Directing Policy Search with Interactively Taught Via-Points

Yannick Schroecker
College of Computing
Georgia Tech
Atlanta, GA 30332, USA
yschroecker3@gatech.edu

Heni Ben Amor
Interactive Robotics Lab
Arizona State University
Tempe, AZ 85282, USA
hbenamor@asu.edu

Andrea Thomaz
Department of ECE
University of Texas at Austin
Austin, TX 78701, USA
athomaz@ece.utexas.edu

ABSTRACT

Policy search has been successfully applied to robot motor learning problems. However, for moderately complex tasks the necessity of good heuristics or initialization still arises. One method that has been used to alleviate this problem is to utilize demonstrations obtained by a human teacher as a starting point for policy search in the space of trajectories. In this paper we describe an alternative way of giving demonstrations as soft via-points and show how they can be used for initialization as well as for active corrections during the learning process. With this approach, we restrict the search space to trajectories that will be close to the taught via-points at the taught time and thereby significantly reduce the number of samples necessary to learn a good policy. We show with a simulated robot arm that our method can efficiently learn to insert an object in a hole with just a minimal demonstration and evaluate our method further on a synthetic letter reproduction task.

Keywords

Reinforcement Learning, Learning from Demonstration, Reinforcement Learning for Motor Skills, Dynamic Movement Primitives, Keyframe Demonstrations

1. INTRODUCTION

Robotics research in recent years has been working towards employing robots in unknown and often unstructured environments. However, controlling a robot in these environments poses major difficulties as the robot has to adapt its actions to the environment and needs to perform tasks with incomplete knowledge of the domain. Reinforcement learning and policy search methods in particular have shown great promise for autonomous learning of motor skills as trajectories. However, due to the high dimensionality and size of the state-action space, the required amount of samples can be prohibitive. A prominent approach to overcome this challenge is to use Learning from Demonstration to obtain an initial trajectory and to ensure that the learning process will quickly converge to the right optimum, see e.g. [15, 18, 4]. Unfortunately, this approach suffers from three major issues: First, providing trajectories as demonstrations can be difficult due to the available input modalities. Recording the desired trajectories using teleoperation or kinesthetic demonstration can be difficult for the user and thus often leads to undesirable pauses, sprints or imperfections. Second, providing full trajectories as demonstrations does not allow the user to put focus on critical segments and limits exploration for segments that the user cannot demonstrate as well as other parts of the trajectory. Third, demonstrations are typically only provided as an initialization of the policy search process. Refining a learned policy and removing undesirable effects usually requires a repeated recording of the entire demonstration and cannot be limited to specific aspects of the task. This problem is amplified if the policy search process has already been started as the learned policy can usually not be combined with new demonstrations.

To address these issues, we propose a method that uses soft via-points to initialize and interactively shape the policy search process. Recent research [1] has shown that via-point based representations can be efficiently obtained by a human teacher in the form of demonstrations and provide the teacher with a more natural way of teaching the desired trajectory. This kind of demonstration can achieve smoother trajectories by separating the act of moving the robot as a teacher from the intended trajectory that the robot should follow. Furthermore, it allows the teacher to focus on the most important aspects of the trajectory. In this paper, we introduce a method to use demonstrations in this form with policy search methods based on Dynamic Movement Primitives. More specifically, we look at policy search methods that can optimize DMP parameters and thereby improve the policy by evaluating parameters sampled around the current best estimate of the optimal policy. This class of policy search algorithms has shown great promise and includes methods such as PoWER [14], REPS [20], policy search based on CMA-ES [8, 22] and PI^2 [24]. By combining these two approaches we allow the teacher to demonstrate the salient points of a trajectory in a natural way while autonomously learning and improving the shape of the trajectory between those via-points. Our method achieves this goal by learning a single and smooth trajectory that adheres to the demonstrated via-points. Furthermore, we introduce a method to modify existing policy distributions by selecting the most likely trajectories based on the provided demonstration. Modifying the trajectory distribution in this way allows the user to provide corrections to existing policies and thereby shape the learning process. These corrections can be applied to policies learned by demonstration, either continuous demonstrations or via-point based demonstrations,

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

as well as to policies learned during the autonomous learning process. Finally, we propose to use models of via-points to handle variations of tasks that differ in parameters such as the position and orientation of key-objects in a manipulation task. We show that this facilitates efficient contextual policy search [13, 16] and allows to learn classes of tasks.

2. RELATED WORK

The work presented in this paper is focusing on policy search for robotics. For a survey of this topic, see Deisenroth et al. [6]. Specifically, we are looking at utilizing policy search methods for learning Dynamic Movement Primitives [10] that share the characteristic that they are sampling directly from the current estimate of the optimal policy. One example for this is the Covariance Matrix Adaption Evolution Strategy (CMA-ES) which has been used in [22] in order to learn dynamic movement primitives based on a reward signal. In [14], Kober et al. propose Policy Learning by Weighting Exploration with the Returns (PoWER) which uses regression weighted by rewards in order to obtain a new policy. Another method in this class is PI^2 [24] which uses path integrals to improve on the current policy.

In our evaluation, we use Episodic Relative Entropy Policy Search (REPS) [20] as the underlying policy search method. Peters et al. derive a sample based approximation of the optimal distribution of DMP parameters given trajectory- and reward samples as well as a bound on the KL-divergence between the optimized distribution and the distribution from which the trajectories were sampled. Using this approximation, the mean can iteratively be updated using weighted linear regression over the sampled DMP parameters and the variance can be updated by the weighted sample variance. The proper weights can be calculated by solving a convex optimization problem based on the bound on the KL-divergence and the sampled parameters and rewards. We refer to [20] for details.

In this paper, we consider an approach based on directing the policy search with demonstrations obtained by a human teacher. The field of Learning from Demonstration(LfD) has been extensively covered in [4] and LfD methods have successfully been combined with policy search for Dynamic Movement Primitives [15, 18]. In particular, we are looking at recording partial demonstrations in the form of via-points which have a long history in trajectory generation. Teaching via-points by demonstrations is also known as keyframe demonstration and has been shown to constitute a user-friendly and efficient way to obtain demonstrations [1]. Wada et al. extract via-points from a continuous demonstration and show that these can be used to create a trajectory that minimizes torque change [25]. Miyamoto et al. extend this approach and apply it to learning robot motor skills [17]. Bitzer et al. introduce an approach that combines keyframe demonstrations with reinforcement learning by learning a lower-dimensional manifold to simplify the state-space for a non-episodic reinforcement learner [3]. This method differs from our approach in that it only learns a simpler state-space representation and does not learn a heuristic for specific trajectories. Utilizing corrections in order to change a policy learned from demonstration has been introduced by Argall et al. who propose to use tactile corrections [2]. While this approach is interesting, it cannot be straight-forwardly integrated into autonomous policy search algorithms such as the ones utilized in our approach.

Utilizing feedback from human teachers has been investigated in the field of interactive reinforcement learning. Knox and Stone [12] introduce TAMER+RL, a reinforcement learning framework that utilizes a reward signal obtained by a human teacher in order to learn a regression model that can fulfill a role similar to a Q-function. Another example is given by Griffith et al. [7] who have introduced Policy Shaping. Policy shaping is utilizing rewards obtained by a human teacher in order to learn a separate policy. This policy can then be combined with the policy learned by standard reinforcement learning methods. Judah et al. [11] propose an integrated approach which optimizes a modified objective function based on the reward as well as on human feedback in the form of binary labels. All three methods utilize feedback provided by a human teacher but are different from our approach in that the feedback takes the shape of a reward-like signal instead of demonstrations. One can see both, the learning from demonstration based approach as well as the interactive reinforcement learning approach as belonging to a generalized class of algorithms that utilize insight obtained by a human teacher in order to improve the policy. In this view, the interactive reinforcement learning is online and less structured whereas the classical learning from demonstration based approach is offline and utilizes structured feedback. Our approach is structured as well but can be used in an offline manner as well as online.

Another representation of distributions over trajectories that can be restricted to go through specified via points is called Probabilistic Movement Primitives and has been proposed in [19]. Probabilistic Movement Primitives define feed-forward trajectories directly as combination of basis-functions and define operations on Gaussian distributions over such trajectories. The conditioning operation defined in this approach is similar to the operation for distributions over DMPs introduced in section 3.2. However, while it is likely that Probabilistic Movement Primitives could also be used with our approach, we are focusing on Dynamic Movement Primitives as they are more popular and better understood.

3. APPROACH

In this paper, we want to utilize a human teacher in order to provide corrections and suggestions before and during the learning process. Our goal is to use this information to help the learning algorithm converge to a better solution after seeing fewer samples. Specifically, we propose a setup where the teacher is observing the reinforcement learning process and can, before starting the learning process or in-between iterations, inspect the current estimate of the best policy and provide suggestions by physically or remotely moving the robot. Suggestions are recorded as soft via-points y^* that the trajectories have to pass through at a specified time t^*. In the case of corrections, the time t^* can be naturally recorded by having the user stop the robot during an execution in order to provide the correction. This process is illustrated in Algorithm 1. In the case of initial demonstrations, the time t^* can be estimated manually based on domain knowledge. While choosing the right t^* in this case may require some thought, we have found that simple heuristics such as distributing via-points equally in time or choosing t^* to be proportional to the spatial distance of the via-point are usually sufficient.

We base our method on episodic policy search in the space of trajectories represented as Dynamic Movement Primitives (DMPs) [10, 21] which we describe in detail in section 3.1. These algorithms optimize the parameters of the DMP w.r.t. a given reward function, often by a weighted average with weights obtained based on a transformation of the rewards. The parameters of the DMP uniquely define a policy and thus, we loosely refer to the parameters $\boldsymbol{\theta}$ as policy and to the distribution over parameters $\pi(\boldsymbol{\theta})$ as policy distribution. To incorporate demonstrations and corrections into this framework, we use the given via-points as a heuristic to guide the learning process to the solution without directly modifying the given objective function. To this end, we obtain a modified policy distribution and sample only trajectories that are close to the demonstrated via-points. By modifying the policy directly, the effects of demonstrations and corrections are immediate and the learning process never samples trajectories that are far from the demonstrated via-points. This leads to faster convergence and safer samples. As we modify the policy directly, we require policy search methods that improve upon the given policy in independent iterations based on samples obtained directly from the policy such as PoWER [14], REPS [20] and CMA-ES [22] which obtain a new policy based on a weighted average of the samples obtained from the old policy distribution as well as PI^2 [24].

In many cases it can be desirable to consider parameterized tasks as this allows us to learn variations of a task instead of optimizing for a single trajectory. For example, the optimal trajectory in a manipulation task may depend on the location of key objects. One way to solve tasks such as these is to learn a linear model of the parameters $\mu_\pi = A_\pi \boldsymbol{\Phi}$ for some features $\boldsymbol{\Phi}$ to serve as the mean of the policy distribution [16]. To handle parameterized tasks, we generalize our notion of via-points to models that are linear in $\boldsymbol{\Phi}$, i.e. $y^* = B\boldsymbol{\Phi}$ where B is either derived from domain knowledge or learned with linear regression.

Algorithm 1 Policy search with interactive demonstrations

```
Initialize $\pi^{(0)}(\boldsymbol{\theta}) \leftarrow \mathcal{N}(\boldsymbol{\theta}; \mu_{\pi^{(0)}}, \Sigma_{\pi^{(0)}})$
Obtain initial via-points $\mathbf{y}^*, \mathbf{t}^*$ from demonstration
for $y^*, t^* \leftarrow \mathbf{y}^*, \mathbf{t}^*$ do
    $\pi^{(0)}(\boldsymbol{\theta}) \leftarrow p(\boldsymbol{\theta}|y^*, t^*, \mu_{\pi^{(0)}}, \Sigma_{\pi^{(0)}})$ (see Eqs. 18-21)
end for
for $k \leftarrow 1$ to $N$ iterations do
    $\pi^{(k)}(\boldsymbol{\theta}) \leftarrow$ POLICY_SEARCH_ITERATION($\pi^{(k)}$)
    Execute trajectory defined by mean parameters $\mu_{\pi^{(k)}}$
    while teacher stops execution do
        Record stop time $t^*$
        Let teacher move the robot, obtain correction $y^*$
        $\pi^{(k)}(\boldsymbol{\theta}) \leftarrow p(\boldsymbol{\theta}|y^*, t^*, \mu_{\pi^{(k)}}, \Sigma_{\pi^{(k)}})$
        Execute mean trajectory defined by $\mu_{\pi^{(k)}}$
    end while
end for
```

3.1 Dynamic Movement Primitives

Policy search relies on optimizing the parameters of a parametric policy representation. One such policy representation are Dynamic Movement Primitives which have been introduced by Ijspeert et al. in [10]. DMPs are given as dynamical systems that are attracted by a goal position while following a superimposed trajectory. As such they have the property that they can adjust the goal position of the trajectory separately from the shape of the trajectory. Furthermore, DMPs have the concept of a phase which is a function of time which can be adapted by changing the time-scale. This is meant to reduce the dependency on the time. DMPs are defined as

$$\ddot{y} = \tau^2 \alpha \left(\beta(g - y) - \frac{\dot{y}}{\tau} \right) + \tau^2 f_{\mathbf{w}}(z), \tag{1}$$

$$\dot{z} = -\tau \alpha_z z. \tag{2}$$

where τ is the time scaling parameter and α_z is a parameter that shapes the phase function. α and β are parameters that are analogous to the gains of a PD-controller and define how the system is drawn to the goal of the trajectory which is defined by g. $f_{\mathbf{w}}(z)$ is the forcing function which determines the shape of the trajectory and is defined as a mixture of radial basis functions with parameters $K \in \mathbb{N}, \mathbf{c}, \mathbf{h} \in \mathbb{R}^K$

$$f_{\mathbf{w}}(z) \doteq \frac{\sum_{i=1}^K \varphi_i(z) w_i}{\sum_{i=1}^K \varphi_i(z)} z, \tag{3}$$

$$\varphi_i(z) \doteq \exp\left(-\frac{1}{2} \frac{(z - c_i)^2}{h_i} \right). \tag{4}$$

Commonly, the weights $w_i; 0 \leq i < K$ are taken as the parameters of the DMP that are learned by demonstration or autonomously whereas the other parameters are given as hyper-parameters. However, in some cases the goals are not known. We therefore include the goal position in the parameters $\boldsymbol{\theta} \doteq \left(\begin{smallmatrix} g \\ \boldsymbol{w} \end{smallmatrix} \right)$. K then determines the dimensionality of the parameter-space. In this paper, we are utilizing Gaussian distributions over DMP parameters as policy distributions and optimize this distribution with respect to the reward. The mean of this distribution π will be given as a linear function of the features of the task parameters: $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}, A_\pi \boldsymbol{\Phi}, \Sigma_\pi)$.

3.2 Obtaining a Sample Distribution

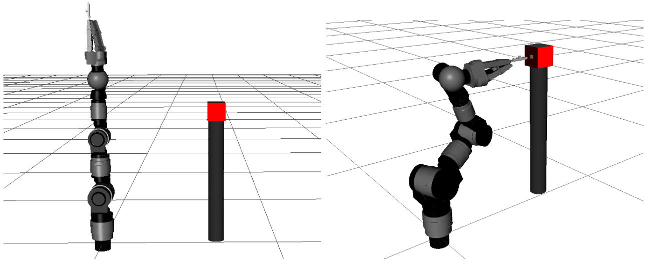
Assuming that we have via-points obtained from demonstration as described above, we derive a modified policy distribution for the underlying policy search that passes close to this via-point at the specified time t^*:

$$\pi_H(\boldsymbol{\theta}) \doteq p(\boldsymbol{\theta}|t^*, y^*) \propto p(y^*|t^*, \sigma_y \boldsymbol{I}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|t^*). \tag{5}$$

The latter distribution over parameters $\boldsymbol{\theta}$ denotes the prior of where we assume that the human teacher would want the samples to lie. A possible prior is the current policy $\mathcal{N}(\boldsymbol{\theta}|\mu_{\pi^{(k)}}, \Sigma_{\pi^{(k)}})$ where $\mu_{\pi^{(k)}} = A_{\pi^{(k)}} \boldsymbol{\Phi}$. This prior is reasonable as we want our samples to still follow the current policy where no via points are given. The modified sampling distribution then depends on the previous policy distribution and is given by

$$\begin{aligned} \pi_H(\boldsymbol{\theta}) &= p(\boldsymbol{\theta}|t^*, y^*, A_{\pi^{(k)}}, \Sigma_{\pi^{(k)}}) \\ &\propto p\left(y^*|t^*, \sigma_y \boldsymbol{I}, \boldsymbol{\theta}\right) \pi(\boldsymbol{\theta}|A_{\pi^{(k)}}, \Sigma_{\pi^{(k)}}). \end{aligned} \tag{6}$$

The former distribution $p(y^*|t^*, \sigma_y \boldsymbol{I}, \boldsymbol{\theta})$ denotes the probability of a given DMP going through the specified via point with a specified variance σ_y. This distribution is dependent on the trajectory that the DMP is following. As DMPs are defining accelerations as a linear differential equation, they can be solved for y given a starting position and velocity in order to obtain an estimate of the position at any given time. Note that the solution will not be exact as a real robotic system always has noise and the DMP will react to that noise



(a) (b)

Figure 1: **a)** Rest position of the robot. The robot has to find a trajectory that puts the object in the box. **b)** Via-point with the key placed in front of the hole, as it was given to the robot. All trajectories have to be close to this via-point which gives significant aid for finding the opening in the box.

as well as inaccuracies in the underlying controller. Only the goal position is being tracked exactly. However, using this estimate is reasonable as long as the noise of the estimate is significantly smaller than the inaccuracies introduced by the human teacher when recording a demonstration. Assuming that we start from a rest position where the position y_0 and velocity $\dot{y}_0$ are zero (we make this assumption for the sake of simplicity. See the appendix for a derivation of the sampling distribution for arbitrary y_0 and $\dot{y}_0$), the dynamical system can be solved for the position y using Duhamel's principle [23]:

$$\left(\begin{smallmatrix}\boldsymbol{y}\\ \dot{\boldsymbol{y}}\end{smallmatrix}\right) = \int_0^t h_t(s)\left(\begin{smallmatrix}0\\ \tau^2(\alpha\beta g+\boldsymbol{\Psi}(z(s))^T\boldsymbol{w})\end{smallmatrix}\right) ds, \tag{7}$$

where

$$h_t(s) \doteq e^{(t-s)\left(\begin{smallmatrix}0 & 1\\ -\tau^2\alpha\beta & -\tau\alpha\end{smallmatrix}\right)}, \tag{8}$$

$$z(t) \doteq e^{-\tau\alpha_z t}, \tag{9}$$

$$\boldsymbol{\Psi}_i(z) \doteq \frac{\varphi_i(z)z}{\sum_{j=0}^K \varphi_j(z)}. \tag{10}$$

This equation is linear w.r.t the parameters $\boldsymbol{\theta}$ and can therefore be written as:

$$\boldsymbol{y} = \left(\begin{smallmatrix}1 & 0\end{smallmatrix}\right)\left(\int_0^t h_t(s)\left(\begin{smallmatrix}0\\ \tau^2(\alpha\beta g+\boldsymbol{\Psi}(z(s))^T\boldsymbol{w})\end{smallmatrix}\right) ds\right) \tag{11}$$

$$= \left(\begin{smallmatrix}1 & 0\end{smallmatrix}\right)\left(g\alpha\beta\int_0^t h_t(s)\left(\begin{smallmatrix}0\\ \tau^2\end{smallmatrix}\right) ds \right. \tag{12}$$

$$\left. \sum_{i=0}^N w_i \int_0^t \Psi_i(z(s))h_t(s)\left(\begin{smallmatrix}0\\ \tau^2\end{smallmatrix}\right) ds\right)$$

$$= \boldsymbol{M}_t\boldsymbol{\theta}, \tag{13}$$

where

$$\boldsymbol{M}_t = \left(\begin{smallmatrix}m_0 & m_1 & \cdots & m_{N+1}\end{smallmatrix}\right), \tag{14}$$

$$m_0 = \int_0^t \alpha\beta h_t(s)\left(\begin{smallmatrix}0\\ \tau^2\end{smallmatrix}\right) ds, \tag{15}$$

$$m_{i;0<i<N+1} = \int_0^t \Psi_{i-1}(z(s))h_t(s)\left(\begin{smallmatrix}0\\ \tau^2\end{smallmatrix}\right) ds. \tag{16}$$

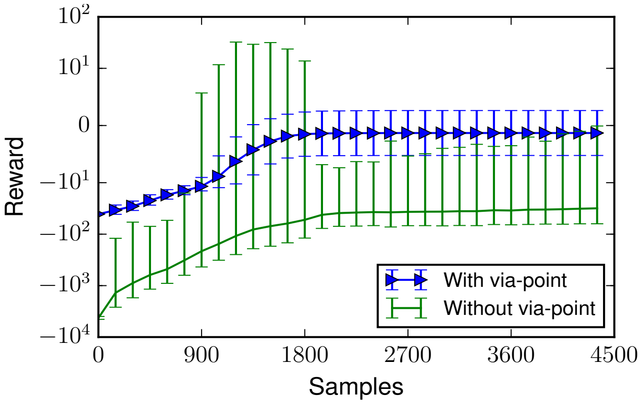


Figure 2: Rewards obtained over 20 trials during the learning process with and without via-points, plotted on a scale that is linear from -10 to 10 and logarithmic outside of that span. In 18 out of 20 trials, the learner with the via-point finds the opening in the box and obtains a reward close to zero. This results in a low standard deviation (error bars). Without the via-point the learning process converges to a local optimum.

Note that this equation is for a single dimension but can straightforwardly be extended to multiple dimensions by extending $\boldsymbol{M}$ diagonally and adding rows to $\boldsymbol{\theta}$ such that

$$\boldsymbol{y} = \begin{pmatrix}\boldsymbol{M}_t & \boldsymbol{0} & \cdots\\ \boldsymbol{0} & \boldsymbol{M}_t & \cdots\\ \vdots & \vdots & \ddots\end{pmatrix}\begin{pmatrix}\boldsymbol{\theta_0}\\ \boldsymbol{\theta_1}\\ \vdots\end{pmatrix},$$

where $\boldsymbol{\theta_j}$ denotes the parameters of the DMP for dimension j. Therefore for DMPs, we can write the likelihood distribution $p(y^*|t^*, \sigma_y\boldsymbol{I}, \boldsymbol{\theta})$ as

$$p(y^*|t^*, \sigma_y\boldsymbol{I}, \boldsymbol{\theta}) = \mathcal{N}(y^*|\boldsymbol{M}_{t^*}\boldsymbol{\theta}, \sigma_y\boldsymbol{I}). \tag{17}$$

Now we can calculate the sampling distribution as the posterior distribution of the likelihood $p(y^*|t^*, \sigma_y\boldsymbol{I}, \boldsymbol{\theta})$, encoding information about the via-point, and the prior distribution π which consists of the previously learned policy. For Gaussian distributions, the sampling distribution is therefore given as

$$\pi_H(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|A_H\boldsymbol{\Phi}, \Sigma_H), \tag{18}$$

$$\Sigma_H \doteq \left(\Sigma_{\pi(k)}^{-1} + \boldsymbol{M}_{t^*}^T\sigma_y^{-1}\boldsymbol{I}\boldsymbol{M}_{t^*}\right) \tag{19}$$

$$= \Sigma_{\pi(k)} - \Sigma_{\pi(k)}\boldsymbol{M}_{t^*}^T\left(\sigma_y\boldsymbol{I} + \right. \tag{20}$$

$$\left. \boldsymbol{M}_{t^*}\Sigma_{\pi(k)}\boldsymbol{M}_{t^*}^T\right)^{-1}\boldsymbol{M}_{t^*}\Sigma_{\pi(k)},$$

$$A_H \doteq \Sigma_H\left(\boldsymbol{M}_{t^*}^T\sigma_y^{-1}\boldsymbol{I}B + \Sigma_{\pi(k)}^{-1}A_{\pi(k)}\right). \tag{21}$$

Where the application of the Woodbury matrix identity in Eq. 20 allows for numerically stable computation of Σ_H. This sampling distribution can then be used in place of the policy in order to obtain the samples used in the next iteration of the policy search as described in Algorithm 1.

4. EXPERIMENTS

In section 3, we showed how to derive a modified sampling distribution based on via-points with timing information that are provided by a human demonstration. In this section, we first utilize a simulated robot arm to show that

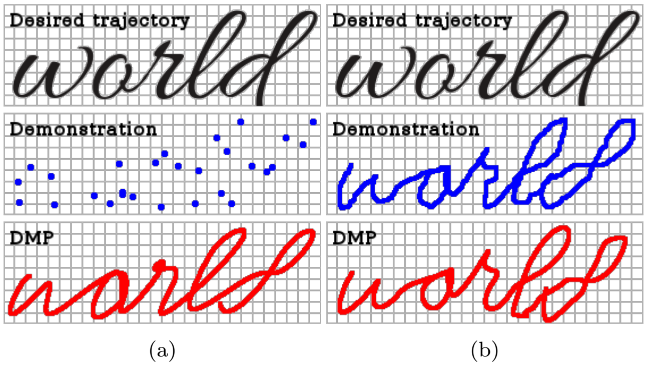


Figure 3: **a)** Mean trajectory of initial policy (bottom) derived from via-points (center). The trajectory is smooth and barely deviates from the desired trajectory (top) even in-between via-points. **b)** Mean trajectory (bottom) of initial policy derived from a continuous demonstration (center). Especially the last letter shows how the trajectory mimics imperfections of the demonstration.

such a modified sampling distribution can drastically improve the outcome of a reinforcement learner by having it learn how to insert an object in a hole while requiring only minimal information from the user. We then utilize a word reproduction task in order to provide a comparison to continuous demonstrations and to exhaustively analyze the key-properties of our algorithm.

4.1 Object Insertion with a Robot Arm

In our first experiment, we are evaluating the influence of a small number of demonstrated via-points on the reinforcement learning process and show that it can drastically improve the convergence of the policy search. To this end, we are utilizing a simulated 7 DoF robot arm and have it learn how to insert an object into a hole without any knowledge about the environment. We provide a single via-point (see Figure 1b) and show that this is sufficient to lead the learning process to the right solution which cannot reliably be found without a demonstration. To learn this task, we are utilizing our method with REPS as the underlying policy search algorithm to optimize the distribution over trajectories. These trajectories are represented by 7 Dynamic Movement Primitives with 6-dimensional weights leading to a 49-dimensional action vector $\boldsymbol{\theta}$. For evaluation, the outcome of the learning process is averaged over 20 trials with 30 policy search iterations per trial and 150 samples for each iterations. As can be seen in Figure 2, the reward curve observed by using the via-point is converging to a value close to 0 and therefore the distance of the trajectory to the goal position is converging to 0 as well. The learning process initialized with a zero mean policy, on the other hand, is converging to different values. This can easily be explained by the local optima that arise around the box, i.e. the robot is converging to solutions where the end-effector is pressing against the middle of other the sides of the box in order to get closer to the goal position. As a consequence, it stops exploring the box and does not find the opening. Note that the learning process always exceeds the initial reward obtained by our approach as the reinforcement learner always finds at least the closest points outside of the box which cannot be derived by the provided via-point alone, i.e. the provided via-point is further away from the box than the local optima.

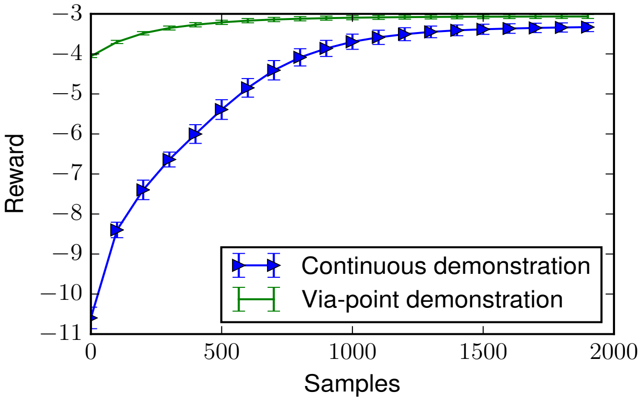


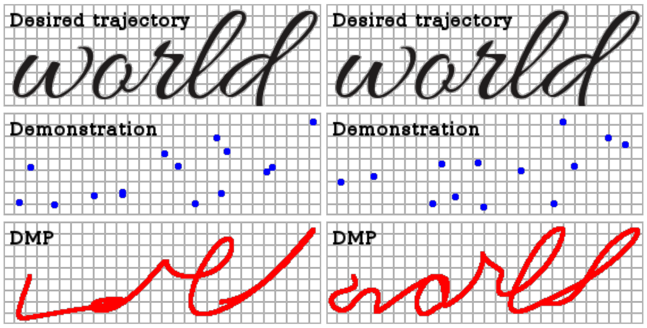
Figure 4: Comparison of reward obtained within 2000 samples, averaged over 30 trials. Error bars are showing the standard deviation. The reward function is defined as the minimum squared distance to the goal position. Learning initialized with via-points consistently outperforms the learning process initialized with a continuous demonstration.

4.2 Letter Writing

To further investigate the properties of this algorithm we are evaluating our approach on a letter reproduction task as well. Variations of this task have been used in the past to evaluate different properties of Dynamic Movement Primitives as it has many similarities with learning trajectories for robot arms while allowing for intuitive visualization, easy recording of demonstrations and thus good conditions for a comparison to conventional demonstrations as well as fast execution [10, 9]. The objective of the letter reproduction task is to learn trajectories for both dimensions which, when followed by a simulated pen, can accurately reproduce a sequence of letters. We are representing those trajectories as DMPs with a 60 dimensional weight-vector for each dimension and initialize the policy by a continuous demonstration or via-points before optimizing it using REPS. For the policy search, we defined the reward function as the number of overlapping black pixels: $-\frac{10^4}{\#IntersectingPixels+1}$. Note that solving this task requires the learning algorithm to learn trajectories that are far more complex than in the previous task while utilizing a sparser reward signal. First, we will compare our algorithm to learning trajectories from continuous demonstrations as a baseline, then we will evaluate the use of corrections after the learning process has started and finally we will show that linear models of via-points can be used to learn policies that can reproduce trajectories with different rotation and scaling factors.

4.2.1 Comparison to Continuous Demonstrations

The first instance of the letter reproduction task compares learning initialized with a normal distribution around a continuous demonstration to learning initialized on a fixed set of via-points at equi-distant points in time. In this experiment we show that despite the lack of information between via-points, our approach will converge to a more accurate solution in fewer iterations. To obtain an initial policy from a continuous demonstration, we use standard least squares



(a) (b)

Figure 5: **a)** Mean trajectory of the initial policy (bottom) derived by using only every second via-point (center). This trajectory omits whole letters when compared to the desired trajectory (top). **b)** Trajectory after 3 iterations (bottom), when the second half of the via-points have been added (center). The policy now shows the desired word and can be improved in future iterations. The "r" is not fully formed immediately after the correction due to learned behavior.

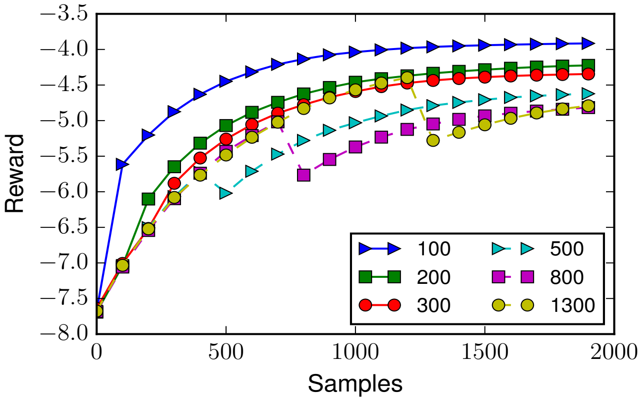


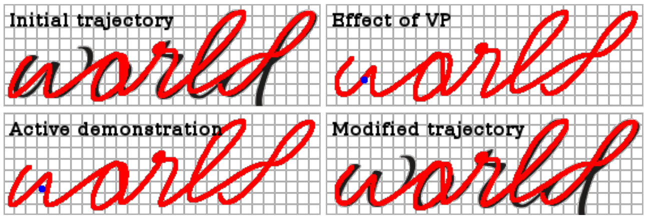
Figure 6: Comparison of rewards obtained with the rest of the via-points added after different numbers of samples. Early demonstrations are clearly better than late demonstrations but late demonstrations can still be effective.

to learn the mean and then add an initial variance of 10^3 for the weights and 5 for the goals of the DMPs. We have found these values to yield the best result in the continuous case. For policy search initialized with via-points, we start with a multivariate normal distribution with mean 0 and a variance of 10^5 for the weights and 500 for the goal positions. Note that the higher initial variance is necessary as conditioning on the via-points will otherwise lead to an overly narrow distribution with each added via-point decreasing the variance of the policy. This initial distribution is then used as the prior distribution to obtain a modified initial policy based on the via-points that can be seen in Figure 3a. In Figures 3a and b, it can be seen that the initial policy derived from the via-points is very smooth whereas the initial policy derived from a continuous demonstration mirrors the imperfections of that demonstration. Note that these imperfections are often much larger in practice as policy search is unnecessary in domains where given demonstrations already solve the task perfectly. Furthermore, the figures show that the errors introduced by missing information between the via-points is of the same order as the errors that can be introduced by linear regression and that the mean of the initial policy is already describing a good trajectory which leads to a fast learning process. Finally, Figure 4 shows that our approach yields both, higher initial reward as well as higher final reward and therefore better trajectories before and after the learning process, when compared to the baseline. Note that the higher initial reward is tied to the amount of exploration that is necessary in the beginning and can, in many cases, be a desirable property when it comes to safe exploration. While our approach only explores the areas in-between the via-points, a reinforcement learner that has been initialized with a continuous demonstration has to explore around the full trajectory. It is possible to reduce the exploration around the continuous demonstrations; however this would also reduce the final reward that can be obtained.

4.2.2 *Active Corrections*

One important aspect of the approach presented in this paper is that via-points can be provided at any time and can be used to modify an already trained policy. This is especially useful in situations where the optimal trajectory is not immediately apparent to the user. To investigate the impact of providing via-points at later iterations we are looking at a variation of the letter writing task where only every second of the initial via-points are provided from the start. Figure 5a shows that this constitutes a far worse initial policy and that we would expect large gains by providing the second half of the via-points. As can be seen in Figure 5b as well as in the reward curve in Figure 6, via-points provided at a later point can indeed improve the policy significantly.

However, while giving the via-points after some number of iterations can still cause a significant jump in reward, the size of this jump decreases for later iterations. After some time, the modified sample distribution will even decrease the performance of the learning process. This effect can be attributed to a mismatch of the chosen prior distribution, i.e. the current policy, with the optimal prior distribution which is the unknown policy according to which the human teacher is sampling his via-points. As the learning process converges to a sub optimal policy, it is impossible to sample trajectories that adhere to both, the learned policy as well as the specified via-points. Depending on the value of the variance parameter, the learner can then either sam-



(a) (b)

Figure 7: Example of giving via-points interactively. **a)** Original policy in comparison to the desired trajectory (top) and the via-point that has been provided as a correction (bottom). **b)** Mean of the modified policy. The trajectory goes through the via-point (top) and thereby matches the desired trajectory more closely (bottom).

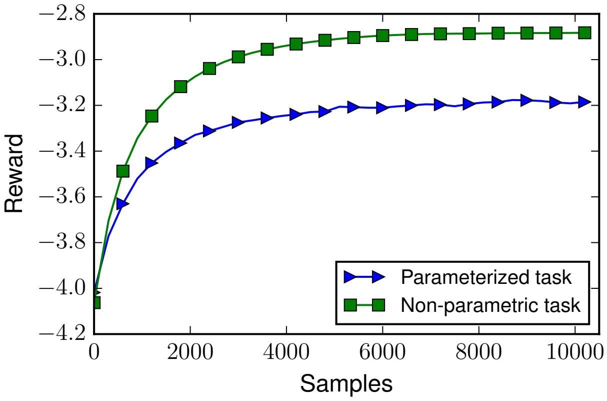


Figure 8: Reward obtained by contextual REPS shows that via-point models are an effective initialization. Rewards on the non-parametric task are displayed for comparison. The initial reward is identical due to the linear model and is improved upon significantly in both cases.

ple degenerate trajectories from low-probability areas of the current policy or ignore the given suggestion. The variance parameter is thus a measure of safety and specified how far the new policy can stray from the learned policy in order to adhere to the correction. To avoid this effect, via-points should always be given while the uncertainty in the current policy is relatively high in comparison to the deviation of the via-points from this policy. Note that in this experiment, the via-points are already known from the start even if the agent doesn't utilize them. This allowed us to analyze the effect of giving late demonstrations without having to account for the human factor. However, in practice, late demonstrations should be given depending on trajectories sampled from the current policy. This way, the user can actively shape the trajectory and guide the learning process to the right solution. We illustrate the process of providing via-points interactively and show the impact of a correction in Figure 7.

4.2.3 Evaluation of Learning with Linear Models

For the third experiment, we investigated the properties of learning a parametric generalization of the above task with a linear via-point model. We are looking at learning a model that can recreate words with respect to rotation and scaling of the target image. Note that similar kinds of parametric tasks can be found in practice, where the parameters often denote the location and orientation of an object. We are assuming that good features are known as this would be a necessity for obtaining a via-point model in practice. In this case we are using the feature vector $(s_x cos(\theta), s_y cos(\theta), s_x sin(\theta), s_y sin(\theta), 1)$ with s_x and s_y representing the scaling and lie between 0.6 and 1.4 whereas θ represents the rotation of the target image which lies between -45 and 45 degrees. The linear via-point models are then given w.r.t. these features and are used for learning a linear Gaussian policy using locally linear weighted regression and contextual REPS. Figure 8 shows that the initial policy adapts to the task parameters and that it can be used in conjunction with contextual REPS to obtain a similar reward curve for arbitrary rotations and scaling as for a single instance of the task.

5. CONCLUSION

In this paper we introduced an approach to utilize partial demonstrations to interactively guide the policy search process. We have shown that our approach of using soft via-points significantly outperforms continuous demonstrations when used to initialize the learning process. Furthermore, our results show that our approach can be used to guide the learning process in an interactive way, utilizing the knowledge gained from observing the robot to change specific aspects of the policy. Finally, we have shown that we can use linear models of via-points to generalize over variations of a task.

One key insight is that when applying our method during the learning process, the results are largely dependent on the covariance of the already learned policy. While sampling completely new trajectories ensures that the robot continues exploration and does not return to the original trajectory after reaching the via-point, it also requires a prior distribution that specifies sensible trajectories. In our approach, this distribution is given by the policy search. In the future, we plan to investigate other prior distributions based on different models of how humans give via-points to allow providing corrections to already converged policies. Furthermore, for each via-point the user is required to specify an exact point in time. We plan to relax this assumption and allow the user to specify distributions in time as the importance of preserving the timing is heavily dependent on the task. Finally, the method presented in this paper is based on the assumption that the policy is a Gaussian distribution. While this can be a reasonable assumption, there are cases where other distributions would be preferable. Multi-modal policies, for example, can be used to learn tasks with multiple solutions [5]. In the future, we would like to explore this avenue and extend our approach to different types of policies.

Acknowledgments

This work was conducted as a part of the OpenLabs project 1436618 sponsored by PSA Peugeot and partially funded under ONR grant number N000141410003.

APPENDIX

A. DERIVING $\pi_H(\boldsymbol{\theta})$ FOR ARBITRARY INITIAL POSITIONS AND VELOCITIES

In section 3, we derived a sampling distribution under the assumption that $y_0 = 0$ and $\dot{y}_0 = 0$. However, while $y_0 = 0$ can be assumed w.l.o.g., $\dot{y}_0 = 0$ is only true for trajectories that start from a rest position. Here, we derive an equivalent sampling distribution for the general case. In the general case, the closed form for dynamic movement primitives is given by

$$\begin{pmatrix} \boldsymbol{y} \\ \dot{\boldsymbol{y}} \end{pmatrix} = \int_0^t e^{(t-s)\begin{pmatrix} 0 & 1 \\ -\tau^2\alpha\beta & -\tau\alpha \end{pmatrix}} \begin{pmatrix} 0 \\ \tau^2(\alpha\beta g + \boldsymbol{\Psi}(z(s))^T \boldsymbol{w}) \end{pmatrix} ds + e^{t\begin{pmatrix} 0 & 1 \\ -\tau^2\alpha\beta & -\tau\alpha \end{pmatrix}} \begin{pmatrix} y_0 \\ \dot{y}_0 \end{pmatrix}. \quad (22)$$

And therefore

$$\boldsymbol{y} = \boldsymbol{M}_t \boldsymbol{\theta} + c \quad (23)$$

where

$$c \doteq \left(1 \ 0 \right) e^{t \begin{pmatrix} 0 & 1 \\ -\tau^2\alpha\beta & -\tau\alpha \end{pmatrix}} \begin{pmatrix} y_0 \\ \dot{y}_0 \end{pmatrix} . \tag{24}$$

The sampling distribution $p(\boldsymbol{\theta}|t^*, y^*) = \mathcal{N}(\boldsymbol{\theta}|A_H\boldsymbol{\Phi}, \Sigma_H)$ is then computed with the modified likelihood distribution

$$p(y^*|t^*, \boldsymbol{\theta}) = \mathcal{N}(y^*|\boldsymbol{M}_{t^*}\boldsymbol{\theta} + c, \sigma_y \boldsymbol{I}). \tag{25}$$

The mean of this distribution differs slightly from the distribution derived in section 3 so that

$$A_H\boldsymbol{\Phi} = \Sigma_H \left(\boldsymbol{M}_{t^*}^T \sigma_y^{-1} \boldsymbol{I} \left(B\boldsymbol{\Phi} - c \right) + \Sigma_{\pi(k)}^{-1} A_{\pi(k)} \boldsymbol{\Phi} \right). \tag{26}$$

Note that we can assume w.l.o.g. that $\boldsymbol{\Phi}$ is of the form $\boldsymbol{\Phi} = \left(\Phi_1 \ \Phi_2 \ \cdots \ 1 \right)^T$. The sampling distribution is then defined by

$$\Sigma_H \doteq \Sigma_{\pi(k)} - \Sigma_{\pi(k)} \boldsymbol{M}_{t^*}^T \tag{27}$$

$$\left(\sigma_y \boldsymbol{I} + \boldsymbol{M}_{t^*} \Sigma_{\pi(k)} \boldsymbol{M}_{t^*}^T \right)^{-1} \boldsymbol{M}_{t^*} \Sigma_{\pi(k)},$$

$$A_H \doteq \Sigma_H \left(\boldsymbol{M}_{t^*}^T \sigma_y^{-1} \boldsymbol{I} \left(B - \left(0 \ \ldots \ 0 \ c \right) \right) + \Sigma_{\pi(k)}^{-1} A_{\pi(k)} \right). \tag{28}$$

REFERENCES

[1] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz. Trajectories and keyframes for kinesthetic teaching: a human-robot interaction perspective. In *International Conference on Human-Robot Interaction*, pages 391–398, 2012.

[2] B. Argall, E. Sauser, and A. Billard. Policy adaptation through tactile correction. In *Annual Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB)*, 2010.

[3] S. Bitzer, M. Howard, and S. Vijayakumar. Using dimensionality reduction to exploit constraints in reinforcement learning. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 3219–3225, 2010.

[4] S. Chernova and A. L. Thomaz. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121, 2014.

[5] C. Daniel, G. Neumann, and J. Peters. Hierarchical relative entropy policy search. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.

[6] M. P. Deisenroth, G. Neumann, and J. Peters. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142, 2013.

[7] S. Griffith, K. Subramanian, J. Scholz, C. Isbell, and A. L. Thomaz. Policy shaping: integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2625–2633, 2013.

[8] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *International Conference on Evolutionary Computation (ICEC)*, pages 312–317, 1996.

[9] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Computation*, 25(2):328–373, 2013.

[10] A. J. A. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1547–1554, 2002.

[11] K. Judah, S. Roy, A. Fern, and T. G. Dietterich. Reinforcement Learning Via Practice and Critique Advice. *AAAI*, 2010.

[12] W. B. Knox and P. Stone. Reinforcement learning from simultaneous human and MDP reward categories and subject descriptors. In *Autonomous Agents and Multiagent Systems (AAMAS)*, pages 475–482, 2012.

[13] J. Kober, E. Oztop, and J. Peters. Reinforcement learning to adjust robot movements to new situations. In *Robotics: Science and Systems (RSS)*, 2010.

[14] J. Kober and J. Peters. Policy search for motor primitives in robotics. In *Advances in Neural Information Processing Systems (NIPS).*, pages 849–856, 2008.

[15] P. Kormushev, S. Calinon, and D. G. Caldwell. Robot motor skill coordination with EM-based reinforcement learning. In *Intelligent Robots and Systems (IROS)*, pages 3232–3237, 2010.

[16] A. G. Kupcsik, M. P. Deisenroth, J. Peters, and G. Neumann. Data-Efficient Generalization of Robot Skills with Contextual Policy Search. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2013.

[17] H. Miyamoto, S. Schaal, F. Gandolfo, H. Gomi, Y. Koike, R. Osu, E. Nakano, Y. Wada, and M. Kawato. A Kendama learning robot based on bi-directional theory. *Neural Networks*, 9(8):1281–1302, 1996.

[18] K. Mülling, J. Kober, O. Kroemer, and J. Peters. Learning to select and generalize striking movements in robot table tennis. *International Journal of Robotics Research*, 32(3):263–279, 2013.

[19] A. Paraschos, C. Daniel, J. Peters, and G. Neumann. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2616–2624, 2013.

[20] J. Peters, K. Mülling, and Y. Altun. Relative Entropy Policy Search. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 1607–1612, 2010.

[21] S. Schaal. Dynamic movement primitives - a framework for motor control in humans and humanoid robotics. In *Adaptive Motion of Animals and Machines*, pages 261–280. Springer Tokyo, 2003.

[22] F. Stulp and O. Sigaud. Path integral policy improvement with covariance matrix adaptation. In *International Conference on Machine Learning (ICML)*, pages 281–288, 2012.

[23] G. Teschl. *Ordinary differential equations and dynamical systems*, volume 140. American Mathematical Soc., 2012.

[24] E. Theodorou, J. Buchli, and S. Schaal. Learning policy improvements with path integrals. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 828–835, 2010.

[25] Y. Wada, Y. Koike, E. Vatikiotis-Bateson, and M. Kawato. A computational model for cursive handwriting based on the minimization principle. In *Advances in Neural Information Processing Systems (NIPS)*, pages 727–734, 1993.