Acquiring and Generalizing the Embodiment Mapping from Human Observations to Robot Skills

Guilherme Maeda[1], Marco Ewerton[2], Dorothea Koert[3], and Jan Peters[1,2]version final RAL version

***Abstract*—Robot imitation based on observations of the human movement is a challenging problem as the structure of the human demonstrator and the robot learner are usually different. A movement that can be demonstrated well by a human may not be kinematically feasible for robot reproduction. A common approach to solve this kinematic mapping is to retarget pre-defined corresponding parts of the human and the robot kinematic structure. When such a correspondence is not available, manual scaling of the movement amplitude and the positioning of the demonstration in relation to the reference frame of the robot may be required. This paper's contribution is a method that eliminates both the need of human-robot structural associations—and therefore is less sensitive to the type of robot kinematics—and searches for the optimal location and adaptation of the human demonstration, such that the robot can accurately execute the optimized solution. The method defines a cost that quantifies the quality of the kinematic mapping and decreases it in conjunction with task-specific costs such as via-points and obstacles. We demonstrate the method experimentally where a real golf swing recorded via marker tracking is generalized to different speeds on the embodiment of a 7 degree-of-freedom (DoF) arm. In simulation, we compare solutions of robots with different kinematic structures.**

***Index Terms*—Learning and Adaptive Systems, Kinematics, Optimization and Optimal Control, Motion and Path Planning, Human Factors and Human-in-the-Loop**

I. Introduction

MAPPING human movements to executable robot trajectories is an essential part of imitation learning and programming by demonstration. Teleoperation and kinesthetic teaching have been widely used to avoid the correspondence problem in imitation learning. Despite being popular, these methods are not ideal for a variety of reasons. As illustrated in Fig. 1(b), during kinesthetic teaching, the human must decide how to move the several joints of a redundant manipulator when demonstrating a golf swing. This procedure not only leads to a solution biased by the demonstrator, but even if a champion player would be in the laboratory available to demonstrate such a task, the demonstration is compromised as the robot is kinematically different in size and has a limited number of degrees-of-freedom (DoFs). Also, among the very few back-drivable robots available, some are too fragile to accept aggressive demonstrations, while others are too bulky to be handled naturally (see Fig. 1(b)).

(a) Natural demonstration (b) Obtrusive robot presence (c) Embodiment variations

Fig. 1. (a) Learning from human observations requires the solution of a non-trivial correspondence mapping. (b) Methods such as kinesthetic teaching avoid the correspondence problem, but the physical presence of the robot hinders the movements of truly natural demonstrations. (c) In this paper different embodiments are used to execute the same human demonstration.

A. Learning Kinematic Skills from Human Observations

Reproducing human observations is unarguably the most natural and non-disruptive form of acquiring robot skills. A common procedure to map upper limb movements from a human to a robot relies on manually defining the positioning and scaling of the recorded human demonstration into the robot's workspace, often with the help of heuristics such as pose similarity; or by using some form of retargeting [1]. The solution to this problem is not trivial under dissimilar or arbitrary robot structures or when the structure of the human arm is not fully observed. In practice, to facilitate the correspondence solution, the human demonstrator may also need to adapt his/her movement to take into consideration the robot's kinematic constraints.

To automate this process, one could initially optimize the transformation between the reference frame of demonstrator and the robot, aiming at a feasible inverse kinematics solution (e.g. using a normalized pose [2] or the T-pose [3]). As a second step, one could then consider optimizing the demonstration at the location found in the first step, as a trajectory optimization problem; to avoid obstacles in the robot's workspace and to scale the human trajectory to fit the kinematics of the robot. These two steps should not be solved independently, however, as modifications on the trajectory in the latter step

Manuscript received: August, 31, 2015; Revised December, 08, 2015; Accepted January, 11, 2016.

This paper was recommended for publication by Antonio Bicchi upon evaluation of the Associate Editor and Reviewers' comments.

[1]Guilherme Maeda, Marco Ewerton, Dorothea Koert, and Jan Peters are with the Intelligent Autonomous Systems Lab, Technische Universitaet Darmstadt, Darmstadt, Germany {maeda, ewerton, koert, peters}@ias.tu-darmstadt.de

[2]Jan Peters is with the Max Planck Institute for Intelligent Systems, Tuebingen, Germany jan.peters@tuebingen.mpg.de

Digital Object Identifier (DOI): see top of this page.

could allow for subsequent improvements in the reference frame transformation of the first step. Another problem is that of defining a metric that quantifies the suitability of the human demonstration in relation to the robot structure.

The contribution of this paper is an off-line algorithm for trajectory generation for robotic arms based on human observations. Given the kinematic structure of the robot arm, the method searches for the optimal combination of reference frame transformation and shape adaptation of a human demonstration. The basic idea is to use the residual error that is output by an inverse kinematics (IK) algorithm to quantify the "mapping error". This error arises when a demonstrated trajectory cannot be suitably reproduced by a robot learner, due to differences in the kinematic structures and/or in the relative location between the robot and the demonstrated trajectory. Since robot-specific information is only applied by the IK function, the method is general and can optimally adapt demonstrations to match different robot embodiments, such as the lightweight 7-DoF KUKA LWR4, an industrial 6-DoF ABB IRB140, and a fictitious 4-DoF SCARA arm shown in Fig. 1(c).

B. Problem Statement and Proposed Optimization

Assume a Cartesian trajectory of T time steps provided by a human demonstrator as $\boldsymbol{X}^H_{1:T}$. The trajectory is recorded w.r.t. to the reference frame of the demonstration $\{D\}$. A robot arm in an arbitrary location (e.g. in another laboratory), defined by the reference frame $\{W\}$, is to reproduce the shape of the demonstration w.r.t. its own reference frame as close as possible. The transformation ${}^W_D\boldsymbol{T}$ that relates the coordinates from $\{D\}$ to $\{W\}$ is unknown. Also, the robot may be limited in DoFs and reach such that the original demonstration may not be exactly reproducible, regardless of the choice of ${}^W_D\boldsymbol{T}$.

This paper proposes solving the mapping from demonstration to robot joint trajectories $M\colon \boldsymbol{X}^H_{1:T} \rightarrow \boldsymbol{q}_{1:T}$.The goal is to find a trajectory $\boldsymbol{X}_{1:T}$ similar to $\boldsymbol{X}^H_{1:T}$, and a transformation ${}^W_D\boldsymbol{T}$ that expresses $\boldsymbol{X}_{1:T}$ in $\{\mathrm{W}\}$ such that ${}^W_D\boldsymbol{T}\boldsymbol{X}_{1:T}$ can be accurately reproduced by the robot. In other words, the joint trajectories of the IK solution $\boldsymbol{q}_{1:T} = IK({}^W_D\boldsymbol{T}(\boldsymbol{\theta}_L)\boldsymbol{X}_{1:T}(\boldsymbol{\theta}_S))$ lead to the exact reproduction of the Cartesian input ${}^W_D\boldsymbol{T}(\boldsymbol{\theta}_L)\boldsymbol{X}_{1:T}(\boldsymbol{\theta}_S)$.

In its simplest form, we propose solving the optimization

$$\min_{\boldsymbol{\theta}=\{\boldsymbol{\theta}_L,\boldsymbol{\theta}_S\}} \sum_{t=1}^{T} \|{}^W_D\boldsymbol{T}(\boldsymbol{\theta}_L)\boldsymbol{X}_t(\boldsymbol{\theta}_S) - FK(\boldsymbol{q}_t)\| + \sum_{t=1}^{T} \|\boldsymbol{X}^H_t - \boldsymbol{X}_t(\boldsymbol{\theta}_S)\| \quad (1)$$

$$\text{s.t.} \quad \boldsymbol{q}_{1:T} \in [\boldsymbol{q}_{min}, \boldsymbol{q}_{max}],$$

where $[\boldsymbol{q}_{min}, \boldsymbol{q}_{max}]$ are the limits for each of the joints of the robot, and $FK(\cdot)$ is a forward kinematics function. The first term quantifies, at each time step, the error between the adapted trajectory in the reference frame of the robot and the trajectory that the robot end-effector describes when reproducing $\boldsymbol{q}_t$. As it will be shown in more detail in Section III-B, this error can be obtained as the output of a numerical IK solver. The second term penalizes for dissimilarities between the original trajectory and the current trajectory in the reference frame of the demonstration. The policy parameters $\boldsymbol{\theta}$ account for shape and location with $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_L$, respectively. This simultaneous optimization contrasts with conventional trajectory optimization and motion planning methods where only $\boldsymbol{\theta}_S$ is taken into account, usually in the joint space of the robot learner.

II. Related Work: From Human Observations to Robot Skills

The problem of kinematically mapping trajectories to different embodiments is well known to the computer graphics and animation community where a particular approach, referred to as motion retargeting [1], has found applications in robotics. The retargeting approach consists in matching skeletons with different link lengths but approximately similar kinematic structures. Much of the focus in retargeting has been proposed to deal with stylistic and human-like mappings both in animation [4] and robotics [5]. Retargeting with real robots often needs to address dynamics and mechanical limitations [6]. More recently, retargeting-based approaches have also been proposed to program industrial robots [7].

Since the knowledge of the kinematic structure of the demonstrator is a basic requirement, retargeting methods require a planned setup with well positioned cameras and extensive use of markers for tracking. The recent use of off-the-shelf depth cameras for retargeting (e.g. [8], [9]) requires the whole human body to be facing the camera, free from occlusion. In both cases, the demonstration of several tasks of interest—such as the assembly of a product in an occluded factory line, or the interaction between two humans—is compromised by setup requirements and the need in estimating the kinematic structure of the demonstrator.

Here, we relax the requirement in tracking (parts of) the human skeleton and focus only on the task achievement of a manipulator arm; where the end-effector trajectory (e.g. the hand of the demonstrator) is the only recorded element. This approach greatly widens the range of possible setups and demonstration scenarios. Learning directly from observations in task space was presented in [10] where the focus was on the improvement of a policy initialized by the human demonstration. Their work did not address, however, how to map the human demonstration into the initial policy on the robot's frame—the goal of this paper. In fact, that work used inverse kinematics directly to follow the straight line trajectory of the human hand, and therefore their movements had a one-to-one task space reproduction.

Inverse kinematics has an important role in the human/robot kinematic mapping. Closed-form solutions [11] have the advantage of being fast enough for on-line applications at the expense of being robot-specific. Engineered solutions based on IK have been used to map intricate systems such as in [12] and used to generate stylistic and human-like mappings in animation [13]. In whole body control, IK is usually followed by an optimization step to satisfy dynamic constraints, such as balance [3].

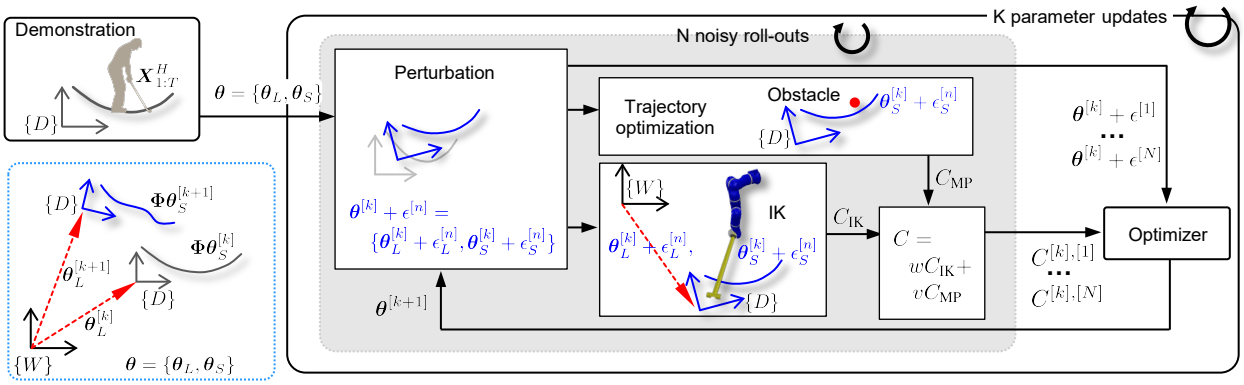

Fig. 2. The proposed method Imitation by Stochastic Optimization of the Embodiment Mapping (ISOEMP). The demonstration is parameterized in both shape $\boldsymbol{\theta}_S$ and location of the reference frame $\boldsymbol{\theta}_L$. Stochastic optimization is used to find the optimal shape and location such that the solution is feasible in relation to the kinematics of the robot, and also respects task constraints such as obstacles and via-points.

When learning from demonstration, an issue associated with IK is the re-location of the demonstration of the teacher in relation to the learner's body. This is usually achieved manually when obvious, for example, when the kinematic dimensions of the teacher and learner are similar; or by using some heuristic such as the T-pose [3] where both human and robot stand-up with the arms wide open, or other pre-defined starting poses [2]. In this paper, while recording only the end-effector trajectory relaxes the knowledge of the teacher's kinematic structure, it also brings the issue that references useful for positioning the observed trajectory w.r.t. the robot are now lost. Our method provides an automated way to search for the optimal location of the human demonstration based on the robot capability in reproducing it.

Since the problem of this paper is essentially to make the robot to reproduce a trajectory obtained from demonstrations, another approach is to use trajectory optimization [14], or motion planning techniques such as [15], [16], [17]. A problem that arises with such methods is that their formulations require the knowledge of initial and goal states. While these two states are known in the reference frame of the demonstrator $\{D\}$, we usually do not know their transformation on the frame of another agent $\{W\}$, especially if kinematic constraints have to be taken into account. This would then require some pre-processing step where the demonstration can be correctly located into the reference frame of the learner agent, but may lead to suboptimal solutions (see a comparison in Section IV where [17] was used). Our method explicit incorporates this transformation as part of the simultaneous optimization of location and trajectory shape.

III. Imitation by Stochastic Optimization of the Embodiment Mapping

Fig. 2 illustrates the workflow of the proposed method, referred to as Imitation by Stochastic Optimization of the Embodiment Mapping (ISOEMP). Given a trajectory of the human demonstration, we define the set of parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_L, \boldsymbol{\theta}_S\}$. $\boldsymbol{\theta}_L$ governs the location of the reference frame of the demonstration $\{D\}$ in relation to the robot's world frame $\{W\}$. The parameter $\boldsymbol{\theta}_S$ governs the shape of the trajectory and is defined relative to the demonstration frame $\{D\}$.

A. Policy Parameterization and Improvement

Assume a demonstration recorded as a Cartesian trajectory $\boldsymbol{X}^H_{1:T}$, where $\boldsymbol{X}_t = [x_t, y_t, z_t]^T$. As an initial policy, the trajectory is parameterized in a lower dimensional space of shape parameters

$$\boldsymbol{\theta}_S = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{X}^H_{1:T}, \tag{2}$$

where each Cartesian dimension is encoded by a column in $\boldsymbol{\theta}_S \in \mathbb{R}^{N\times 3}$. The design matrix $\boldsymbol{\Phi} \in \mathbb{R}^{T\times N}$ is composed of N Gaussian basis functions with centers spread evenly in the interval $[1,\ T]$,

$$\boldsymbol{\Phi} = \begin{bmatrix} \Phi_{1,1} & \dots & \Phi_{1,N} \\ \vdots & \ddots & \vdots \\ \Phi_{T,1} & \dots & \Phi_{T,N} \end{bmatrix} \tag{3}$$

which also enforces smooth solutions. In practice, the size N is usually one order of magnitude lower than the raw size of the sampled trajectory, for example, in the golf swing case, a 3 second trajectory sampled at 120 Hz is decreased from $3 \times 120 = 360$ points to $N = 20$ parameters.

The location of the demonstration's reference frame $\{D\}$ in relation to the robot's frame $\{W\}$ is given by the parameter vector $\boldsymbol{\theta}_L$, whose elements comprise the rotation and translation entries of a homogeneous transformation between frames ${}^W_D\boldsymbol{T}(\boldsymbol{\theta}_L)$ such that

$${}^W_D\boldsymbol{T}\colon \boldsymbol{X}_{1:T} \rightarrow \bar{\boldsymbol{X}}_{1:T}, \tag{4}$$

where $\bar{\boldsymbol{X}}_{1:T}$ are the Cartesian trajectory coordinates expressed in the robot's frame.

Thus, the trajectory $\bar{\boldsymbol{X}}_{1:T}$ is obtained from $\boldsymbol{\theta} = \{\boldsymbol{\theta}_L, \boldsymbol{\theta}_S\}$ by first recovering the trajectory in the reference frame of the demonstration $\boldsymbol{X}_{1:T}(\boldsymbol{\theta}_S) = \boldsymbol{\Phi}\boldsymbol{\theta}_S$ and then by using (4) to describe its coordinates in $\{W\}$. The bottom left of Fig. 2 illustrates two consecutive updates of the policy. Note that the trajectory optimization is executed in the local reference frame $\{D\}$ but, as it shall be clear in Section III-B, the shape is influenced by the absolute location of the trajectory since the cost that updates the policy $\boldsymbol{\theta}_S$ accounts for the accuracy of the inverse kinematics solution.

To optimize the policy parameters we rely on stochastic optimization based on episodic model-free reinforcement learning formulations, such as REPS [18] and PI^2 [19]. Such methods are based on perturbing the policy parameters to run a N number of noisy roll-outs, rendering a set of exploration trajectories and their respective costs. A policy improvement is obtained by a weighted average of the previous noisy policies where low-cost samples are attributed higher weights (refer to [20] for an in-depth survey).

Stochastic optimization methods are suited for our problem since discontinuity of the cost (due to inverse kinematics) is to be expected, making the use of gradient-based methods harder, if not impractical. Also, here we focus on kinematic mappings where roll-outs can be done numerically and assume that a feedback controller will suitably track the final solution. However, the use of RL methods keep open the possibility to run the roll-outs directly on the real robot, which would then allow us to take into account the dynamics of the robot under poor control performance.

B. Inverse Kinematics and Task Costs

The goal of the optimization is to decrease a cost where one of the components, C_{IK}, quantifies the robot's reproduction error of the current modification of the demonstration $\boldsymbol{X}_{1:T}$ when mapped to the robot's workspace with ${}^{W}_{D}\boldsymbol{T}(\boldsymbol{\theta}_L)\boldsymbol{X}_{1:T}$. Other costs that account for task constraints, such as obstacle avoidance and similarity with the original demonstration are represented in C_{MP},

$$C = C_{\text{IK}} + C_{\text{MP}}. \tag{5}$$

1) Inverse Kinematics Cost: Algorithm 1 shows a generic numerical inverse kinematics function for a redundant manipulator. Given a trajectory $\bar{\boldsymbol{X}}_{1:T}$ representing the current noisy roll-out of $\pi(\boldsymbol{\theta}+\epsilon)$ in the robot's reference frame, an initial guess of joint configuration $\boldsymbol{q}_i$, and a learning rate α, for each time step, the algorithm attempts to iteratively decrease the Cartesian error by updating the joint configurations until a maximum number of iterations $nIterMax$ is achieved.

Algorithm 1 InverseKinematics($\bar{\boldsymbol{X}}_{1:T}$, $\boldsymbol{q}_i$)

$\boldsymbol{q}_t \leftarrow \boldsymbol{q}_i$
for $t < T-1$ **do**
  while $\sim nIterMax$ **do**
    $\bar{\boldsymbol{X}}'_t \leftarrow$ ForwarKinematics($\boldsymbol{q}_t$)
    $\boldsymbol{e}_t \leftarrow$ computeFKError($\bar{\boldsymbol{X}}_t, \bar{\boldsymbol{X}}'_t$)
    $\boldsymbol{J}_t \leftarrow$ Jacobian($\boldsymbol{q}_t$)
    $\boldsymbol{\Delta q}_t \leftarrow \alpha \boldsymbol{J}_t^{-1}\boldsymbol{e}_t$.
    $\boldsymbol{q}_t \leftarrow \boldsymbol{q}_t + \alpha\boldsymbol{\Delta q}_t$

For each time step, $\boldsymbol{e}_t$ stores the minimum error achieved by the while loop, which yields a trajectory of Cartesian error $\boldsymbol{e}_{1:T}$. Thus we define the IK cost of the k-th iteration as

$$C^{[k]}_{\text{IK}} = w\sum_{t=1}^{T}\|\boldsymbol{e}^{[k]}_t\|. \tag{6}$$

Note that $\bar{\boldsymbol{X}}_t = {}^{W}_{D}\boldsymbol{T}(\boldsymbol{\theta}_L)\boldsymbol{X}_t(\boldsymbol{\theta}_S)$, and therefore $\boldsymbol{e}^{[k]}_t$ is the summand in the first term of (1). The scalar w weights the importance of the IK error with other terms in C_{MP}. In addition, commonly used terms in trajectory optimization problems such as joint accelerations can be added depending on the requirements of the task. Hereinafter, we will use the superscript $(\cdot)^{[k]}$ to explicitly indicate a k-th iteration when relevant, and omit it otherwise.

While Algorithm 1 is an optimization procedure by itself, a typical inverse kinematics solver has no influence on the input $\bar{\boldsymbol{X}}_{1:T}$. Our method aims at modifying $\bar{\boldsymbol{X}}_{1:T}$ such that it minimizes $\boldsymbol{e}_{1:T}$ by both changing the shape of the demonstration $\boldsymbol{X}_{1:T}(\boldsymbol{\theta}_S)$ and its position relative to the robot with ${}^{W}_{D}\boldsymbol{T}(\boldsymbol{\theta}_L)$. We define a kinematically feasible solution when $\boldsymbol{e}_{1:T}$ becomes a vector of zero values.

2) Task Achievement Cost: Since C_{IK} accounts for the kinematic feasibility of the solution, costs related to via-point requirements and obstacle avoidance can be computed and optimized directly in task space. Conventional motion planning and trajectory optimization penalties are used. In our specific application the cost takes the form

$$\begin{aligned} C^{[k]}_{\text{MP}} = v_1\sum_{t=1}^{T}\|\boldsymbol{X}^{H}_t - \boldsymbol{X}^{[k]}_t\| + v_2\text{Collision}(\bar{\boldsymbol{X}}^{[k]}_{1:T}) + \\ v_3\text{Viapoint}(\bar{\boldsymbol{X}}^{[k]}_{1:T}), \end{aligned} \tag{7}$$

where the first term penalizes for solutions that look dissimilar to the original human demonstration in the frame $\{D\}$. The second term computes obstacle collision cost in the frame of the robot $\{W\}$. The last term can optionally penalize for via-points, including desired start and initial states particular to the current robot's task, thus the trajectory must be expressed in $\{W\}$.

C. Adding Noise for Exploration

It is critical in stochastic optimization methods to perturb the policy parameters in a way that induces efficient exploration. To optimize the demonstrated trajectory, a noise model to disturb $\boldsymbol{\theta}_S$ suitable for motion planning problems was proposed in STOMP [17] where the covariance matrix from which exploration noise is sampled is given by

$$\begin{aligned} \boldsymbol{\Sigma}_{\epsilon,S} = (\mathbf{A}^T\mathbf{A})^{-1}, \\ \text{with } \mathbf{A} = \text{blockdiag ([1 -2 1])}, \end{aligned} \tag{8}$$

where $\mathbf{A}$ is a finite-differencing matrix. A characteristic of $\boldsymbol{\Sigma}_{\epsilon,S}$ is to create smooth perturbations spread over the whole trajectory. Particularly important for our method is that the exploration provided by (8) has the effect of generating gradual changes of shape between consecutive iterations of the optimizer, which is useful to maintain the similarity with the original human demonstration. We borrow from STOMP the same kind of exploration, but instead of adding the sampled noise at each time step of the trajectory, we add the noise on each of the shape parameters in $\boldsymbol{\theta}_S$. Thus, the covariance has dimensions $\mathbb{R}^{N\times N}$ rather than $\mathbb{R}^{T\times T}$.

The original STOMP algorithm was designed for start and goal-oriented exploration of trajectories. While this form of structure is suitable when kinematically feasible start and goal configurations are known in advance, this form of exploration

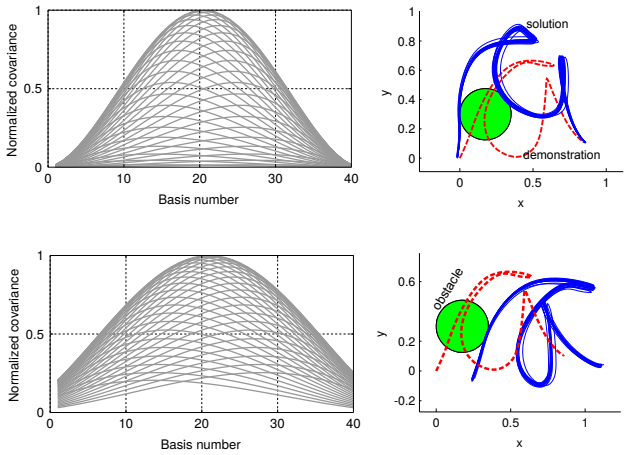

Fig. 3. The effect of clipping the covariance matrix allows the stochastic optimizer to search for solutions where the trajectory floats around the obstacle, leading to less deformations of the trajectory. Differently from goal oriented motion planning, this effect is beneficial since it adds an extra degree of freedom when searching for the optimal location of the demonstration.

does not apply to our problem as a bad initial guess on the location of the trajectory may render start and goal positions whose IK solution does not exist. Therefore, we propose a modification of the covariance obtained by clipping $\mathbf{\Sigma}_{\epsilon,S}$ such that the covariance at the extremities do not collapse to zero. This modification and its effect can be visualized in Fig. 3 where the upper row shows the plot of the rows of $\mathbf{\Sigma}_{\epsilon,S}$ and a simple toy-problem, free from kinematic constraints, solved by REPS where a letter "a" is modified from perturbations sampled from $\mathbf{\Sigma}_{\epsilon,S}$ such that it does not pass over the obstacle. The bottom row shows the effect of using the clipped covariance and the letter solution. Note that the first solution requires large deformations of the trajectory since the start and goal states are not explored.

To optimize the location of the reference frame $\{D\}$, each of the elements in $\boldsymbol{\theta}_L$ are perturbed independently with additive Gaussian noise, $\mathcal{N}(0, \mathbf{\Sigma}_{\epsilon,L})$, where $\mathbf{\Sigma}_{\epsilon,L}$ is diagonal. As a requirement of REPS, PI^2, and other stochastic optimizers, the scaling of the covariances $\mathbf{\Sigma}_{\epsilon,L}$ and $\mathbf{\Sigma}_{\epsilon,S}$ must be adjusted by the user given the problem at hand.

D. The ISOEMP Algorithm

Algorithm 2 ISOEMP($\boldsymbol{\theta}_L,\ \mathbf{\Sigma}_{\epsilon,L},\ \boldsymbol{\theta}_S,\ \mathbf{\Sigma}_{\epsilon,S}, (\cdot)$)

1: **for** $k < K$ **do** // Updates
2: $[C_{\text{IK}}^{[k]}, C_{\text{MP}}^{[k]}] \leftarrow$ rollOut($[\boldsymbol{\theta}_L^{[k]},\ \boldsymbol{\theta}_S^{[k]}]$) // Clean roll-out
3: **if** $[C_{\text{IK}}^{[k]} + C_{\text{MP}}^{[k]}] < minTotalCost$ **then**
4: return success
5: **for** $n < N$ **do** // Noisy roll-outs
6: $[\boldsymbol{\epsilon}_L, \boldsymbol{\epsilon}_S] \leftarrow$ sample($\mathbf{\Sigma}_{\epsilon,L}, \mathbf{\Sigma}_{\epsilon,S}$)
7: $[C_{\text{IK}}, C_{\text{MP}}] \leftarrow$ rollOut($[\boldsymbol{\theta}_L^{[k]} + \boldsymbol{\epsilon}_L,\ \boldsymbol{\theta}_S^{[k]} + \boldsymbol{\epsilon}_S]$)
8: $C^{[k][n]} \leftarrow [C_{\text{IK}} + C_{\text{MP}}]$
9: $[\boldsymbol{\theta}_L^{[k+1]}, \boldsymbol{\theta}_S^{[k+1]}] \leftarrow$Optimizer($\boldsymbol{\theta}^{[k][1:N]}$,$C^{[k][1:N]}$)

Algorithm 2 shows the ISOEMP pseudo-code. Line 2 quantifies the total cost with a "clean" roll-out, when the current policy parameters are free from noise. This clean solution allows us to compute its cost, which is compared to a minimum desired value $minTotalCost$. The loop in line 5 generates a N number of policy variations by perturbing the current parameters with additive noise sampled from $\mathbf{\Sigma}_{\epsilon,L}$ and $\mathbf{\Sigma}_{\epsilon,S}$. A stochastic optimizer is then given access to the policy variations and their respective costs to update the next policy. Note that in the case the covariance matrix $\mathbf{\Sigma}_{\epsilon,S}$ is zero, the algorithm will search for the transformation ${}^{W}_{D}\boldsymbol{T}(\boldsymbol{\theta}_L)$ that best matches the robot kinematics without changing the shape of the original human demonstration (see Fig. 11(a)).

IV. Experiments: Imitating Human Demonstrations of Writing and Golf Swing

In this section we evaluate our method in two different tasks of imitating natural human observations. First with a planar toy-problem, and later with real robot experiments.

A. Illustrative Toy-Problem: Drawing the Demonstration of a Letter on a Constrained Space

As shown in Fig. 4, in this simulated toy-problem a hand-written letter "a" written by a human using a mouse on a computer screen must be reproduced by a planar serial-link manipulator with three revolute joints. Joint limits were set such that the robot's workspace largely overlaps with an area where the robot was forbidden to write. Under those constraints, it is not possible to draw the same shape of the original letter without going through the forbidden area. An alternative shape must be found while resembling the original human writing. The bottom plots show the trajectory of inverse kinematic error $\boldsymbol{e}_{1:T}$ for each policy update. Note from the error in (b) that several solutions have discontinuities when parts of the trajectory are out of the arm reach, and gradient-free stochastic optimizers prove suitable. As a local method, the quality of the final solution depends on the initial guess. The solution in (a) has a final shape that better resembles the initially demonstrated letter when compared to the solution found in (b).

On the same problem, we compared our method with STOMP. As a motion planner based on trajectory optimization, STOMP assumes that the initial and final states in joint space are given. This is an ill-posed problem for STOMP as initial and final states are only known in respect to $\{D\}$, but not $\{W\}$. This would require an initial search of the positioning of the demonstration in the robot's workspace, which by itself could be solved as an optimization problem. To make the comparison possible, we used the clipped noise covariance described in Section III-C such that the trajectory is allowed to "float" in the workspace and initial and final conditions do not need to be assumed.

Fig. 5(a) shows the ISOEMP and STOMP solutions found given the same initial conditions and cost. In the case of STOMP, the cost C is computed from the forward kinematics of the current solution, whose joint positions are saturated at the joint limits. As shown in the figure, due to the bad placement of the initial demonstration in relation to the robot's workspace, the robot can only write the letter partially. Both

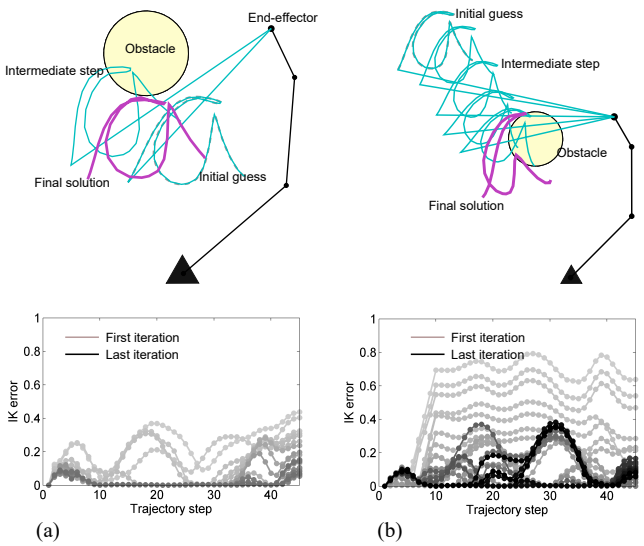

Fig. 4. The robot reproduces a letter "a" demonstrated by a human. The robot must find where to position the reference frame of the demonstration in relation to itself, and also how to modify the shape of the letter such that it does not pass over the forbidden circular area. (a) An initial guess close to the robot. (b) A initial guess that starts far from the robot. The axes on the top plots were removed to improve clarity.

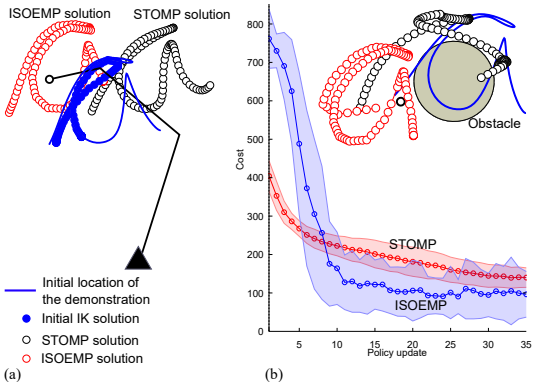

Fig. 5. A comparison of ISOEMP and STOMP. (a) In free space both methods find the reference frame transformation that leads to minimal trajectory deformations. (b) When an obstacle is placed, STOMP leads to large trajectory deformations, that during the first iterations achieve lower costs, but leads to poor final results. The explicit search of the reference frame transformation allows ISOEMP to find locations that require less deformations of the demonstration.

methods successfully search for a location of the letter while preserving the original shape. Fig. 5(b) shows the case where an obstacle is added. A total of 25 random initial placements of the demonstration are made and the plot shows the average cost C_{total} as a function of the number of parameter updates. The patches represent the mean $\pm$ one standard deviation. The trajectories shown in (b) illustrate one of the solutions. ISOEMP has larger cost during the first iterations when the original demonstration tends to be placed on the top of the obstacle, which is a consequence of the explicit search of the location of $\{D\}$. STOMP, on the other hand, deviates from the obstacles from the first iteration and therefore the cost is lower at the beginning, but this also leads to poor convergence and large trajectory deformations. After the 10-th iteration in average, ISOEMP finds a location of lower cost and the small changes in the shape dominates the subsequent updates.

B. Generalizing Golf Skills from Human Demonstration

The method was evaluated on a real golf swing task where a human demonstration using motion capture data was mapped to the kinematics of different robots. We chose the golf swing as a task that although simple and easy to demonstrate as natural human movement, is not easily achievable by kinesthetic teaching and teleoperation. A swing trajectory of the tip of the golf club was recorded with a marker tracking system shown in Fig. 6 as a sequence of snapshots.

Fig. 7 shows the ability of the method in mapping the recorded demonstration to a trajectory that fits the kinematics of an industrial 6-DoF ABB IRB140. As shown in Fig. 7(a), the trajectory is too far from the robot's reach and the end-effector trajectory cannot reproduce it. After the optimization, Fig. 7(b) shows that the demonstration was reshaped such that it fits the robot's kinematics and also allows the golf-club to pass through a via-point that represents the ball location, which includes the orientation of the hitting. Fig. 7(c) shows the initial and final solutions where the orientation of the final trajectory is aligned with the via-point pose. Fig. 7(d) shows traces of the updates.

We evaluated the ability of the algorithm to scale the demonstration under different dissimilarity weights (v_1 in (7)). Initially, we scaled the original golf swing by a factor of 3 in the XY directions, resulting in a trajectory that spans roughly 7 meters in the longest direction. Note that given such a large span, there is no frame transformation ${}^{W}_{D}\boldsymbol{T}(\boldsymbol{\theta}_L)$ that results in a low C_{IK} cost unless the demonstration is scaled down in certain directions. To simplify the analysis, we ignored the presence of obstacles. The weight w used in (6) was set to a fixed value of 100. The weight of the dissimilarity v_1 was set to 1, 10, and 100 values. Their effect can be seen in Fig. 8 where the table shows the IK error values normalized to the highest value. The deviation was computed as the average Euclidean distance between the original demonstration and the current solution in the frame $\{D\}$, over the whole trajectory. The figure shows that low penalty in the dissimilarity leads to accurate IK solutions with shapes that have a large deviation from the original demonstration, and the opposite effect when the dissimilarity is heavily penalized. Note that trajectories have a large deformation in the direction of the swing, which generates high costs in C_{IK}, while having less deformations on the width of the swing. The method provides a form of adaptive scaling, where the directions that can be readily reproduced maintain the maintain closer similarity with the demonstration.

Fig. 9 shows under the same costs, initial conditions, and number of iterations, the solutions achieved by robots with very different kinematics (see Fig. 1(c)). Due to the limited DoFs of the SCARA robot, the deviations required from the original demonstration and the reproduction error are the largest. The IRB140 and the LWR4 robots achieve roughly the same cost C_{IK}, however, note that their final trajectory solutions are not the same. The LWR4 final trajectory has more similarity with the original demonstration than the solution

Fig. 6. A sequence of snapshots where the human demonstrates a golf swing movement while the tip of the club are recorded in Cartesian coordinates.

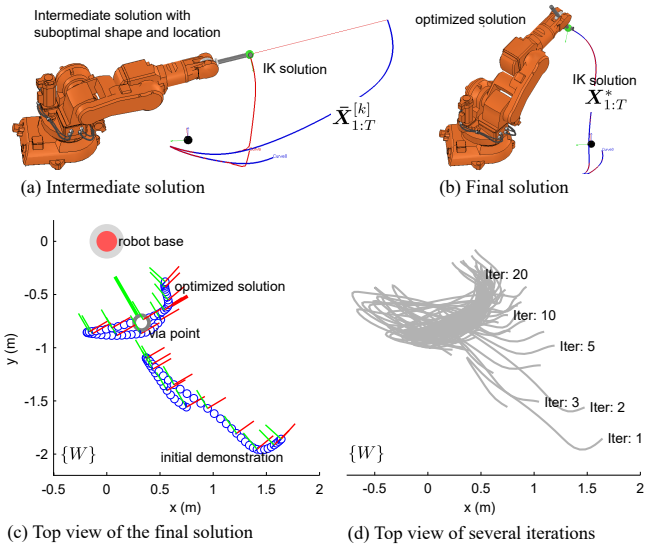

Fig. 7. The human demonstration is mapped into a feasible trajectory that the robot can reproduced accurately while passing through a via-point that represents the location of the ball hitting.

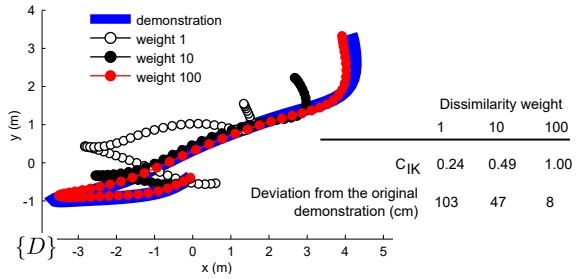

Fig. 8. The effect of different weightings that punish the dissimilarity with the original demonstration using the ABB IRB140 model. The initial demonstration was scaled to be 3 times larger in the XY directions. The method provides automatic scaling of the demonstration as the directions that can be readily reproduced by the robot will contribute with a lower cost in C_{IK}.

found by the IRB140 robot. This could be given by the fact that, compared to the IRB140, the LWR4 arm is structurally closer to that of the human demonstrator, with an extra DoF and longer reach.

Finally, we evaluated the method using a real LWR4 arm. To avoid large dynamical disturbances and damages to the robot hands, a light, plastic golf club was used. Due to the limited space available in the laboratory, we fixed the yaw rotation such that the club could only swing on a plane that is parallel to the base frame of the robot. Fig. 11 shows three different solutions obtained by our method which differ by via-point constraints where in each case, the initial human

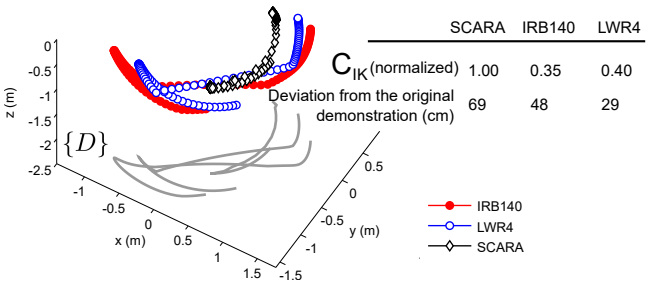

Fig. 9. Comparison of solutions found when using the same costs, initial conditions and number of iterations under different robot structures. The plots are given on the reference frame of the demonstration.

demonstration (traced curves), was initially set at an arbitrary location somewhat close to the robot.

In Fig. 11(a) the shape of the original human demonstration was preserved by freezing updates in $\boldsymbol{\theta}_S$ by setting $\boldsymbol{\Sigma}_{\epsilon,S} = \mathbf{0}$. In this case, ISOEMP simply searches for the optimal location of the original demonstration in relation to the robot's frame. The optimized location of $\{D\}$ was found to be [68.8, -0.3, 47.0] cm in the XYZ directions away from the initial guess location. Fig. 11(b-c) shows the generalized trajectories for the case where the instant of the ball hitting was used as a temporal via-point with varying speeds. The speed refers to the magnitude of the projected velocity vector onto the horizontal plane. The hitting instant was set to have a speed 50% lower in (b) and 50% higher in (c), both in relation to the original human demonstration. Large deformations of the movement are observed since all trajectories are constrained to have the same duration of the original demonstration. The method could still find kinematically feasible solutions[1]. A sequence of snapshots of the robot executing the mapped swing trajectory can be seen in Fig. 10. A video of the experiment accompanies this paper and will be available at `http://ieeexplore.ieee.org`. A high resolution version is available at `http://youtu.be/6o8was_tZZ4`.

V. Conclusion

This paper proposed a method that allows a robot to acquire kinematics skills directly from natural human movements recorded as Cartesian trajectories. To solve the correspondence problem between the different teacher-learner embodiments, we proposed using the residual error of the IK algorithm as a metric to quantify the accuracy achieved by the robot when

[1]The experiment was set in this manner to validate the IK adaptation. A trivial solution would be to accelerate/decelerate the whole trajectory uniformly without any shape changes.

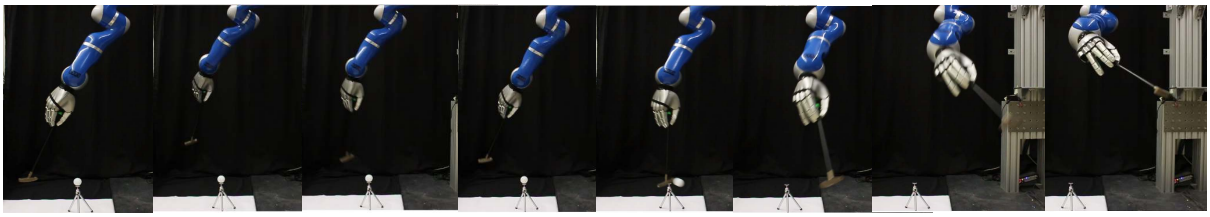

Fig. 10. A sequence of snapshots where the robot hits the ball with a velocity 50% faster than the hitting originally demonstrated by the human.

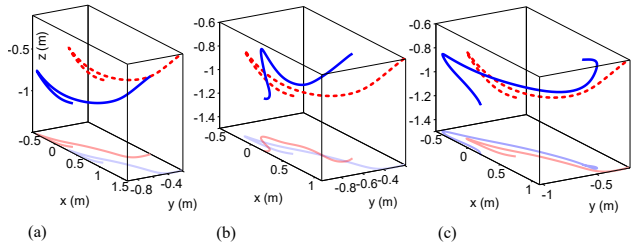

Fig. 11. The traced curves represent the original human demonstration. The solid curves represent the solution of ISOEMP when generalizing the demonstration to different speeds at the via-point. (a) Original velocity profile preserved, (b) 50% slower at the hitting instant, (c) 50% faster at the hitting instant.

reproducing perturbed trajectory variations. Using stochastic optimization, the method searches for a local optimum where the error in the kinematic mapping and the satisfaction of task constraints are minimized. We showed that our framework is capable of mapping and generalizing skills with robots of different kinematics given the same trajectory of an observed human. A promising application of the method is learning human-robot collaborative tasks from the observation of two human coworkers. Finally, we envision that the contribution of the method can be greatly boosted with a combination of video image analysis; when the method here proposed will allow the use of the massive amount of truly optimal human demonstrations—for example, of professional player's movements—already documented in videos.

VI. Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7-ICT-2013-10) under grant agreement 610878 (3rdHand) and from the BIMROB project of the "Forum für interdisziplinäre Forschung" (FiF) of the TU Darmstadt.

References

[1] M. Gleicher, "Retargetting motion to new characters," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 1998, pp. 33–42.

[2] T. Tosun, R. Mead, and R. Stengel, "A general method for kinematic retargeting: Adapting poses between humans and robots," in *ASME 2014 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, 2014, pp. 1–10.

[3] J. Koenemann, F. Burget, and M. Bennewitz, "Real-time imitation of human whole-body motions by humanoids," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2806–2812.

[4] Z. Song, J. Yu, C. Zhou, D. Tao, and Y. Xie, "Skeleton correspondence construction and its applications in animation style reusing," *Neurocomputing*, vol. 120, pp. 461–468, 2013.

[5] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson, "Adapting human motion for the control of a humanoid robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2002, pp. 1390–1397.

[6] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin, "On human motion imitation by humanoid robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 2697–2704.

[7] S. Michieletto, N. Chessa, and E. Menegatti, "Learning how to approach industrial robot tasks from natural demonstrations," in *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 2013, pp. 255–260.

[8] M. Do, P. Azad, T. Asfour, and R. Dillmann, "Imitation of human motion on a humanoid robot using non-linear optimization," in *Proceedings of the IEEE/RAS International Conference on Humanoids Robots (HUMANOIDS)*, 2008, pp. 545–552.

[9] A. Dragan and S. Srinivasa, "Online customization of teleoperation interfaces," in *IEEE International Symposium on Robot and Human Interactive Communication (Ro-Man)*, September 2012.

[10] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *International Conference on Machine Learning (ICML)*, vol. 97, 1997, pp. 12–20.

[11] T. Asfour and R. Dillmann, "Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, 2003, pp. 1407–1412.

[12] K. Yamane, J. K. Hodgins, and H. B. Brown, "Controlling a motorized marionette with human motion capture data," *Proceedings of the IEEE/RAS International Conference on Humanoids Robots (HUMANOIDS)*, vol. 1, no. 04, pp. 651–669, 2004.

[13] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović, "Style-based inverse kinematics," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 522–531.

[14] J. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[15] D. Berenson, S. S. Srinivasa, D. Ferguson, A. Collet, and J. J. Kuffner, "Manipulation planning with workspace goal regions," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 618–624.

[16] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 489–494.

[17] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 4569–4574.

[18] J. Peters, K. Mülling, and Y. Altun, "Relative entropy policy search." in *Advancement of Artificial Intelligence (AAAI)*, 2010.

[19] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *The Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.

[20] M. P. Deisenroth, G. Neumann, J. Peters *et al.*, "A survey on policy search for robotics." *Foundations and Trends in Robotics*, vol. 2, no. 1-2, pp. 1–142, 2013.