity and acceleration at the start and endpoint of our path are combined in $\overline{\mathbf{R}}$ and $\overline{\mathbf{N}}$. The remaining lines are

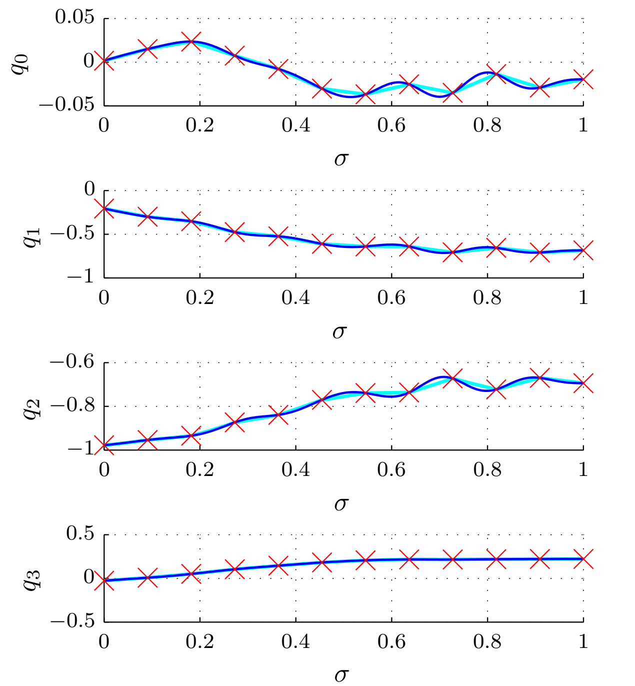

Fig. 2. Interpolation path with × the desired quaternions, — the smooth path and — SLERP

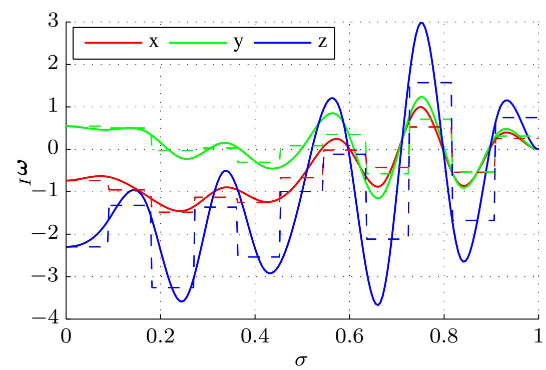

Fig. 3. Angular velocities for the interpolation path using the smooth path generation (solid lines) and SLERP (dashed lines)

stacked into $\hat{\mathbf{N}}$ and $\hat{\mathbf{R}}$ forming the set of equations for the approximation. Finally, the quaternion path can be calculated and the resultant path is printed in Fig. 5. The according angular velocity and acceleration is shown in Fig. 6 and Fig. 7, respectively.

By the separation in points which should be passed exactly and in points which only should be passed in the vicinity, the angular velocity and acceleration decrease significantly in contrast to the interpolation method.

CONCLUSION

In this paper a new method for the calculating of smooth quaternion paths was presented. The benefits of our method are highlighted by comparing the results obtained with the standard method of quaternion path planning (SLERP) and

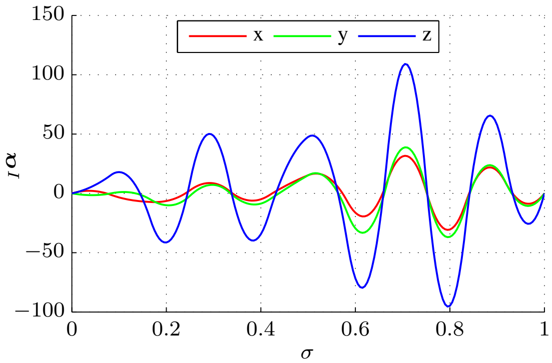

Fig. 4. Angular acceleration for the interpolation path using the smooth path generation

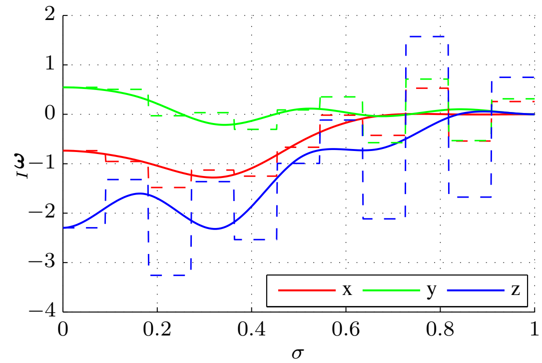

Fig. 6. Angular velocities for the approximation and interpolation path using the smooth path generation (solid lines) and SLERP (dashed lines)

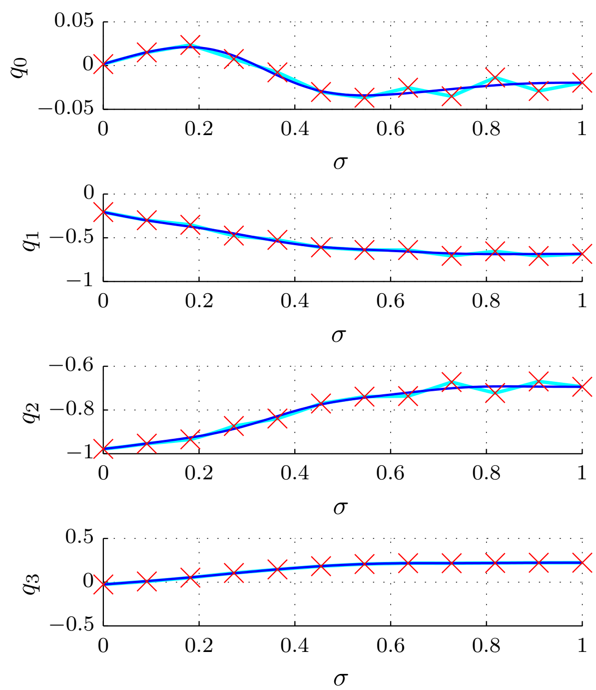

Fig. 5. Approximation and interpolation path with × the desired quaternions, — the smooth path and — SLERP

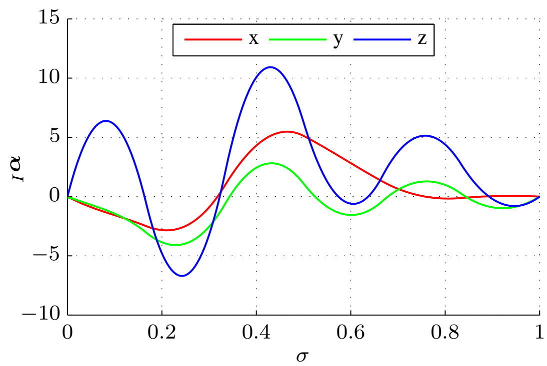

Fig. 7. Angular acceleration for the approximation and interpolation path using the smooth path generation

our approach. Especially the possibility to satisfy a predefined degree of continuity distinguishes the two methods. The ability to ensure a prescribed order of continuity (desired angular velocities, accelerations or even higher order derivatives) in the path planning is a singnificant advantage. By choosing an appropriate method for the calculation of the control points, the path planning allows for reduction of the angular acceleration, for example. Finally it shall be remarked that the method can be straightforwardly extended to dual quaternions, i.e. to full frame motion.

References

[1] L. Biagiotti, C. Melchiorri, Trajectory Planning for Automatic Machines and Robots, Berlin Heidelberg: Springer-Verlag, 2008, ch. 4.

[2] S. M. LaValle, Planning Algorithms. New York: Cambrigde University Press, 2006, ch. 2.

[3] B. Siciliano and O. Khatib, Handbook of robotics, Berlin Heidelberg: Springer-Verlag, 2008, ch. 1.

[4] A. Morawiec, Orientations and Rotations: Computations in Crystallographic Textures, Berlin Heidelberg: Springer-Verlag, 2004.

[5] K. Shoemake, Animating Rotation with quaternion curves, in Proc. 12nd Annu. Conf. on Computer Graphics and interactive techniques, New York, 1985, pp. 245.

[6] A. Kecskeméthy, M. Hiller, Einflüsse der Parametrisierung auf die Interpolation von Drehbewegungen, ZAMM - Journal of Applied Mathematics and Mechanics, 4-5, T 129 - T131, 1993.

[7] L. Piegl, W. Tiller, The Nurbs Book, Berlin Heidelberg: Springer-Verlag, 1997, ch. 2.

[8] A. H. Barr, B. Currin, S. Gabriel, J. F. Hughes, Smooth Interpolation of Orientations with Angular Velocity Constraints using Quaternions, in Proc. 19nd Annu. Conf. on Computer Graphics and interactive techniques, New York, 1992, pp. 313.

[9] G. M. Nielson, ν-quaternion splines for the smooth interpolation of orientations, IEEE Transactions on Visualization and Computer Graphics, vol. 10, no. 2, pp. 224, March/April 2004.

[10] M.-J. Kim, M.-S. Kim, A general construction scheme for unit quaternion curves with simple high order derivatives, in Proc. 22nd Annu. Conf. on Computer Graphics and interactive techniques, New York, 1995, pp. 369.

[11] E. B. Dam, M. Koch, M. Lillholm, Quaternions, Interpolation and Animation, Technical Report, July 1998.

[12] I. G. Kang, F.C. Park, Cubic spline algorithms for orientation interpolation, Int. J. Numer. Meth. Engng.. vol. 46, no. 1, 1999.

[13] O. A. Bauchau, S. Han, Interpolation of rotation and motion, Multibody System Dynamics, vol. 31, no. 3, pp. 339-370, March 2014.