

Movement Primitives via Optimization

Anca D. Dragan, Katharina Muelling, J. Andrew Bagnell, and Siddhartha S. Srinivasa
The Robotics Institute, Carnegie Mellon University

Abstract—We formalize the problem of adapting a demonstrated trajectory to a new start and goal configuration as an optimization problem over a Hilbert space of trajectories: minimize the distance between the demonstration and the new trajectory subject to the new end point constraints. We show that the commonly used version of Dynamic Movement Primitives (DMPs) implement this minimization in the way they adapt demonstrations, for a particular choice of the Hilbert space norm. The generalization to arbitrary norms enables the robot to select a more appropriate norm for the task, as well as learn how to adapt the demonstration from the user. Our experiments show that this can significantly improve the robot’s ability to accurately generalize the demonstration.

I. INTRODUCTION

We focus on the problem of learning motor skills from demonstration, in which a user demonstrates a trajectory to a robot (like the gray trajectory in Fig.1) (for instance, through kinesthetic demonstration), and the robot adapts it to new conditions that it faces, such as a new start or goal configuration (the blue trajectory in Fig.1). This problem is important in learning motion skills from demonstration [1], as well as in learning from experience using trajectory libraries [2, 3].

Among several tools for addressing this problem [1, 4–6], a commonly used one is a Dynamic Movement Primitive (DMP) [7, 8]. DMPs have seen wide application across a variety of domains, including biped locomotion [9], grasping [10], placing and pouring [11], dart throwing [12], ball paddling [13], pancake flipping [14], playing pool [15], and handing over an object [16].

DMPs represent a demonstration as a dynamical system *tracking a moving target configuration*, and adapt it to new start and goal constraints by simply changing the start and goal parameters in the equation of the moving target. The adaptation process is the same, regardless of the task and of the user, and is merely one instance of a larger problem.

Our work introduces a generalization of this adaptation process. We provide a variational characterization of the problem by formalizing the adaptation of a demonstrated trajectory to new endpoints as an optimization over a Hilbert space of trajectories (Sec. II). We find the *closest* trajectory to the demonstration, in the linear subspace induced by the new endpoint constraints (Fig.2). Distance (the notion of “closer”) is measured by the norm induced by the inner product in the space.

Using this formalism, different choices for the inner product lead to different adaptation processes. We prove that DMPs implement this optimization in the way they adapt trajectories, for a *particular* choice of a norm (Sec. III). We do so by proving that when updating the endpoints, the moving target tracked by the dynamical system adapts (as in Fig.1(b)) using the very same norm that we often use in trajectory optimizers like CHOMP [17] (we denote this norm by A). We then

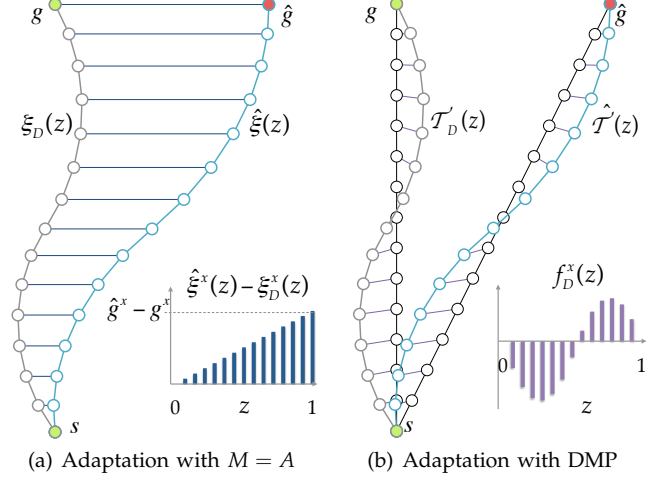


Fig. 1: (a) Using a norm M for adaptation propagates the change in the start and goal, from $\{s, g\}$ to $\{\hat{s}, \hat{g}\}$, to the rest of the trajectory, changing ξ_D into $\hat{\xi}$. The difference between the two as a function of time is plotted in blue. (b) In contrast, DMPs represent the demonstration as a spring damper system tracking a moving target trajectory \mathcal{T}_D , compute differences f_D (purple) between \mathcal{T}_D and the straight line trajectory, and apply the same differences to the new straight line trajectory between the new endpoints. This results in a new target trajectory $\hat{\mathcal{T}}$ for the dynamical system to track. When $M = A$, the velocity norm typically used in CHOMP [17], the two adaptations are equivalent. In general, different norms M would lead to different adaptations.

show that this also implies that the adaptation in the trajectory space, obtained by then tracking the adapted target, is also the result of optimizing a norm based on A .

Beyond providing a deeper understanding of DMPs and what criteria they are inherently optimizing when adapting demonstrations, our generalization frees the robot from a fixed adaptation process by enabling it to use *any* inner product (or norm). Because computing the minimum norm adaptation is near-instant, any such adaptation process can be used in the DMP to obtain the new moving target trajectory.

Thus, we can select a more appropriate norm based on the task at hand (Sec. IV-A). What is more, if the user is willing to provide a few examples of how to adapt the trajectory as well, then the robot can *learn* the desired norm (Sec. IV-B): the robot can learn, from the user, not only the trajectory, but also *how to adapt the trajectory* to new situations.

We conduct an experimental analysis of the benefit of learning a norm both with synthetic data where we have ground truth, as well as with kinesthetic demonstrations on a robot arm. Our results show a significant improvement in how well the norm that the robot learns is able to reconstruct a holdout set of demonstrations, compared to the default DMP norm.

Overall, we contribute a deeper theoretical understanding of DMPs that relates them to trajectory opti-

mization, and also leads to practical benefits for learning from demonstration that help broaden the use of DMP-like algorithms.

II. HILBERT NORM MINIMIZATION

In this section, we formalize trajectory adaptation as a Hilbert norm minimization problem. We then derive the solution to this problem, and study the case in which translating trajectories carries no penalty. This is the case for the norm stemming from a common trajectory optimization objective, as well as for the norm DMPs use in their adaptation process.

A. Problem Statement

Trajectories are functions $\xi : [0, 1] \rightarrow \mathcal{Q}$ mapping time to robot configurations. We allow the time index to be either discrete or continuous. Given a demonstrated trajectory ξ_D , we propose to adapt it to a new start \hat{s} (the robot's starting configuration) and a new goal \hat{g} by solving:

$$\begin{aligned} \hat{\xi} &= \arg \min_{\xi \in \Xi} \|\xi_D - \xi\|_M^2 \\ \text{s.t. } \xi(0) &= \hat{s} \\ \xi(1) &= \hat{g} \end{aligned} \quad (1)$$

where M is the norm defined by the inner product in the Hilbert space of trajectories, $\|\xi\|_M^2 = \langle \xi, \xi \rangle$. Fig.2 illustrates this problem. Different inner products lead to different M s, which in turn lead to different adaptations.

B. Solution

In general, M induces a linear operator. When time is discrete, M is a matrix, and $\|\xi\|_M^2 = \xi^T M \xi$.

The Lagrangian of Eq.(1) is

$$\mathcal{L} = (\xi_D - \xi)^T M (\xi_D - \xi) + \lambda^T (\xi(0) - \hat{s}) + \gamma^T (\xi(1) - \hat{g}) \quad (2)$$

Taking the gradient w.r.t. ξ , λ , and γ :

$$\nabla_{\xi} \mathcal{L} = M(\xi_D - \xi) + (\lambda, 0, \dots, 0)^T + (0, \dots, 0, \gamma)^T \quad (3)$$

$$\nabla_{\lambda} \mathcal{L} = \xi(0) - \hat{s}, \quad \nabla_{\gamma} \mathcal{L} = \xi(1) - \hat{g} \quad (4)$$

Thus, the solution is:

$$\hat{\xi} = \xi_D + M^{-1}(\lambda, 0, \dots, 0, \gamma)^T \quad (5)$$

where the vectors λ and γ are set by Eq.(4).

This has an intuitive interpretation: *correct the start and the goal, and propagate the differences across the trajectory* in a manner dictated by the norm M (Fig.1(a)). Fig.2 depicts the geometry of the space.

C. Free Translations

Often times, we are interested in being able to translate trajectories at no cost, i.e. if $\hat{\xi} = \xi + \xi_k$, with $\xi_k(t) = k, \forall t$ (a constant valued trajectory), then $\|\hat{\xi} - \xi\|_M = 0, \forall k$. However, that makes M a *semi-norm*, as $\langle \xi_k, \xi_k \rangle = 0, \forall k$, which makes the problem ill posed.

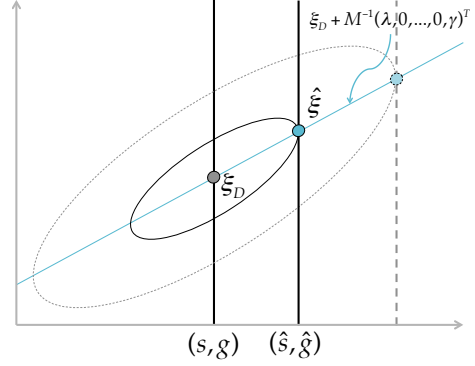


Fig. 2: We adapt ξ_D by finding the closest trajectory to it that satisfies the new end point constraints. The x axis is the start-goal tuple, and the y axis is the rest of the trajectory. M warps the space, transforming (hyper)spheres into (hyper)ellipsoids. The space of all adaptations of ξ_D is a linear subspace of Ξ .

1) *Why Free Translations:* One such example that is of wide applicability is the one stemming from the integral over squared velocities along the trajectory, a criteria often used in trajectory optimization [17–19]. Let

$$C[\xi] = \frac{1}{2} \int \dot{\xi}(z)^T \dot{\xi}(z) dz = \frac{1}{2} \xi^T A \xi \quad (6)$$

with $A = K^T K$, K being the finite differencing matrix. Then $M = A$ is such a semi-norm, as every constant trajectory ξ_k has norm 0:

$$\xi_k^T A \xi_k = 2C[\xi_k] = 2 \int 0 dz = 0, \forall k$$

The CHOMP trajectory optimizer [17] is often implemented using this velocity norm to measure distances between trajectories.

In the next section, we show that this norm A is the norm that DMPs minimize in the way they adapt the trajectory being tracked by the spring damper system to new start and goal configurations.

2) *Handling Free Translations:* CHOMP bypasses the semi-norm problem because the trajectory endpoints are constants. Similarly, the key to free translations while maintaining a full norm is fixing one of the endpoints, e.g. the starting configuration: one can adapt the trajectory's goal in a restricted space of trajectories that all have the same (constant) start, and then translate the result to the new starting configuration.

Let $\Xi_{s=k}$ be the subspace of trajectories s.t. the starting configuration is a constant k : $\xi(0) = k, \xi \in \Xi_{s=k} \subset \Xi$. M is a full norm in $\Xi_{s=k}$, as no translations are allowed.

Let $\sigma_k : \Xi_{s=k} \rightarrow \Xi_{s=0}, \sigma_k(\xi) = \xi - \xi_k$ be the function that translates trajectories from $\Xi_{s=k}$ to start at $s = 0$. This function is bijective, $\sigma_k^{-1}(\xi) = \xi + \xi_k$.

We can reformulate Eq.(1) to finding the closest trajectory within $\Xi_{s=0}$ that ends at $\hat{g} - \hat{s}$, and translating this trajectory to the new start \hat{s} , thereby obtaining a trajectory from \hat{s} to $\hat{g} - \hat{s} + \hat{s} = \hat{g}$:

$$\begin{aligned} \hat{\xi} &= \sigma_s^{-1} \left(\arg \min_{\xi \in \Xi_{s=0}} \|\sigma_s(\xi_D) - \xi\|_M^2 \right) \\ \text{s.t. } \xi(1) &= \hat{g} - \hat{s} \end{aligned} \quad (7)$$

The solution to this, following an analogous derivation to Sec. II-B, is to take the demonstration translated to 0, correct the goal to $\hat{g} - \hat{s}$, propagate this change to the rest of the trajectory via M , and then translate the result to the new start:

$$\hat{\xi} = \sigma_s^{-1} \left(\sigma_s(\xi_D) + M^{-1}(0, \dots, 0, \gamma)^T \right) \quad (8)$$

with γ s.t. $\xi(1) = \hat{g}$. For a norm M with no coupling between joints, and m the last entry in M , this becomes:

$$\hat{\xi} = \sigma_s^{-1} \left(\sigma_s(\xi_D) + \frac{1}{m} M^{-1}(0, \dots, 0, (\hat{g} - \hat{s}) - (g - s))^T \right) \quad (9)$$

This corrects the goal in $\Xi_{s=0}$ from $g - s$ to $\hat{g} - \hat{s}$, effectively changing the goal in Ξ from g to \hat{g} .¹

III. DMP ADAPTATION AS A SPECIAL CASE OF HILBERT NORM MINIMIZATION

In this section, we summarize a commonly used version of DMPs, and write it as a target tracker with a moving target. Next, we show that the adaptation of the tracked target to a new start and goal is an instance of Hilbert norm minimization (Theorem 1). Finally, we show that this induces an adaption in trajectory space that is an instance of norm minimization (Theorem 2).

A. DMPs

A commonly used version [10, 11, 15, 16] of a DMP is a second order linear dynamical system which is stimulated with a non-linear forcing term:

$$\tau^2 \ddot{\xi}(t) = K(g - \xi(t)) - D\tau \dot{\xi}(t) - K(g - s)u + K\bar{f}(u) \quad (10)$$

where $K(g - \xi(t))$ is an attractor towards the goal, $K(g - s)u$ avoids jumps at the beginning of the movement, $D\dot{\xi}(t)$ is a damper, and $K\bar{f}(u)$ is a nonlinear forcing term. u is a phase variable generated by the dynamical system

$$\tau \dot{u} = -\alpha u$$

Thus, u maps time from 1 to (almost) 0:

$$u(t) = e^{-\frac{\alpha}{\tau} t} \quad (11)$$

B. DMP Adaptation as Tracked Target Adaptation

Let $z = 1 - u$. We can reformulate a DMP as a target tracker with a moving target, $\mathcal{T}(z)$:

$$\tau^2 \ddot{\xi}(t) = K(\mathcal{T}(z) - \xi(t)) - D\tau \dot{\xi}(t) \quad (12)$$

with $\mathcal{T}(z)$ moving from s to g as a function of z on a straight line constant speed in z plus a deviation f as a function of z , $f(z) = \bar{f}(u)$:

$$\mathcal{T}(z) = s + z(g - s) + f(z) \quad (13)$$

¹Note that here we are overloading M . In Eq.(7), we are measuring norms in a space of trajectories with constant start 0, which is a lower dimensional space of trajectories $\tilde{\xi} : (0, 1] \rightarrow \mathcal{Q}$ that do not contain the starting configuration (which is not a variable). In this space, we can define a norm \tilde{M} by $\|\tilde{\xi}\|_{\tilde{M}} = \|\xi\|_M$, with $\xi(0) = 0$ and $\xi(z) = \tilde{\xi}(z) \forall z \in (0, 1]$. \tilde{M} is then of dimensionality one less than M and full rank, and what we actually use in Eq.(9).

Given a demonstration ξ_D , one forms a DMP by computing $f_D(z)$ from Eq.(12).² To generalize to a new \hat{s} and \hat{g} , the target changes from Eq.(14) to Eq.(15):

$$\mathcal{T}_D(z) = s + z(g - s) + f_D(z) \quad (14)$$

$$\hat{\mathcal{T}}(z) = \hat{s} + z(\hat{g} - \hat{s}) + f_D(z) \quad (15)$$

The linear function from s to g is adapted to the new endpoints, becoming $\hat{s} + z(\hat{g} - \hat{s})$ (black trajectories in Fig.1(b)), and the deviation f_D remains fixed (purple deviations in Fig.1(b)).

C. Relation to Hilbert Norm Minimization

The adaptation of the target being tracked by the DMP, from \mathcal{T}_D to $\hat{\mathcal{T}}$, is a special case of the Hilbert norm adaptation from ξ_D to $\hat{\xi}$, when the norm $M = A$ from Eq.(6).

To prove this, we show the equivalence between the DMP adapted trajectory $\hat{\mathcal{T}}$ and the outcome of the Hilbert norm minimization $\hat{\xi}$ from Eq.(9), for $\mathcal{T} = \xi$.

We do this in two steps. Since $\hat{\mathcal{T}}$ is the sum of a straight line trajectory (as a function of z) and a fixed deviation, we first show that the Eq.(7) will adapt a straight line trajectory to another straight line when M is the norm A . Next we show that when adding a nonzero deviation to the initial trajectory, the same deviation is added by Eq.(7) to the adapted trajectory.

Therefore, we first focus on the case when $f_D = 0$. In this case, the targets are straight lines from the start to the goal, moving at constant speed: $\mathcal{T}(z) = \xi_{\text{straight}}(z) = (g - s)z + s$, and $\hat{\mathcal{T}}(z) = \hat{\xi}_{\text{straight}}(z) = (\hat{g} - \hat{s})z + \hat{s}$.

In Lemma 3, we show that the adaptation of ξ_{straight} to a new start \hat{s} and a new goal \hat{g} with respect to the norm A matches $\hat{\xi}_{\text{straight}}$. We build to this via two other lemmas, where the key is to represent straight lines in terms of the norm A . We first prove that ξ_{straight} minimizes $\xi^T A \xi$ (Lemma 1). This enables us to write out ξ_{straight} in terms of A (Lemma 2).

We then generalize this to non-zero f_D using that f_D is not actually changed by the norm M in Eq.(9).

Lemma 1: ξ_{straight} is the solution to minimizing Eq.(6): constant speed straight line trajectories have minimum norm under A .

Proof: We show this by showing that the solution to Eq.(6) is a straight line with constant velocity, just like ξ_{straight} . The gradient of C is

$$\nabla_{\xi} C = -\ddot{\xi}$$

and setting this to 0 results in $\xi = az + b$. $\xi(0) = s \Rightarrow b = s$, and $\xi(1) = g \Rightarrow a = g - s$. Thus, $\xi = (g - s)z + s = \xi_{\text{straight}}$. ■

Lemma 2: $\sigma_s(\xi_{\text{straight}}) = \frac{1}{m} A^{-1}(0, \dots, 0, g)^T$ with m the last entry of A as in Eq.(9): we can write constant speed straight line trajectories in closed form in terms of A .

²Typically, there is a smoothing step before adaptation where \tilde{f}_D is fitted by some basis functions, $\tilde{f}_D(u) = \frac{\sum \psi_i(u) \theta_i u}{\sum \psi_i(u)}$. The same smoothing can be applied to a trajectory before performing Hilbert norm minimization.

Proof: From Lemma 1 and from $C[\tilde{\zeta}] = \tilde{\zeta}^T A \tilde{\zeta}$, we infer that $\sigma_s(\tilde{\zeta}_{straight})$, which is the straight line from 0 to $g - s$, is the solution to

$$\begin{aligned} \min_{\tilde{\zeta} \in \Xi_{s=0}} \tilde{\zeta}^T A \tilde{\zeta} \\ \text{s.t. } \tilde{\zeta}(1) = g - s \end{aligned} \quad (16)$$

Writing the Lagrangian and taking the gradient like before, we get that $\sigma_s(\tilde{\zeta}_{straight}) = \frac{1}{m} A^{-1}(0, \dots, 0, g - s)^T$: this term is the straight line from 0 to $g - s$. ■

Lemma 3: $\hat{\zeta}_{straight}$ is the solution to Eq.(7) for $\zeta_D = \tilde{\zeta}_{straight}$: constant speed straight lines get adapted by A to constant speed straight lines.

Proof: From Lemma 2, the term $\frac{1}{m} M^{-1}(0, \dots, 0, (\hat{g} - \hat{s}) - g)^T$ from Eq.(9) is the straight line from 0 to $(\hat{g} - \hat{s}) - (g - s)$, i.e. $((\hat{g} - \hat{s}) - g)t$.

Thus, Eq.(9) becomes

$$\begin{aligned} \hat{\zeta} &= \sigma_s^{-1}(\sigma_s(\tilde{\zeta}_{straight}) + \\ &\quad + \frac{1}{m} M^{-1}(0, \dots, 0, (\hat{g} - \hat{s}) - (g - s))^T) \Rightarrow \\ \hat{\zeta} &= \sigma_s^{-1}((g - s)z + ((\hat{g} - \hat{s}) - (g - s))z) \Rightarrow \\ \hat{\zeta} &= \sigma_s^{-1}((\hat{g} - \hat{s})z) \Rightarrow \\ \hat{\zeta} &= (\hat{g} - \hat{s})z + \hat{s} \Rightarrow \\ \hat{\zeta} &= \hat{\zeta}_{straight} \end{aligned}$$

Theorem 1: $\hat{\mathcal{T}}$ is the solution to Eq.(7) for $\zeta_D = \mathcal{T}$: straight lines plus deviations get adapted by A to straight lines plus the same deviations, like the target trajectories in DMPs.

Proof: When $f_D = 0$, $\mathcal{T} = \tilde{\zeta}_{straight}$, and $\hat{\mathcal{T}} = \hat{\zeta}_{straight}$. The theorem follows from Lemma 3.

When $f_D \neq 0$, the demonstrated target is $\mathcal{T}_D = \tilde{\zeta}_{straight} + f_D$, and the adapted target is $\hat{\mathcal{T}} = \hat{\zeta}_{straight} + f_D$. This adapted target still matches the solution in Eq.(9):

$$\begin{aligned} \hat{\zeta} &= \sigma_s^{-1}(\sigma_s(\tilde{\zeta}_{straight} + f_D) + \\ &\quad + \frac{1}{m} M^{-1}(0, \dots, 0, (\hat{g} - \hat{s}) - (g - s))^T) \Rightarrow \\ \hat{\zeta} &= \sigma_s^{-1}(\sigma_s(\tilde{\zeta}_{straight}) + f_D + \\ &\quad + \frac{1}{m} M^{-1}(0, \dots, 0, (\hat{g} - \hat{s}) - (g - s))^T) \Rightarrow \\ \hat{\zeta} &= \sigma_s^{-1}(f_D + (\hat{g} - \hat{s})z) \Rightarrow \\ \hat{\zeta} &= f_D + (\hat{g} - \hat{s})z + \hat{s} \Rightarrow \\ \hat{\zeta} &= \hat{\mathcal{T}} \end{aligned}$$

Therefore, the target adaptation that the DMP does, from \mathcal{T} to $\hat{\mathcal{T}}$, is none other than the Hilbert norm minimization from Eq.(1), with the same norm as the one often used in trajectory optimization algorithms like CHOMP.

Norm Minimization Directly in the Trajectory Space. Because the tracked target adaption from \mathcal{T} to $\hat{\mathcal{T}}$ is

a Hilbert norm minimization, then the corresponding adaptation in the space of trajectories, which adapts ζ_D into $\hat{\zeta}$ by tracking $\hat{\mathcal{T}}$, is also the result of a Hilbert norm minimization.

To see this, let $\beta : \zeta \mapsto \mathcal{T}$ be the function mapping a demonstrated trajectory to the corresponding tracked target like in Eq.(12). Given a particular spring damper system, β is a bijection: every demonstrated trajectory maps to a unique tracked target, and every tracked target maps to a unique trajectory when tracked by the spring damper. Furthermore, β is linear, due to Eq.(12) and additivity and homogeneity of differentiation.

Because β is bijective and linear, the norm A in the tracked target spaces induces a norm P in the trajectory space: $\|\tilde{\zeta}\|_P = \|\beta(\tilde{\zeta})\|_A$.

Theorem 2: The final trajectory obtained by tracking the adapted target $\hat{\mathcal{T}}$, $\hat{\zeta} = \beta^{-1}(\hat{\mathcal{T}})$, is the closest trajectory to ζ_D that satisfies the new endpoint constraints with respect to the norm P : the final trajectory in a DMP is the result of Eq.(1) for $M = \hat{P}$.

Proof: Assume $\exists \tilde{\zeta}$ with endpoints \hat{s} and \hat{g} s.t. $\|\zeta_D - \tilde{\zeta}\|_P < \|\zeta_D - \hat{\zeta}\|_P$, i.e. $\tilde{\zeta}$ is closer to ζ_D than $\hat{\zeta}$ is. Then $\|\beta(\zeta_D - \tilde{\zeta})\|_A < \|\beta(\zeta_D - \hat{\zeta})\|_A \Rightarrow \|\beta(\zeta_D) - \beta(\tilde{\zeta})\|_A < \|\beta(\zeta_D) - \beta(\hat{\zeta})\|_A \Rightarrow \|\mathcal{T}_D - \beta(\tilde{\zeta})\|_A < \|\mathcal{T}_D - \hat{\mathcal{T}}\|_A$, which contradicts Theorem 1: we know that $\hat{\mathcal{T}}$ is the closest to \mathcal{T}_D w.r.t. the norm A given the endpoint constraints, thus $\beta(\tilde{\zeta})$ cannot be closer. ■

Therefore, DMPs adapt trajectories by minimizing a norm that depends on both A (the norm used to adapt the tracked target), as well as the particulars of the dynamical system (represented here by the function β).

IV. IMPLICATIONS

Theoretical Implications. Our work connects DMPs to trajectory optimization, providing an understanding of what objective the DMP adaptation process is inherently optimizing.

Our work also opens the door for handling obstacle avoidance via planning. Currently with DMPs, obstacles that appear as part of new situations influence the adapted trajectory in a reactive manner, akin to a potential field. Certain more difficult situations, however, require using a motion planner for successful obstacle avoidance, which reasons about the entire trajectory and not just the current configuration. Using our generalization, a trajectory optimizer akin to CHOMP can search for a trajectory that minimizes the adaptation norm (as opposed to the trajectory norm, as in CHOMP) while avoiding collisions.

Practical Implications. First, the generalization frees us from the default A norm, and enables us to select more appropriate norms for each task. We discuss this benefit in Sec. IV-A.

Second, the generalization gives the robot the opportunity to *learn* how to adapt trajectories from the user. If the user is willing to provide not only a demonstration, but also a few adaptations of that demonstration to different start and goal configurations, then the robot can use this set of trajectories to learn the desired norm M . We describe an algorithm for doing so in Sec. IV-B.

Aside 1 — Computation. The adaptation in a DMP happens instantly, by instantiating the start and goal variables with new values. Hilbert norm minimization

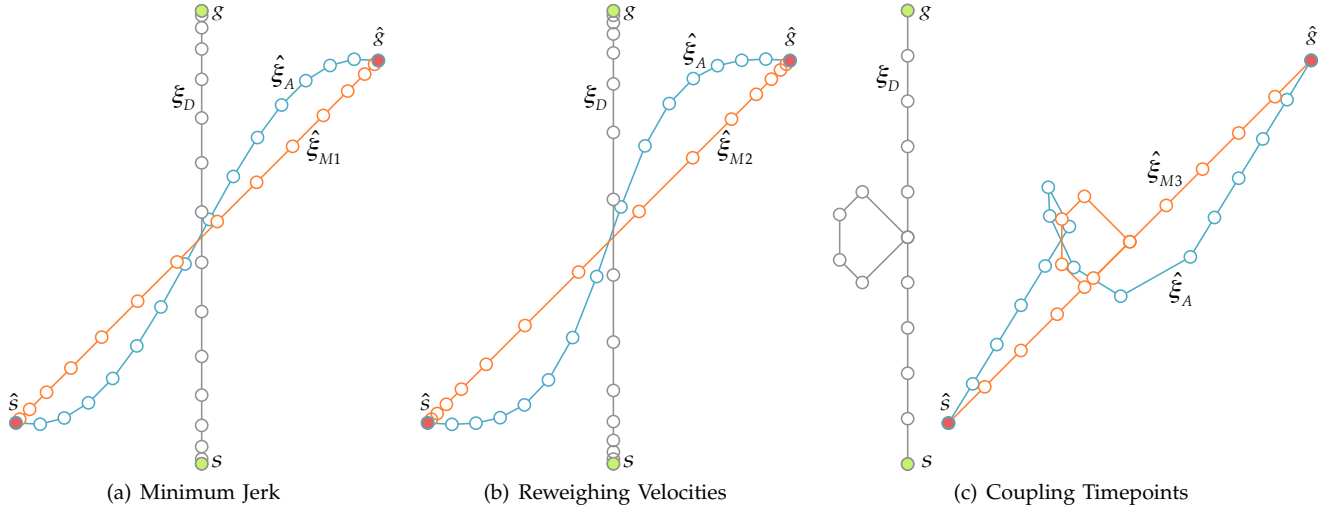


Fig. 3: A is not always an appropriate choice for the Hilbert norm. Each of the plots compares adapting a different original trajectory ξ_D (gray) using A (ξ_A , blue) vs. using a better norm (ξ_M , orange). The norms we used, much like A , do not allow free rotations, but free rotations could be obtained similarly to free translations (Sec. II-C).

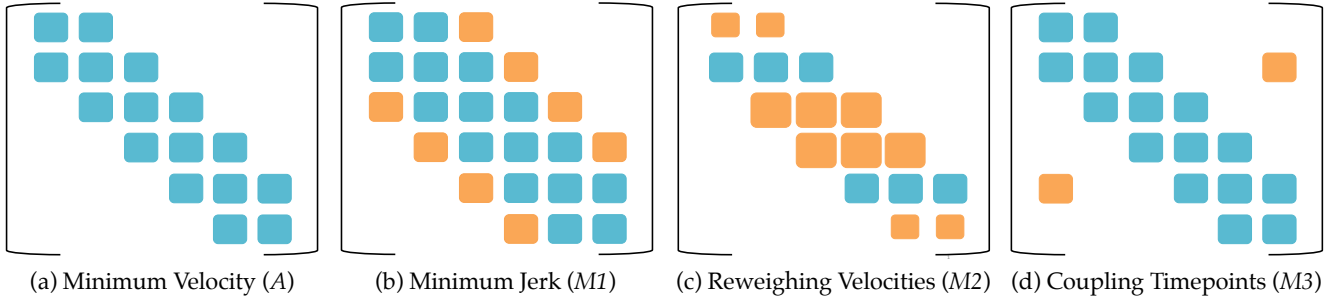


Fig. 4: The different changes to the norm structure result in different adaptation effects.

has an analytical solution, with computational complexity in the discrete case dominated by a single matrix multiplication. This means any DMP can adapt its moving target using norm minimization.

Aside 2 — Using a Spring Damper. DMPs first cast the trajectory as a moving target tracked by a spring damper, and adapt the moving target trajectory. Hilbert norm minimization can be used to adapt trajectories both for the moving target, as well as for the demonstrated trajectory itself. The decision to use a spring damper is independent from the adaptation process.

A. Selecting a Better Norm

The norm A can lead to good adaptations (see Fig. 1), but it is not always the most suitable norm. Fig. 3 shows three cases where a different norm leads to better adaptations. In all three cases, the better norm is a modification of the matrix structure of A (as shown in Fig. 4).

The first case, Fig. 3(a), uses a demonstrated trajectory that minimizes jerk. Therefore, using a norm that stems from jerk as opposed to velocities, results in the correct adaptation — the minimum jerk trajectory (orange). This norm is band diagonal, like A , but has a wider band because computing the jerk requires terms further away from the current trajectory point than computing velocities (Fig. 4(b)).

The second case, Fig. 3(b), uses a demonstrated trajectory that moves faster in the middle than it does in

the beginning and end. Therefore, a norm that weighs velocities in middle of the trajectory less than velocities at the endpoints (unlike A , for which the velocities at every time point matter equally), results in the adaptation in orange: the trajectory remains a straight line, and follows a similar velocity profile as the demonstration. This norm is a reweighing of the rows of A (Fig. 4(c)).

The third case, Fig. 3(c), uses a loop as the demonstrated trajectory. The demonstration itself is not necessarily minimizing any L^2 norm. However, a more appropriate norm for adapting this demonstration couples waypoints that are distant in time but close in space: instead of only minimizing velocities, it also minimizes the distance between the two points that begin and end the loop. Unlike A , which is band diagonal, this norm also has entries far from the diagonal, depending on how far apart in time these two waypoints are (Fig. 4(d)).

B. Learning a Better Norm

As we saw in the previous section, different norms result in different ways of adapting a demonstrated trajectory. If the user providing the demonstration is willing to also provide example adaptations to new endpoints, then the robot can learn the norm M from these examples: *instead of adapting trajectories in a pre-defined way, the robot can learn from the user how it should adapt trajectories.*

Let $\mathcal{D} = \{\xi_i\}$ be the set of user demonstrations, each of them corresponding to a different tuple of endpoints $(\xi_i(0), \xi_i(1))$. The robot needs to find a norm M such that for each pair of trajectories $(\xi_i, \xi_j) \in \mathcal{D} \times \mathcal{D}$, ξ_j is the closest trajectory to ξ_i out of all trajectories between the new endpoints, $\xi_j(0)$ and $\xi_j(1)$, i.e. find a norm that explains why the user adapted ξ_i into ξ_j and *not* into any other trajectory:

$$\|\xi_i - \xi_j\|_M \leq \|\xi_i - \xi\|_M, \quad \forall \xi \in \Xi_{s=\xi_j(0)}^{g=\xi_j(1)} \quad (17)$$

Equivalently:

$$\begin{aligned} \|\xi_i - \xi_j\|_M^2 &\leq \min_{\xi \in \Xi} \|\xi_i - \xi\|_M^2 \\ \text{s.t. } \xi(0) &= \xi_j(0) \\ \xi(1) &= \xi_j(1) \end{aligned} \quad (18)$$

One way to find an M under these constraints is to follow Maximum Margin Planning [20]. We find M by minimizing the following expression:

$$\begin{aligned} \min_M \sum_{i,j} \|\xi_i - \xi_j\|_M^2 - \min_{\xi \in \Xi} [\|\xi_i - \xi\|_M^2 - \mathcal{L}(\xi, \xi_j)] \\ \text{s.t. } \xi(0) &= \xi_j(0) \\ \xi(1) &= \xi_j(1) \\ \text{s.t. } M &\succ 0 \end{aligned} \quad (19)$$

with \mathcal{L} a loss function, e.g. a function evaluating to 0 when the trajectory matches ξ_j and to 1 otherwise, and $M \succ 0$ the positive-definiteness constraint.

If ξ_{ij}^* is the optimal solution to the inner minimization problem, then the gradient update is:

$$M = M - \alpha \sum_{i,j} [(\xi_i - \xi_j)(\xi_i - \xi_j)^T - (\xi_i - \xi_{ij}^*)(\xi_i - \xi_{ij}^*)^T] \quad (20)$$

followed by a projection onto the space of positive definite matrices.

Aside 3 — Geometry. An M that satisfies all the constraints only exists if the demonstrations in \mathcal{D} lie in a linear subspace of Ξ of dimensionality $2d$, with d the number of degrees of freedom: the adaptation induces a foliation of the space, with each linear subspace of a demonstration and all its adaptations to new endpoints forming a plaque of the foliation. Fig.2 depicts such a linear subspace, obtained by adapting $\xi_{\mathcal{D}}$.

This follows from Eq.(5): the space of all adaptations of a trajectory is parametrized by the vectors λ and γ . Similarly, when we allow free translations, the linear subspace has dimensionality d (Eq.(8)). Note that there are many norms that satisfy the constraints in this case, because only a subset of the rows of M^{-1} are used in the adaptation.

When the demonstrations do not form such a linear subspace, the algorithm will find an approximate M that minimizes the criterion in Eq.(19). We study the effects of noise in the next section. Other techniques for finding an approximate M , such as least squares or PCA, would also apply, but they would minimize different criteria, e.g. the difference between the trajectories themselves ($\sum \|\xi_j - \xi_{ij}^*\|^2$), and not the difference between the norms.

V. EXPERIMENTAL ANALYSIS

We divide our experiments in two parts. The first experiment (Sec. V-A) analyzes the ability to learn a norm from only a few demonstrations, under different noise conditions. We do this on synthetically generated data so that we can manipulate the noise and compare the results to ground truth. We assume an underlying norm, generate noisy demonstrations based on it, and test the learner's ability to recover the norm. The second experiment tests the benefit of learning the norm with real kinesthetic demonstrations on a robot arm (Sec. V-B).

A. Synthetic Data

To analyze the dependency of learning the norm on the number of demonstrations, we generate demonstrations for different endpoints using a given norm M and some arbitrary initial trajectory. We then use the training data to learn a norm \tilde{M} . For simplicity, we focus on norms that allow free translations, and that do not couple different joints (similar to A).

1) *Dependent Measures:* We test the quality of a learned norm \tilde{M} using two measures (which significantly correlate, see Analysis): one is about the norm itself, and the other is about the effect it has on adaptations.

Waypoint Error: This measure captures deviations of the behavior induced by the learned norm from desired behavior. We generate a test set of 1200 new start and goal configuration tuples for testing, leading to 1200 adapted trajectories using M as ground truth. We then adapt the demonstrated trajectory to each tuple using the learned norm \tilde{M} . For each obtained trajectory, we measure the mean waypoint deviation from the ground truth trajectories, and combine these into an average across the entire set.

Norm Error: This measure captures deviations in the learned norm itself (between M and \tilde{M}). Because only the last row of M^{-1} (which we denote M_N^{-1}) affects the resulting adaptation, we compute the norm of the component of the normalized \tilde{M}_N^{-1} that is orthogonal to the true normalized M_N^{-1} .

2) *Ideal Demonstrations:* We first test learning from ideal demonstrations, meaning perfectly adapted using M , without any noise.

Because of the structure that M imposes on the optimal adaptations (a linear subspace of dimensionality $2d$ in general, d for free translations), only a few ideal demonstrations are necessary to perfectly retrieve M : 3 in the general case, and 2 in the case of free translations.

As a sanity check, we ran an experiment in which we chose the starting trajectory from Fig.1 and generated 100 random norms. For each norm, we computed the two measures above. The resulting error was exactly 0 in each case: the learning algorithm perfectly retrieved the underlying norm.

3) *Tolerance to Noise:* Real demonstrations will not be perfect adaptations – they will be noisy. With noise comes the necessity for more than the minimal number

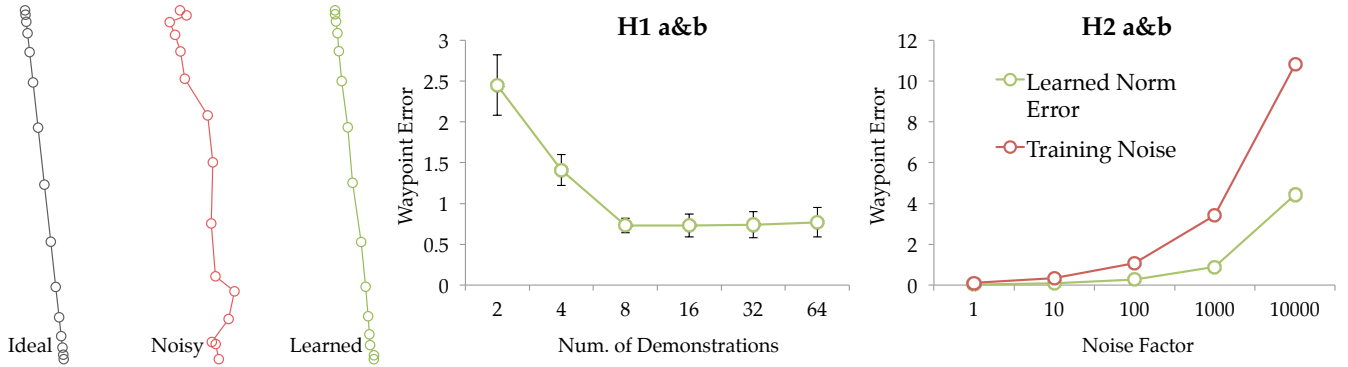


Fig. 5: Left: an ideal adapted trajectory (gray), a noisy adapted trajectory (red) that we use for training, and the reproduction using the learned norm (green), with a 6-fold average reduction in noise. Center: the error on a test set as a function of the number of training examples. Right: the error on a test set as a function of the amount of noise, compared to the magnitude of the noise (red). Error bars show standard error on the mean – when not visible, the error bars are smaller than the marker size.

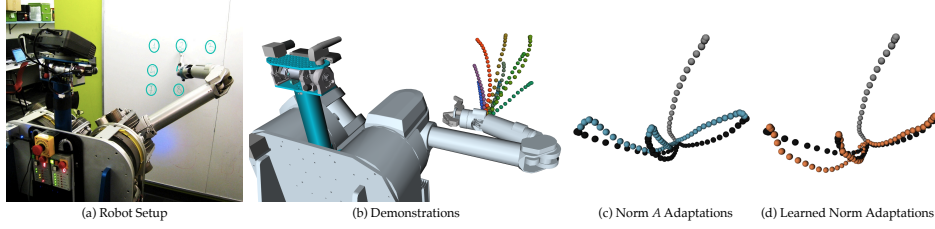


Fig. 6: A comparison between adapting trajectories with the default A metric (c) and adapting using a learned metric (d) on a holdout set of demonstrated pointing gestures (shown in black). The trajectory ζ_D used for adaptation is in gray. Note that the adaption happens in the full configuration space of the robot, but here we plot the end effector traces for visualization. The learned norm more closely reproduces two of the trajectories, and has higher error in the third. Overall, the error decreases significantly (see Fig. 7).

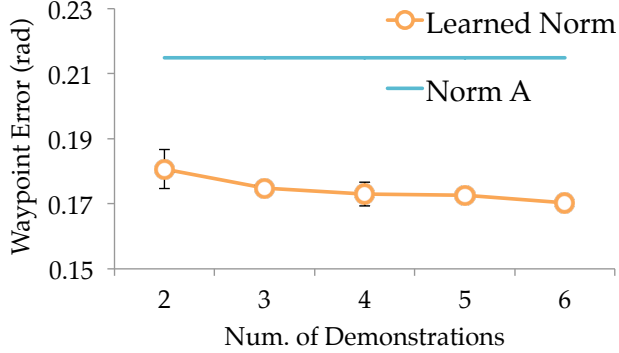


Fig. 7: The average waypoint error on a holdout set of pointing gesture demonstrations on the HERB robot, for the adaptations obtained using the learned norm, compared to error when using the default A .

of demonstrations, and the questions of how many demonstrations are needed and how robust the learning is to the amount of noise.

Manipulated Variables. In this experiment, we study these questions by manipulating two factors: (1) the **number of demonstrations**, and (2) the **amount of noise** we add to the adaptations in the training data.

We added Gaussian noise to the ideal adaptations using a covariance matrix that adds more noise to the middle of the trajectory than the endpoints (since the endpoints are fixed when requesting an adaptation).

For the first factor – number of demonstrations – we started at 2 (the minimum number required), and chose exponentially increasing levels (2, 4, 8, 16, 32, 64) to get an idea for what the scale of the number of demonstration should be. For the second factor, we scaled the covariance matrix (by 1, 10, 100, 1000, 10000)

up to the point where the average noise for a trajectory waypoint was 50% of the average distance from start to goal (which we considered an extreme amount that exceeds by far levels we expect to see in practice). This resulted in 30 total conditions, and we ran the experiment with 30 different random seeds for each condition.

Hypotheses:

H1a. (Sanity Check) *The number of demonstration positively affects the learned norm quality.*

H1b. (We Only Need a Small Number of Examples) *There is a point beyond which increasing the number of examples results in practically equivalent norm quality.*

H2a. (Sanity Check) *The amount of noise negatively affects norm quality.*

H2b. (Learning is Tolerant to Noise) *The waypoint error is significantly lower than the noise on the training examples.*

Analysis. The waypoint error and norm error measures were indeed significantly correlated (standardized Cronbach's $\alpha = 0.95$), suggesting that the waypoint error also captures the deviation from the real norm.

A factorial least squares regression revealed that, in line with H1a and H2a, both factors were significant: as the number of demonstrations increased, the error did decrease ($F(1, 867) = 24.07$, $p < .0001$), and as the amount of noise increased, the error did increase ($F(1, 867) = 628.35$, $p < .0001$).

Fig. 5 plots these two effects. In support of H1b, the error stops decreasing after 8 demonstrations (it takes a difference threshold of 0.3 for an equivalence test between the error at 8 and the error at 16 to reject the hypothesis that they are practically the same with $p = .04$). This suggests that learning the norm can

happen from relatively few demonstrations.

In support of H2b, the error was significantly lower than the noise in the training trajectories ($t(899) = 19.35$, $p < .0001$): on average, the error was lower by a factor of 6.71, and this factor increased significantly with the number of demonstrations ($F(1, 869) = 869.01$, $p < .0001$).

B. Real Data

Our simulation study compared the learned norm to ground truth. Next, we were interested in studying the benefits of learning the norm with real kinesthetic demonstrations on a robot arm.

We collected 9 expert demonstrations of pointing gestures on the HERB robot [21], where the task was to point to a particular location on a board, as in Fig.6(a). We chose pointing as a task because the shape of the adapted trajectories is important for such gestures. We used up to 6 of these trajectories for training, and held out 3 for testing.

Dependent Measures. We use the waypoint error measure from before, this time from the noisy holdout set as opposed to ground truth. We cannot use the norm error since we no longer have access to the true norm M .

Manipulated Variables. We used both the learned norm, as well as the default A norm from Eq.(6), to generate adaptations of the same original demonstration (its end effector trace is shown in gray in Fig.6(c and d)). Note that even though the learned norm has access to more than the original demonstration, we used this demonstration only when testing the adaptation, to remain fair to the default norm. In practice, if the user provides multiple demonstrations, the one corresponding the situation closest to the test situation could be used for adaptation.

We also manipulated how many of the 6 demonstrations the learning algorithm used.

Hypotheses:

H3. (Data Improves Performance) *As before, we expect that the number of demonstrations positively affects performance of the learned norm, i.e. error in reproducing the holdout trajectories decreases as the number of demonstrations increases.*

H4. (Learned Norm > Default A) *The learned norm has smaller error in reproducing the holdout demonstrations than the default A norm.*

Analysis. Fig.6 qualitatively compares the learned and the default norm, and Fig.7 plots our results.

Overall, the performance did tend to improve with the number of demonstrations, but the effect was not significant ($F(4, 26) = 1.31$, $p = .29$). In support of H4, the error was significantly lower overall when learning the norm than when using the DMP default ($t(30) = 31.96$, $p < .0001$), suggesting that for real kinesthetic demonstrations, there is indeed a practical benefit to the generalization we propose in this paper.

VI. DISCUSSION AND FUTURE WORK

In this paper, we formalized the trajectory adaptation problem as Hilbert norm minimization and showed this is a generalization of the process DMPs use to adapt demonstrated trajectories to new endpoints. Our work

can be used for tasks in which DMPs are typically employed, in particular those tasks for which the shape of the trajectory is important (such as gestures): by learning the adaptation norm from the user, the robot can produce trajectories that better match the desired shape in new situations (as in our experiments).

We also envision an alternate use, geared towards skills for which success is the main driver (such as closing a lid or hitting a tennis ball): rather than learning the norm from the user, the robot can learn a good norm through reinforcement, targeting an increase in the success rate compared to the default norm.

Further avenues of future work include testing the norm learning on data from novice users. This would require addressing the problem of enabling novices to provide good demonstrations, which is a separate line of research and the reason we opted for expert demonstrations in this paper.

Finally, we plan on investigating skills beyond pointing gestures. In particular, since minimum L^2 norm adaptations lie in a low dimensional linear subspace of the trajectory space, we aim to explore what types of skills best meet this assumption.

REFERENCES

- [1] 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.
- [2] Nikolay Jetchev and Marc Toussaint. Trajectory prediction: learning to map situations to robot trajectories. In *ICML*, 2009.
- [3] Martin Stolle and Christopher G Atkeson. Policies based on trajectory libraries. In *ICRA*, 2006.
- [4] Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2):286–298, 2007.
- [5] Gu Ye and Ron Alterovitz. Demonstration-guided motion planning. In *ISRR*, 2011.
- [6] John Schulman, Jonathan Ho, Cameron Lee, and Pieter Abbeel. Learning from demonstrations through the use of non-rigid registration. In *ISRR*, 2013.
- [7] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. Learning attractor landscapes for learning motor primitives. In *NIPS*, 2003.
- [8] Auke Jan 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.
- [9] Jun Nakanishi, Jun Morimoto, Gen Endo, Gordon Cheng, Stefan Schaal, and Mitsuo Kawato. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47(2):79–91, 2004.
- [10] Peter Pastor, Ludovic Righetti, Mrinal Kalakrishnan, and Stefan Schaal. Online movement adaptation based on previous sensor experiences. In *IROS*, 2011.
- [11] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *ICRA*, 2009.
- [12] Jens Kober, Erhan Oztop, and Jan Peters. Reinforcement learning to adjust robot movements to new situations. In *IJCAI*, 2011.
- [13] Jens Kober and Jan Peters. Learning motor primitives for robotics. In *ICRA*, 2009.
- [14] Peter Kormushev, Sylvain Calinon, and Darwin G Caldwell. Robot motor skill coordination with em-based reinforcement learning. In *IROS*, 2010.
- [15] Peter Pastor, Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, and Stefan Schaal. Skill learning and task outcome prediction for manipulation. In *ICRA*, 2011.
- [16] Miguel Prada, Anthony Remazeilles, Ansgar Koene, and Satoshi Endo. Dynamic movement primitives for human-robot interaction: comparison with human behavioral observation. In *IROS*, 2013.
- [17] Matt Zucker, Nathan Ratliff, Anca D Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher M Dellin, J Andrew Bagnell, and Siddhartha S Srinivasa. Chomp: Covariant hamiltonian optimization for motion planning. *IJRR*, 32(9-10):1164–1193, 2013.
- [18] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. Stomp: Stochastic trajectory optimization for motion planning. In *ICRA*, 2011.
- [19] Chonhyon Park, Jia Pan, and Dinesh Manocha. Itomp: Incremental trajectory optimization for real-time replanning in dynamic environments. In *ICAPS*, 2012.
- [20] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *ICML*, 2006.
- [21] Siddhartha S Srinivasa, Dmitry Berenson, Maya Cakmak, Alvaro Collet, Mehmet Remzi Dogar, Anca D Dragan, Ross A Knepper, Tim Niemueller, Kyle Strabala, Mike Vande Weghe, et al. Herb 2.0: Lessons learned from developing a mobile manipulator for the home. *Proceedings of the IEEE*, 100(8), 2012.