# Generalized Cylinders for Learning, Reproduction, Generalization, and Refinement of Robot Skills

S. Reza Ahmadzadeh, Muhammad Asif Rana, Sonia Chernova
School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA
Email: {reza.ahmadzadeh, asif.rana, chernova}@gatech.edu

***Abstract*—This paper presents a novel geometric approach for learning and reproducing trajectory-based skills from human demonstrations. Our approach models a skill as a Generalized Cylinder, a geometric representation composed of an arbitrary space curve called spine and a smoothly varying cross-section. While this model has been utilized to solve other robotics problems, this is the first application of Generalized Cylinders to manipulation. The strengths of our approach are the model's ability to identify and extract the implicit characteristics of the demonstrated skill, support for reproduction of multiple trajectories that maintain those characteristics, generalization to new situations through nonrigid registration, and interactive human refinement of the resulting model through kinesthetic teaching. We validate our approach through several real-world experiments with a Jaco 6-DOF robotic arm.**

## I. Introduction

Learning from Demonstration (LfD) provides the ability to interactively teach robots new skills, eliminating the need for manual programming of the desired behavior [4]. 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 [30], highlighting the need for further advances in this area.

Existing trajectory learning representations are limited by a number of challenges. First, the number of parameters and the need to tune them for different types of trajectories present a challenge to many users, especially non-experts. Second, 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) demand a significant level of expertise to perform near-optimal demonstrations. Third, because of the assumption on near-optimal trajectories, few techniques support robust methods of refinement of the resulting model, assuming instead that the input demonstrations should be treated as the only input to the system. Finally, many existing techniques fail to generalize over start/end states, limiting the technique's ability to generalize to new situations.

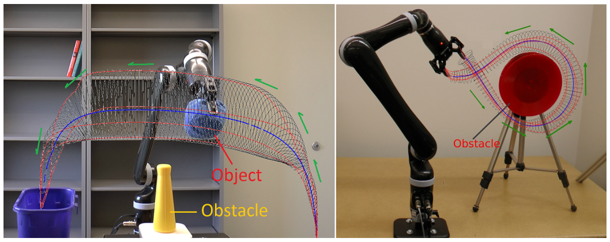


Fig. 1: Robot reproducing two trajectory-based skills encoded and learned using the proposed approach.

In this paper, we present a novel LfD approach that addresses the above challenges through the use of a geometric representation composed of a regular curve and a surface in 3D Cartesian space. The presented approach requires minimal parameter tuning and can reproduce a variety of successful movements inside the boundaries of the encoded model by exploiting the whole demonstration space. To tackle the problem of generalization over terminal states, we use a nonrigid registration method to transfer the encoded model accordingly. This generalization approach preserves the main characteristics of the demonstrated skill while achieving the goal of the task. 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 techniques, 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 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 trajectory using a simple geometric rule. We validate our approach in seven experiments using a physical 6-DOF robot, as well as demonstrate its use in comparison to Dynamic Movement Primitives [16] and Gaussian Mixture Models [9].

## II. Related Work

Existing trajectory-based LfD approaches use a variety of techniques to encode demonstrations and retrieve a generalized form of the skill [4]. One category of approaches uses regression-based techniques to generate a probabilistic representation of the given demonstrations [29],[11]. Grimes *et. al.,* employed Gaussian Process (GP) regression to learn and generalize over a set of demonstrated trajectories [11]. In follow-on work, to overcome the computational cost of GP, Schneider and Ertel used local Gaussian process regression [24]. Another approach similar to GP called LfD by Averaging Trajectories (LAT), used only one-dimensional normal distributions [23]. Both GP and LAT cannot extract constraints from the demonstrations with objects aligned parallel to a Cartesian coordinate axis, and none of the above approaches can deal with generalizing the learned skill to new situations (e.g. terminal states).

Work by Calinon *et. al.,* built a probabilistic representation of the skill using a Gaussian Mixture Model (GMM) and retrieved a smooth trajectory using Gaussian Mixture Regression(GMR) [9]. To add generalization, GMM/GMR was later extended to task-parameterized GMM [8]. The resulting method is more robust in its execution but requires extensive parameter tuning for each trajectory (e.g. number of Gaussian components, scale, weight, kernel).

An alternate approach is to use dynamic systems to encode/reproduce trajectories [16, 13]. 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 [16]. 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.

Generalization to new situations is another necessary feature for an LfD approach. DMPs can generalize by attaching an attractor to a new terminal point. However, the implicit definition of time as a canonical system makes the changes in the output (i.e. the speed of the movement) very small as time increases [15]. Our work overcomes this challenge through the use of nonrigid registration [7]. Finally, to maintain the shape of the movement during generalization, DMPs require significant tuning of continuous parameters, including those of the dynamical systems, such as time constants and scaling factors [15].

Several other techniques utilize models with characteristics similar to generalized cylinders. The real-time motion planning approach proposed by Majumdar and Tedrake approximates a boundary around a trajectory, which is visualized as a funnel [20]. 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 [10]. 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.

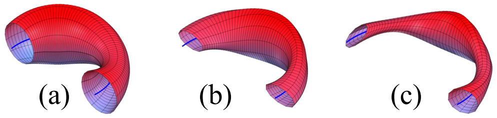


Fig. 2: Three generalized cylinders with identical directrix and different cross-section types. (a) is a canal surface.

Finally, regardless of the technique used for learning from demonstration, the capability of improving the learned model by refining its shape or spatial constraints, through an interface or kinesthetic teaching, is highly desirable. 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 [5]. 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) [18]. Our proposed approach goes beyond prior work by enabling the user to refine the skill interactively both during and after the learning process.

## 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 [26]. In other words, by moving a cross-sectional curve, defined on a plane, along the directrix while keeping the plane perpendicular to the directrix at each point, the swept volume represents a generalized cylinder. The shape and size of the cross-sectional curve can vary smoothly from point to point. 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 [12]. In robotics, generalized cylinders have been used for finding flyable paths for unmanned aerial vehicles [27]. This is the first application we are aware of to manipulation. In this section, we first outline the mathematical definition and parameterized formulation of *Canal Surfaces* (CS) [14], which are a simpler form of GCs, 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[1] of regular implicit surfaces[2] with real-valued parameter $u$. Two surfaces corresponding to different values of $u$ intersect in a common curve. As $u$ varies, the generated surface is the envelope[3] of the given pencil of surfaces [1]. The envelope can be defined using the following equations:

$$\Phi_u : F(x_1, x_2, x_3, u) = 0, \tag{1}$$

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

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

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

where the spheres are centered on a regular curve $\Gamma : \mathbf{x} = \mathbf{c}(u) \in \mathbb{R}^3$ in Cartesian space. The radiuses 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 $|\dot{r}| < \|\dot{\mathbf{c}}\|$ [12]. $\Gamma$ is the *directrix* (spine curve) and $r(u)$ is called the *radii* function. For the one-parameter pencil of **spheres**, (3) can be written as

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

### 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,

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

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 an orthonormal basis 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[4].By defining the cross-section in the TNB frame, we form a parametric representation of generalized cylinders as follows:

$$\mathcal{G}_{u,v} : f(u, v) = \mathbf{c}(u) + \rho_{x_1}(u, v)\mathbf{e}_N(u) + \rho_{x_2}(u, v)\mathbf{e}_B(u). \tag{6}$$

As mentioned earlier, canal surfaces are formed by circular cross-sections even though they can vary in size. Generalized cylinders, on the other hand, can be formed using cross-sectional curves with various shapes (e.g. a closed-spline [26]). These variations 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, reproduce, and generalize trajectory-based skills from demonstrations. We assume that multiple examples of a skill are demonstrated and captured as a set of trajectories in task-space. To capture demonstrations we use kinesthetic teaching (Fig. 3), however, alternate 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 is used for generating new trajectories starting from arbitrary initial poses. We also use nonrigid registration [7] to generalize the encoded skill over terminal constraints (i.e. initial and final poses). Algorithm 1 shows a pseudo code of our approach.

### 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 $\hat{\boldsymbol{\xi}}^j = \{\xi_1^j, \xi_2^j, \xi_3^j\} \in \mathbb{R}^{3\times T^j}$, where $T^j$ is the number of data-points within the $j^{th}$ demonstrated trajectory ($j = 1 \dots n$). Since $T^j$ can vary among 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 achieve this, a set of piecewise polynomials is obtained using cubic spline interpolation for

[1] A pencil is a family of geometric objects sharing a common property (e.g. spheres).

[2] An implicit surface is a surface in Euclidean space that can be represented in $F(x_1(u), x_2(u), x_3(u)) = 0$ form.

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

[4] Calculating Frenet-Serret frames for real data is prone to noise. The reason is that at some points the derivative vector $\frac{d\mathbf{e}_T}{du}$ 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 $\mathbf{e}_T$.

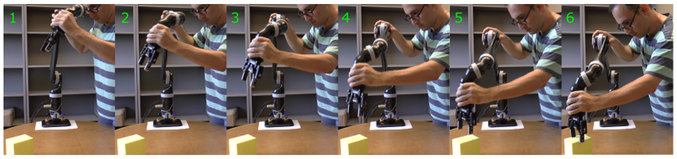
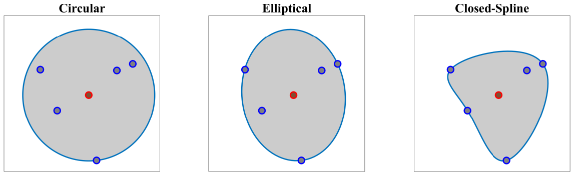


Fig. 3: Kinesthetic demonstration of reaching to an object (yellow box). Task-space pose of end-effector is used as input.


Circular

Elliptical

Closed-Spline

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.

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 $\boldsymbol{\xi} \in \mathbb{R}^{3\times N\times n}$, 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 solution is to employ Dynamic Time Warping [21].

***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 $\boldsymbol{\xi}$. 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 [9]. 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 estimated directrix, in this step, we explain methods for calculating the cross-section function $\rho(u,v)$. Fig. 4 illustrates three different types of cross-sections calculated for the same set of data. For 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* (the blue points in Fig. 4). 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. 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. As shown in Fig. 4(left), the estimated cross-section bounds other points as well and therefore the formed canal surface encloses all the demonstrations. The radii function $r(u) \in \mathbb{R}$ produces a radius for each point $u$, and $v$ parameterizes the circumference of the circular cross-section (e.g. $v = [0\ 2\pi]$). More detail on encoding skills using canal surfaces can be found in [2]. To cover the cross-sectional area more effectively and precisely while maintaining the implicit local constraints of the task, we can also construct generalized cylinders with elliptical cross-sections (see Fig. 4(middle)). The radii function for elliptical cross-section $\mathbf{r}(u) : \mathbb{R} \mapsto \mathbb{R}^3$ produces the major and minor axes and the rotation angle of the ellipse for each step $u$.

**Algorithm 1** Skill Learning using Generalized Cylinders

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

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

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

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

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

$$p(v) = a_0 + a_1(v-v_0) + a_2(v-v_0)^2 + a_3(v-v_0)^3. \qquad (7)$$

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. We denote the closed-spline as $\mathcal{P}(u,v)$. Therefore, the obtained spline is $C^2-$continuous within each interval and at each interpolating nodes. Fig. 4(right) shows a closed-spline cross-section constructed on the same set of effective points.

### 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. By drawing a ray starting from $c_0$ and passing through $p_0$, we find $g_0$, the intersection of the ray and the cross-sectional curve (See Fig. 5). The distance between the given point $p_0$ to $g_0$ reflects the similarity of the movement we want to reproduce to the nearest neighbor on the GC. We measure the ratio $\eta$ (Line 10) by calculating the distance $\|\overline{p_0 c_0}\|$ and the distance $\|\overline{g_0 c_0}\|$. 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 adjusted by applying the measured ratio as:

$$p_1 = \eta \|\overline{g_1 c_1}\| T_{\mathcal{F}_1,\mathcal{F}_0}. \tag{8}$$

Since the ratio $\eta$ is kept fixed throughout the process, thus we call this reproduction method the *ratio rule*. The ratio rule can generate new trajectories from any point inside the generalized cylinder and serves to ensure that the essential characteristics of the demonstrated skill are preserved. An illustration of a single-step reproduction process using the ratio rule can be seen in Fig. 5.

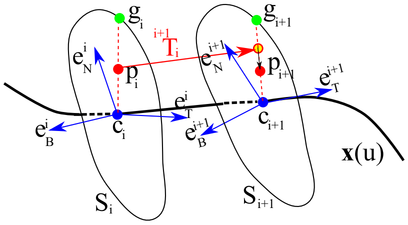
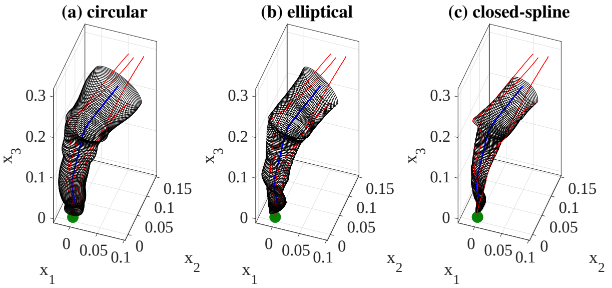


Fig. 5: Reproduction from a random initial pose $p_i$ on the $i^{\text{th}}$ cross-section $S_i$.

Finally, as mentioned earlier, the presented approach requires minimal parameter tuning in that only the shape of the cross-section needs to be defined. However, we have found the closed-spline cross-section to be most effective in encoding a wide range of trajectories, thus serving as a useful default for this single parameter.


(a) circular

(b) elliptical

(c) closed-spline

Fig. 6: A reaching skill encoded using three generalized cylinders with different cross-section types.

### C. *Generalization*

The approach described thus far enables a robot to reproduce the skill under similar conditions (i.e. start and goal states within the GC cross-section). However, to have a robust model we must ensure it generalizes to novel start and terminal states. We use a nonrigid registration technique to achieve this goal. Given a set of points in source geometry and a corresponding set of points in target geometry, nonrigid registration computes a spatial deformation function. Nonrigid registration techniques have been widely used in medical imaging [19], computer vision [6] and 3D modeling communities [22]. Recently, Schulman et al. [25] demonstrated the usefulness of nonrigid registration in LfD by employing it for autonomous knot tying. Their proposed *trajectory transfer* method is based on the classic Thin Plate Splines (TPS) [7] registration algorithm extended to 3D Cartesian space, which we also utilize here.

Consider a source geometry composed of a set of $N$ landmark points in 3D Cartesian space, $\mathcal{L} = \{L_n \in \mathbb{R}^3 | n = 1,2,...,N\}$, and a target geometry composed of the corresponding set of landmark points, $\mathcal{L}^{'} = \{L^{'}_n \in \mathbb{R}^3 | n = 1,2,...,N\}$. The nonrigid registration problem then is to find an interpolation function $\mathbf{z}: \mathbb{R}^3 \mapsto \mathbb{R}^3$, constrained to map the points in $\mathcal{L}$ to the points in $\mathcal{L}^{'}$. However, there are infinitely many such interpolation functions. To address this issue, TPS finds an interpolation function that achieves the optimal trade-off between minimizing the distance between the landmarks and minimizing the so-called *bending energy*, in effect finding a smooth interpolator. The TPS formulation is as follows,

$$\min_{\mathbf{z}} \left\{ \sum_{n=1}^{N} \|L^{'}_n - \mathbf{z}(L_n)\|^2 + \lambda \int_{\mathbb{R}^3} d\mathbf{x} \sum_{i \in \{1,2,3\}} \| \bigtriangledown^2 z_i(\mathbf{x}) \|^2_F \right\} \tag{9}$$

where, $\bigtriangledown^2 z_i$ represents the Hessian matrix of the $i^{\text{th}}$ dimension of the image of $\mathbf{z}$, $\lambda$ is a regularization parameter, and $\|.\|_F$ is the Frobenius norm. The integral term represents the bending energy.

The required function $\mathbf{z}$ which solves (9) is known to constitute of two parts: an affine part and a non-affine part. The affine part approximates the overall deformation of the geometry acting globally, while the non-affine part represents the local residual adjustments forced by individual landmark points. With the non-affine part expanded in terms of the basis function, $\mathbf{z}$ can be represented as,

$$\mathbf{z}(\mathbf{x}) = \mathbf{b} + \mathbf{A}\mathbf{x} + \sum_{n=1}^{N} \mathbf{w}_n \phi(L_n, \mathbf{x}) \tag{10}$$

where, $\mathbf{b} \in \mathbb{R}^3$, $\mathbf{A} \in \mathbb{R}^{3\times 3}$ and $\mathbf{w}_n \in \mathbb{R}^3$ are the unknown parameters while $\phi$ is the basis function defined as $\phi(L_n, \mathbf{x}) = \|L_n - \mathbf{x}\|$ for all $\mathbf{x} \in \mathbb{R}^3$. The unknown parameters in (10) can be found using matrix manipulation [7].

The generalization procedure is detailed in Algorithm 3. The source geometry is composed of the locations of the important landmarks in the workspace during the demonstration. The corresponding target geometry is composed of the new locations of the landmark points in the environment. We first find a mapping $\mathbf{z}$ of the form in (10) using the nonrigid registration method (line 4). We then use the resulting function to map the directrix $\mathbf{m}$ to $\mathbf{m}^{'}$, and the cross-sectional function $\mathcal{P}$ to $\mathcal{P}^{'}$ (lines 5 and 6). The new generalized cylinder $\mathcal{G}^{'}_{u,v}$ is then found using the mapped parameters (line 7). The ratio rule can be employed to perform reproduction in $\mathcal{G}^{'}_{u,v}$.

**Algorithm 3** Generalization of GC

```
1: procedure GeneralizeGC
2:     Input:$\mathcal{G}_{u,v}$, frames $\mathcal{F}_u$, source & target landmarks $\mathcal{L}$, $\mathcal{L}^{'}$
3:     Output:target $\mathcal{G}^{'}_{u,v}$, target frames $\mathcal{F}^{'}_u$
4:     $\mathbf{z} \leftarrow findTPS(\mathcal{L}, \mathcal{L}^{'})$      $Using Eq.$ 9
5:     $\mathbf{m}^{'}(u) \leftarrow \mathbf{z}\big(\mathbf{m}(u)\big)$
6:     $\mathcal{P}^{'}(u,v) \leftarrow \mathbf{z}\big(\mathcal{P}(u,v)\big)$
7:     $\mathcal{G}^{'}_{u,v}, \mathcal{F}^{'} \leftarrow makeGeneralizedCylinder(\mathbf{m}^{'}(u), \mathcal{P}^{'}(u,v))$
```

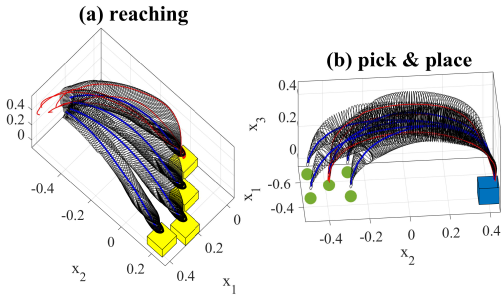


Fig. 7: Generalization of generalized cylinders over initial and final poses using nonrigid registration.

## V. Experimental Results

We conducted eight experiments to demonstrate the GC model, as well as its reproduction and generalization capabilities, on multiple trajectory-based skills. For each experiment, we gathered a set of demonstrations through kinesthetic teaching using a 6-DOF Kinova Jaco2 robotic arm (Fig. 3). The data was recorded at 100Hz.

### A. Learning and Reproduction

In this section, we present examples of five trajectory-based skills encoded using the generalized cylinder model. In the first experiment, we performed a reaching skill towards an object (green sphere) from above (Fig. 6)[5]. We present circular, elliptical and closed-spline cross-sections to showcase how GCs with different cross-section types encode the boundaries of the movement.

The demonstrations recorded for the second experiment (Fig. 8(a)) show an example of a movement that can be started and ended in a wide task-space but in the middle 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. The obtained GC extracts and preserves the important characteristics of the demonstrated skill, i.e. the precision and shape of the trajectory throughout the movement. Given an arbitrary initial pose of the end-effector, the robot successfully reproduces the learned skill using the ratio rule.

The third experiment (Fig. 8(b)) shows a reaching/placing skill similar to the first experiment with a curved trajectory. The robot learns to exploit a wider space while reaching the object while maintaining trajectories with precision near the object. The accompanying video shows the execution of this task [28].

The fourth experiment (Fig. 8(c)) shows a circular movement around an obstacle, which is unknown to the robot. Since the given demonstrations avoid the obstacle, and the encoded GC guarantees that all the reproductions of the task remain inside the cylinder, the reproduced path is guaranteed to be collision-free. Fig. 1 (right) shows a snapshot during the reproduction of the skill.

The fifth task represents a pick-and-place movement, in which the robot picks up an object and places it in a box (Fig. 8(d)). The encoded GC shows that the initial and final poses of the movement are the main constraints of the tasks while in the middle of the trajectory, the end-effector can pass through a wider space while preserving the shape of the movement. Fig. 1 (left) shows a snapshot during the reproduction of this skill.

### B. Generalization

In this section, we demonstrate the generalization capability of the proposed approach using data from the third and fifth experiments (Fig. 7). After encoding the skill in the third experiment, we relocated the target three times and each time used the nonrigid registration approach to adapt the encoded model to the new situation as explained in Section IV-C. In the fifth experiment, we relocated the initial pose of the object four times and used nonrigid registration to adapt the encoded model to the new situations. As noticeable in both generalization experiments, the overall shape of the generalized cylinder is preserved while accordingly expanding or contracting for different initial and final poses. As a direct consequence of the ratio rule for reproduction, this successfully enables shape preservation of the reproduced trajectories. Nonrigid registration can also be applied in the case that the initial and target points both differ.

### C. Comparison to DMP and GMM/GMR

In this section, we compare the presented approach to two widely-used LfD techniques, DMPs and GMM/GMR. Fig. 9 shows a comparison of both results on the data from experiment four. For the GC, we encoded the demonstrations using a generalized cylinder with a closed-spline

[5] In all of the figures, the demonstrations, directrix, and reproductions are plotted in red, blue, and magenta, respectively.

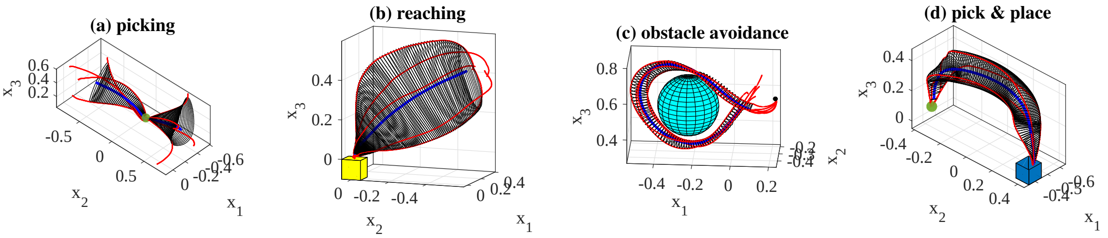


Fig. 8: Four real-world experiments performed using our approach. The demonstrations (red) are used to extract the main characteristics of the skills. The obtained directrix (blue) and generalized cylinders (gray) represent the skills.

cross-section and generated five reproductions from various initial poses (Fig. 9(b)). As mentioned before, our approach requires no parameter tuning beyond specifying the cross-section type, and by extracting the characteristics of the movement it learns to avoid the obstacle. For GMM/GMR we tuned the number of Gaussian components (5 and 10, in figures (c) and (d), respectively), and for DMPs, we tuned the number of attractors (5 and 10, in figures (e) and (f), respectively). As can be seen in the results, both DMPs and GMM/GMR can learn the skill. In (c), and to a lesser degree in (d), GMM/GMR produces a more angular trajectory than seen in the demonstrations. In (e) we see the DMP deviating from the demonstrations and colliding with the object. In all four GMM/GMR and DMP examples, we also observe that the five different start locations all converge to a single trajectory path. In contrast, the trajectories reproduced by the generalized cylinder produce a more natural set of motions that are not identical and exploit the whole cylinder while maintaining the important characteristics of the skill. It has to be noted that the GC can reproduce analogous trajectories by using the directrix directly if that behavior is desirable.

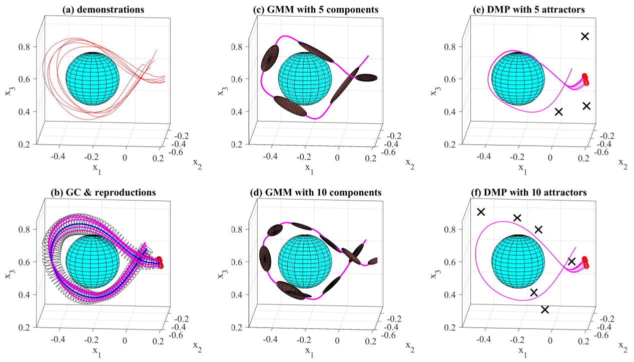


Fig. 9: Results of the comparison between GMM, DMP, and GC. (a) five demonstrations, (b) GC and five reproductions, (c) GMM with five components, (d) GMM with ten components, (e) DMP with five attractors, and (f) DMP with ten attractors.

## VI. Skill Refinement

The above examples show that generalized cylinders can be used to extract, generalize and reproduce skills from reliable human demonstrations. In practice, however, user-provided demonstrations of a task are usually sub-optimal. Multiple solutions to this problem have been proposed. Argall et al. showed that behavior can be corrected by having the teacher assign weights to each demonstration based on its quality [3, 5]. It has also been shown that integrating LfD with Reinforcement Learning can be used for refinement [17]. However, assigning weights to demonstrations is not trivial and exploring the state-action to find an improved solution requires extensive trial and error. An alternate approach is to refine the skill through physical human-robot interaction [5]. In this work, we differentiate two types of refinement. *Incremental refinement* occurs during the learning process, in which the robot replays a demonstrated trajectory (or a reproduction), the user can modify it in some way, and then the model is retrained on the updated data. Once a model is learned, *constraint-based refinement* can be used to refine the model further by applying new constraints. In this section, we show that both approaches can be applied to generalized cylinders. Note that, we have selected simple movements for analysis and illustrative purpose.

### A. Incremental Refinement

In its first form, skill refinement can be performed during the learning process incrementally. After gathering the set of demonstrations and encoding the skill using generalized cylinders, the user identifies a *target trajectory* (either a demonstration or a reproduction) that needs to be modified. The user executes the target trajectory with the robot in compliant mode, allowing the joints and the end-effector position to be adjusted while the robot is moving. While the robot is replaying the trajectory, the teacher can reshape the target trajectory through kinesthetic correction. The obtained trajectory is added to the set and replaces the initial demonstration. Given the new set, the algorithm updates the model and reproduces new trajectories that inherit the applied corrections.

The following experiment illustrates this approach on GCs. Initially, we demonstrated three simple trajectories, encoded the skill as a GC with spline cross-section (Fig. 10(a)), and reproduced the skill using the ratio rule (Fig. 10(b)). Now, we assume we would like the arm of the robot to dip downwards in the middle of the first (top) demonstration. 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

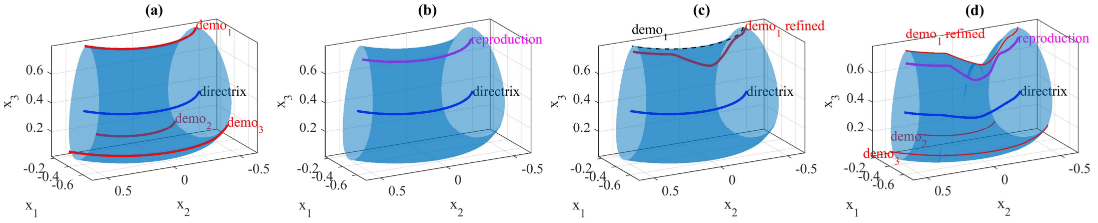


Fig. 10: Incremental refinement of a skill by correcting a demonstration. (a) demonstrations (red), directrix (blue) and the obtained GC, (b) reproduction from a random pose (magenta), (c) first demonstration was refined by the teacher (red), (d) updated GC, directrix, and a new reproduction.

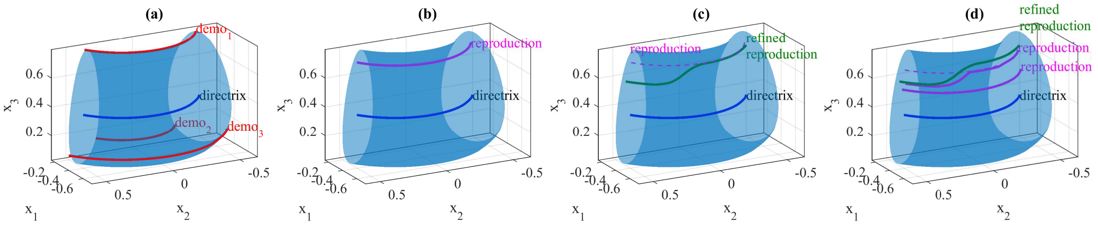


Fig. 11: Constraint-based refinement of a skill by correcting a reproduction. (a) demonstrations, (b) reproduction (c) reproduction refined by the teacher (green), (d) two new reproductions; the upper one is affected by the refinement, while the lower is not.

after replacing the target with the refined demonstration. Given an arbitrary initial pose, the algorithm reproduces a new trajectory that reflects the performed refinements. This experiment shows that our approach can deal with the refinements incrementally. Although many approaches can benefit from a similar process [5], our representation is visually perceivable and enables even non-experts to observe and interpret the effect of the refinements on the model. The accompanying video shows the execution of this task [28].

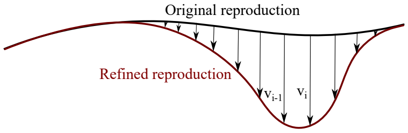


Fig. 12: Formation of the refinement matrix from the original and modified reproductions.

### B. Constraint-based Refinement

In this section, we show that skill refinement can also be performed after the model has been encoded by applying new constraints. During the compliant reproduction phase, the user observes and kinesthetically modifies the reproduced trajectory. When this occurs, we compare the original and the modified reproductions, calculate point-to-point translation vectors, $\mathbf{v}_i$, and form a refinement matrix, $\hat{V}$ by concatenating the vectors. Fig. 12 depicts the formation of the refinement matrix from the original and the modified trajectories. The refinement matrix acts as a geometric constraint on the GC that would affect future reproductions.

Fig. 11 presents an example of this technique. The green trajectory in Fig. 11(c) shows how the original reproduction is refined by the teacher through kinesthetic correction; the teacher has applied downward forces ($-x_3$ direction) to keep the end-effector at a certain level. We calculate the refinement matrix $\hat{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n] \in \mathbb{R}^{3\times n}$ 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. 11(d). On the other hand, if a reproduction intersects with the constraining plane, the refinement matrix applies to it. The upper reproduction in Fig. 11(d) shows the effect of the constraint while the dashed line shows reproduction without applying the constraint. This experiment indicates that using constraint-based refinement, the user can apply new constraints to the model without modifying it. One of the advantages of this approach is that the imposed constraint later can be removed or combined with other constraints without updating the encoded model. To our knowledge, there is no other LfD approach with similar capabilities.

## VII. Conclusion

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 minimal parameter tuning that not only simplifies the usage of the algorithm and makes the result consistent but also can make the approach more convenient for non-expert users. We also have shown that our approach enables users to refine a learned skill both through incremental and constraint-based refinement interactively.

## Acknowledgments

This work is supported in part by the Office of Naval Research Award N000141410795.

# References

[1] Elsa Abbena, Simon Salamon, and Alfred Gray. *Modern differential geometry of curves and surfaces with Mathematica*. CRC press, 2006.

[2] S. Reza Ahmadzadeh, Roshni Kaushik, and Sonia Chernova. Trajectory learning from demonstration with canal surfaces: A parameter-free approach. In *Humanoid Robots (Humanoids), IEEE-RAS International Conference on*, pages 544–549. IEEE, 2016.

[3] Brenna D Argall, Brett Browning, and Manuela Veloso. Automatic weight learning for multiple data sources when learning from demonstration. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 226–231. IEEE, 2009.

[4] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57 (5):469–483, 2009.

[5] Brenna D Argall, Eric L Sauser, and Aude G Billard. Tactile guidance for policy refinement and reuse. In *Development and Learning (ICDL), IEEE 9th International Conference on*, pages 7–12. IEEE, 2010.

[6] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

[7] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989.

[8] Sylvain Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 9(1):1–29, 2016.

[9] Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2): 286–298, 2007.

[10] Shuonan Dong and Brian Williams. Learning and recognition of hybrid manipulation motions in variable environments using probabilistic flow tubes. *International Journal of Social Robotics*, 4(4):357–368, 2012.

[11] David B. Grimes, Rawichote Chalodhorn, and Rajesh PN Rao. Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In *Robotics: Science and Systems (RSS)*, pages 199–206, 2006.

[12] Erich Hartmann. Geometry and algorithms for computer aided design. *Darmstadt University of Technology*, 2003.

[13] Micha Hersch, Florent Guenter, Sylvain Calinon, and Aude G Billard. Learning dynamical system modulation for constrained reaching tasks. In *Humanoid Robots (Humanoids), 6th IEEE-RAS International Conference on*, pages 444–449. IEEE, 2006.

[14] David Hilbert and Stephan Cohn-Vossen. *Geometry and the Imagination*, volume 87. American Mathematical Society, 1952.

[15] Auke J. Ijspeert, Jun Nakanishi, and Stefan Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 1398–1403. IEEE, 2002.

[16] Auke J. Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

[17] Petar Kormushev, Sylvain Calinon, and Darwin G. Caldwell. Robot motor skill coordination with EM-based reinforcement learning. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 3232–3237. IEEE, 2010.

[18] Dongheui Lee and Christian Ott. Incremental kinesthetic teaching of motion primitives using the motion refinement tube. *Autonomous Robots*, 31(2-3):115–131, 2011.

[19] JB Antoine Maintz and Max A Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[20] Anirudha Majumdar and Russ Tedrake. Funnel libraries for real-time robust feedback motion planning. *arXiv:1601.04037*, 2016.

[21] Cory Myers, Lawrence Rabiner, and Aaron Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(6):623–635, 1980.

[22] Mark Pauly, Niloy J Mitra, Joachim Giesen, Markus H Gross, and Leonidas J Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing*, pages 23–32, 2005.

[23] Benjamin Reiner, Wolfgang Ertel, Heiko Posenauer, and Markus Schneider. Lat: A simple learning from demonstration method. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 4436–4441. IEEE, 2014.

[24] Markus Schneider and Wolfgang Ertel. Robot learning by demonstration with local gaussian process regression. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 255–260. IEEE, 2010.

[25] John Schulman, Jonathan Ho, Cameron Lee, and Pieter Abbeel. Learning from demonstrations through the use of non-rigid registration. In *Robotics Research*, pages 339–354. Springer, 2016.

[26] Uri Shani and Dana H Ballard. Splines as embeddings for generalized cylinders. *Computer Vision, Graphics, and Image Processing*, 27(2):129–156, 1984.

[27] Madhavan Shanmugavel, Antonios Tsourdos, Rafał Zbikowski, and Brian A White. 3D path planning for

multiple UAVs using pythagorean hodograph curves. In *AIAA Guidance, Navigation, and Control Conference*, pages 20–23, 2007.

[28] Video. video accompanying this paper, available online. https://youtu.be/0ANvgLkqdaM, 2017.

[29] Sethu Vijayakumar, Aaron D'souza, and Stefan Schaal. Incremental online learning in high dimensions. *Neural computation*, 17(12):2602–2634, 2005.

[30] Holly Yanco, Adam Norton, Willard Ober, David Shane, Anna Skinner, and Jack Vice. Analysis of human-robot interaction at the DARPA robotics challenge trials. *Journal of Field Robotics*, 32(3):420–444, 2015.