

# Generalized Cylinders for Learning, Reproduction, and Refinement of Trajectories from Human Demonstrations\*

S. Reza Ahmadzadeh and Sonia Chernova

**Abstract**—This paper presents a novel parameter-free geometric approach to learning and reproducing trajectory-based skills from human demonstrations. We model a skill as a Generalized Cylinder which is a geometric representation composed of an arbitrary space curve called spine and a smoothly varying cross-section. To form the spine and the cross-section function of the generalized cylinder, our approach identifies and extracts the main characteristics of the demonstrated skill. These characteristics are spatial correlations across different demonstrations. Using a geometric rule, the encoded model reproduces the skill, as a time-independent trajectory, and generalizes it to unforeseen situations while its main characteristics are preserved. We also show that with a slight modification to the reproduction rule, our approach enables a human teacher to refine the characteristics of the learned skill through physical interaction. We validate the feasibility and efficiency of the proposed approach through several real-world experiments with a Jaco robotic arm.

## I. INTRODUCTION

Learning from Demonstration (LfD) facilitates teaching new skills to robots interactively by eliminating the need for manual programming of the desired behavior [1]. By observing a set of human-provided demonstrations, LfD approaches learn a model and generalize the encoded skill to novel situations autonomously. These capabilities make LfD a powerful approach that has the potential to enable even non-experts to teach new skills to robots with little effort. However, despite the existence of several trajectory-based LfD approaches, the vast majority of the existing robotic platforms rely on motion-level actions that are either hand-coded or teleoperated by experts [2], highlighting the need for further advances in this area.

One of the main factors that make existing approaches challenging (especially for non-experts) is the level of complexity of the existing representations. The number of parameters to be tuned and their corresponding tuning methods make it almost impossible for naïve users to tune the system and also understand how the system reacts to such adjustments. The other critical factor is that most available approaches require near-optimal demonstrations in order to perform effectively, while the redundancy and complexity of the current robotic platforms (i.e. high degrees of freedom), together with uncertainty and noise from the environment, demand a significant level of expertise to perform near-optimal demonstrations.

\*This work is supported in part by the Office of Naval Research award N000141410795.

Authors are with the School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA, 30332. {reza.ahmadzadeh, chernova}@gatech.edu

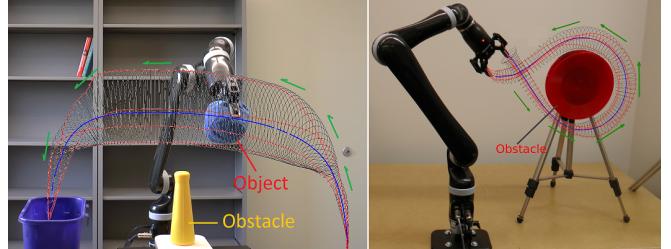


Fig. 1: Robot reproducing two trajectory-based skills encoded and learned using the proposed approach.

In addition, being able to refine the given demonstrations to improve the learned model is highly desirable. Very few attempts have been made to enable the teacher to refine the given demonstrations through incremental learning.

In this paper, we propose a novel parameter-free LfD approach with a geometric representation composed of a regular curve and a surface in 3D Cartesian space. In addition to capturing the demonstrated trajectories, the constructed model extracts and represents the main characteristics of the demonstrated skill, which are the spatial correlations across different demonstrations. These underlying characteristics are extracted from the raw data and are not specified by the user explicitly, thereby minimizing the effort of the user. Additionally, the proposed representation is visually perceivable and can reproduce the learned skill as a time-independent trajectory using a simple geometric rule. To overcome the issue of sub-optimal demonstrations, our approach enables the user to improve the learned model through physical motion refinement. Unlike other existing approaches, refinements can be applied to both the demonstrations and reproductions of the skill. Consequently, the user can start from a set of (sub-optimal) demonstrations and refine the learned model interactively to reach the desired behavior. In total, the proposed skill learning approach presented in this paper: *a)* maintains the important characteristics and implicit boundaries of the skill, *b)* generalizes the learned skill over the initial condition of the movement while exploiting the whole demonstration space to reproduce a variety of successful movements. *c)* requires no parameter tuning, *d)* enables users to provide physical feedback to improve the characteristics/quality of the learned skill interactively. We validate our approach in seven experiments using a physical 6-DOF robot, as well as demonstrate its use in comparison to prior work.

## II. RELATED WORK

Existing trajectory-based LfD approaches use various techniques to encode demonstrations and retrieve a generalized form of the skill [1]. A category of approaches use regression-based techniques to generate a probabilistic representation of the given demonstrations [3],[4]. Grimes *et. al.*, employed Gaussian Process (GP) regression to learn and generalize over a set of demonstrated trajectories [4]. In follow-on work, to overcome the computational cost of GP, Schneider and Ertel used local Gaussian process regression [5]. Another approach similar to GP called LfD by Averaging Trajectories (LAT), uses only one-dimensional normal distributions [6]. Both GP and LAT cannot extract constraints from the demonstrations with objects aligned parallel to a Cartesian coordinate axis. A well-known work in this category by Calinon *et. al.*, builds a probabilistic representation of the skill using a Gaussian Mixture Model (GMM) and retrieves a smooth trajectory using Gaussian Mixture Regression(GMR) [7]. GMM/GMR, GP, local GP, and LAT are time-dependent approaches that require explicit time-indexing and are useful for encoding skills that are deemed to be performed in a fixed amount of time. The skills learned using time-independent approaches, on the other hand, can be temporally scaled [8]. Another drawback of these approaches is that they require parameter tuning (e.g. number of Gaussian components, scale, weight, kernel).

Another category of approaches uses dynamical systems to encode and reproduce trajectories [9], [10]. Dynamic Movement Primitives (DMPs) represent demonstrations as movements of a particle subject to a set of damped linear spring systems perturbed by an external force [9]. The shape of the movement is approximated using Gaussian basis functions and the weights are calculated using locally weighted regression. DMPs are implicitly time-dependent and this makes the system sensitive to temporal perturbations. In addition, one has to tune the parameters of the dynamical systems, such as time constants and scaling factors.

The real-time motion planning approach proposed by Majumdar and Tedrake approximates a boundary around a trajectory, which is visualized as a funnel [11]. By computing a library of funnels and their corresponding controllers off-line, their approach generates trajectories from the library using a closed-loop system. The generated funnels illustrate a similar representation to our approach, however, we do not require extensive off-line computation. Dong and Williams proposed probabilistic flow tubes to represent trajectories by extracting covariance data [12]. The learned flow tube consists of a spine trajectory and 2D covariance data at each corresponding time-step. Although the approach was applied to extract a human’s intention, the flow tube representation can be seen as a special case of our approach in which the cross-sections are formed using covariance data.

Regardless of the technique used for learning from demonstration, the capability of improving the learned model by refining its shape or spatial constraints is highly desirable. This can become available through human-robot physical

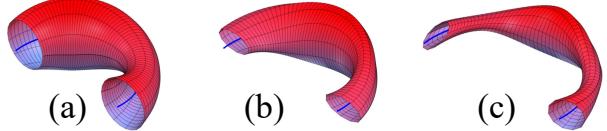


Fig. 2: Three generalized cylinders with identical directrix and different cross-section types. (a) is a canal surface.

interaction. There exist few approaches that enable the human to refine the initially given demonstrations. Argall *et. al.*, used tactile feedback for refining a given set of demonstrations and reusing the modified demonstrations to reproduce the skill through incremental learning [13]. They apply this approach to teaching a robot to position its hand for grasping of different objects. Lee and Ott proposed an incremental learning approach for iterative motion refinement. Their approach combines kinesthetic teaching with impedance control and represents the skill using a Hidden Markov Model (HMM) [14]. Our proposed approach, on the other hand, can be applied in both task-space and joint-space, and also it can be used to refine both demonstrations and reproductions interactively.

## III. BACKGROUND

A *Generalized Cylinder* (GC) represents an elongated object composed of an arbitrary space curve called a *directrix* (spine) and a smoothly varying cross-section [15]. In other words, a cross-sectional plane is translated along the directrix perpendicular to it and a cross-sectional curve is defined on the plane that may vary in shape and size during translation. Fig. 2 illustrates three GCs with an identical directrix and different cross-section types. Generalized cylinders play a fundamental role in Differential Geometry, and in the context of Computer Aided Graphic Design they are used for the construction of smooth blending surfaces, shape reconstruction and transition surfaces between pipes [16]. In robotics, generalized cylinders have been used for finding flyable paths for unmanned aerial vehicles [17]. A generalized cylinder with a circular cross-section is called a *Canal Surface* (CS) [18]. It is generated by sweeping a two-dimensional circle along the directrix perpendicular to the circle at its center. The radius of the circle may vary along the curve (Fig. 2(a)). Since canal surfaces are more intuitive, we first outline their mathematical definition and parameterized formulation and then extend the formulae to generalized cylinders.

### A. Canal Surfaces

Let  $\mathbb{R}^3$  be Euclidean 3–space with Cartesian coordinates  $x_1, x_2, x_3$ . Let  $\Phi_u$  be the one-parameter pencil<sup>1</sup> of regular implicit surfaces<sup>2</sup> with real-valued parameter  $u$ . Two surfaces corresponding to different values of  $u$  intersect in a common

<sup>1</sup>A pencil is a family of geometric objects sharing a common property (e.g. spheres).

<sup>2</sup>An implicit surface is a surface in Euclidean space that can be represented in  $F(x(u), y(u), z(u)) = 0$  form.

curve. As  $u$  varies, the generated surface is the envelope<sup>3</sup> of the given pencil of surfaces [19]. The envelope can be defined using the following equations:

$$\Phi_u : F(x_1, x_2, x_3, u) = 0, \quad (1)$$

$$\frac{\partial F(x_1, x_2, x_3, u)}{\partial u} = 0, \quad (2)$$

where  $\Phi_u$  consists of implicit  $C^2$ -surfaces which are at least twice continuously differentiable.

A **canal surface**,  $\mathcal{C}_u$ , is defined as an envelope of the one-parameter pencil of spheres and can be written as

$$\mathcal{C}_u : f(\mathbf{x}; u) := \{(\mathbf{c}(u), r(u)) \in \mathbb{R}^{3,1} | u \in \mathbb{R}\}, \quad (3)$$

where the spheres are centered on a regular curve  $\Gamma : \mathbf{x} = \mathbf{c}(u) \in \mathbb{R}^3$  in Cartesian space. The radius of the spheres are given by the function  $r(u) \in \mathbb{R}$ , which is a  $C^1$ -function. The non-degeneracy condition is satisfied by assuming  $r > 0$  and  $|\dot{r}| < \|\dot{\mathbf{c}}\|$  [16].  $\Gamma$  is the *directrix* (spine curve) and  $r(u)$  is called the *radii* function. For the one-parameter pencil of spheres, Equation (3) can be written as

$$\mathcal{C}_u : f(\mathbf{x}; u) := (\mathbf{x} - \mathbf{c}(u))^2 - r(u)^2 = 0. \quad (4)$$

### B. Generalized Cylinders

Since canal surfaces are constructed using the one-parameter pencil of spheres, the cross-sectional curve is always a circle even though its radius can vary along the directrix. Generalized cylinders generalize this idea by considering an arbitrary cross-sectional curve that can vary in both shape and size while sweeping along the directrix  $\Gamma$ .

A **generalized cylinder** can be defined as follows,

$$\mathcal{G}_{u,v} : f(\mathbf{x}; u, v) := \{\mathbf{c}(u), \rho(u, v) \in \mathbb{R}^{3,2} | u, v \in \mathbb{R}\}, \quad (5)$$

where  $\rho(u, v)$  represents the cross-sectional curve that is defined by two parameters,  $u$  the arc length on the directrix, and  $v$  the arc length on the cross-sectional curve. The dependence on  $u$  reflects the fact that the cross-section's shape may vary along the directrix. To obtain a parametric representation of generalized cylinders, it is useful to employ a local coordinate system defined with origin at each point on the directrix. A convenient choice is the Frenet-Serret (or TNB) frame which is suitable for describing the kinematic properties of a particle moving along a continuous, differentiable curve in  $\mathbb{R}^3$ . TNB is composed of three unit vectors  $\mathbf{e}_T$ ,  $\mathbf{e}_N$ , and  $\mathbf{e}_B$ , where  $\mathbf{e}_T$  is the unit tangent vector, and  $\mathbf{e}_N$  and  $\mathbf{e}_B$  are the unit normal and unit binormal vectors, respectively. For a non-degenerate directrix curve  $\Gamma : \mathbf{x}(u)$ , the TNB frame

<sup>3</sup>An envelope is a curve/surface tangent to a family of curves/surfaces (2D or 3D).

can be defined <sup>4</sup> using the following equations:

$$\mathbf{e}_T(u) = \frac{d\mathbf{x}(u)}{du}, \quad (6)$$

$$\mathbf{e}_N(u) = \frac{d\mathbf{e}_T}{du} / \left\| \frac{d\mathbf{e}_T}{du} \right\|, \quad (7)$$

$$\mathbf{e}_B(u) = \mathbf{e}_T \times \mathbf{e}_N. \quad (8)$$

By defining the cross-section in the TNB frame, we form a parametric representation of generalized cylinders as follows:

$$\mathcal{G}_{u,v} : f(u, v) = \mathbf{c}(u) + \rho_{x_1}(u, v)\mathbf{e}_N(u) + \rho_{x_2}(u, v)\mathbf{e}_B(u). \quad (9)$$

As mentioned earlier, generalized cylinders can be formed with cross-sections that not only vary in size but also in shape. The cross-section can be constructed using different shapes (such as circles, ellipses, polygons, closed-polynomials, and closed-splines [15]) and techniques (such as parabolic blending, and cone sections). These variations and characteristics make generalized cylinders a powerful candidate for modeling complicated constraints of trajectory-based skills captured through demonstrations.

## IV. SKILL LEARNING USING GENERALIZED CYLINDERS

In this section, we explain how generalized cylinders can be used to encode and reproduce trajectory-based skills from demonstrations. We assume that multiple examples of a skill are demonstrated and captured as a set of trajectories. To capture demonstrations we use kinesthetic teaching, however, various demonstration techniques such as teleoperation and shadowing can be employed. Given the set of captured demonstrations, our approach first calculates the directrix (i.e. an average form of the movements) and then extracts the main characteristics of the set (i.e. spatial correlations across demonstrations) and forms the cross-section function by identifying its boundaries. When the generalized cylinder is constructed, a geometric approach (i.e. ratio rule) is used for generating new trajectories starting from arbitrary initial poses. In this section, we explain both learning and reproduction phases of the proposed approach in detail. Algorithm 1 shows a pseudo code of our approach for encoding and reproducing skills by constructing generalized cylinders with closed-spline cross-sections.

### A. Skill Encoding

Consider  $n$  different demonstrations of a task performed and captured in task-space. For each demonstration, the 3D Cartesian position of the target (e.g. robot's end-effector) is recorded over time as  $\hat{\xi}^j = \{\xi_1^j, \xi_2^j, \xi_3^j\} \in \mathbb{R}^{3 \times T^j}$ , where  $j = 1 \dots n$  denotes the  $j^{\text{th}}$  demonstration and  $T^j$  is the number of data-points within the trajectory. Since  $T^j$  can vary between demonstrations, we use interpolation and resampling in order to gain a frame-by-frame correspondence mapping among the recorded demonstrations and align them temporally. To

<sup>4</sup>Calculating Frenet-Serret frames for real data is prone to noise. The reason is that at some points the derivative vector  $\frac{d\mathbf{e}_T}{ds}$  vanishes and the formulae cannot be applied anymore (i.e.  $\mathbf{e}_N$  cannot be calculated). This problem can be addressed by calculating the unit normal vector  $\mathbf{e}_N$  as the cross product of a random vector by the unit tangent vector  $\mathbf{e}_T$ .



Fig. 3: Kinesthetic demonstration of reaching to an object (yellow box). Captured task-space pose of the end-effector is used as input.

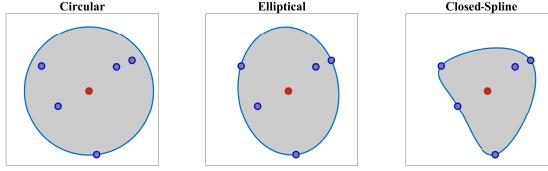


Fig. 4: Different types of cross-section on the same set of data. Point on the directrix and the effective points are shown in red and blue respectively.

achieve this, a set of piecewise polynomials is obtained using cubic spline interpolation for each demonstration. Then, we generate a set of temporally aligned trajectories by resampling  $N$  new data-points from each obtained polynomial. This process provides us with the set of  $n$  resampled demonstrations  $\xi \in \mathbb{R}^{3 \times N \times n}$ , each of which consists of  $N$  data-points. An advantage of this technique is that when the velocity and acceleration data are unavailable, the smoothed first and second derivatives of the obtained piecewise polynomials can be used instead. An alternate widely used solution is to employ Dynamic Time Warping [20].

**Estimating the directrix:** In order to estimate the directrix,  $\Gamma$ , we calculate the directional mean (axis-wise arithmetic mean) value for the given set of demonstrations. Let  $\mathbf{m} \in \mathbb{R}^{3 \times N}$  be the arithmetic mean of  $\xi$ . Note that  $\mathbf{m}$  is the space curve that all the cross-sections are centered on to form a generalized cylinder (Line 4 in Algorithm 1). Alternatively, the directrix can be produced using GMR [7]. In that case, GMR generates the directrix by sampling from the learned statistical model using GMM. However, using GMR requires defining a time vector explicitly. In this work we use the directional mean.

**Estimating the cross-section function:** Given  $n$  demonstrations and the directrix from the previous step, in this step, we explain methods for calculating the cross-section function  $\rho(u, v)$ . At each point  $u$  on the directrix, we gather one corresponding point (aligned with  $u$  on the same cross-sectional plane) from each demonstration; we call this set the *effective points*. We use the effective points to calculate the cross-section at each step with a smooth closed curve. The circumference of a cross-section represents the implicit local constraints of the task (i.e. boundaries) imposed by the set of demonstrations. In its simplest form, we can employ (4) and construct a canal surface which has a circular cross-section [21]. In this case, we calculate the distances from the point on the directrix to the effective points and use the maximum distance as the radius of the circle. The estimated cross-section bounds other points as well and therefore the

---

### Algorithm 1 Skill Learning using Generalized Cylinders

---

```

1: procedure ENCODING DEMONSTRATIONS
2:   Input: set of  $n$  demonstrations  $\xi \in \mathbb{R}^{3 \times N \times n}$ 
3:   Output:Generalized cylinder  $\mathcal{G}_{u,v}$ , TNB frames  $\mathcal{F}(u)$ 
4:    $\mathbf{m}(u) \leftarrow \text{mean}(\xi)$ 
5:    $\mathcal{P}(u, v) \leftarrow \text{estimateCSpline}(\xi)$ 
6:    $\mathcal{G}_{u,v}, \mathcal{F} \leftarrow \text{makeGeneralizedCylinder}(\mathbf{m}(u), \mathcal{P}(u, v))$ 
```

7: **procedure** REPRODUCING TRAJECTORY

```

8:   Input:initial point  $\mathbf{p}_0 \in \mathbb{R}^3$ ,  $\mathcal{G}_{u,y}$ ,  $\mathcal{F}$ 
9:   Output:New trajectory  $\rho \in \mathbb{R}^{3 \times N}$ 
10:   $\eta \leftarrow \frac{\|\mathbf{p}_0 - \mathbf{e}_0\|}{\|\mathbf{g}_0 - \mathbf{e}_0\|}$ 
11:   $\mathbf{p}_i \leftarrow \mathbf{p}_0$  ,  $\rho \leftarrow \mathbf{p}_0$ 
12:  for each frame  $\mathcal{F}_i$  do
13:     $\mathbf{p}_{i+1} \leftarrow \text{project}(\mathbf{p}_i, \eta, \mathcal{F}_{i+1}, \mathcal{F}_i)$ 
14:     $\rho \leftarrow \mathbf{p}_{i+1}$ 
15:     $i \leftarrow i + 1$ 
```

---

### Algorithm 2 Generating GC with arbitrary cross-section

---

```

1: procedure MAKEGENERALIZEDCYLINDER
2:   Input:directrix  $\mathbf{m}(u)$ , boundary function  $\mathcal{P}(u, v)$ 
3:   Output:Generalized cylinder  $\mathcal{G}_{u,v}$ , TNB frames  $\mathcal{F}_u$ 
4:   for each  $u_i$  do
5:      $\{\mathbf{e}_T(u_i), \mathbf{e}_N(u_i), \mathbf{e}_B(u_i)\} \leftarrow \text{estimateFrame}(\mathbf{m}(u_i))$ 
6:      $\mathcal{F} \leftarrow \{\mathbf{e}_T(u_i), \mathbf{e}_N(u_i), \mathbf{e}_B(u_i)\}$ 
7:      $\mathcal{G}_{u,v} \leftarrow \mathbf{m}(u_i) + \mathcal{P}^x(u_i, v)\mathbf{e}_N(u_i) + \mathcal{P}^y(u_i, v)\mathbf{e}_B(u_i)$ 
```

---

formed generalized cylinder (i.e. canal surface) encloses all the demonstrations. The radii function  $r(u) \in \mathbb{R}$  produces a radius for each step  $u$ , and  $v$  can be defined as an implicit parameter independently (e.g.  $v = [0, 2\pi]$ ). To cover a smaller yet more reasonable area while maintaining all the implicit local constraints of the task more efficiently, we can also construct generalized cylinders with elliptical cross-sections. The radii function for elliptical cross-section  $r(u) \mathbb{R} \mapsto \mathbb{R}^3$  produces the major and minor axes and the rotation angle of the ellipse for each step  $u$ .

In a more general form, we generate cross-sections by interpolating closed splines to the data. Given a set of break points  $v_j$ , ( $j = 1, \dots, n$ ) on the interval  $[v_0, v_n]$  such that  $v_0 < v_1 < \dots < v_{n-1} < v_n$ , we can fit a cubic polynomial to each interval.

$$p(v) = a_0 + a_1(v - v_0) + a_2(v - v_0)^2 + a_3(v - v_0)^3. \quad (10)$$

Each polynomial  $p(v)$  is described with four coefficients  $a_0, a_1, a_2, a_3$ . The accumulated square root of chord length is used to find the breaks and the number of polynomials. Since each polynomial is  $C^2$ -continuous, by applying the boundary condition  $p''(t_0) = p''(t_n) = 0$  and joining the polynomials we construct a smooth piecewise polynomial curve called a closed cubic spline. Therefore, the obtained spline is  $C^2$ -continuous within each interval and at each interpolating nodes. Fig. 4 illustrates three different types of cross-sections calculated for the same set of data.

## B. Skill Reproduction

During the reproduction phase, the initial position of the end-effector  $p_0$  in the cross-sectional plane  $S_0$  (perpendicular to the directrix at  $c_0$ ) is used as input. We measure the ratio  $\eta$  (Line 10) by calculating the distance  $\overline{p_0 c_0}$  and the distance  $\overline{g_0 c_0}$ , where  $g_0$  is located at the boundary of the corresponding cross-section (See Fig. 5). We use  $\eta$  to generate the next pose of the end-effector by transforming  $p_0$  from the current TNB frame  $\mathcal{F}_0$  to the next (Line 13). The pose on the second frame is then calculated as:  $p_1 = \eta(\overline{g_1 c_1})(T_{\mathcal{F}_1, \mathcal{F}_0})$ . An illustration of a single-step reproduction process using this *ratio rule* can be seen in Fig. 5. The ratio rule can generate new time-independent trajectories from any point inside the generalized cylinder. It also ensures that the essential characteristics of the demonstrated skill are applied to the reproduced time-independent trajectory.

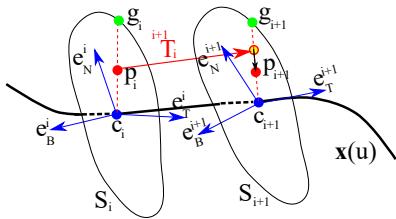


Fig. 5: Reproduction from a random initial pose  $p_i$  on the  $i^{\text{th}}$  cross-section  $S_i$ . Firstly the ratio of the initial point is calculated as  $\eta = \frac{\overline{p_i c_i}}{\overline{g_i c_i}}$ . Then the point is transferred to the next cross-section,  $S_{i+1}$  and scaled by  $\overline{g_{i+1} c_{i+1}}$  such that  $p_{i+1} = \eta \cdot \overline{g_{i+1} c_{i+1}} \cdot (T_{\mathcal{F}_{i+1}, \mathcal{F}_i})$ , where  $T$  is the transformation matrix between two frames.

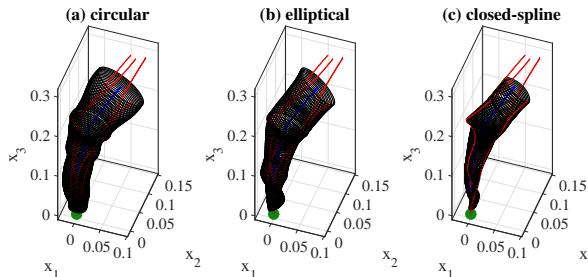


Fig. 6: A reaching skill encoded using three canal surfaces with different cross-section types. Demonstrations and the object are shown in red and green respectively.

## V. EXPERIMENTAL RESULTS

We conducted five experiments to demonstrate the validity of the proposed approach for representing and reproducing a variety of simple trajectory-based skills. For each experiment, a set of demonstrations was gathered through kinesthetic teaching using a 6DOF Kinova Jaco2 robotic arm (as shown in Fig. 3). The data was recorded at a sampling rate of 30Hz. For each experiment, the captured demonstrations together with the obtained generalized cylinder are illustrated<sup>5</sup> in Fig. 6 and Fig. 7.

<sup>5</sup>In all of the figures, the demonstrations, directrix, and reproductions are plotted in red, blue, and magenta, respectively. The generalized cylinders are either shown by their cross-sectional curves or continuous surfaces.

In the first experiment, a reaching skill towards an object (green sphere) from above is performed. The object, the recorded demonstrations, and the directrix are illustrated in Fig. 6. We encoded the skill using our approach with three different cross-section types, circular, elliptical, and closed-spline. This experiment shows how generalized cylinders with different cross-section types encode the main characteristics and implicit boundaries of the movement.

The demonstrations recorded for the second experiment (Fig. 7(a)) imply that the movement can be started and ended in a wide task-space; in the middle, however, it is constrained to pass through a narrow area. This movement resembles threading a needle or picking up an object in the middle of the movement. We estimate the directrix and the GC with a spline cross-section using Algorithm 1. The obtained GC extracts and preserves the important characteristics of the demonstrated skill. Given an arbitrary initial pose of the end-effector, the robot is capable of reproducing the learned skill using the ratio rule (as explained in Section IV-B).

The third experiment (Fig. 7(b)) shows a reaching/placing skill similar to the first experiment with a curvy movement. The robot learns to exploit a wider space while reaching the object but it has to be precise near the object. The video accompanying this paper shows the execution of this task.

The fourth experiment shows a circular movement around an obstacle (unknown to the robot) while avoiding collision (Fig. 7(c)). Our result shows that since the given demonstrations avoid the obstacle, the encoded GC guarantees that all the reproductions of the task remain inside the cylinder. Fig. 8 shows the boundaries of the generalized cylinder for this experiment. The first row depicts the demonstrations along three axes and the second row shows five reproductions of the skill using the ratio rule. It can be seen that all the reproductions stay inside the boundaries while exploiting the space represented by GC. Fig. 1 (right) shows a snapshot during the reproduction of the skill.

The fifth task represents a pick-and-place movement, where can be used for picking up an object and placing it in a box (Fig. 7(d)). The encoded GC shows that the initial and final pose of the movement are the main constraints of the tasks while in the middle the end-effector can pass a wider space while preserving the shape of the movement. Fig. 1 (left) shows a snapshot during the reproduction of the skill. In all of the experiments, the obtained GCs formed with closed-spline cross-sections, represent the demonstrated skills continuously and enable the robot to reproduce the movement from any initial pose.

## VI. COMPARISON TO DMPs

In this section, we compare the proposed approach to Dynamic Movement Primitives (DMPs) [9]. We have chosen the DMPs representation because it is widely-used and it handles the time-dependency implicitly (i.e. time-independent). Although we employed DMPs to learn and generalize the five tasks explained in Section V, due to lack of space, we only present results for the obstacle avoidance task and highlight the differences between the two approaches. The

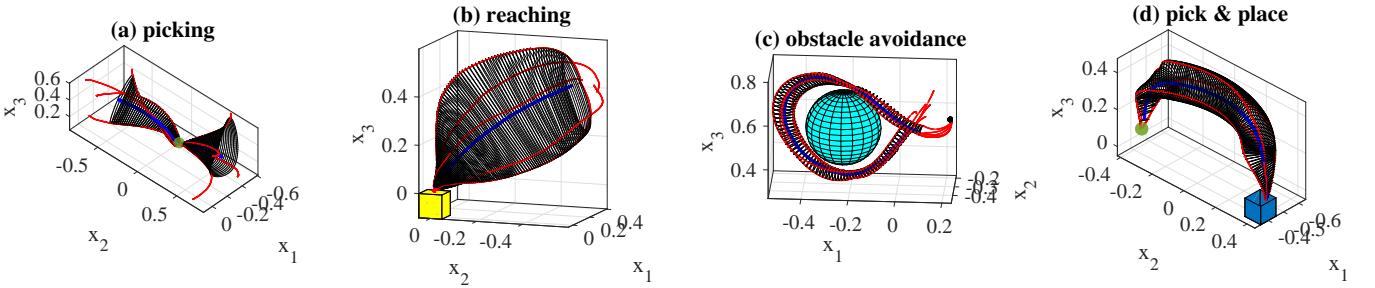


Fig. 7: Four real-world experiments performed using our approach. The demonstrations (red) are used to extract the main characteristics of the skills. The obtained directrix (blue) and generalized cylinders (gray) represent the skills.

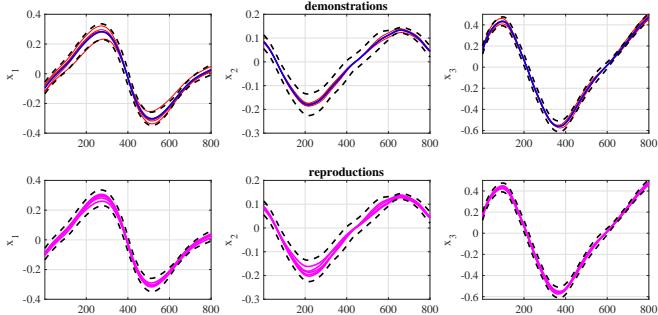


Fig. 8: Upper row shows the demonstrations (red), directrix (blue), and boundaries of the generalized cylinder (dashed) for the obstacle avoidance experiment. Lower row shows five reproductions of the skill (magenta).

recorded demonstrations and the obstacle, which is unknown to the robot, are depicted in Fig. 9(a). We encoded the demonstrations using a generalized cylinder with closed-spline cross-section and generated five reproductions from various initial poses (Fig. 9(b)). As mentioned before, our approach requires no parameter tuning and by extracting the characteristics of the movement it learns to avoid the obstacle. Also, all the reproduced trajectories are guaranteed to remain inside the GC. For DMPs, however, several parameters, such as proportional and damping gains, have to be tuned. In our first attempt, we trained DMPs with five attractors and reproduced five trajectories (from the same initial poses used before) using the learned model. Although the given demonstrations avoid the obstacle, it can be seen in Fig. 9(c) that the reproduced trajectories could not accurately mimic the movement. In our next attempt, we increased the number of DMPs' attractors to ten. The reproductions depicted in Fig. 9(d) indicate that the learned model is improved. This experiment shows the crucial and sensitive role of parameter tuning in DMPs.

In addition, the results show that all of the reproduced DMP trajectories converge toward the first attractor and exhibited a similar behavior afterward. It has to be noted that the reproduced trajectories using DMPs are similar to the directrix of the obtained GC (which by itself can be used to reproduce the motion). Whereas the reproduced trajectories by the generalized cylinder are not identical and exploit the whole cylinder while maintaining the important

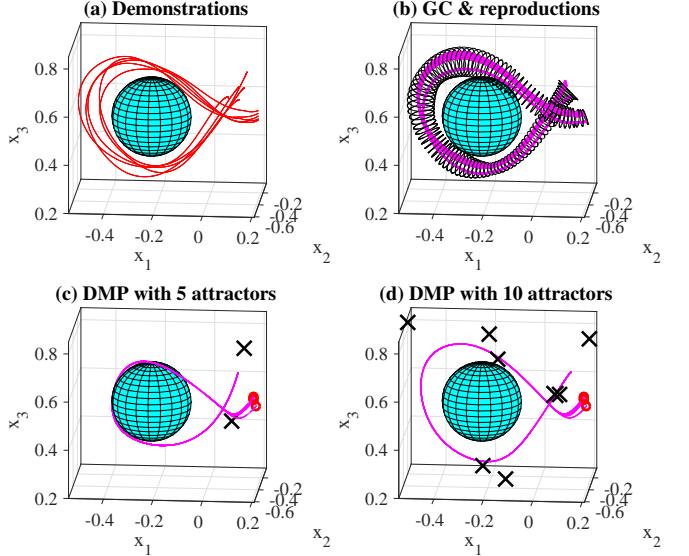


Fig. 9: Comparing to DMPs. (a) five demonstrations and the obstacle, (b) obtained generalized cylinder and five reproductions, (c) DMP with five attractors and five reproductions, and (d) DMP with ten attractors and five reproductions (see Section VI).

characteristics of the skill. In summary, without requiring any parameter tuning, our approach encodes the skill and can generate a variety of valid reproductions.

## VII. SKILL REFINEMENT

In practice, user-provided demonstrations of a task are usually sub-optimal. The demonstrations either include extra unnatural movements or do not represent the main intention of the user. As a result, the learned model will not embody the main characteristics of the intended movements and the reproductions would be dissimilar to the desired skill. One of the most popular (yet inefficient) solutions used even by experts, is to gather a larger set of demonstrations and throw out the ones with poor quality. Another approach is to refine the skill through human-robot physical interaction. A skill can be refined in two ways. First, during the learning process by modifying one or more of the demonstrations. The refined set then is fed back into the algorithm. Almost all existing approaches can benefit from this technique to improve the learned model. Second, a skill can be refined after the learning process by refining the reproductions. In

this section, we show that our approach can be used for refining skills through kinesthetic correction applied on both demonstrations and reproductions. We validate the feasibility of these techniques with two experiments. We have selected simple movements for analysis and illustrative purpose.

#### A. Refining Demonstrations

In its first form, skill refinement can be performed on demonstrations. After gathering the demonstrations and encoding the skill, the user identifies the demonstration(s) that needs to be modified (target demonstration) and executes it on the robot. While the robot is replaying the target demonstration, the teacher can apply a set of physical corrections. To achieve this capability, we activate our robot in compliant control mode that lets the joints and the end-effector be moved around while the robot is trying to follow the desired trajectory (target demonstration). The compliant behavior of the robot during the interaction can be adjusted by tuning the joints' stiffness and damping parameters. The teacher then can reshape the actual trajectory of the robot through kinesthetic correction. The obtained trajectory replaces the initial demonstration in the set. Given the new set of demonstrations, the algorithm updates the generalized cylinder and reproduces new trajectories that inherit the applied corrections.

The following experiment illustrates the validity of this process using our approach. Initially, we demonstrated three simple trajectories and encoded the skill as a GC with spline cross-section (using Algorithm 1). The demonstrations, directrix, and GC are illustrated in Fig. 10(a). As shown in Fig. 10(b), the skill can be reproduced using the ratio rule. Now, we assume there is an obstacle in the middle of the movement that requires the first demonstration (target demonstration) to bend downwards in the middle. While the robot is replaying the target demonstration, the teacher reshapes the demonstration through kinesthetic correction in the middle of the movement. Fig. 10(c) illustrates the original and refined demonstrations. Fig. 10(d) shows the updated GC and directrix after replacing the target with the refined demonstration. Given an arbitrary initial pose, the algorithm reproduces a new trajectory that reflects the performed refinements. The video accompanying this paper shows the execution of this task. This experiment shows that our approach can deal with the refinements applied to the demonstrations. Although many approaches can benefit from this process similarly, our representation is visually perceivable and enables even non-experts to observe and interpret the effect of the refinements on the model.

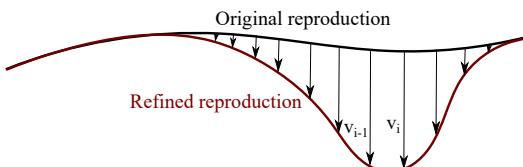


Fig. 11: Formation of the refinement matrix from the original and modified reproductions.

#### B. Refining Reproductions

In this section, we demonstrate that skill refinement can be performed by refining reproductions. It means that after the set of demonstrations are gathered and the skill is encoded, we can still refine the skill and apply new constraints to the model. As mentioned before, given an arbitrary point and using the ratio rule, our approach can reproduce a new trajectory that stays inside the GC. Now if during the reproduction, the movement is modified by the teacher, we can compare the original<sup>6</sup> and the modified reproductions, calculate point-to-point translation vectors, and form a refinement matrix by concatenating the vectors. The refinement matrix acts as a geometric constraint on the GC that would affect future reproductions.

To validate this process we conduct a new experiment using the previous dataset. The demonstrations, directrix, GC, and a reproduction of the skill are shown in Fig. 12(a),(b). The green trajectory in Fig. 12(c) shows the reproduction refined by the teacher through kinesthetic correction during the execution of the original reproduction. It can be seen that the teacher has applied downward forces ( $-x_3$  direction) to keep the end-effector at a certain level. We calculate the refinement matrix  $\hat{V} = [v_1, \dots, v_n] \in \mathbb{R}^{3 \times n}$  and apply it as a constraint to our ratio rule. In other words, a reproduction remains unaffected if it is generated below the constraining plane. This case can be seen as the lower reproduction in Fig. 12(d). On the other hand, if a reproduction intersects with the constraining plane, the refinement matrix applies to it. The upper reproduction in Fig. 12(d) shows the effect of the constraint while the dashed line shows reproduction without applying the constraint. The video accompanying this paper shows the execution of this task. This experiment indicates that using the proposed approach, the user can apply new constraints to the model without modifying it. The constraint later can be removed or combined with other constraints. To our knowledge there is no other LfD approach with similar capabilities.

## VIII. CONCLUSIONS

We have presented a novel LfD approach for learning and reproducing trajectory-based skills. Our geometric representation maintains the important characteristics and implicit boundaries of the skill and generalizes it over the initial condition of the movement. By exploiting the whole demonstration space, it reproduces a variety of successful movements. In addition, the proposed approach requires no parameter tuning that not only simplifies the usage of the algorithm and makes the result consistent, but also makes the approach more convenient for non-expert users. We also have shown that our approach enables users to refine the learned skill interactively through kinesthetic correction. We demonstrated that the skill refinement can be performed both on demonstrations and reproductions. During the demonstration refinement process the algorithm updates the learned model. While in the reproduction refinement

<sup>6</sup>Notice that we do not need to execute the original reproduction.

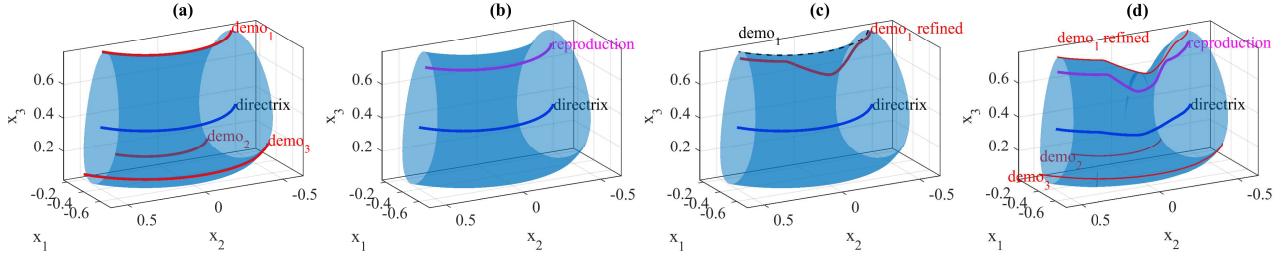


Fig. 10: Skill refinement by correcting demonstrations. (a) demonstrations (red), directrix (blue) and the obtained GC, (b) reproduction from a random pose (magenta), (c) first demonstration was refined (red) by the teacher, (d) updated GC, directrix, and a new reproduction.

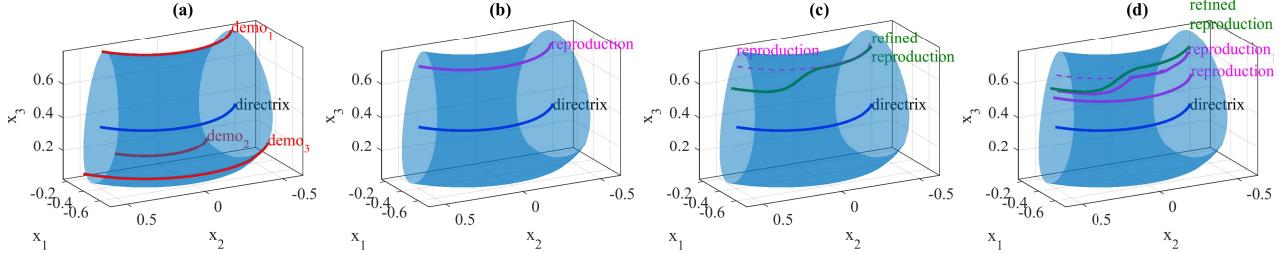


Fig. 12: Skill refinement by correcting reproductions. (a) demonstrations (red), directrix (blue) and the obtained GC, (b) reproduction from a random pose (magenta), (c) reproduction was refined (green) by the teacher, (d) two new reproductions; the upper one is affected by the refinement, while the lower one is not.

the model stays intact and the applied physical corrections form a constraint that affects the reproduction rule. Our representation also facilitates the visual interpretation of the reproductions and refinements. Further work will concentrate on applying our method to more complex skills such as bimanual and collaborative tasks, and compare it to other existing approaches. We also plan to conduct a user study to measure the efficiency of the proposed approach for non-expert users as well as experts.

## REFERENCES

- [1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.
- [2] H. A. Yanco, A. Norton, W. Ober, D. Shane, A. Skinner, and J. Vice, "Analysis of human-robot interaction at the darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 3, pp. 420–444, 2015.
- [3] S. Vijayakumar, A. D'souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural computation*, vol. 17, no. 12, pp. 2602–2634, 2005.
- [4] D. B. Grimes, R. Chalodhorn, and R. P. Rao, "Dynamic imitation in a humanoid robot through nonparametric probabilistic inference," in *Robotics: science and systems*. Cambridge, MA, 2006, pp. 199–206.
- [5] M. Schneider and W. Ertel, "Robot learning by demonstration with local gaussian process regression," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 255–260.
- [6] B. Reiner, W. Ertel, H. Posenauer, and M. Schneider, "Lat: A simple learning from demonstration method," in *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 4436–4441.
- [7] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 286–298, 2007.
- [8] E. Gribkovskaya, S. M. Khansari-Zadeh, and A. Billard, "Learning nonlinear multivariate dynamics of motion in robotic manipulators," *The International Journal of Robotics Research*, pp. 1–37, 2010.
- [9] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Robotics and Automation (ICRA), 2002 IEEE International Conference on*, vol. 2. IEEE, 2002, pp. 1398–1403.
- [10] M. Hersch, F. Guenter, S. Calinon, and A. G. Billard, "Learning dynamical system modulation for constrained reaching tasks," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2006, pp. 444–449.
- [11] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *arXiv preprint arXiv:1601.04037*, 2016.
- [12] S. Dong and B. Williams, "Learning and recognition of hybrid manipulation motions in variable environments using probabilistic flow tubes," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 357–368, 2012.
- [13] B. D. Argall, E. L. Sauser, and A. G. Billard, "Tactile guidance for policy refinement and reuse," in *2010 IEEE 9th International Conference on Development and Learning*. IEEE, 2010, pp. 7–12.
- [14] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Autonomous Robots*, vol. 31, no. 2-3, pp. 115–131, 2011.
- [15] U. Shani and D. H. Ballard, "Splines as embeddings for generalized cylinders," *Computer Vision, Graphics, and Image Processing*, vol. 27, no. 2, pp. 129–156, 1984.
- [16] E. Hartmann, "Geometry and algorithms for computer aided design," *Darmstadt University of Technology*, 2003.
- [17] M. Shanmugavel, A. Tsourdos, R. Zbikowski, and B. A. White, "3d path planning for multiple uavs using pythagorean hodograph curves," in *AIAA Guidance, Navigation, and Control Conference and Exhibit, Hilton Head, South Carolina*, 2007, pp. 20–23.
- [18] D. Hilbert and S. Cohn-Vossen, "Geometry and the imagination," *Chelsea, New York*, 1952.
- [19] E. Abbena, S. Salamon, and A. Gray, *Modern differential geometry of curves and surfaces with Mathematica*. CRC press, 2006.
- [20] C. Myers, L. Rabiner, and A. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 6, pp. 623–635, 1980.
- [21] S. R. Ahmadzadeh, R. Kaushik, and S. Chernova, "Trajectory learning from demonstration with canal surfaces: A parameter-free approach," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS International Conference on*. IEEE, November 2016, (under review).