2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

On Screw Linear Interpolation for Point-to-Point Path Planning

Anik Sarker[1], Anirban Sinha[2], and Nilanjan Chakraborty[3]

***Abstract*— Robot motion is controlled in the joint space whereas the robots have to perform tasks in their task space. Many tasks like carrying a glass of liquid, pouring liquid, opening a drawer requires constraints on the end-effector during the motion. The forward and inverse kinematic mappings between joint space and task space are highly nonlinear and multi-valued (for IK). Consequently, modeling task space constraints like keeping the orientation of the end-effector fixed while changing its position (which is required for carrying a cup of liquid without dropping it) is quite complex in the joint space. In this paper, we show that the use of screw linear interpolation to plan motions in the task space combined with resolved motion rate control to compute the corresponding joint space path, allows one to satisfy many common task space motion constraints in motion planning, without explicitly modeling them. In particular, any motion constraint that forms a subgroup of the group of rigid body motions can be incorporated in our planning scheme, without explicit modeling. We present simulation and experimental results on Baxter robot for different tasks with task space constraints that demonstrates the usefulness of our approach.**

I. Introduction

Planning and control of motion of a robot manipulator to go from an initial to a goal configuration is a fundamental problem in robotics. Additionally, during motion, different constraints on the end-effector may have to be satisfied. For example, consider the dual-handed manipulation task shown in Figure 1, where the robot has to hold a tray with a glass of liquid with both hands and move it from an initial to goal configuration. To accomplish the task successfully, the robot has to compute a path (i.e., a sequence of joint angles) such that the relative pose (position and orientation) between the two end-effectors as well as the orientation of the tray do not change along the entire path. The *goal of this paper is to study such motion planning problems with end-effector constraints (also called task space constraints) that have to be satisfied during the the motion*.

The *task space* of a manipulator is the set of all end-effector poses that it can reach. The *joint space* of a manipulator is the set of all feasible joint angles of the manipulator. A joint space configuration can be converted to a task space pose through the forward kinematics mapping, which is a nonlinear mapping. Joint space based (randomized) planning approaches are a widely used class of motion planning techniques [1], [2]. They have been developed to obtain feasible motion plans in the presence of obstacles. However, handling task space constraints in joint space based planning approaches is quite complicated [3], [4], [5], [6], [7] because they lead to nonlinear constraints that the joint angles should satisfy.

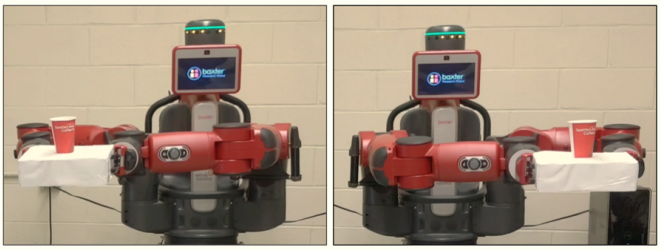

Fig. 1: The robot has to compute joint space path for both arms to transfer a tray with a liquid filled glass from an initial (right) to a goal (left) configuration. To accomplish the task, the relative configuration between the two end-effectors and the orientation of the tray should not change during motion.

Task space based planning approaches are historically older than joint space based approaches and rose out of the resolved motion rate control (RMRC) approach in [8]. In this approach, for a given start and goal pose, a path is first planned in the task space and then the (weighted) pseudoinverse of the manipulator Jacobian is used to compute the joint angles [9]. The second step is also known as redundancy resolution for redundant manipulators [10], [11]. Related to the task space based planning approaches are the operational space based control approaches [12], where the redundancy resolution may be done at the velocity level or acceleration level [13]. A key difference between the task space based path planning and the operational space control is that in the former it is assumed that the robots are position-controlled (and thus the output of planning is a sequence of joint angles), whereas in the latter it is assumed that the robots are torque controlled (and thus the output is a sequence of joint torques).

Since specifying tasks are more natural in the task space, much work has been done in development of task specification and prioritization in the context of operation space control [14], [15], [16]. For task space based planning, most papers assume that an end-effector path between start and goal poses is given. When just the start and end poses are given, it is usually assumed that only the position of the end-effector is of interest and thus the path can be computed by interpolation in $\mathbb{R}^3$ [11]. Note that the task space is a subset of $SE(3)$, which is a group and not a vector space.

This work was supported in part by NSF award CMMI 1853454. The authors are with the Department of Mechanical Engineering, Stony Brook University, 100 Nicolls Rd, Stony Brook, NY 11794.
[1] anik.sarker.me@gmail.com
[2] anirban.sinha@stonybrook.edu
[3] nilanjan.chakraborty@stonybrook.edu.

Thus, care needs to be taken to respect the group structure of $SE(3)$ during interpolation. To the best of our knowledge, this aspect has not been carefully considered in the context of task space based motion planning. However, in the theoretical kinematics, computer graphics, and CAD literature, many interpolation methods between rigid body poses in $SE(3)$ using a unit dual quaternion representation of $SE(3)$ have been studied. One such interpolation method is the screw linear interpolation (ScLERP), which is the analogue of straight line interpolation between two points in Euclidean space.

Contributions: In this paper we present a task space based path planning approach, where we use screw linear interpolation to compute a path in the task space and use pseudoinverse of the Jacobian to compute the corresponding path in the joint space. Note that although RMRC and screw linear interpolation have been both around for quite some time, the combination of these two for local path planning in task space has not been explored in the literature. Furthermore, we prove that by using ScLERP, any task space constraint that allows motion within subgroups of $SE(3)$ can be handled without explicitly considering them. This is rather surprising, because there has been significant effort in the literature on incorporating task space constraints like that shown in Table 1 [4], [5], [17], [18], [19]. Figure 2 shows one such exemplar problem of opening a door.

In our method, any task constraint that can be decomposed into a sequence or combination of these motion subgroup constraints can also be satisfied without explicitly enforcing them. For example, the solutions for dual handed manipulation example in Figure 1 can be obtained by planning the path in task space as follows: (1) Set the initial and goal configuration of each end-effector such that (a) the relative configuration between them at start and goal are same and (b) for each end-effector the orientation at the start and goal are same. (2) Plan the path for each end-effector using ScLERP with the same interpolation parameter. The resulting path will satisfy the desired motion constraints, i.e., the end effectors will have constant relative configuration and maintain constant orientation of the tray during the entire path. We also show other examples of transferring a glass of liquid with a single manipulator and pouring it into another container, which can be accomplished very simply without enforcing the task constraints along the path explicitly. We also validate our results experimentally using Baxter robot from ReThink Robotics. The videos of our experiments are presented in the video attachment to the paper.

Thus, *our key contribution is the use of ScLERP in the context of task space based motion planning and showing both mathematically and experimentally that this approach simplifies motion planning with task space constraints in some applications*. The use of dual quaternions to represent tasks have been proposed before in the context of operation space control [20]. However they do not use ScLERP. The use of linear interpolation for dual quaternion in the context of task space based path planning was proposed in [21]. The shortcomings of linear interpolation of unit dual quaternions compared to ScLERP have been pointed out in the graphics literature [22]. Furthermore, popular heuristics of decoupling interpolation of translation and rotation of the end-effector pose also does not ensure satisfaction of task space constraints. To the best of our knowledge, none of the previous work have shown that by using ScLERP, we can essentially satisfy some task constraints without explicitly considering them.

II. Mathematical Preliminaries

Let $\mathbb{R}^n$ be the real Euclidean space of dimension n, $\mathbb{R}^{m\times n}$ be the set of all $m \times n$ matrices with real entries. The set of all joint angles, $\mathcal{J} \in \mathbb{R}^n$, is called the *joint space* or the *configuration space of the robot* where n is the number of degrees of freedom (DoF) of the robot. The Special Orthogonal group of dimension 3, which is the space of all rigid body rotations is denoted as $SO(3)$. The Special Euclidean group of dimension 3, which is the space of rigid motions (i.e., rotations and translations) is denoted as $SE(3)$. Mathematically, $SO(3)$ and $SE(3)$ are defined as: $SO(3) = \left\{\mathbf{R} \in \mathbb{R}^{3\times3} | \mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}, |\mathbf{R}| = 1\right\}$, $SE(3) = \left\{(\mathbf{R}, \mathbf{p}) | \mathbf{R} \in SO(3), \mathbf{p} \in \mathbb{R}^3\right\}$, where $|\mathbf{R}|$ is the determinant of $\mathbf{R}$ and $\mathbf{I}$ is a 3×3 identity matrix. The set of all end-effector or hand configurations is called the *end-effector space* or *task space* of the robot and is a subset of $SE(3)$. A task space configuration $\mathbf{g} \in SE(3)$ can be written either as the pair $(\mathbf{p}, \mathbf{R})$ or as a 4×4 homogeneous transformation, i.e., $\mathbf{g} = \begin{bmatrix}\mathbf{R} & \mathbf{p}\\ \mathbf{0} & 1\end{bmatrix}$, where $\mathbf{0}$ is a 1×3 vector with all components as 0. An element of $SO(3)$, i.e., a rigid body rotation, can also be represented by a *unit quaternion* and an element of $SE(3)$, i.e., a rigid body motion can be represented by *a unit dual quaternion*.

Quaternion: A quaternion is a hypercomplex number, which can be represented by a tuple $\mathbf{Q} = (q_0, q_1, q_2, q_3)$ or as $\mathbf{Q} = q_1 i + q_2 j + q_3 k + q_0$ or $\mathbf{Q} = \mathbf{q} + q_0$ where q_0 is the real scalar part and $\mathbf{q}$ is the imaginary vector part with components q_1, q_2, q_3. Here, i, j, k are the imaginary numbers which have the following properties: i^2 = j^2 = k^2 = -1 and $ij = k$, $ji = -k$, $jk = i$, $kj = -i$, $ki = j$, $ik = -j$. Note that scalars, as well as vectors in $\mathbb{R}^3$, can also be represented as quaternions. For a scalar $q_1 = q_2 = q_3 = 0$, and for a vector $q_0 = 0$.

A unit quaternion is a quaternion where $\|\mathbf{Q}\| = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. The *conjugate* of a quaternion is defined as $\mathbf{Q}^* = q_0 - q_1 i - q_2 j - q_3 k$ whereas inverse is $\mathbf{Q}^{-1} = \mathbf{Q}^*/||\mathbf{Q}||$. For a unit-quaternion $\mathbf{Q}^{-1} = \mathbf{Q}^*$. A rotation can be represented as a unit quaternion: $\mathbf{Q} = \mathbf{u}\sin\theta/2 + \cos\theta/2$ where θ and $\mathbf{u}$ are angle and axis of the rotation. Let $\mathbf{P}$ and $\mathbf{Q}$ be the unit quaternion representations of two rotations. The composition of the two rotations is equivalent to the quaternion product given by $\mathbf{P} \otimes \mathbf{Q} = (p_0 q_0 - \mathbf{p} \cdot \mathbf{q}) + (p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q})$.

Dual-Quaternion: A dual number is a number of the form $a + \epsilon b$, where $a, b \in \mathbb{R}$, and $\epsilon^2 = 0$, although $\epsilon \neq 0$. The concept of dual numbers can be extended to dual vectors as well as dual quaternions [23]. A dual quaternion is defined as

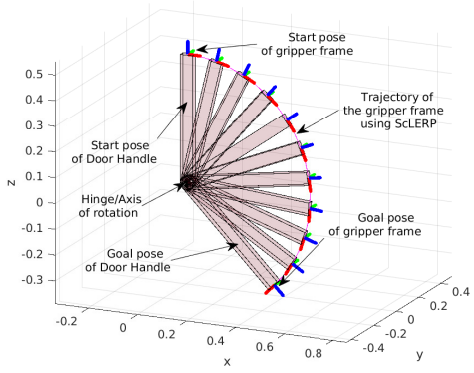

Fig. 2: Motion of a door handle (rectangular bars) obtained by the proposed motion planner using ScLERP based interpolation to generate end-effector poses. The ScLERP based interpolation implicitly maintain the rigid constraint imposed by the door handle, i.e. at all intermediate poses of the end-effector frame the distance between the end-effector frame and hinge of the handle are fixed.

$\mathbf{A} = \mathbf{A}_r + \epsilon\mathbf{A}_d$ where $\mathbf{A}_r$ and $\mathbf{A}_d$ are both quaternions and $\epsilon^2 = 0$, $\epsilon \neq 0$. The conjugate of a dual quaternion is $\mathbf{A}^* = \mathbf{A}_r^* + \epsilon\mathbf{A}_d^*$, where the conjugate on the right hand side is the conjugate operation on quaternions. The sum of two dual quaternions $\mathbf{A}$ and $\mathbf{B}$ is $\mathbf{A}+\mathbf{B} = (\mathbf{A}_r+\mathbf{B}_r)+\epsilon(\mathbf{A}_d+\mathbf{B}_d)$. Product of two dual quaternions $\mathbf{A}$ and $\mathbf{B}$ is obtained as $\mathbf{A}\otimes\mathbf{B} = \mathbf{A}_r\otimes\mathbf{B}_r+\epsilon(\mathbf{A}_d\otimes\mathbf{B}_r+\mathbf{A}_r\otimes\mathbf{B}_d)$, where $\otimes$ on the left denote dual quaternion product and $\otimes$ on the right denote quaternion product. For a *unit dual quaternion*, $\mathbf{A}\otimes\mathbf{A}^* = \mathbf{1}$, i.e., $||\mathbf{A}_r|| = 1$ and $\mathbf{A}_r \cdot \mathbf{A}_d = 0$. A unit dual quaternion $\mathbf{A} = \mathbf{A}_r + \frac{\epsilon}{2}\mathbf{A}_t \otimes \mathbf{A}_r$ represents rigid body motions or transformation where $\mathbf{A}_r$ is the rotation unit-quaternion and $\mathbf{A}_t$ is the linear translation vector (represented as a quaternion). If $\mathbf{A} = \mathbf{A}_r + \epsilon\mathbf{A}_d$ with $\mathbf{A}_r = \cos\frac{\theta}{2} + \mathbf{u}\sin\frac{\theta}{2}$ then power of $\mathbf{A}$ with respect to a fractional exponent $\tau \in [0,1]$ can be obtained as $\mathbf{A}^\tau = \cos\tau\bar{\theta} + \bar{u}\sin\tau\bar{\theta}$ where $\cos\bar{\theta} = \cos\frac{\theta}{2} - \epsilon\frac{d}{2}\sin\frac{\theta}{2}, \sin\bar{\theta} = \sin\frac{\theta}{2} + \epsilon\frac{d}{2}\cos\frac{\theta}{2}$. We find d as $d = (2\mathbf{A}_d \otimes \mathbf{A}_r^*)\cdot[0,1]^T$. Also $\bar{\mathbf{u}} = \mathbf{u} + \epsilon\mathbf{m}$ where $\mathbf{m} = \frac{1}{2}\left(\mathbf{p}\times\mathbf{u} + (\mathbf{p} - d\mathbf{u})\cot\frac{\theta}{2}\right)$ and $\mathbf{p}$ is the last three rows of $2\mathbf{A}_d \otimes \mathbf{A}_r^*$.

III. Problem Statement

Let $\mathbf{g}(0) = \mathbf{g}_0 \in SE(3)$ be the initial pose of the end-effector frame of a robot and $\mathbf{g}_d \in SE(3)$ be the desired end-effector pose. Let $\mathbf{\Theta} = [\theta_1, \theta_2, \cdots, \theta_{\mathrm{n}}]^{\mathrm{T}} \in \mathbb{R}^{\mathrm{n}}$ be the vector formed by concatenating the joint angles, where n is the DoF of the manipulator. Let $\mathbf{\Theta}(0) \in \mathbb{R}^{\mathrm{n}}$, be the initial joint space configuration of the manipulator. The *motion planning problem* that we consider is defined as: *Given* $\mathbf{g}_0$ *and* $\mathbf{g}_d$ *for a manipulator, compute a sequence of joint angles,* $\mathbf{\Theta}(\mathrm{i})$, $\mathrm{i} = 0, \ldots, \mathrm{m}$, *such that* $\mathcal{FK}(\Theta(\mathrm{m})) = \mathbf{g}_{\mathrm{d}}$, *where* $\mathcal{FK}$ *is the forward kinematics map of the manipulator. The sequence of joint angles or end-effector configurations may also need to satisfy different constraints like joint limits, collision avoidance with obstacles, and constraints on motion of the end-effector.*

Motion Subgroup	Exemplar Tasks
$\mathbb{R}^n, n = 1,2,3$	Opening a drawer ($\mathbb{R}$), Carrying a glass of water ($\mathbb{R}^3$).
$SO(n), n = 2,3$	Opening a hinged door, valve ($SO(2)$), Re-orienting an object while maintaining contact with a plane $SO(3)$.
$SE(2)$	wiping a flat surface, e.g., window pane
$SO(2)\times\mathbb{R}^1$	All rotations in a plane and translations perpendicular to the plane.
Screw Motion ($H(1)$)	Opening or tightening a bolt with a wrench.

TABLE I: Common Motion Constraints and Exemplar Tasks.

The motion planning problem becomes challenging in the presence of constraints on the motion of the end-effector as well as obstacles in the environment that need to be avoided. In this paper, we will be presenting a *local planning* or *point-to-point* planning approach that can take into consideration the constraints on the motion of the end-effector. Incorporation of joint limits and collision avoidance constraints with the proposed planner are described in [24]. .

Overview of Solution Approach: We propose to use a generalization of the Resolved motion rate control approach proposed in [8]. We use an iterative two-step approach: (a) In the first step we use a dual quaternion representation of the end-effector pose and use screw linear interpolation to compute a path in the task space for the robot to follow. (b) In the second step, we use the (weighted) pseudoinverse of the Jacobian to convert the task space path to a joint space path. The key innovation in this paper is the use of ScLERP to generate the interpolating poses and showing that for a wide class of constraints on the path, where the motion is restricted to sub-groups of $SE(3)$, ScLERP can automatically generate paths that satisfy the constraints.

The set of unit dual quaternions is a group under the operation of dual quaternion multiplication. Thus, the unit dual quaternion representation of $SE(3)$ respects the group structure of the rigid body motion. Furthermore (as we show below), when the motion is constrained to be in a subgroup of $SE(3)$, any interpolated pose obtained from ScLERP is guaranteed to be in the subgroup. In other words, without explicit representation or enforcement of the task constraints (that constrain the motion to be in a sub-group), the interpolated poses will always satisfy the task constraints along the path.

IV. Screw Linear Interpolation (ScLERP) and its properties

In this section, we present the basics of Screw Linear Interpolation (ScLERP) and some useful properties of ScLERP. We also compare ScLERP to two baseline approaches that are commonly used for path planning in task space ($\mathbb{T}$-space). Let $\mathbf{A}$ and $\mathbf{B}$ be two unit dual quaternions corresponding to two rigid body poses. ScLERP is a method to compute a curve in $SE(3)$ that interpolates between these two poses. It is the analogue of the shortest path motion in $\mathbb{R}^3$ for motion in $SE(3)$. Let $\tau \in [0,1]$ be the interpolation parameter. Then,

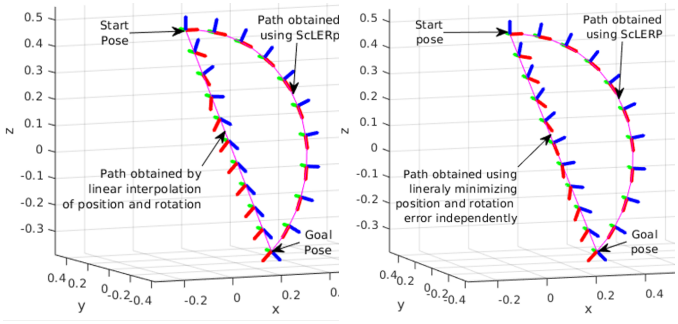

Fig. 3: Comparison of ScLERp based path generation for door-handle twisting task with (a) independent interpolation of position and orientation and (b) velocity controller by minimizing position and orientation error. The path generated by ScLERP (circular arc) maintained the path constraint imposed by the door handle while other interpolation approaches does not maintain the constraint.

any point on the path between $\mathbf{A}$ and $\mathbf{B}$ can be obtained for a given value of τ is $\mathbf{C}(\tau) = \mathbf{A} \otimes (\mathbf{A}^* \otimes \mathbf{B})^\tau$.

Given any two end-effector poses $\mathbf{A}$ and $\mathbf{B}$, there is a screw motion corresponding to the end-effector motion. In other words, there is an axis such that a rotation about that axis (say by angle, θ) and a translation along the axis (say by distance, d) transforms the $\mathbb{T}$-space pose from $\mathbf{A}$ to $\mathbf{B}$. This screw is an intrinsic property of rigid body displacement and independent of the choice of coordinate frames [22], [23]. Thus, the interpolated poses are also independent of the choice of coordinate frames. ScLERP has a very nice geometric interpretation with regard to this screw. For any choice of the parameter τ, the interpolated pose is such that the rotation is $\tau\theta$ and the translation is τd, i.e., the rotational and translational displacement are by the same fraction τ. This is the reason the motion is called screw linear, and note that the resultant motion need not be linear in $\mathbb{T}$-space.

Task Constraints: The key subgroups of $SE(3)$ and examples of the types of motion constraints they encode are given in Table I. These include the subgroup of pure translations (first row), pure rotations (second row), motion in a plane (third row), set of all motions that can be generated by a cylindrical joint (fourth row) and motions that include screwing or unscrewing operations (fifth row). Please note that when we say that the motion is constrained to be in a subgroup, it implies that the screw corresponding to the motion is in the subgroup; it *does not imply that the end-effector pose has to be in the subgroup*. For example, consider the motion required to open a refrigerator door or a kitchen cabinet door or turn a handle (as shown in Figure 3). The motion is in $SO(2)$, since it is a pure rotation about the hinge axis (the screw axis in this case). However, the end-effector pose has changes in both translation and orientation parameter. For existing task space based planning methods, which are coordinate dependent, the constraint that the end-effector pose lies on a circular arc has to be encoded explicitly by using an algebraic equation. However, as we will show below, since ScLERP is coordinate independent, we just need to provide the initial and final pose and the interpolated poses are guaranteed to be on a circular arc.

Interpolating Translation and Orientation Independently: A common heuristic used in robotics and also in many motion planning software packages is to interpolate the positions and orientations independently using the same interpolation parameter. Linear interpolation is used for the translation and spherical linear interpolation (SLERP) is used for orientation with a unit quaternion representation of orientation. However, this decoupling leads to a path that depends on the choice of the coordinate frame. The interpolated path does not always satisfy the motion group constraints. Thus, constraints for motions like opening a door or rotating a handle as shown in Figure 3 will not be satisfied when generating the path. As Figure 3 shows, the generated path does not lie on a circular arc and thus would not satisfy the $\mathbb{T}$-space constraint.

Minimizing the error to the goal in parameter space: Another approach to motion planning in task space is to use a kinematic controller to reduce the error between the initial and the goal pose. This approach can be used with any representation of $SE(3)$. Let $\boldsymbol{\gamma}$ be the end-effector pose parameters formed by concatenating the Cartesian position and unit quaternion representation of rotation. Let $\boldsymbol{\gamma}_0$ and $\boldsymbol{\gamma}_d$ be the initial and final end-effector pose respectively. Then, this approach tries to reduce the error $\mathbf{e} = \boldsymbol{\gamma}_d - \boldsymbol{\gamma}_0$ in each parameter and it is equivalent to (weighted) linear interpolation in the parameter space. As shown in Figure 3, for turning the handle, it also does not ensure that the $\mathbb{T}$-space path will lie on a circular arc. Thus motion constraints would not be satisfied along the path, unless explicitly enforced.

To summarize, there are multiple advantages of using ScLERP: (1) ScLERP is the generalization of straight line interpolation between two points in $\mathbb{R}^3$ to $SE(3)$. Thus, we have the analogue of the shortest path motion in $\mathbb{R}^3$ for the motion in $SE(3)$, when there are no constraints on the end-effector pose [25]. (2) If the motion of the end-effector has to lie within a subgroup of $SE(3)$, ScLERP with the given start and goal poses ensures that the path generated will always be in the same subgroup. Thus, motion constraints can be enforced without explicitly representing them. (3) Even when the motion constraints do not form a subgroup of $SE(3)$ or there are multiple sub-group constraints along a path, we can introduce intermediate way-points to ensure that the motion constraints are respected.

A. *Useful properties of ScLERP to satisfy task constraints*

For different motion subgroups, we now prove the claim that if the initial and final poses are in the subgroup, then all intermediate configurations will be in the subgroup. Because of space constraints we only provide some of the proofs (for $\mathbb{R}^3$ and $SO(3)$). The others (for $SE(2)$, cylindrical pair, and screw displacement) can be obtained as simple corollaries from the given proofs or in an analogous manner. (a) During a ScLERP, if the orientation of initial and goal location is kept same, i.e., if the motion is constrained to

$\mathbb{R}^3$, all intermediate locations will have the same orientation (see Lemma 1). The proof of this lemma can be extended to the subgroups $\mathbb{R}^2$ and $\mathbb{R}$ (b) During a ScLERP, if the position vector of initial and goal location is kept same, i.e., if the motion is constrained to $SO(3)$, then all intermediate configurations of will have the same position (see Lemma 2). This proof can be extended to $SO(2)$. (c) Given a pair of initial and a pair of goal configurations in dual-quaternion form such that the relative configurations between the initial and goal configuration pairs are the same, then performing ScLERP independently between the two start-goal-pairs preserves the same relative configurations between any intermediate configuration pair(see Lemma 3).

Lemma 1: Let $\mathbf{A} = \mathbf{A}_r + \epsilon\mathbf{A}_d$ and $\mathbf{B} = \mathbf{B}_r + \epsilon\mathbf{B}_d$ be two unit dual quaternions representing two task space configurations with $\mathbf{A}_r = \mathbf{B}_r = \mathbf{Z}$ (say). Let $\mathbf{C}(\tau) = \mathbf{C}_r(\tau) + \epsilon\mathbf{C}_d(\tau) = \mathbf{A} \otimes (\mathbf{A}^* \otimes \mathbf{B})^\tau$ be the configuration at any value of the interpolation parameter $\tau \in [0, 1]$. Then $\mathbf{C}_r(\tau) = \mathbf{Z}$, for all values of $\tau \in [0, 1]$.

Proof: Since $\mathbf{A} = \mathbf{A}_r + \epsilon\mathbf{A}_d = \mathbf{Z} + \epsilon\mathbf{A}_d$, therefore $\mathbf{A}^* = \mathbf{Z}^* + \epsilon\mathbf{A}_d^*$. Then

$$\begin{aligned}\mathbf{A}^* \otimes \mathbf{B} &= (\mathbf{Z}^* + \epsilon\mathbf{A}_d^*) \otimes (\mathbf{Z} + \epsilon\mathbf{B}_d) \\ &= \mathbf{Z}^* \otimes \mathbf{Z} + \epsilon\mathbf{K} \quad (\because \epsilon^2 = 0) \\ &= \mathbf{1} + \epsilon\mathbf{K} \quad (\because \mathbf{Z}^* \otimes \mathbf{Z} = \mathbf{1})\end{aligned} \tag{1}$$

where $\mathbf{K} = \mathbf{A}_d^* \otimes \mathbf{Z} + \mathbf{Z}^* \otimes \mathbf{B}_d$. Using algebraic manipulation (which we omit due to space constraints) it can be shown that for any dual quaternion of the form $\mathbf{Q} = \mathbf{1} + \epsilon\mathbf{K}$, $\mathbf{Q}^\tau = \mathbf{1} + \epsilon\bar{\mathbf{K}}(\tau)$ for any value of $\tau \in [0, 1]$. Therefore, using (1), we get $(\mathbf{A}^* \otimes \mathbf{B})^\tau = (\mathbf{1} + \epsilon\mathbf{K})^\tau = \mathbf{1} + \epsilon\bar{\mathbf{K}}$.

$$\begin{aligned}\therefore \mathbf{C}(\tau) &= \mathbf{A} \otimes (\mathbf{A}^* \otimes \mathbf{B})^\tau = (\mathbf{Z} + \epsilon\mathbf{A}_d) \otimes \left(\mathbf{1} + \epsilon\bar{\mathbf{K}}\right) \\ &= \mathbf{Z} + \epsilon(\mathbf{A}_d + \mathbf{Z} \otimes \bar{\mathbf{K}})\end{aligned} \tag{2}$$

Since (2) is valid for all $\tau \in [0, 1]$, $\mathbf{C}_r(\tau) = \mathbf{Z}, \forall\, \tau \in [0, 1]$. ■

Lemma 2: Let $\mathbf{A} = \mathbf{A}_r + \frac{\epsilon}{2}\mathbf{A}_t \otimes \mathbf{A}_r$ and $\mathbf{B} = \mathbf{B}_r + \frac{\epsilon}{2}\mathbf{B}_t \otimes \mathbf{B}_r$ be two unit dual quaternions representing two task space configurations with $\mathbf{A}_t = \mathbf{B}_t = \mathbf{Z}$ (say). Let $\mathbf{C}(\tau) = \mathbf{C}_r(\tau) + \frac{\epsilon}{2}\mathbf{C}_t(\tau) \otimes \mathbf{C}_r(\tau) = \mathbf{A} \otimes (\mathbf{A}^* \otimes \mathbf{B})^\tau$ be the configuration at any value of the interpolation parameter $\tau \in [0, 1]$. Then $\mathbf{C}_t(\tau) = \mathbf{Z}$, for all values of $\tau \in [0, 1]$.

Proof: Since $\mathbf{A} = \mathbf{A}_r + \frac{\epsilon}{2}\mathbf{Z} \otimes \mathbf{A}_r$,

$$\mathbf{A}^* = \mathbf{A}_r^* + \frac{\epsilon}{2}(\mathbf{Z} \otimes \mathbf{A}_r)^* = \mathbf{A}_\mathrm{r}^* + \frac{\epsilon}{2}(\mathbf{A}_\mathrm{r}^* \otimes \mathbf{Z}^*).$$

$$\therefore \mathbf{A}^* \otimes \mathbf{B} = \left(\mathbf{A}_r^* + \frac{\epsilon}{2}(\mathbf{A}_r^* \otimes \mathbf{Z}^*)\right) \otimes \left(\mathbf{B}_r + \frac{\epsilon}{2}(\mathbf{Z} \otimes \mathbf{B}_r)\right)$$

$$\begin{aligned}&= \mathbf{A}_r^* \otimes \mathbf{B}_r + \frac{\epsilon}{2}(\mathbf{A}_r^* \otimes \mathbf{Z}^* \otimes \mathbf{B}_r + \mathbf{A}_r^* \otimes \mathbf{Z} \otimes \mathbf{B}_r) \\ &= \mathbf{A}_r^* \otimes \mathbf{B}_r + \frac{\epsilon}{2}(\mathbf{A}_r^* \otimes (\mathbf{Z}^* + \mathbf{Z}) \otimes \mathbf{B}_r) \\ &= \mathbf{A}_r^* \otimes \mathbf{B}_r, \qquad \text{since } \mathbf{Z} + \mathbf{Z}^* = 0\end{aligned} \tag{3}$$

$$\therefore (\mathbf{A}^* \otimes \mathbf{B})^\tau = (\mathbf{A}_r^* \otimes \mathbf{B}_r)^\tau = \mathbf{D} \text{ (say)} \tag{4}$$

Note that $\mathbf{D}$ is a pure rotation with dual part 0.

$$\begin{aligned}\therefore \mathbf{C}(\tau) &= \mathbf{A} \otimes (\mathbf{A}^* \otimes \mathbf{B})^\tau = \mathbf{A} \otimes (\mathbf{A}_r^* \otimes \mathbf{B}_r)^\tau \\ &= \left(\mathbf{A}_r + \frac{\epsilon}{2}(\mathbf{Z} \otimes \mathbf{A}_r)\right) \otimes \mathbf{D} \\ &= \mathbf{A}_r \otimes \mathbf{D} + \frac{\epsilon}{2}\mathbf{Z} \otimes (\mathbf{A}_r \otimes \mathbf{D})\end{aligned} \tag{5}$$

From (5), $\mathbf{C}_r(\tau) = (\mathbf{A}_r \otimes \mathbf{D})$ and $\mathbf{C}_t(\tau) = \mathbf{Z}$, $\forall\, \tau \in [0, 1]$. ■

Lemma 3: Let $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{C}, \mathbf{D})$ be unit dual quaternions representing two pairs of initial and goal configurations such that the relative configurations between starts and goals are the same, i.e., $\mathbf{C} \otimes \mathbf{A}^* = \mathbf{D} \otimes \mathbf{B}^* = \mathbf{G}$. Let $\mathbf{E}(\tau) = \mathbf{A} \otimes (\mathbf{A}^* \otimes \mathbf{B})^\tau$ and $\mathbf{F}(\tau) = \mathbf{C} \otimes (\mathbf{C}^* \otimes \mathbf{D})^\tau$ be two poses on the path between $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{C}, \mathbf{D})$ respectively at the same value of the interpolation parameter τ. Then the relative configuration between $\mathbf{E}(\tau)$ and $\mathbf{F}(\tau)$ is $\mathbf{G}$, *i.e.*, $\mathbf{F}(\tau) \otimes \mathbf{E}^*(\tau) = \mathbf{G}$, $\forall\, \tau \in [0, 1]$.

Proof: Let $\mathbf{A} = \mathbf{A}_r + \epsilon\mathbf{A}_d$, $\mathbf{B} = \mathbf{B}_r + \epsilon\mathbf{B}_d$, $\mathbf{C} = \mathbf{C}_r + \epsilon\mathbf{C}_d$, and $\mathbf{D} = \mathbf{D}_r + \epsilon\mathbf{D}_d$. Since $\mathbf{C} \otimes \mathbf{A}^* = \mathbf{D} \otimes \mathbf{B}^*$, by pre-multiplying by $\mathbf{C}^*$ and post-multiplying by $\mathbf{B}$, we get $\mathbf{A}^* \otimes \mathbf{B} = \mathbf{C}^* \otimes \mathbf{D}$. Therefore, $(\mathbf{A}^* \otimes \mathbf{B})^\tau = (\mathbf{C}^* \otimes \mathbf{D})^\tau = \mathbf{Z}$ (say). Thus, $\mathbf{E}(\tau) = \mathbf{A} \otimes \mathbf{Z}$ and $\mathbf{F}(\tau) = \mathbf{C} \otimes \mathbf{Z}$. Thus

$$\begin{aligned}\mathbf{F} \otimes \mathbf{E}^* &= \mathbf{C} \otimes \mathbf{Z} \otimes (\mathbf{A} \otimes \mathbf{Z})^* = \mathbf{C} \otimes \mathbf{Z} \otimes \mathbf{Z}^* \otimes \mathbf{A}^* \\ &= \mathbf{C} \otimes \mathbf{A}^* = \mathbf{G}\end{aligned} \tag{6}$$

Therefore the relative configuration between $\mathbf{E}$ and $\mathbf{F}$ is same as that of $\mathbf{A}$ and $\mathbf{C}$ (or $\mathbf{B}$ and $\mathbf{D}$). ■

V. ScLERP based Motion Planning

A. Jacobian-based Inverse Kinematics

Let $\mathbf{p}$ denote the position of the end-effector and $\mathbf{Q}$ be the unit quaternion representing the orientation of the end-effector. We know that the spatial angular velocity, ω_s and the spatial linear velocity $\mathbf{v}_s$ is related to $\dot{\mathbf{p}}$ and $\dot{\mathbf{Q}}$ as

$$\omega_s = 2\mathbf{J}_1\dot{\mathbf{Q}}, \tag{7}$$

$$\mathbf{v}_s = \dot{\mathbf{p}} - \hat{\omega}_s\mathbf{p} = \dot{\mathbf{p}} + \hat{\mathbf{p}}\omega_s = \dot{\mathbf{p}} + 2\hat{\mathbf{p}}\mathbf{J}_1\dot{\mathbf{Q}} \tag{8}$$

In the above equation, the $\hat{\cdot}$ operator, converts any 3×1 vector to a 3×3 skew-symmetric matrix form, which is used to represent the cross product of vector ω_s and vector $\mathbf{p}$. $\mathbf{J}_1$ is the representation Jacobian that maps rate of change in rotation quaternion into angular velocity. Writing the above equations in matrix form, we obtain:

$$\begin{bmatrix}\mathbf{v}_s \\ \omega_s\end{bmatrix} = \underbrace{\begin{bmatrix}\mathbf{I} & 2\hat{\mathbf{p}}\mathbf{J}_1 \\ \mathbf{0} & 2\mathbf{J}_1\end{bmatrix}}_{\mathbf{J}_2}\begin{bmatrix}\dot{\mathbf{p}} \\ \dot{\mathbf{Q}}\end{bmatrix} = \underbrace{\mathbf{J}_2}_{6\times7}\begin{bmatrix}\dot{\mathbf{p}} \\ \dot{\mathbf{Q}}\end{bmatrix} \tag{9}$$

We know that $\mathbf{V} = [\mathbf{v}^s \quad \omega^s]^\mathrm{T} = \mathbf{J}_s\dot{\mathbf{\Theta}}$, where $\dot{\mathbf{\Theta}}$ is the vector of joint velocities and $\mathbf{J}_s$ is the spatial Jacobian of the manipulator. Therefore

$$\dot{\mathbf{\Theta}} = \mathbf{J}_s^\mathrm{T}\left(\mathbf{J}_s\mathbf{J}_s^\mathrm{T}\right)^{-1}\begin{bmatrix}\mathbf{v}_s \\ \omega_s\end{bmatrix} = \mathbf{J}_s^\mathrm{T}\left(\mathbf{J}_s\mathbf{J}_s^\mathrm{T}\right)^{-1}\mathbf{J}_2\begin{bmatrix}\dot{\mathbf{p}} \\ \dot{\mathbf{Q}}\end{bmatrix} \tag{10}$$

Let $\gamma = [\mathbf{p} \quad \mathbf{Q}]^{\mathrm{T}}, \dot{\gamma} = [\dot{\mathbf{p}} \quad \dot{\mathbf{Q}}]^{\mathrm{T}}, \mathbf{B} = \mathbf{J}_s^{\mathrm{T}} \left(\mathbf{J}_s \mathbf{J}_s^{\mathrm{T}}\right)^{-1} \mathbf{J}_2$. Therefore, from Equation (10), we can have

$$\dot{\mathbf{\Theta}} = \mathbf{B}\dot{\gamma} \tag{11}$$

Equation (11) can be used for both kinematics based motion planning and inverse kinematics (or redundancy resolution) for redundant manipulators. Let $\mathbf{\Theta}^{\mathrm{t}}$ and $\mathbf{\Theta}^{\mathrm{t+h}}$ be the joint angles at time t and $\mathrm{t} + \mathrm{h}$ respectively, where h is the interpolation time-step. Using a Euler time-step to discretize the Equation (11) and simpifying, we obtain

$$\mathbf{\Theta}^{\mathrm{t+h}} = \mathbf{\Theta}^{\mathrm{t}} + \mathbf{B}(\mathbf{\Theta}^{\mathrm{t}}) \left(\gamma^{\mathrm{t+h}} - \gamma^{\mathrm{t}}\right) \tag{12}$$

B. Steps in Motion Planning Algorithm

Input: Initial configuration of manipulator $\mathbf{\Theta}(0)$ (and therefore, initial configuration of end-effector, $\mathbf{g}_0$) and desired configuration, $\mathbf{g}_d$.

Output: A sequence of $\mathrm{n} \times 1$ joint-angle vectors, $\mathbf{\Theta}(0), \ldots, \mathbf{\Theta}(\mathrm{m})$ where n is number of degrees of freedom of arm.

1) Convert $\mathbf{g}_0$ and $\mathbf{g}_d$ to a dual quaternion representation $\mathbf{A}_0$, $\mathbf{A}_f$. Initialize $\mathrm{t} = 0$.
2) For $\mathrm{t} = 0$, $\mathbf{A}^{\mathrm{t}} = \mathbf{A}_0$, otherwise $\mathbf{A}^{\mathrm{t}}$ is known from the previous iteration. Perform a dual quaternion interpolation between $\mathbf{A}^{\mathrm{t}}$ and $\mathbf{A}_{\mathrm{f}}$ by choosing a value of the interpolation parameter (say, $\tau = 0.01$) to obtain the next configuration $\mathbf{A}^{\mathrm{t+h}}$.
3) Compute $\gamma^{\mathrm{t+h}}$ from $\mathbf{A}^{\mathrm{t+h}}$ and γ^{t} from $\mathbf{A}^{\mathrm{t}}$. Recall that γ is a 7×1 array of numbers with the first 3 denoting the position of the end-effector frame in the world coordinates and the last 4 denoting the unit quaternion representation of the tool frame with respect to the world frame. Compute $\mathbf{\Theta}^{\mathrm{t+h}}$ using Equation (13).

$$\mathbf{\Theta}^{\mathrm{t+h}} = \mathbf{\Theta}^{\mathrm{t}} + \beta \mathbf{B}(\mathbf{\Theta}^{\mathrm{t}}) \left(\gamma^{\mathrm{t+h}} - \gamma^{\mathrm{t}}\right) \tag{13}$$

   where $\beta \leq 1$ is a step length parameter for how far one is moving from $\mathbf{\Theta}^{\mathrm{t}}$.
4) Use forward kinematics to compute the actual configuration that the robot would reach, $\bar{\mathbf{g}}^{\mathrm{t+h}}$.
5) Check if $\bar{\mathbf{g}}^{\mathrm{t+h}}$ is *close enough* to $\mathbf{g}_d$. If yes, then stop, and return output. Otherwise, set $\mathrm{t} \leftarrow \mathrm{t} + \mathrm{h}$, convert $\bar{\mathbf{g}}^{\mathrm{t+h}}$ to the dual quaternion representation $\mathbf{A}^{\mathrm{t+h}}$, set $\mathbf{A}^{\mathrm{t}} \leftarrow \bar{\mathbf{A}}^{\mathrm{t+h}}$, set $\mathbf{\Theta}^{\mathrm{t}} \leftarrow \bar{\mathbf{\Theta}}^{\mathrm{t+h}}$ and go back to step 2.

In the supplementary document [26] we have presented the above steps in algorithmic format for the ease of implementation.

VI. Results and Discussions

In this section we provide simulation and experimental results obtained by using the proposed motion planner. We use the Baxter robot from ReThink Robotics as our experimental platform. The tasks that we consider are (a) turning a door handle (shown in Figure 2), (b) moving a glass of liquid from one position to another and pouring it into another container, and (c) moving a tray with a glass of water by using two hands. We have done multiple simulations and experiments for all the three cases. Because of space constraints, we provide simulation results for the example (a) and experimental results for (b) and (c). In scenario (b), the end-effector has to maintain the constraint that the motion should be pure translation so that the liquid is not spilled. Also, in example (b), part of the motion should be pure translation and part should be pure rotation. In example (c), both arms have to perform pure translation while also maintaining the same relative configuration between each other. As stated earlier, we do not put in any of these constraints explicitly, but just specify the start and goal end-effector configurations. The resulting paths automatically satisfies the task level constraints. Please see the supplementary video file of this paper demonstrating the proposed planner working in practice.

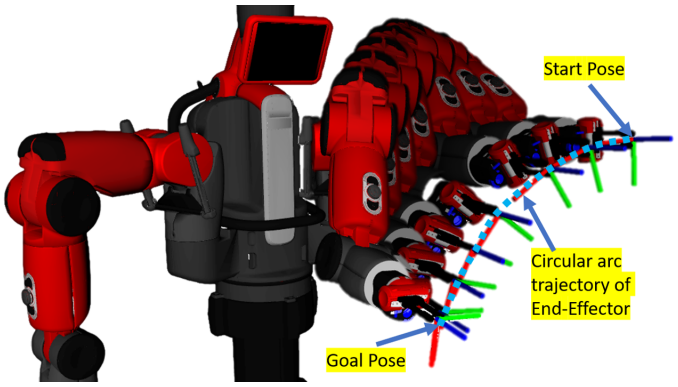

Fig. 4: Simulated Baxter arm executing circular end-effector path generated using ScLERP based proposed planner.

1) Planning for end-effector path when position and orientation of the end-effector changes simultaneously: The objective of this example is to compute a plan for the end-effector to turn a door handle to open a door. As discussed earlier, ScLERP based pose interpolation to generate intermediate poses has the advantage of naturally maintaining end-effector constraints as opposed to independent interpolation of the position and orientation methods (see Figure 3). So this example also serves as a case where ScLERP can only produce poses respecting desired circular nature of the desired end-effector path whereas other pose interpolation techniques cannot. The initial joint angle vector and intial and goal end-effector poses used to compute the plan using Algorithm V-B can be found in the supplementary document [26]. In Figure 4 we have shown the arm motion while executing the plan obtained using Algorithm V-B with a simulated Baxter robot. Notice that although we did not explicitly mention the task constraint, that the intermediate end-effector poses should lie on an arc.

2) Transferring and pouring a glass of liquid: In this example the robot transfers a glass of liquid and pours it into another container (see Figure 5). In this task the robot's position or orientation does not remain fixed during the entire task. However, it can be decomposed into two sub-tasks, where, in the first sub-task the robot moves the glass keeping its orientation fixed, and in the second sub-task, it pours the liquid by rotating the glass while keeping the position of its end-effector fixed. Note that the first sub-task is same as the

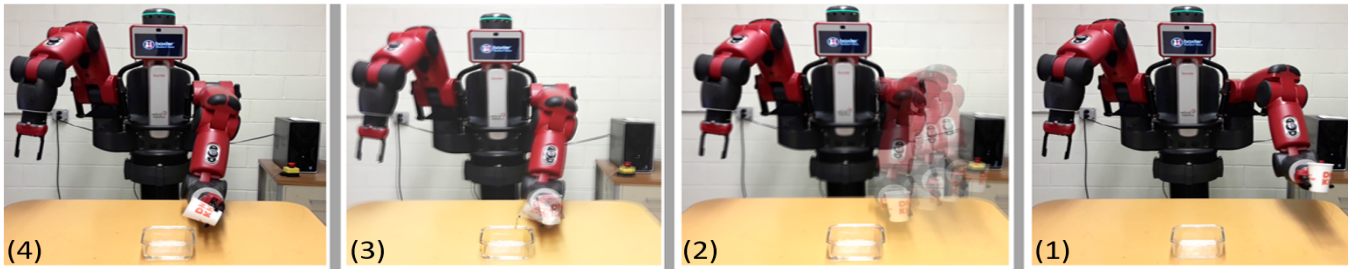

Fig. 5: From right to left: (1) Initial pose for the liquid transfer (sub-task-1) and pouring (sub-task-2). (2) Intermediate poses for sub-task-1 with fixed orientation of the end-effector as motion constraint. (3) Intermediate poses for sub-task-1 with fixed position of the end-effector as motion constraint. (4) Achieved Goal pose after successful completion of the task.

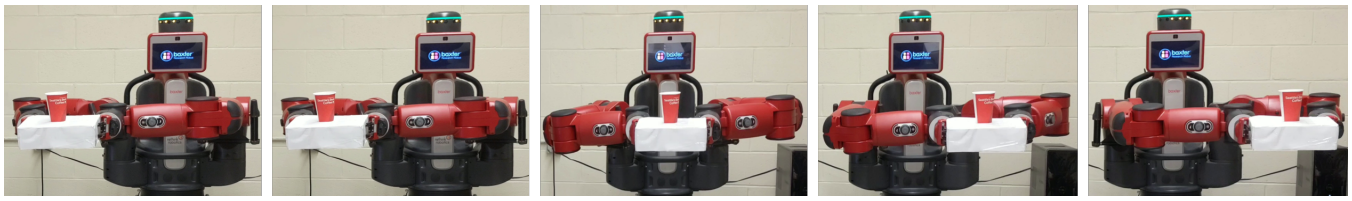

Fig. 6: From right to left: Baxter robot transferring glass of water (red) by moving a tray (white) held between left and right end-effectors from the initial to the goal pose. During the entire motion the rigidity constraint between the left and right gripper is maintained and glass stays upright.

previous example. We will call the first sub-task *the transfer task* and the second sub-task *the pouring task*. The initial and goal end-effector poses of the task and the corresponding values of the sub-tasks are given in [26]. We computed the plan for both the first and second sub-tasks. The obtained sequence of joint angles are then executed in Baxter robot and the motion is shown in Figure 5. Panel 1 shows the initial configuration and panel 2 shows some intermediate configurations to the end of the transfer. Panel 3 shows some intermediate configurations for pouring and panel 4 shows the goal configuration. The attached video submission also shows the full motion for this example along with couple of other instances of this task.

In Figure 7 we have plotted the path of all the 7 joints of Baxter robot while executing transfer and pouring tasks. The pouring task started at time corresponding to iteration 440 in Figure 7. Note that beyond this time stamp value, only the 7^{th} joint angle changes, which generates the rotational motion for pouring. All other joints have almost no change in values. *Notice again that we have not explicitly mentioned the task constraints while computing the plan yet those constraints were satisfied by the proposed planner*. This also corroborates the claim in Lemma 2.

3) Transferring a glass of liquid on a tray using dual arm manipulator: In this example, we consider a dual-armed manipulation task where the robot has to carry a tray with a glass of liquid from one position to another using both hands. There are two constraints here, (a) the orientation of the tray should not change during motion and (b) the relative pose between the two hands should not change during motion. The objective of this example is to show that for this dual-armed manipulation task, the plan for both the arms can be obtained independently using the proposed algorithm, as long

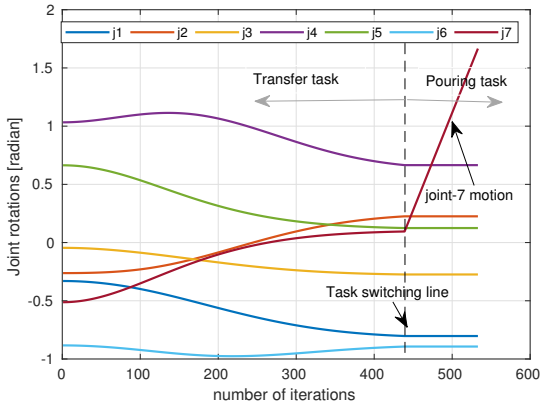

Fig. 7: Joint space path during the transfer and pouring task.

as we perform ScLERP with a common parameter τ for both hands. The resulting path for the end-effectors of both arms will implicitly satisfy constraints (a) and (b) above, at any intermediate point of the path.

We compute joint paths for left and right arms, that satisfy the task constraints, independently once with input tuple $(\boldsymbol{\Theta}_L(0), \mathbf{g}_{\text{L0}}, \mathbf{g}_{\text{Ld}})$ and once with input tuple $(\boldsymbol{\Theta}_R(0), \mathbf{g}_{\text{R0}}, \mathbf{g}_{\text{Rd}})$ respectively where subscript L and R denote respective quantities of left and right arms. The actual values used in the experiment are given in a supplementary document [26], due to space constraints. In Figure 6, we have shown the snapshots of the plan execution with Baxter robot. Please note the paths of the left and right end-effectors have been computed independently, but the relatives pose between the end-effectors is always fixed to the initial (or goal) relative poses at any intermediate time instant. It's worth

emphasizing that planning on a single arm and performing rigid body transformation between the end-effectors will provide us the relative pose of the second arm. However, we have to perform inverse kinematics on the second arm followed by forward kinematics. Thus, there will be an error between the actual pose of the second end-effector and the computed pose from the rigid body transformation. The error will accumulate over time and the relative configuration constraint between the two end-effector will not be properly maintained.

VII. Conclusion

In this paper, we have proposed a novel method for task-space based path planning by combining screw linear interpolation in $SE(3)$ with resolved motion rate control to convert $SE(3)$ paths to the joint space. The key advantage of our method is that it makes motion planning problems with task space constraints quite simple and efficient when the end effector motion is constrained to be in a subgroup of $SE(3)$. The manipulation of many human designed objects with joints, like opening a door, opening a drawer, naturally fits into this category, since joints restrict the relative motion of two objects to a subgroup of $SE(3)$. Furthermore, if the motion is constrained to be in a sequence of subgroups or combination of subgroups of $SE(3)$, our method can be used. Since our method for satisfying constraints is task space based, it is insensitive to the number of DoF of the manipulator. This is in contrast to joint space based methods where the motion subgroup constraints leads to nonlinear equations that the joint angles must satisfy during the motion, which increases the complexity of planning. We have presented theoretical proofs to show that ScLERP generates a path that satisfies motion subgroup constraints, given that the initial and final poses are in the same subgroup. The motion generated is coordinate invariant, i.e., it does not depend on choice of reference frames. We also presented multiple simulation and experimental results that demonstrate our algorithm.

Future Work: Although, the planner can be used even when the motion constraints do not lie in a subgroup of $SE(3)$ by selecting intermediate waypoints, we are currently performing this manually. We believe that the geometry of the screw motion can allow us to develop automated algorithms to select the waypoints and this is our future research. We also plan to integrate collision avoidance within this task space based motion planning scheme and some advances in this direction is reported in [24].

References

[1] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[2] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *International journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.

[3] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 625–632.

[4] D. Berenson, S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *The International Journal of Robotics Research*, vol. 30, no. 12, pp. 1435–1460, 2011.

[5] L. Jaillet and J. M. Porta, "Path planning under kinematic constraints by rapidly exploring manifolds," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 105–117, Feb 2013.

[6] M. Stilman, "Global manipulation planning in robot joint space with task constraints," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 576–584, June 2010.

[7] B. Kim, T. T. Um, C. Suh, and F. C. Park, "Tangent bundle rrt: A randomized algorithm for constrained motion planning," *Robotica*, vol. 34, no. 1, p. 202225, 2016.

[8] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on man-machine systems*, vol. 10, no. 2, pp. 47–53, 1969.

[9] C. A. Klein and C. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 2, pp. 245–250, March 1983.

[10] D. P. Martin, J. Baillieul, and J. M. Hollerbach, "Resolution of kinematic redundancy using optimization techniques," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 529–533, Aug 1989.

[11] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of intelligent and robotic systems*, vol. 3, no. 3, pp. 201–212, 1990.

[12] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, February 1987.

[13] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[14] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[15] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, June 1997.

[16] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, June 2009.

[17] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 625–632.

[18] M. Stilman, "Global manipulation planning in robot joint space with task constraints," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 576–584, 2010.

[19] O. Brock and O. Khatib, "Elastic strips: A framework for integrated planning and execution," in *Experimental Robotics VI*. Springer, 2000, pp. 329–338.

[20] B. V. Adorno, P. Fraisse, and S. Druon, "Dual position control strategies using the cooperative dual task-space framework," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 3955–3960.

[21] T. H. Connolly and F. Pfeiffer, "Cooperating manipulator control using dual quaternion coordinates," in *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, vol. 3. IEEE, 1994, pp. 2417–2418.

[22] L. Kavan, S. Collins, C. OSullivan, and J. Zara, "Dual quaternions for rigid transformation blending," *Trinity College Dublin, Tech. Rep. TCD-CS-2006–46*, 2006.

[23] K. Daniilidis, "Hand-eye calibration using dual quaternions," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.

[24] A. Sarker, "Gepmetri motion planing in task space using complementarity constraints to avoid collisions," Master's thesis, Stony Brook University, 2018.

[25] M. efran, V. Kumar, and C. Croke, "Metrics and connections for rigid-body kinematics," *The International Journal of Robotics Research*, vol. 18, no. 2, pp. 242–1–242–16, 1999.

[26] A. Sarker, A. Sinha, and N. Chakraborty, "Supplementary document: On screw linear interpolation for point-to-pointpath planning," https://bit.ly/32NJqtm.