2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Smooth Invariant Interpolation on Lie groups with Prescribed Terminal Conditions for Robot Motion Planning and Modeling of Soft Robots

Andreas Müller and Tobias Marauli and Hubert Gattringer *

***Abstract*— Interpolation of rigid body motions, or a general frame motion in Euclidean space, is a recurring topic in robotics. It boils down to generating trajectories in a Lie group, either $SE(3)$ or $SO(3)\times\mathbb{R}^3$, with given initial and/or terminal values. To this end, spline interpolation schemes were developed where canonical coordinates are represented by cubic splines. They allow for prescribing initial velocity and acceleration only. In many robotic applications, terminal conditions are prescribed, however. In this paper, a novel interpolation scheme is presented that admits prescribing the terminal pose, velocity and acceleration, or the initial condition. As example, the scheme is applied to a rendezvous task of a UAV and describing the deformation of a Cosserat beam as relevant for soft robotics. The presented interpolation scheme can be directly applied to the motion parameterization in terms of (dual) quaternions.**

***Keywords*– Motion interpolation, splines, Lie group $SE(3)$, screws, geodesics, UAV, robotic manipulator, soft robotics**

I. Introduction and Literature Overview

Interpolating trajectories along given spatial poses is a frequent task in robotics. There are two different situation, the trajectory planning and the reconstruction of motions from given samples. Either task necessitates using a certain geometric model, i.e. configuration space Lie group G, of rigid body motions. The pose of a moving frame $\mathcal{F}$ relative to a reference frame $\mathcal{F}_0$, e.g. an inertial frame, is represented by $\mathbf{C}\in G$, where G is a Lie group (with algebra $\mathfrak{g}$) that is used to represent motions of rigid frames. Typical choices for G are the group of proper rigid body motions $SE(3)$ and the direct product group $SO(3)\times\mathbb{R}^3$, each representing a different geometric model of spatial motions (Sec. II). Rotation matrices and homogenous transformation matrices or quaternions and dual quaternions are different (homomorphic) representations of either group. Interpolation of spatial frame motions has manifold applications in trajectory planning of robotic manipulators, UAVs, legged robots, humanoids etc. but also for modeling compliant elements in soft robotics.

The *geometric point-to-point interpolation problem* is to find a curve in G connecting a prescribed initial configuration $\mathbf{C}_0$ and terminal configuration $\mathbf{C}_T$, where $\mathbf{C}(\tau)\in G$ describes the pose of a moving frame $\mathcal{F}$ relative to a reference frame $\mathcal{F}_0$, e.g. inertial frame. This curve is parameterized by a normalized path parameter $\tau\in[0,1]$ so that $\mathbf{C}_0:=\mathbf{C}(0)$ and $\mathbf{C}_T:=\mathbf{C}(1)$. A solution to the geometric interpolation problem is given by $\mathbf{C}(\tau)=\mathbf{C}_0\exp(\tau\hat{\boldsymbol{\xi}})$ with the constant vector[1] $\boldsymbol{\xi}=\log(\mathbf{C}_0^{-1}\mathbf{C}_T)\in\mathfrak{g}$ of canonical coordinates. Notice that $\boldsymbol{\xi}$ is represented in the moving frame $\mathcal{F}$ at its initial configuration (which ensures left-invariance of the interpolation scheme). This is known as spherical linear interpolation (SLERP), when interpolating pure rotation, i.e. on $SO(3)$ with $\hat{\boldsymbol{\xi}}\in so(3)$, which was first introduced in [4] using quaternions, and as Screw Linear Interpolation (ScLERP), when interpolating rigid body motions on $SE(3)$, with $\hat{\boldsymbol{\xi}}\in se(3)$, as introduced for dual quaternions in [5]. The latter was used in [6] for point-to-point path planning of robots by restriction to $SE(3)$ subgroups corresponding to motion tasks. Notice that SLERP yields a geodesic on $SO(3)$, i.e. the motion with shortest path, while ScLERP does not lead to a geodesic on $SE(3)$ w.r.t. an (left-)invariant metric. Whether a geodesic is desirable depends on the application, i.e. on the geometric model of motion (Sec. II). The curve in $SE(3)$, according to $\boldsymbol{\xi}=\log(\mathbf{C}_0^{-1}\mathbf{C}_T)$ of the ScLERP, is the correct interpolation when a screw motion is to be reconstructed, although it is not the shortest path from $\mathbf{C}_0$ to $\mathbf{C}_T$, which would be a translation along a straight line plus a decoupled rotation [7] (Sec. VI-B). By nesting two SLERP interpolations, an interpolation through four quaternions, called spherical quadrangle (SQUAT) interpolation, was proposed in [8], equivalent to a cubic Bézier curve.

The point-to-point *trajectory interpolation problem* is to find a solution of the geometric interpolation problem which additionally satisfies initial and/or terminal conditions of the rigid body velocity and its derivatives. Solutions to the trajectory interpolation problem can be expressed as

$$\mathbf{C}(\tau)=\mathbf{C}_0\exp\hat{\mathbf{X}}(\tau) \tag{1}$$

where $\hat{\mathbf{X}}\in se(3)$ is the matrix associated to the canonical coordinates $\mathbf{X}\in\mathbb{R}^6$ [1]. It describes the motion relative to the initial configuration $\mathbf{C}_0$. The condition $\mathbf{C}(0)=\mathbf{C}_0$ implies that $\mathbf{X}(0)=\mathbf{0}$. From a general point of view, the trajectory interpolation on $SE(3)$, and its Riemannian geometry, was addressed in [9], where the differential equations were derived whose solutions are curves with minimal acceleration and jerk, and solved for special cases. This was also addressed in [10], where conditions for a curve in $SE(3)$ to have a stationary acceleration (i.e. with minimal average acceleration w.r.t. a left-invariant metric) were derived. Various interpolation schemes were developed where the interpolation on the Lie group is lifted to an interpolation of canonical coordinates $\mathbf{X}(\tau)$ in (1). Quaternion Bézier

*Institute of Robotics, Johannes Kepler University Linz, Altenberger Str. 69, 4040 Linz, Austria a.mueller@jku.at

This work was supported by the LCM-K2 Center within the Austrian COMET-K2 program, and the Austrian Science Fund (FWF) [I 4452-N]

[1]If $G=SE(3)$, then $\hat{\mathbf{X}}\in se(3)$ denotes the 4×4 matrix associated to a twist/screw coordinate vector $\mathbf{X}\in\mathbb{R}^6$. $[\mathbf{X},\mathbf{Y}]$ denotes the Lie bracket (screw product) of two twist vectors. Details can be found in [1], [2], [3].

curves for orientation interpolation were used in [11], and for rigid body motion interpolation in terms of dual quaternions in [12]. In [13], a quintic B-spline description was used. A shortcoming of these approaches is that they do not allow prescribing boundary values for velocities and derivatives. A B-spline formulation that admits combined interpolation and approximation of orientations, and allows prescribing the velocity and arbitrary time derivatives at the all knot points was presented in [14]. A practical guide to using SLERP, SQUAD, and B-splines was reported in [15]. The above formulations involve various computational steps, and a de Casteljau algorithm. Computationally efficient interpolation schemes can be derived by expressing $\mathbf{X}(\tau)$ as cubic polynomial. This was introduced in [16], [17] for orientation trajectories, i.e. canonical coordinates are expressed as cubic spline curves in $so(3)$. However, in all cubic interpolation schemes on Lie groups reported in the literature, it is assumed that the initial values of velocity and acceleration are prescribed. This is a rather limiting assumption since in many applications, not only the pose but also the velocity and acceleration at the terminal time are prescribed. Moreover, it is in general necessary to combine spline interpolations with given initial and terminal twists and accelerations.

In this paper, a novel cubic spline interpolation on Lie groups is presented that allows prescribing the terminal velocity and acceleration. First, the interpolation between two poses is derived directly from the solution of the governing kinematic equation. This naturally yields the minimum acceleration curve as a special case, as derived from a variational principle in [9]. Then, the spline interpolation is obtained by smoothly combining point-to-point interpolations. The method inherits the simplicity of the known cubic spline method. The examples in Sec. VI show the applicability of the proposed method. One application is the interpolation of deformation fields of Cosserat beams, a highly relevant topic in soft robot modeling and kinematics. In this case, the motion of a frame (cross section) is represented in $SE(3)$. The interpolation scheme will be derived for a general Lie group, which can then be applied to the particular choice. A motion is usually expressed as function of time t (or arc length), $\mathbf{C}(t)$. For given terminal time T (or total arc length), it will be convenient to use the normalized path parameter $\tau := t/T$, so that $\tau \in [0,1]$, and to express the motion as $\mathbf{C}(\tau)$. Then $\mathbf{C}' = \frac{d}{d\tau}\mathbf{C} = T\mathbf{C}$ denotes the derivative w.r.t. τ.

II. Spatial Motions in terms of Canonical Coordinates

Denote with $\mathcal{F}$ a moving frame (e.g. representing a rigid body or the cross section of a Cosserat beam), and with $\mathcal{F}_0$ another (e.g. inertial) frame. The *configuration* (also called pose) of $\mathcal{F}$ relative to $\mathcal{F}_0$ is determined by the tuple $\mathbf{r}, \mathbf{R}$, where $\mathbf{r} \in \mathbb{R}^3$ is the position vector of $\mathcal{F}$ resolved in $\mathcal{F}_0$, and $\mathbf{R} \in SO(3)$ the rotation matrix transforming coordinates from $\mathcal{F}$ to $\mathcal{F}_0$. The *motion* of $\mathcal{F}$ is encoded in the concatenation of relative configurations. Two different geometric models are used to this end: $SE(3)$ and $SO(3) \times \mathbb{R}^3$.

The special Euclidean group $SE(3)$ represents proper rigid body motions, usually represented by 4×4 matrices, denoted $\mathbf{C} \in SE(3)$, that describe homogenous transformations in E^3. Starting from initial $\mathbf{C}_0$, the configuration of $\mathcal{F}$, as a function of time, is expressed with (1), where $\hat{\mathbf{X}} \in se(3)$ is the 4×4 matrix associated to the vector $\mathbf{X} \in \mathbb{R}^6$ of canonical coordinates on $SE(3)$. The latter are referred to as instantaneous screw coordinates as any rigid body motion is a screw motion [2]. As $SE(3)$ is the semidirect product, it correctly captures the effect of rotations on translations.

In the direct product $SO(3)\times\mathbb{R}^3$, on the other hand, rotations and translations are decoupled. Configurations are written as tuple $\mathbf{C} = (\mathbf{R}, \mathbf{r}) \in SO(3) \times \mathbb{R}^3$, and the concatenation of configurations is $(\mathbf{R}_1, \mathbf{r}_1)\cdot(\mathbf{R}_2, \mathbf{r}_2) = (\mathbf{R}_1\mathbf{R}_2, \mathbf{r}_1 + \mathbf{r}_2)$. This geometric model is often used in robotic motion planning assuming that translation are not affected by rotations. This is true for floating base systems if $\mathcal{F}$ is located at the COM. Prominent example is the motion planning and control of the centroidal frame [18]. The assumption behind this model is that the shortest path between two positions is a straight line. The conical coordinates $\mathbf{X} = (\mathbf{x}, \mathbf{r}) \in \mathbb{R}^6$ are the scaled rotation axis $\mathbf{x}$ and the position vector, and in (1) it is $\exp\hat{\mathbf{X}} = (\exp\hat{\mathbf{x}}, \mathbf{r})$, with $\exp\hat{\mathbf{x}}$ being the exponential on $SO(3)$.

Both are legitimate models depending on the application. For motion planning between two poses, $SO(3) \times \mathbb{R}^3$ is an obvious choice. For reconstructing a motion, $SE(3)$ should be used. This is relevant when motion is to be reconstructed from motion samples (e.g. motion capture) or when dealing with general screw motions (e.g. beam deformations).

The (normalized) rigid body velocity $\mathcal{V} \in \mathbb{R}^6$, i.e. the velocity of frame $\mathcal{F}$ relative to $\mathcal{F}_0$, in *body-fixed representation* is defined by the left Poisson-Darboux equation on G

$$\mathbf{C}' = \mathbf{C}\hat{\mathcal{V}}. \tag{2}$$

For given velocity, and initial pose, this equation can be solved to determine the pose of a body. It is therefore referred to as the *kinematic reconstruction equation*. As shown by Magnus [19], if $\mathbf{C}$ in (1) satisfies the ODE (2) on Lie group G, then $\hat{\mathbf{X}}$ satisfies the first-order ODE on its Lie algebra $\mathfrak{g}$

$$\mathcal{V} = \mathbf{dexp}_{-\hat{\mathbf{X}}}\mathbf{X}' \tag{3}$$

where $\mathbf{dexp}_{\mathbf{X}} : \mathfrak{g} \to \mathfrak{g}$ is the right-trivialized differential of the exp map [20], [21], also called tangent map [22]. The inverse relation

$$\mathbf{X}' = \mathbf{dexp}^{-1}_{-\hat{\mathbf{X}}}\mathcal{V} \tag{4}$$

relates velocity to the time derivative of the coordinates.

If $G = SE(3)$, then $\mathcal{V} = (\boldsymbol{\omega}, \mathbf{v})$ is the (normalized) *body-fixed twist*, with angular velocity $\boldsymbol{\omega}$ and linear velocity $\mathbf{v} = \mathbf{R}^T\mathbf{r}'$ expressed in $\mathcal{F}$, and $\mathbf{dexp}$ is defined on $se(3)$. If $G = SO(3) \times \mathbb{R}^3$, then $\mathcal{V} = (\boldsymbol{\omega}, \mathbf{r}')$ is the *mixed velocity*, with angular velocity $\boldsymbol{\omega}$ expressed in $\mathcal{F}$ and (normalized) linear velocity $\mathbf{r}'$ expressed in $\mathcal{F}_0$. The differential $\mathbf{dexp}_{\hat{\mathbf{X}}}\mathbf{X}' = (\mathbf{dexp}_{\hat{\mathbf{x}}}\mathbf{x}', \mathbf{r}')$ reflects the decoupling (linear velocity is simply the derivative of position coordinates),

where $\mathbf{dexp}_{\hat{\mathbf{x}}}$ is the differential on $SO(3)$. The right-trivialized differential on $SO(3)$ and $SE(3)$, their inverse, as well as their derivatives, admit closed form relations [20].

With slight abuse of notation, $\mathcal{V}$ and its derivatives $\mathcal{V}'$ and $\mathcal{V}''$ are referred to as velocity, acceleration and jerk, respectively. They are related to the time derivatives by $\frac{d^i}{d\tau^i}\mathcal{V} = T^{i+1}\frac{d^i}{dt^i}\mathbf{V}$. It must be mentioned that, on $SE(3)$, $\dot{\mathbf{V}}$ and $\ddot{\mathbf{V}}$ are not the acceleration and jerk that would determine the linear acceleration and jerk of the origin of frame $\mathcal{F}$. Yet they serve to express the change of the velocity, and are therefore called reduced acceleration and jerk [23], [24], [10].

For sake of brevity, in the following, the differential of the exp map on G is abbreviated with $\mathbf{A}(\boldsymbol{\xi}) := \mathbf{dexp}_{-\hat{\boldsymbol{\xi}}}$, and its directional derivative with $\mathbf{B}(\boldsymbol{\xi}, \mathbf{X}) := -(\mathrm{D}_{-\hat{\boldsymbol{\xi}}}\mathbf{dexp})(\mathbf{X})$. Then the normalized velocity and acceleration follows from (3) as $\mathcal{V} = \mathbf{A}(\mathbf{X})\mathbf{X}'$ and $\mathcal{V}' = \mathbf{A}(\mathbf{X})\mathbf{X}' + \mathbf{B}(\mathbf{X}, \mathbf{X}')\mathbf{X}'$.

III. Higher-Order Extrapolation with given Initial Conditions

Using the Hausdorff-Magnus expansion of the right-trivialized differential, it was shown in [25] that the solution of the left Poisson equation (2) can be expressed as a series expansion $\mathbf{X}(t) = \sum_{i\geq 0}\frac{t^i}{i!}\mathbf{X}_i$, where $\mathbf{X}_i$ are determined recursively in terms of the initial velocity and its derivatives $\mathcal{V}_0, \mathcal{V}_0', \mathcal{V}_0'', \ldots$ the 3rd- and 4th-order approximations, in terms of normalized velocity $\mathcal{V}$ and path parameter, are

$$\mathbf{X}^{[3]}(\tau) = \tau\mathcal{V}_0 + \frac{1}{2}\tau^2\mathcal{V}_0' + \frac{1}{6}\tau^3\mathcal{V}_0'' + \frac{1}{12}\tau^3[\mathcal{V}_0, \mathcal{V}_0'] \quad (5)$$

$$\mathbf{X}^{[4]}(\tau) = \mathbf{X}^{[3]}(\tau) + \frac{1}{24}\tau^4\mathcal{V}_0''' + \frac{1}{24}\tau^4[\mathcal{V}_0, \mathcal{V}_0'']. \quad (6)$$

These expressions allow for extrapolating solution (1) With given initial values $\mathcal{V}_0, \mathcal{V}_0', \mathcal{V}_0''$: They do not lead to a prescribed configuration $\mathbf{C}_T$ at specified terminal time, however.

IV. Cubic Interpolation

In this section, the above formulae are used to derive interpolation formulae for describing a curve between initial and terminal pose. Three interpolations are derived that differ in the boundary conditions that are satisfied.

A. Prescribed Initial Velocity and Acceleration

The motion relative to an initial pose is described with the formula (1). The screw coordinate vector corresponding to the terminal configuration is $\boldsymbol{\xi} = \log(\mathbf{C}_0^{-1}\mathbf{C}_T)$. A third-order approximation is obtained from (5), with $\tau = 1$, as

$$\boldsymbol{\xi} = \mathbf{X}^{[3]}(1) = \mathcal{V}_0 + \frac{1}{2}\mathcal{V}_0' + \frac{1}{6}\mathcal{V}_0'' + \frac{1}{12}[\mathcal{V}_0', \mathcal{V}_0]. \quad (7)$$

Solving (7) for $\mathcal{V}_0''$ and inserting this into (5) yields

$$\mathbf{X}^{[3]}(\tau) = \tau^3\boldsymbol{\xi} + (\tau - \tau^3)\mathcal{V}_0 + \frac{1}{2}(\tau^2 - \tau^3)\mathcal{V}_0'. \quad (8)$$

Clearly, the approximation (8) satisfies the initial conditions at $\tau = 0$, since $\mathbf{X}'(0) = \mathcal{V}_0$ and $\mathbf{X}''(0) = \mathcal{V}_0'$. Thus, $\mathbf{C}(\tau) = \mathbf{C}_0\exp(\mathbf{X}^{[3]}(\tau))$ is a cubic interpolation formula interpolating the trajectory between initial value $\mathbf{X}_0 = \mathbf{0}$ and terminal value $\boldsymbol{\xi}$, with initial velocity $\mathcal{V}_0$ and derivative $\mathcal{V}_0'$. For zero initial twist and acceleration, the motion described by (8) is $\mathbf{C}(\tau) = \mathbf{C}_0\exp(\tau^3\boldsymbol{\xi})$. The latter is a geodesic on $SO(3)$, which is a rotation about a constant axis but with a trajectory depending on path parameter according to τ^3, and on $SE(3)$ it is a screw motion about a constant screw axis.

B. Prescribed Initial and Terminal Velocity

The above formulae allows interpolating spatial motions between initial and terminal configuration, described by $\mathbf{X}(1) = \boldsymbol{\xi}$, with given initial twist and derivative. A cubic interpolation that allows prescribing the initial as well as terminal velocity can be derived from (8). The motion is again described by (1) with given initial pose. To this end, $\mathcal{V}_0'$ is related to the terminal velocity $\mathcal{V}_T := \mathcal{V}(1)$. The terminal velocity $\mathcal{V}_T$ is expressed using (3). The coordinate vector at terminal $\tau = 1$ is obtained from the cubic interpolation formula (8). Its time derivative is $\frac{d}{d\tau}\mathbf{X}^{(3)}\big|_{\tau=1} = \frac{1}{T}(3\boldsymbol{\xi} - 2\mathcal{V}_0 - \frac{1}{2}\mathcal{V}_0')$, and thus $\mathcal{V}_T = \mathbf{dexp}_{-\hat{\boldsymbol{\xi}}}(3\boldsymbol{\xi}_T - 2\mathcal{V}_0 - \frac{1}{2}\mathcal{V}_0')$. Solving this for $\mathcal{V}_0'$ and inserting into (8) yields

$$\mathbf{X}^{[3]}(\tau) = (3\tau^2 - 2\tau^3)\boldsymbol{\xi} + \tau(1-\tau)^2\mathcal{V}_0 + (\tau^3 - \tau^2)\mathbf{A}^{-1}(\boldsymbol{\xi})\mathcal{V}_T. \quad (9)$$

Formula (9) is a cubic interpolation between two poses with given initial and terminal velocity. If initial and terminal velocity are zero, $\mathbf{X}^{[3]}(\tau)$ defined in (9) yields the minimal acceleration curve as shown by Zefran [9], which is also a geodesic on $SO(3)$ w.r.t. the left invariant metric on $SE(3)$. A similar formula was presented in [16], [17], where the point of departure was a general cubic approximation of $\mathbf{X}(\tau)$, rather than a solution of the kinematic reconstruction equation. However, the interpolation proposed in [16], [17] does not yield the minimal acceleration curve.

C. Prescribed Terminal Velocity and Acceleration

A novel interpolation in terms of terminal velocity and acceleration can be derived by evaluating the extrapolation (5) backward along the trajectories. To this end, the shifted normalized path parameter $\bar{\tau} := \tau - 1 \in [-1, 0]$ is introduced, where $\bar{\tau} = -1$ corresponds to the initial, and $\bar{\tau} = 0$ to the terminal pose. The solution of the interpolation problem is expressed as

$$\mathbf{C}(\bar{\tau}) = \mathbf{C}_T\exp\hat{\bar{\mathbf{X}}}(\bar{\tau}) \quad (10)$$

where $\mathbf{C}_T = \mathbf{C}(0)$. This formulation describes the motion of moving frame $\mathcal{F}$ relative to its terminal pose $\mathbf{C}_T$. Evaluated at $\bar{\tau} = -1$, it yields the initial pose $\mathbf{C}_0 = \mathbf{C}_T\exp\hat{\bar{\mathbf{X}}}(-1)$, and the coordinate vector $\bar{\boldsymbol{\xi}} := \bar{\mathbf{X}}(-1)$ that produces the initial pose is thus $\bar{\boldsymbol{\xi}} = \log(\mathbf{C}_T^{-1}\mathbf{C}_0)$. Magnus expansion of the solution at the terminal pose with $\bar{\tau} = -1$ yields

$$\bar{\mathbf{X}}^{[3]}(\bar{\tau}) = \bar{\tau}\mathcal{V}_T + \frac{1}{2}\bar{\tau}^2\mathcal{V}_T' + \frac{1}{6}\bar{\tau}^3\mathcal{V}_T'' + \frac{1}{12}\bar{\tau}^3[\mathcal{V}_T, \mathcal{V}_T'] \quad (11)$$

in analogy to (5), where now $\bar{\tau} \in [-1, 0]$. Evaluated at the initial configuration at $\bar{\tau} = -1$, this yields the coordinate vector $\bar{\boldsymbol{\xi}} = -\mathcal{V}_T + \frac{1}{2}\mathcal{V}_T' - \frac{1}{6}\mathcal{V}_T'' - \frac{1}{12}[\mathcal{V}_T, \mathcal{V}_T']$. Solving this for $\mathcal{V}_T''$, and backsubstitution into (11), yields

$$\bar{\mathbf{X}}^{[3]}(\bar{\tau}) = -\bar{\tau}^3\bar{\boldsymbol{\xi}} + (\bar{\tau} - \bar{\tau}^3)\mathcal{V}_T + \frac{1}{2}(\bar{\tau}^2 + \bar{\tau}^3)\mathcal{V}_T'. \quad (12)$$

The final form of the interpolation formula is obtained by noting that $\bar{\boldsymbol{\xi}} = -\boldsymbol{\xi}$, and replacing the shifted path parameter $\bar{\tau}$ with the original normalized path parameter, as

$$\bar{\mathbf{X}}^{[3]}(\tau) = (\tau-1)^3 \boldsymbol{\xi} + \left(\tau^2-\tau\right)(2-\tau)\mathcal{V}_T + \frac{1}{2}\tau(\tau-1)^2 \mathcal{V}'_T \tag{13}$$

which satisfies $\bar{\mathbf{X}}^{[3]}(\tau=0) = -\boldsymbol{\xi}$ and $\bar{\mathbf{X}}^{[3]}(\tau=1) = \mathbf{0}$. The two-point interpolation is thus $\mathbf{C}(\tau) = \mathbf{C}_T \exp \hat{\bar{\mathbf{X}}}(\tau)$, which will be the basis for a novel spline interpolation with prescribed terminal velocity and acceleration in Sec. V-B.

V. Cubic Spline Interpolation

In the following, the cubic interpolation formulae (8) and (13) are employed to derive a cubic spline interpolation scheme. The spline interpolation presented in [16], [17] is re-derived from relation (8) that describes a curve in G between prescribed $\mathbf{C}_0$ and $\mathbf{C}_T$, with given initial values $\mathcal{V}_0, \mathcal{V}'_0$. A novel interpolation scheme is derived from (13), which yields a curve satisfying the terminal conditions $\mathcal{V}_T, \mathcal{V}'_T$.

The motion is expressed as function of time, and the trajectory is split into N time intervals. At N sampling time instants $t_0, t_1, \ldots, t_N$, the configurations $\mathbf{C}_i = \mathbf{C}(t_i), i = 0, \ldots, N$ are given. A kth-order continuous spline in the Lie group G is a curve $\mathbf{C}(t)$, passing through $\mathbf{C}_i, i = 0, \ldots, N$, with continuous $k-1$ derivatives of the twist. Expressing the motion as $\mathbf{C}(t) = \mathbf{C}_0 \exp(\mathbf{X}(t))$ implies that $\mathbf{X}(t)$ is C^k continuous, thus needs to be a polynomial of degree $k+1$ or higher. The standard spline interpolation is C^2 continuos, and $\mathbf{X}(t)$ is expressed by concatenation of cubic polynomials. C^2 continuity implies that the velocity and acceleration match at the knot points, which is referred to as compatibility condition. Cubic spline interpolation of $\mathbf{X}$ for rotations was addressed in [16], [17]. Bézier interpolation of $\mathbf{X}$ for rigid body motions was presented in [12].

A. Prescribed Initial Velocity and Acceleration

Denote with $T_i = t_i - t_{i-1}$ the length of time segment i, which covers the time duration $t \in [t_{i-1}, t_i]$, and the normalized path parameter with $\tau_i(t) := (t - t_{i-1})/T_i$. The trajectory in segment i, between successive poses $\mathbf{C}_i$ and $\mathbf{C}_{i-1}$, is expressed as $\mathbf{C}(\tau_i) = \mathbf{C}_{i-1} \exp(\mathbf{X}_i(\tau_i))$. The coordinates describing the relative transformation from the pose at t_i to that at t_{i-1}, within segment i, according to $\mathbf{C}_i = \mathbf{C}_{i-1}\exp(\hat{\boldsymbol{\xi}}_i)$, is $\boldsymbol{\xi}_i := \log(\mathbf{C}_{i-1}^{-1}\mathbf{C}_i)$. Both, $\mathbf{X}_i$ and $\boldsymbol{\xi}_i$ are represented in the moving frame at t_{i-1}.

The velocity and acceleration in segment i, normalized w.r.t. the duration T_i, are denoted with $\mathcal{V}_i(\tau_i)$ and $\mathcal{V}'_i(\tau_i)$. The terminal values $\mathcal{V}_{i-1}(1)$ and $\mathcal{V}'_{i-1}(1)$ of segment $i-1$ are the initial values $\mathcal{V}_i(0)$ and $\mathcal{V}'_i(0)$ of segment i. Second-order compatibility is achieved by identifying the velocity and acceleration at the sampling points, i.e. $T_i\mathcal{V}_{i-1}(1) = T_{i-1}\mathcal{V}_i(0)$ and $T_i^2\mathcal{V}'_{i-1}(1) = T_{i-1}^2\mathcal{V}'_i(0)$. With the cubic interpolation (8), the terminal velocity of segment i is

$$\begin{aligned}\mathcal{V}_i(1) &= \mathbf{A}(\boldsymbol{\xi}_i)\mathbf{X}_i'^{[3]}(1) \\ \mathcal{V}'_i(1) &= \mathbf{A}(\boldsymbol{\xi}_i)\mathbf{X}_i''^{[3]}(1) + \mathbf{B}(\boldsymbol{\xi}_i, \mathbf{X}_i'^{[3]}(1))\mathbf{X}_i'^{[3]}(1).\end{aligned} \tag{14}$$

The derivatives of $\mathbf{X}_i^{[3]}$ at $\tau_i = 1$ are obtained from (8), and with the compatibility conditions, as

$$\mathbf{X}_i'^{[3]}(1) = 3\boldsymbol{\xi}_i - 2\delta_i\mathcal{V}_{i-1}(1) - \tfrac{1}{2}\delta_i^2\mathcal{V}'_{i-1}(1) \tag{15}$$

$$\mathbf{X}_i''^{[3]}(1) = 6\boldsymbol{\xi}_i - 6\delta_i\mathcal{V}_{i-1}(1) - 2\delta_i^2\mathcal{V}'_{i-1}(1) \tag{16}$$

with $\delta_i := T_i/T_{i-1}$. This gives rise to the cubic spline interpolation scheme with initial velocity and acceleration:

- Input:
  $t_0, t_1, \ldots, t_N$ – sampling times
  $\mathbf{C}_0, \mathbf{C}_1, \ldots, \mathbf{C}_N$ – poses to be interpolated
  $\mathbf{V}_0, \dot{\mathbf{V}}_0$ – initial velocity and acceleration
- Initialization:
  $\boldsymbol{\alpha}_0^* := \mathbf{V}_0,\ \boldsymbol{\beta}_0^* := \dot{\mathbf{V}}_0,\ T_0 := 1$
- FOR $i = 1, \ldots, N$ DO
  $T_i := t_i - t_{i-1}$
  $\delta_i := T_i/T_{i-1}$
  $\boldsymbol{\xi}_i := \log(\mathbf{C}_{i-1}^{-1}\mathbf{C}_i)$
  $\boldsymbol{\alpha}_{i-1} := \delta_i\boldsymbol{\alpha}_{i-1}^*$
  $\boldsymbol{\beta}_{i-1} := \delta_i^2\boldsymbol{\beta}_{i-1}^*$
  $\mathbf{a}_i := 3\boldsymbol{\xi}_i - 2\boldsymbol{\alpha}_{i-1} - \frac{1}{2}\boldsymbol{\beta}_{i-1}$
  $\mathbf{b}_i := 6\boldsymbol{\xi}_i - 6\boldsymbol{\alpha}_{i-1} - 2\boldsymbol{\beta}_{i-1}$
  $\boldsymbol{\alpha}_i^* := \mathbf{A}(\boldsymbol{\xi}_i)\mathbf{a}_i$
  $\boldsymbol{\beta}_i^* := \mathbf{A}(\boldsymbol{\xi}_i)\mathbf{b}_i + \mathbf{B}(\boldsymbol{\xi}_i, \mathbf{a}_i)\mathbf{a}_i$
- Primary Output: $\boldsymbol{\xi}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i$
- Secondary Output: $\mathbf{V}_i = \frac{1}{T_i}\boldsymbol{\alpha}_i^*, \dot{\mathbf{V}}_i = \frac{1}{T_i^2}\boldsymbol{\beta}_i^*$

With the output of the algorithm, the trajectory is evaluated at $t \in [t_i, t_{i-1}]$ as $\mathbf{C}(t) = \mathbf{C}_{i-1}\exp(\mathbf{X}_i(\tau_i(t)))$, with $\mathbf{X}_i(\tau_i) = \tau_i^3\boldsymbol{\xi}_i + \left(\tau_i - \tau_i^3\right)\boldsymbol{\alpha}_{i-1} + \frac{1}{2}\left(\tau_i^2 - \tau_i^3\right)\boldsymbol{\beta}_{i-1}$, where $\tau_i(t) := \frac{t-t_{i-1}}{T_i}$. The velocity and acceleration can also be evaluated continuously from the obtained $\mathbf{V}_i$ and $\dot{\mathbf{V}}_i$.

The body-fixed formulation of the approximate solution is bi-invariant, which is not true for the spatial representation. This is the reason why the velocity and accelerations computed at the end of an interval can be used directly as the initial values for the subsequent interval, in contrast to the spatial representation as used in [26], [10].

B. Prescribed Terminal Velocity and Acceleration

Instead of the initial conditions, in many applications the twist and acceleration at the terminal pose are specified. However, no such interpolation method was reported in the literature that leads to an invariant spline interpolation algorithm. All interpolation methods that admit prescribing terminal velocities are based on Bézier splines, which leads to higher computational complexity. A novel cubic spline interpolation is derived from (17) and (13).

The motion within time interval i is described relative to the final pose $\mathbf{C}_i$ as $\mathbf{C}(\bar{\tau}_i) = \mathbf{C}_i \exp\hat{\bar{\mathbf{X}}}(\bar{\tau}_i)$, in terms of the shifted normalized path parameter $\bar{\tau}_i := \tau_i - 1$. It is $\bar{\tau}_i(t_{i-1}) = -1$ at the begin, and $\bar{\tau}_i(t_i) = 0$ at the end of segment i. The overall relative configuration attained within time interval i is $\mathbf{C}_{i-1} = \mathbf{C}_i\exp(\bar{\boldsymbol{\xi}}_i)$, with coordinates $\bar{\boldsymbol{\xi}}_i := \log(\mathbf{C}_i^{-1}\mathbf{C}_{i-1}) = -\boldsymbol{\xi}_i$. According to (12), the coordinate

vector describing the motion within time segment i is

$$\bar{\mathbf{X}}_i^{[3]}(\bar{\tau}_i) = -\bar{\tau}_i^3\bar{\boldsymbol{\xi}}_i + \left(\bar{\tau}_i - \bar{\tau}_i^3\right)\mathcal{V}_i(t_i) + \frac{1}{2}\left(\bar{\tau}_i^2 + \bar{\tau}_i^3\right)\mathcal{V}_i'(t_i) \quad (17)$$

where $\mathcal{V}_i\left(\bar{\tau}_i\right)$ is the twist corresponding to this interpolation. Now the compatibility condition is that the velocity and acceleration determined by the interpolation in time segment i and $i+1$, respectively, match at t_i. This implies that $T_{i+1}\mathcal{V}_i\left(1\right) = T_i\mathcal{V}_{i+1}\left(0\right)$ and $\mathcal{V}_i'\left(1\right) = \frac{T_i^2}{T_{i+1}^2}\mathcal{V}_{i+1}'\left(0\right)$. With this interpolation, the velocity and acceleration at t_{i-1} are

$$\begin{aligned}\mathcal{V}_i\left(0\right) &= \mathbf{A}\left(\bar{\boldsymbol{\xi}}_i\right)\bar{\mathbf{X}}_i'^{[3]}\left(0\right) \quad (18)\\ \mathcal{V}_i'\left(0\right) &= \mathbf{A}\left(\bar{\boldsymbol{\xi}}_i\right)\bar{\mathbf{X}}_i''^{[3]}\left(0\right) + \mathbf{B}(\bar{\boldsymbol{\xi}}_i, \bar{\mathbf{X}}_i'^{[3]}(0))\bar{\mathbf{X}}_i'^{[3]}(0)\,.\end{aligned}$$

The derivatives of $\bar{\mathbf{X}}_i^{[3]}$ are obtained from (17). Along with the compatibility condition, at t_{i-1}, i.e. $\bar{\tau}_i = -1$, these are

$$\bar{\mathbf{X}}_i'^{[3]}(0) = -3\bar{\boldsymbol{\xi}}_i - 2\delta_i\mathcal{V}_{i+1}(0) + \tfrac{1}{2}\delta_i^2\mathcal{V}_{i+1}'(0) \quad (19)$$

$$\bar{\mathbf{X}}_i''^{[3]}(0) = 6\bar{\boldsymbol{\xi}}_i + 6\delta_i\mathcal{V}_{i+1}(0) - 2\delta_i^2\mathcal{V}_{i+1}'(0) \quad (20)$$

where $\delta_i := T_i/T_{i+1}$. This yields the cubic spline interpolation through $\mathbf{C}_0, \mathbf{C}_1, \ldots, \mathbf{C}_N$, with terminal values $\mathbf{V}_T, \dot{\mathbf{V}}_T$:

- Input:
  $t_0, t_1, \ldots, t_N$ – sampling times
  $\mathbf{C}_0, \mathbf{C}_1, \ldots, \mathbf{C}_N$ – poses to be interpolated
  $\mathbf{V}_T, \dot{\mathbf{V}}_T$ – terminal velocity and acceleration
- Initialization:
  $\boldsymbol{\alpha}_N := \mathbf{V}_T,\ \boldsymbol{\beta}_N := \dot{\mathbf{V}}_T,\ T_{N+1} := 1$
- FOR $i = N, \ldots, 1$ DO
  $T_i := t_i - t_{i-1}, \delta_i := T_i/T_{i+1}$
  $\boldsymbol{\xi}_i := \log(\mathbf{C}_{i-1}^{-1}\mathbf{C}_i)$
  $\boldsymbol{\alpha}_i := \delta_i\boldsymbol{\alpha}_i^*$
  $\boldsymbol{\beta}_i := \delta_i^2\boldsymbol{\beta}_i^*$
  $\mathbf{a}_{i-1} := 3\boldsymbol{\xi}_i + 2\boldsymbol{\alpha}_i + \frac{1}{2}\boldsymbol{\beta}_i$
  $\mathbf{b}_{i-1} := -6\boldsymbol{\xi}_i - 6\boldsymbol{\alpha}_i - 2\boldsymbol{\beta}_i$
  $\boldsymbol{\alpha}_{i-1}^* := \mathbf{A}\left(-\boldsymbol{\xi}_i\right)\mathbf{a}_{i-1}$
  $\boldsymbol{\beta}_{i-1}^* := \mathbf{A}\left(-\boldsymbol{\xi}_i\right)\mathbf{b}_{i-1} + \mathbf{B}\left(-\boldsymbol{\xi}_i, \mathbf{a}_{i-1}\right)\mathbf{a}_{i-1}$
- Primary Output: $\boldsymbol{\xi}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, i = 0, \ldots, N-1$
- Secondary Output: $\mathbf{V}_i = \frac{1}{T_{i+1}}\boldsymbol{\alpha}_i^*, \dot{\mathbf{V}}_i = \frac{1}{T_{i+1}^2}\boldsymbol{\beta}_i^*$

With $\bar{\mathbf{X}}_i\left(\tau_i\right) = \left(\tau_i - 1\right)^3\boldsymbol{\xi}_i + \left(\tau_i^2 - \tau_i\right)\left(2 - \tau_i\right)\boldsymbol{\alpha}_{i-1} + \frac{1}{2}\tau_i\left(\tau_i - 1\right)^2\boldsymbol{\beta}_{i-1}$, the trajectory is evaluated at $t \in [t_{i-1},\, t_i]$ as $\mathbf{C}\left(t\right) = \mathbf{C}_i \exp\bar{\mathbf{X}}_i^{[3]}(\tau_i\left(t\right))$, where $\tau_i\left(t\right) := \left(t - t_i\right)/T_i$.

VI. Examples

A. UAV performing a Rendezvous Task

As first example, the trajectory planning for a UAV landing on and taking of from a moving platform is considered. The UAV has to start from the initial location (pose $\mathbf{C}_0$), pass through an intermediate location 1 (pose $\mathbf{C}_1$), and land at a mobile platform at 'rendezvous' location 2 (pose $\mathbf{C}_2$), which is moving with a constant velocity $\mathbf{V}_{\text{plat}}$. The rendezvous location must be reached in 4s. After a 1s rest at the platform, the UAV is at location 3 (pose $\mathbf{C}_3$), from where it must take off and reach the final location 5 (pose $\mathbf{C}_5$) within 3s by passing through location 4 (pose $\mathbf{C}_4$), as shown schematically in Fig. 1. This rendezvous maneuver is described

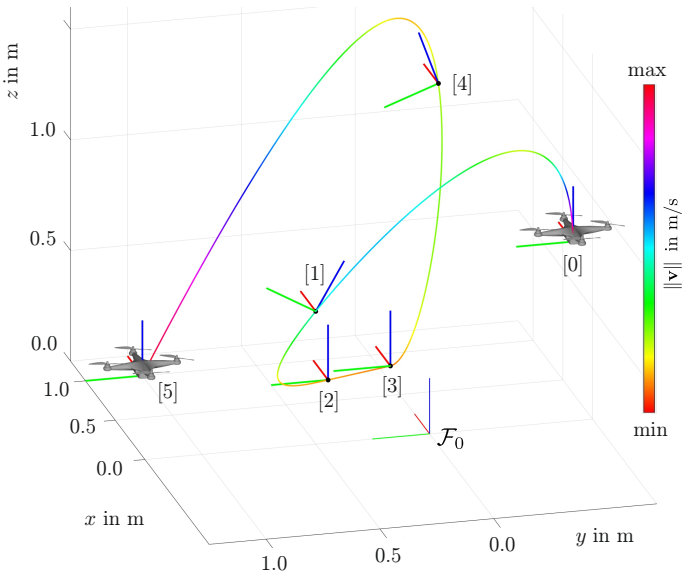

Fig. 1. Trajectory of a UAV performing a rendezvous with a moving platform (pose $\mathbf{C}_2$ and $\mathbf{C}_3$) where velocity and acceleration is prescribed. The UAV must change its attitude at via points (pose $\mathbf{C}_1$ and $\mathbf{C}_4$). Pose i is indicated with $[i]$. The color of the curve indicates the translation speed.

by two subsequent trajectories, for landing and take-off, respectively. The first part, trough $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2$, is interpolated using the spline algorithm in Sec. V-B, with terminal values $\mathbf{V}_{\text{plat}} = (0, 0, 0, 0.1 \text{ m/s}, -0.3 \text{ m/s}, 0)^T$ and $\dot{\mathbf{V}}_{\text{plat}} = \mathbf{0}$. The second sequence through $\mathbf{C}_3, \mathbf{C}_4, \mathbf{C}_5$ is interpolated using the algorithm in Sec. V-A with initial velocity $\mathbf{V}_{\text{plat}}$ and acceleration $\dot{\mathbf{V}}_{\text{takeoff}} = \left(0, 0, 0, 0, 0, 2 \text{ m/s}^2\right)^T$.

In the landing trajectory, the transition from $\mathbf{C}_0$ to $\mathbf{C}_2$ involves an attitude change of the UAV at the via point 1, described by $\mathbf{C}_1$. The subsequent take-off trajectory, from $\mathbf{C}_3$ to $\mathbf{C}_5$, involves an attitude change at via point 4, described by $\mathbf{C}_4$. The necessary initial velocity and acceleration at location 0 are the result of the cubic spline interpolation with prescribed terminal values $\mathbf{V}_{\text{plat}}, \dot{\mathbf{V}}_{\text{plat}}$ at rendezvous location 2. The terminal values at point 5 are the result of the spline interpolation with initial values $\mathbf{V}_{\text{plat}}, \dot{\mathbf{V}}_{\text{takeoff}}$, now at location 3. Fig. 1 shows the trajectory obtained with the two spline interpolations. The coloring of the motion curve indicated the translation speed. For the first sequence, involving $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2$, the speed ranges between min $= 0.311$ m/s and max $= 2.896$ m/s. In the second part, through $\mathbf{C}_3, \mathbf{C}_4, \mathbf{C}_5$, the speed is between min $= 0.316$ m/s and max $= 3.945$ m/s. The absolute values of translational and angular speed along the trajectory are shown in Fig. 2, which also shows the absolute value of the corresponding acceleration. The practical background for this maneuver is the motion planning for a UAV that must land and take off on a moving target, where it must approach through entrance points that require a certain attitude, e.g. small openings. The sequence of configurations $\mathbf{C}_i$, $i = 0 \ldots, 5$ is prescribed with position and orientation, w.r.t. the shown inertial frame $\mathcal{F}_0$. The corresponding numerical values are shown in Tab. I. An animation can be found in the video attachment.

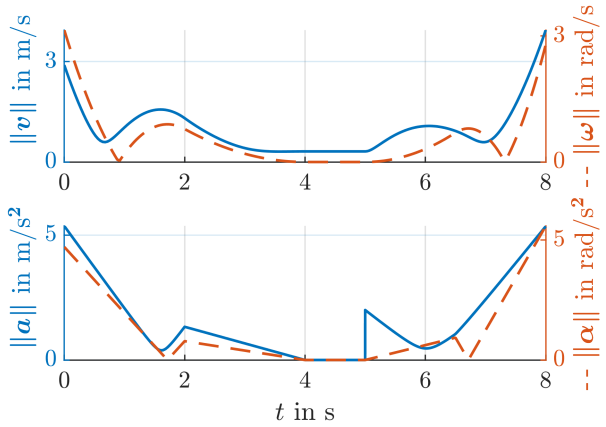

Fig. 2. UAV speed and acceleration during overall motion from initial to terminal pose. Solid lines indicate linear and dashed lines angular part.

TABLE I

NUMERICAL VALUES OF THE PRESCRIBED LOCATIONS.

Location i	Position $\mathbf{r}_i$ in m	Orientation $\mathbf{R}_i$
0	$[-0.5, -0.5, 1]^T$	$\mathbf{I}$
1	$[0, 0.5, 0.6]^T$	$\mathrm{Rot}(x, \pi/6)$
2	$[0.75, 0.25, 0]^T$	$\mathbf{I}$
3	$[0.85, -0.05, 0]^T$	$\mathbf{I}$
4	$[1, -0.25, 1.2]^T$	$\mathrm{Rot}(x, -\pi/9)$
5	$[1, 1, 0]^T$	$\mathbf{I}$

B. Describing the Deformation of Cosserat Beams

Cosserat beam models are extensively used to model and control soft robots as it captures large deflections. The displacement of a Cosserat beam is described by a curve in $SE(3)$, where $\mathbf{C}(\tau)$ describes the displacement field of the beam cross section (frame $\mathcal{F}$). The normalized path parameter is now defined by the arc length s along the beam and the length of the beam as $\tau = s/L$. The beam kinematics is governed by the relation [27], [22], [28]

$$\mathbf{C}' = \mathbf{C}\widehat{\boldsymbol{\chi}} \tag{21}$$

where $\boldsymbol{\chi} \in se(3)$ is the strain field (also called convected generalized curvature [29], [30]). Once the strain $\boldsymbol{\chi}(\tau)$ is known, the displacement field (1) is obtained by solving the kinematic reconstruction equation $\mathbf{X}' = \mathbf{dexp}^{-1}_{-\hat{\mathbf{X}}}\boldsymbol{\chi}$, analogous to (4), where $\mathbf{C}_0$ represents the initial deformation. The displacement can be interpolated when the strain at both ends of the beam or the strain and its derivative are known at the either end. This is an important issue for soft robot modeling and control, but also in flexible multibody dynamics in general [31]. In many robotic applications the strain is known from the boundary conditions or from the loading. Denote the stress with $\boldsymbol{\Lambda} \in se^*(3)$. Assuming linear elastic constitutive relation, the static deflection is governed by

$$\boldsymbol{\Lambda}' - \mathbf{ad}^T_{\boldsymbol{\chi}}\boldsymbol{\Lambda} = \mathbf{W} \tag{22}$$

$$\boldsymbol{\Lambda} = \mathbf{K}(\boldsymbol{\chi} - \boldsymbol{\chi}_0) \tag{23}$$

where, for a beam along the x-axis, the stiffness matrix is $\mathbf{K}(\tau) = \mathrm{diag}(GJ_x, EI_{yy}, EI_{zz}, EA, GA, GA)$, with A being the cross section area, I_{yy}, I_{zz} are the second moment of area, $J_x = I_{yy} + I_{zz}$ the polar momentum about x-axis, G the shear modulus, and E the longitudinal Young modulus. For initially straight beams it is $\boldsymbol{\chi}_0 = (0, 0, 0, 1, 0, 0)$. The 6 DOF of a Reissner beam can be reduced to the 3 DOF of a Kirchhoff beam, if transversal shear and extensibility are neglected. Incorporating the constitutive equation (23) into the 'static' Euler-Poincaré equation (22), yields

$$\boldsymbol{\chi}' = \boldsymbol{\chi}_0' + \mathbf{K}^{-1}(\mathbf{ad}^T_{\boldsymbol{\chi}}\mathbf{K} - \mathbf{K}')(\boldsymbol{\chi} - \boldsymbol{\chi}_0) + \mathbf{W} \tag{24}$$

where $\mathbf{W} \in se^*(3)$ is the load wrench. In this example, distributed loads are neglected. Let $\mathbf{W}_0, \mathbf{W}_T \in se^*(3)$ be the wrench acting at the beam at start and terminal end. From $\boldsymbol{\Lambda}(0) = -\mathbf{W}_0, \boldsymbol{\Lambda}(1) = -\mathbf{W}_T$ follow the boundary values $\boldsymbol{\chi}(0) = \boldsymbol{\chi}_0 - \mathbf{K}^{-1}(0)\mathbf{W}_0, \boldsymbol{\chi}(1) = \boldsymbol{\chi}_0 - \mathbf{K}^{-1}(1)\mathbf{W}_T$.

A rubber beam with a constant $8 \times 8\mathrm{mm}^2$ square cross section, 100 mm length, and $E = 10\,\mathrm{MPa}, G = 0.3\,\mathrm{MPa}$ is considered. The above boundary value problem along with the kinematic reconstruction equation were solved for different boundary loads, and the solution $\mathbf{X}(\tau)$ compared with the result of a cubic interpolation (9). The first example is a pure momentum $\mathbf{W} = (0, -0.05\,\mathrm{Nm}, -0.05\,\mathrm{Nm}, 0, 0, 0)$ acting at both ends. The beam bends about a skew axis with constant curvature, as shown in Fig. 3a). The cubic interpolation reproduces this solution up to machine precision. As second example, a force $\mathbf{f} = (0, 0.03, 0.05)\,\mathrm{N}$ is applied at the terminal end producing $\mathbf{W}_T$. The beam deflection is shown in Fig. 3b). The cubic interpolation well reproduces the deformation. Fig. 4 shows the accuracy measure $\boldsymbol{\varepsilon} = \mathbf{X} - \mathbf{X}^{[3]}$, where $\mathbf{X}^{[3]}$ is the interpolation result. If the beam undergoes more complex deformations the spline interpolation can be used to increase accuracy. The spline interpolation shall be used when displacement/strain values are available at intermediate locations along the beam, e.g. for multi-segment soft robots [32]. Invariance of the spline interpolation ensures strain-continuity.

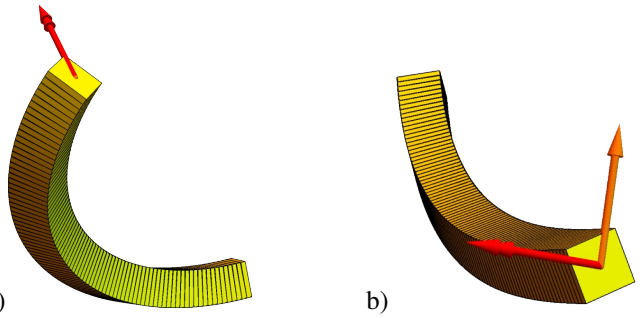

Fig. 3. Beam deformation for two different end loads $\mathbf{W} = (\mathbf{m}, \mathbf{f})$. The moment $\mathbf{m}$ and force $\mathbf{f}$ are indicated by read and orange arrows.

VII. SUMMARY AND OUTLOOK

A cubic spline interpolation scheme for spatial frame motions was introduced that, in contrast to existing formulations, allows to prescribe the terminal twist and acceleration. It

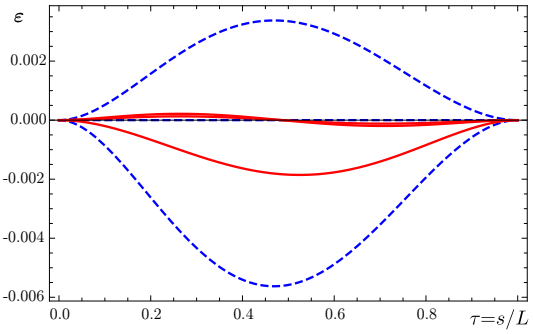

Fig. 4. Difference $\boldsymbol{\varepsilon}(\tau) = \mathbf{X}(\tau) - \mathbf{X}^{[3]}(\tau)$ of the exact solution $\mathbf{X}(\tau)$ and the interpolation $\mathbf{X}^{[3]}(\tau)$ of the beam displacement. Angular components are shown with dashed lines, and translational with solid lines.

is applicable to general trajectory planning tasks in robotics (e.g. UAVs, drones, space robots, and free-form interpolation of end-effector motions of robotic arms), as well as to reconstruct spatial motion (e.g. describe displacements of Cosserat beams models). In the first case, motions are described in $SO(3) \times \mathbb{R}^3$, whereas in the latter case on $SE(3)$. A flight scenario for UAV has been used as an example, where the proposed interpolation was applied to plan a trajectory guiding the UAV toward a rendezvous configuration with a prescribed velocity and acceleration. Interpolation of the displacement of a Cosserat beam was used to show the accuracy when reconstructing a given motion. Motion reconstruction from motion capture data is another typical application.

The proposed interpolation formulae are derived from a 3rd-order approximate solution of the left Poisson-Darboux equation (2) on the configuration Lie group G. In future work, the quartic interpolation formula (6) will be employed to derive 1) a quartic spline interpolation scheme that yields trajectories with continuos third derivatives $\dddot{\mathbf{V}}$, and 2) an interpolation with continuos $\ddot{\mathbf{V}}$ that admits to prescribe the velocities $\dot{\mathbf{V}}_i$ at knot points. The presented interpolation is directly applicable to motion parameterization in terms of quaternions and dual quaternions as the Lie algebras $se(3)$ and $sp(1)$ are isomorphic.

REFERENCES

[1] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[2] J. Selig, *Geometric Fundamentals of Robotics*. Springer, 2005.

[3] A. Müller, "Screw and Lie group theory in multibody dynamics –Motion representation and recursive kinematics of tree-topology systems," *Multibody System Dynamics*, vol. 43, no. 1, pp. 1–34, 2018.

[4] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 1985, pp. 245–254.

[5] L. Kavan, S. Collins, C. O'Sullivan, and J. Zara, "Dual quaternions for rigid transformation blending," *Trinity College Dublin*, vol. 5, 2006.

[6] A. Sarker, A. Sinha, and N. Chakraborty, "On screw linear interpolation for point-to-point path planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9480–9487.

[7] F. C. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," 1995.

[8] K. Shoemake, "Quaternion calculus and fast animation, computer animation: 3-d motion specification and control." Siggraph, 1987.

[9] M. Žefran, V. Kumar, and C. B. Croke, "On the generation of smooth three-dimensional rigid body motions," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 576–589, 1998.

[10] J. Selig, "Curves of stationary acceleration in se (3)," *IMA Journal of Mathematical Control and Information*, vol. 24, no. 1, pp. 95–113, 2007.

[11] F. Park and B. Ravani, "Bézier curves on Riemannian manifolds and Lie groups with kinematics applications," 1995.

[12] Q. Ge and B. Ravani, "Geometric Construction of Bézier Motions," 1994.

[13] J. Tan, Y. Xing, W. Fan, and P. Hong, "Smooth orientation interpolation using parametric quintic-polynomial-based quaternion spline curve," *Journal of Computational and Applied Mathematics*, vol. 329, pp. 256–267, 2018.

[14] M. Neubauer and A. Müller, "Smooth orientation path planning with quaternions using b-splines," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2087–2092.

[15] G. Legnani, I. Fassi, A. Tasora, and D. Fusai, "A practical algorithm for smooth interpolation between different angular positions," *Mechanism and Machine Theory*, vol. 162, p. 104341, 2021.

[16] F. C. Park and B. Ravani, "Smooth invariant interpolation of rotations," *ACM Transactions on Graphics (TOG)*, vol. 16, no. 3, pp. 277–295, 1997.

[17] I. Kang and F. Park, "Cubic spline algorithms for orientation interpolation," *International journal for numerical methods in engineering*, vol. 46, no. 1, pp. 45–64, 1999.

[18] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous robots*, vol. 35, no. 2, pp. 161–176, 2013.

[19] W. Magnus, "On the exponential solution of differential equations for a linear operator," *Communications on pure and applied mathematics*, vol. 7, no. 4, pp. 649–673, 1954.

[20] A. Müller, "Review of the exponential and Cayley map on SE (3) as relevant for Lie group integration of the generalized Poisson equation and flexible multibody systems," *Proceedings of the Royal Society A*, vol. 477, no. 2253, 2021.

[21] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, "Lie-group methods," *Acta Numerica*, vol. 9, p. 215–365, 2000.

[22] V. Sonneville, A. Cardona, and O. Brüls, "Geometrically exact beam finite element formulated on the special Euclidean group SE(3)," *Computer Methods in Applied Mechanics and Engineering*, vol. 268, pp. 451–474, 2014.

[23] L. Brand, *Vector and tensor analysis*. John Wiley & Sons, 1947.

[24] J. R. Martínez and J. Duffy, "An application of screw algebra to the acceleration analysis of serial chains," *Mechanism and Machine theory*, vol. 31, no. 4, pp. 445–457, 1996.

[25] A. Müller, "Approximation of finite rigid body motions from velocity fields," *ZAMM-Journal of Applied Mathematics and Mechanics*, vol. 90, no. 6, pp. 514–521, 2010.

[26] J. M. Selig and Y. Wu, "Interpolated rigid-body motions and robotics," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 1086–1091.

[27] D. C. Rucker, B. A. Jones, and R. J. Webster III, "A geometrically exact model for externally loaded concentric-tube continuum robots," *IEEE transactions on robotics*, vol. 26, no. 5, pp. 769–780, 2010.

[28] S. Briot and F. Boyer, "A geometrically exact assumed strain modes approach for the geometrico-and kinemato-static modelings of continuum parallel robots," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1527–1543, 2022.

[29] M. Borri and C. Bottasso, "An intrinsic beam model based on a helicoidal approximation—Part I: Formulation," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 13, pp. 2267–2289, 1994.

[30] ——, "An intrinsic beam model based on a helicoidal approximation—Part II: Linearization and finite element implementation," *International journal for numerical methods in engineering*, vol. 37, no. 13, pp. 2291–2309, 1994.

[31] L. Greco, A. Scrofani, and M. Cuomo, "A non-linear symmetric g1-conforming bézier finite element formulation for the analysis of kirchhoff beam assemblies," *Computer Methods in Applied Mechanics and Engineering*, vol. 387, p. 114176, 2021.

[32] A. L. Orekhov, G. L. H. Johnston, and N. Simaan, "Task and configuration space compliance of continuum robots via lie group and modal shape formulations," pp. 590–597, 2023.