Incremental Imitation Learning of Context-Dependent Motor Skills

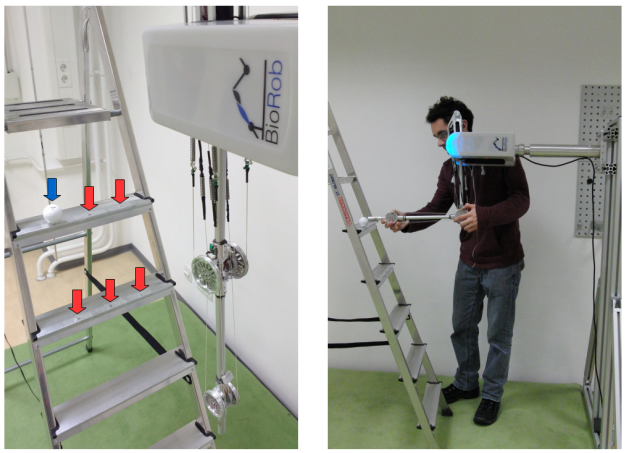
Marco Ewerton[1], Guilherme Maeda[1], Gerrit Kollegger[2], Josef Wiemeyer[2] and Jan Peters[1,3]

***Abstract*— Teaching motor skills to robots through human demonstrations, an approach called "imitation learning", is an alternative to hand coding each new robot behavior. Imitation learning is relatively cheap in terms of time and labor and is a promising route to give robots the necessary functionalities for a widespread use in households, stores, hospitals, etc. However, current imitation learning techniques struggle with a number of challenges that prevent their wide usability. For instance, robots might not be able to accurately reproduce every human demonstration and it is not always clear how robots should generalize a movement to new contexts. This paper addresses those challenges by presenting a method to incrementally teach context-dependent motor skills to robots. The human demonstrates trajectories for different contexts by moving the links of the robot and partially or fully refines those trajectories by disturbing the movements of the robot while it executes the behavior it has learned so far. A joint probability distribution over trajectories and contexts can then be built based on those demonstrations and refinements. Given a new context, the robot computes the most probable trajectory, which can also be refined by the human. The joint probability distribution is incrementally updated with the refined trajectories. We have evaluated our method with experiments in which an elastically actuated robot arm with four degrees of freedom learns how to reach a ball at different positions.**

I. Introduction

Since the beginning of the 1980s, a large amount of research has been done to make robots capable of learning motor skills by imitating human demonstrations. This strategy, called "imitation learning" [1], might become a viable route to quickly train general-purpose robots to perform new tasks on demand in environments such as households, stores, hospitals, etc.

However, a number of challenges hinder the widespread application of imitation learning in those environments. For example, the robot might not be able to accurately reproduce some of the movements demonstrated by the human and the robot should be able to generalize the motor skills it has learned so far to different contexts. In those cases, it may be helpful to structure the imitation learning as an incremental process. In this process, the human could incrementally refine the robot trajectories in order to make it accomplish a certain task or incrementally correct trajectories inferred by the robot in the face of new contexts. In this way, the human would not have to demonstrate a movement all over again when the robot had made a mistake. Instead, the human could just give small incremental feedbacks that would be interpreted by the robot as necessary changes to its movements and to the relation between movements and contexts.



(a) Experimental setup (b) Kinesthetic teaching

Fig. 1: Images of an experiment involving the BioRob, a 4 DoFs elastically actuated robot arm. The objective in this experiment is to teach the robot how to reach a ball at each of the positions pointed by the arrows. **(a)** The blue arrow points to the current position of the ball, while the red arrows point to other possible ball positions. **(b)** The human demonstrates through kinesthetic teaching how to reach the ball at a certain position.

Following this perspective, the main contribution of this paper is an algorithm that allows humans to teach robots context-dependent motor skills through demonstrations in an incremental manner. Our demonstrations have been obtained through kinesthetic teaching, i.e. by letting the human grasp and move the links of the robot. The proposed algorithm allows the human to incrementally refine the movement of the robot, a desirable feature if the robot cannot imitate the original demonstration of the human or if the movement executed by the robot does not yet solve the task at hand.

In our method, a joint probability distribution over refined trajectories and context variables is built. Having built this joint distribution, the robot is able to infer from previous experiences what movement it should execute to accomplish a certain task given a new context. The contexts used in our experiments have been in the form of via points. More generally, the context could for example be positions,

[1]M. Ewerton, G. Maeda and J. Peters are with the Intelligent Autonomous Systems group, department of Computer Science, Technische Universität Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany {ewerton, maeda, peters}@ias.tu-darmstadt.de

[2]G. Kollegger and J. Wiemeyer with the Institute of Sport Science, Technische Universität Darmstadt, Magdalenenstr. 27, 64289 Darmstadt, Germany {kollegger, wiemeyer}@sport.tu-darmstadt.de

[3]J. Peters is also with the Max Planck Institute for Intelligent Systems, Spemannstr. 38, 72076 Tübingen, Germany jan.peters@tuebingen.mpg.de

orientations or other properties of objects in the workspace of the robot. The human can also incrementally correct the inferences of the robot by intervening in its movements. Each new refined trajectory and context is used to update the joint distribution.

The algorithm proposed in this work is fairly general and may be applied to teach different motor skills to robots. In our experiments, this algorithm has been applied in a 2D problem to make a particle pass through certain via points, with the BioRob [2], an elastically actuated robot arm with four degrees of freedom (DoFs), to make it reach a ball at different positions and in a minigolf-like task involving the same robot.

The remainder of this paper is organized as follows: Section II presents related work. Section III introduces the proposed method for incremental imitation learning of context-dependent motor skills. Our method is explained starting with a procedure to learn trajectories for a single context. After that, the necessary extensions to deal with context-dependent motor skills are described. Section IV presents our experiments. Finally, Section V summarizes the paper and discusses ideas for future work.

II. Related Work

While in this work we have been specifically dealing with demonstrations through kinesthetic teaching, another form of imitation learning involves observation. In this case, the movements of a human are recorded by a camera or by a motion capture system. Those movements are mapped to the kinematics of the robot, which reproduces the demonstration or learns motion primitives from multiple demonstrations as in [3]. Usually, though, this mapping does not always produce the most preferable motions. For this reason, researchers have been investigating the idea of incremental imitation learning through kinesthetic teaching to refine the movements initially learned by the robot through observation.

Calinon and Billard [4] presented an approach based on Principal Component Analysis (PCA) and Gaussian Mixture Models (GMMs) to teach gestures to a humanoid robot. The gestures are demonstrated by a human in two ways: the human moves while sensors attached to his/her body record his/her movements or the human performs kinesthetic teaching by grasping and moving the arms of the robot. The robot learns new gestures from multiple human demonstrations and its movements can be incrementally refined by the human. When performing kinesthetic teaching, the human can decide which DoFs he/she wants to drive and which DoFs should be autonomously driven by the robot. The DoFs driven by the human are set in passive mode. In contrast, in our method, the human disturbs the movements of the robot without setting any DoFs in passive mode. In our work, the disturbances introduced by the human lead to changes in the behavior of the robot. The human does not need to demonstrate the whole movement of a DoF again in case it is not moving correctly yet. He/she just needs to apply incremental changes to this movement.

Lee and Ott [5] also presented a method to teach motor skills to a humanoid robot. In their method, first the robot observes the movements of a human. Subsequently, a human may incrementally refine the movement of the robot through kinesthetic teaching. Their work uses the concept of a "motion refinement tube", which is a region around the nominal trajectory followed by the robot where the controller has low stiffness, allowing the human to refine the trajectory. Movements are represented in their method by Hidden Markov Models (HMMs). Changes introduced in a trajectory by the human translate into changes in the parameters of the HMM representing that trajectory. The updated HMM generates an updated trajectory, accounting for the refinements introduced by the human. Their approach offers desirable properties, such as the possibility of defining the "refinement tube" in such a way that changes in the trajectory of one joint do not result in accidental disturbances in the trajectories of other joints. In contrast to our work, their approach requires an accurate model of the robot dynamics in order to identify human intervention. Our approach allows robots to learn context-dependent motor skills in the absence of an accurate model of the robot dynamics at the expense of having the robot execute a movement with and without human intervention at each iteration of the algorithm.

Another approach to incremental imitation learning aims at eliciting human preferences from the feedbacks he/she gives to the robot. Jain et al. [6] have developed a framework that enables robots to learn grocery checkout tasks with the help of human feedback. In their work, it is assumed that the user has an unknown score function quantifying the quality of trajectories in different contexts. This score function is a weighted sum of predefined features describing: 1) interactions between objects, 2) robot arm configurations, 3) orientation and temporal behavior of the object being manipulated, 4) interactions between the object being manipulated and the environment. The robot learns the weights for each of those features from human feedback. The user gives feedback by re-ranking a set of trajectories proposed by the robot or by changing waypoints of trajectories, setting the robot in zero-force gravity-compensation mode. In our method, when giving physical feedback to the robot, the user does not change specific waypoints. Instead, the user disturbs the movement of the robot while it is moving. By doing so, the user can directly interfere in the speed profile of the trajectory of the robot, teaching it for example to hit a golf ball faster or slower. Our work also differs from the cited one by computing a probability distribution between trajectories and contexts. While computation of the distribution requires modeling assumptions (Gaussians, GMMs, etc.), it does not require the design of a score function.

Akgun et al. [7] conducted a study evaluating kinesthetic teaching by demonstrating trajectories as well as kinesthetic teaching by demonstrating keyframes. They concluded that both trajectory and keyframes demonstration are viable methods to teach motor skills to humanoid robots and have their specific advantages. When keyframes were demonstrated, the trajectories were generated by splines connecting the

keyframes. They presented a way to demonstrate keyframes iteratively to the robot, but did not present a way to demonstrate trajectories iteratively. In this paper, we present a method for demonstrating trajectories iteratively. The method presented here might be of interest to future usability studies such as the one presented in [7].

The idea of incremental imitation learning has also been used in conjunction with tactile feedback to allow humans to teach grasp positioning and adaptation to robots [8], [9].

As explained in Section III-B, we have been using as part of our algorithm the framework of Probabilistic Movement Primitives (ProMPs) [10] for achieving generalization. This framework offers a straightforward way of inferring trajectories given contexts. Other methods such as task parameterized models discussed in [11] and [12] have very good generalization properties as well. In task parameterized models, demonstrations are observed from different frames. Different contexts are then represented by different positions and orientations of these frames. For example, by changing the position and orientation of an object of interest in the workspace of the robot, the position and orientation of the frame associated to this object also changes. The product of distributions over trajectories observed from different frames determines the trajectories that should be executed in a new context. The novelty of our work resides not in the generalization to different contexts in itself but in offering the human an intuitive way to teach trajectories to a robot in an incremental manner. The human feedback also changes how the robot responds to different contexts. The incremental learning aspect of this work could be combined with task parameterized models as well.

Reinforcement Learning methods, as in [13], have been used to make robots find successful movements for solving a given task and to generalize those movements to new situations. However, in the absence of a good model of the robot and of its environment, which would allow for optimization in simulation, it might take too long to find a successful policy with the real robot. In this case, our method may be helpful by allowing the human to influence the search for good policies.

III. Incremental Imitation Learning

In this section, our method for incremental imitation learning of context-dependent motor skills is explained. First, in Section III-A, a procedure is explained that allows a human to teach a trajectory to a robot through kinesthetic teaching and to incrementally introduce adjustments to the trajectory. Afterwards, Section III-B explains how this procedure can be combined with a probabilistic framework to provide the robot with the capability of learning context-dependent motor skills from human demonstrations and incremental refinements.

A. Incremental Imitation Learning of a Trajectory for a Single Context

The workflow depicted in Fig. 2 describes our procedure to incrementally teach a trajectory to the robot for a single context. First, an initial desired trajectory $\boldsymbol{\tau}_D$ is defined.

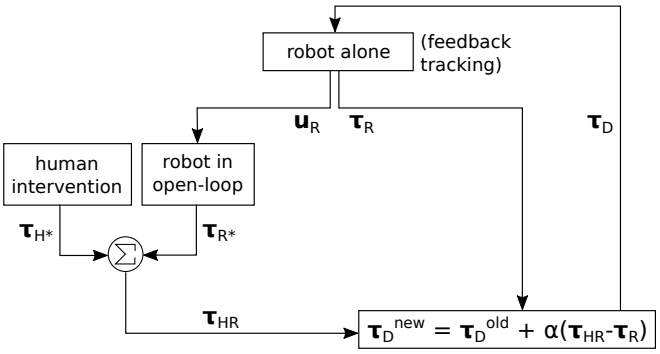


Fig. 2: Workflow of procedure to incrementally teach a trajectory to a robot for a single context.

This initial desired trajectory could have been defined by a vector of Cartesian or joint positions entered by the user or by kinesthetic teaching. We assume the existence of a feedback controller with potentially imperfect dynamics compensation. The robot tries to track $\boldsymbol{\tau}_D$, but executes in general a different trajectory $\boldsymbol{\tau}_R$. The sequence of forces or torques $\boldsymbol{u}_R$ generated by the controller at each time step is recorded. The robot returns then to its initial position and $\boldsymbol{u}_R$ is executed. While the robot executes $\boldsymbol{u}_R$, the human can disturb the trajectory of the robot. The resulting trajectory $\boldsymbol{\tau}_{HR}$ is recorded. We propose an update of the form

$$\boldsymbol{\tau}_D^{new} = \boldsymbol{\tau}_D^{old} + \alpha\left(\boldsymbol{\tau}_{HR} - \boldsymbol{\tau}_R\right), \tag{1}$$

where $\alpha \in [0, 1]$ is a scalar that defines how much the difference between the trajectories $\boldsymbol{\tau}_{HR}$ and $\boldsymbol{\tau}_R$ should change the desired trajectory $\boldsymbol{\tau}_D$. In our experiments, the term α has been set equal to 1. Smaller values of α decrease the update step, what might be useful to adjust for cases in which the human tends to exaggerate his feedback.

The update (1) resembles the format of iterative learning control (ILC) [14], [15]. Fundamentally, ILC learns tracking commands with respect to a given trajectory that the robot should execute. The challenge addressed by our method relates to the acquisition of the desired trajectory updates, which are assumed to be implicitly provided by human interventions. As in ILC, the control signal, in this case the desired trajectory $\boldsymbol{\tau}_D$ for the feedback controller, is updated according to an error, in our case $(\boldsymbol{\tau}_{HR} - \boldsymbol{\tau}_R)$. In other words, the difference between the trajectory that the robot executes without disturbance $\boldsymbol{\tau}_R$ and the trajectory that the robot executes with human disturbance $\boldsymbol{\tau}_{HR}$ indicates how the control signal $\boldsymbol{\tau}_D$ should be changed in order to perform the movement that the human wishes. While in ILC the reference trajectory used in the computation of the tracking error is predefined, in update (1) the reference trajectory $\boldsymbol{\tau}_{HR}$ changes at each iteration when the human is allowed to disturb the trajectory of the robot. The human decides when to stop disturbing the trajectory of the robot, in which case $\boldsymbol{\tau}_{HR} = \boldsymbol{\tau}_R$ and $\boldsymbol{\tau}_D^{new} = \boldsymbol{\tau}_D^{old}$, which means that the desired trajectory does not change anymore.

B. Learning Context-Dependent Motor Skills from Incremental User Feedback

This section presents our method for incremental imitation learning of context-dependent motor skills. This method is based on the procedure explained in the previous section and uses Probabilistic Movement Primitives (ProMPs) [10].

1) Probabilistic Movement Primitives: A ProMP is a probability distribution over trajectories. This probability distribution can be built from multiple demonstrated trajectories. In the ProMP framework, each demonstrated trajectory, which has a duration of T time steps, is approximated by a weighted sum of N normalized Gaussian basis functions. This approximation can be represented by the equation

$$\boldsymbol{\tau} = \boldsymbol{\Psi}\boldsymbol{w} + \boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{\epsilon}$ is a zero-mean i.i.d. Gaussian noise, i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \boldsymbol{I}_{TxT}\right)$, and

$$\boldsymbol{\Psi} = \begin{bmatrix} \psi_1(1) & \psi_2(1) & \cdots & \psi_N(1) \\ \psi_1(2) & \psi_2(2) & \cdots & \psi_N(2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(T) & \psi_2(T) & \cdots & \psi_N(T) \end{bmatrix}. \tag{3}$$

The terms $\psi_n(t)$ correspond to basis functions indexed by n and evaluated at time t. The centers of those basis functions are positioned at regular intervals along the time axis. The vector of weights $\boldsymbol{w} = [w_1, w_2, \cdots, w_N]^T$, containing the weight w_n for each basis function ψ_n, can be computed through linear least squares, according to

$$\boldsymbol{w} = \left(\boldsymbol{\Psi}^T\boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Psi}^T\boldsymbol{\tau}. \tag{4}$$

Once the weight vector $\boldsymbol{w}$ for each demonstrated trajectory $\boldsymbol{\tau}$ has been computed, a probability distribution $p(\boldsymbol{w})$ over weight vectors can be defined. In this work, $p(\boldsymbol{w})$ is assumed to be a Gaussian with mean $\boldsymbol{\mu_w}$ and covariance $\boldsymbol{\Sigma_w}$, i.e.

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{\mu_w}, \boldsymbol{\Sigma_w}). \tag{5}$$

With this assumption, it is possible to compute in closed form the probability distribution $p(\boldsymbol{\tau})$ over trajectories by integrating out the weight vectors $\boldsymbol{w}$ as in the equation

$$p(\boldsymbol{\tau}) = \int p(\boldsymbol{\tau}|\boldsymbol{w})\, p(\boldsymbol{w})\; d\boldsymbol{w}, \tag{6}$$

which results in

$$p(\boldsymbol{\tau}) = \mathcal{N}(\boldsymbol{\mu_\tau}, \boldsymbol{\Sigma_\tau}), \tag{7}$$

where

$$\boldsymbol{\mu_\tau} = \boldsymbol{\Psi}\boldsymbol{\mu_w}, \tag{8}$$

$$\boldsymbol{\Sigma_\tau} = \sigma^2\boldsymbol{I}_{TxT} + \boldsymbol{\Psi}\boldsymbol{\Sigma_w}\boldsymbol{\Psi}^T. \tag{9}$$

2) Modeling Context-Dependent Motor Skills: In this work, ProMPs are used to infer the most probable trajectory for a given context. In order to achieve this, a normal joint probability distribution over weight vectors $\boldsymbol{w}$ and the corresponding contexts, here represented by the vectors $\boldsymbol{c}$, is created,

$$p(\boldsymbol{w}, \boldsymbol{c}) = \mathcal{N}(\boldsymbol{\mu}_{joint}, \boldsymbol{\Sigma}_{joint}), \tag{10}$$

where

$$\boldsymbol{\mu}_{joint} = \begin{bmatrix} \boldsymbol{\mu_w} \\ \boldsymbol{\mu_c} \end{bmatrix}, \boldsymbol{\Sigma}_{joint} = \begin{bmatrix} \boldsymbol{\Sigma_{ww}} & \boldsymbol{\Sigma_{wc}} \\ \boldsymbol{\Sigma_{cw}} & \boldsymbol{\Sigma_{cc}} \end{bmatrix}.$$

Given a specific context $\boldsymbol{c}$, it is then possible to compute the conditional probability distribution

$$p(\boldsymbol{w}|\boldsymbol{c}) = \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{c}}, \boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{c}}\right), \tag{11}$$

where

$$\boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{c}} = \boldsymbol{\mu_w} + \boldsymbol{\Sigma_{wc}}\boldsymbol{\Sigma_{cc}}^{-1}(\boldsymbol{c} - \boldsymbol{\mu_c}), \tag{12}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{c}} = \boldsymbol{\Sigma_{ww}} - \boldsymbol{\Sigma_{wc}}\boldsymbol{\Sigma_{cc}}^{-1}\boldsymbol{\Sigma_{cw}}. \tag{13}$$

Next, the conditional probability distribution $p(\boldsymbol{\tau}|\boldsymbol{c})$ can be computed by solving the equation

$$p(\boldsymbol{\tau}|\boldsymbol{c}) = \int p(\boldsymbol{\tau}|\boldsymbol{w})\, p(\boldsymbol{w}|\boldsymbol{c})\; d\boldsymbol{w}, \tag{14}$$

which results in

$$p(\boldsymbol{\tau}|\boldsymbol{c}) = \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\tau}|\boldsymbol{c}}, \boldsymbol{\Sigma}_{\boldsymbol{\tau}|\boldsymbol{c}}\right), \tag{15}$$

with

$$\boldsymbol{\mu}_{\boldsymbol{\tau}|\boldsymbol{c}} = \boldsymbol{\Psi}\boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{c}}, \tag{16}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\tau}|\boldsymbol{c}} = \sigma^2\boldsymbol{I}_{TxT} + \boldsymbol{\Psi}\boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{c}}\boldsymbol{\Psi}^T. \tag{17}$$

3) Online Learning with Human Feedback: This section explains the proposed algorithm, which allows robots to learn in an online fashion context-dependent motor skills from human demonstrations and refinement. This algorithm uses the refinement procedure illustrated as a workflow in Fig. 2 in conjunction with the probabilistic modeling of context-dependent motor skills previously explained in Section III-B.2.

The robot starts with an initial joint probability distribution $p(\boldsymbol{w}, \boldsymbol{c})$ over weights $\boldsymbol{w}$ and contexts $\boldsymbol{c}$. Given this prior and a certain context $\boldsymbol{c}$, the robot computes $p(\boldsymbol{\tau}|\boldsymbol{c}) = \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\tau}|\boldsymbol{c}}, \boldsymbol{\Sigma}_{\boldsymbol{\tau}|\boldsymbol{c}}\right)$, as in Section III-B.2. The robot's desired trajectory $\boldsymbol{\tau}_D$ is set equal to the mean $\boldsymbol{\mu}_{\boldsymbol{\tau}|\boldsymbol{c}}$. The algorithm iterates over the refinement loop depicted in Fig. 2 as many times as the human wants. By the end of this iteration, the weights $\boldsymbol{w}$ of the new desired trajectory $\boldsymbol{\tau}_D$ are computed, using (4). This new weight vector $\boldsymbol{w}_M$ and the given context vector $\boldsymbol{c}_M$ are concatenated to form the vector $\boldsymbol{x} = \left[\boldsymbol{w}_M^T, \boldsymbol{c}_M^T\right]$, where M is the number of situations experienced so far. The joint distribution $p(\boldsymbol{w}, \boldsymbol{c}) = \mathcal{N}(\boldsymbol{\mu}_{joint}, \boldsymbol{\Sigma}_{joint})$ is then updated according to Welford's method for computing mean and covariance online [16]. According to this method, the mean $\boldsymbol{\mu}_{joint}$ is updated with

$$\boldsymbol{\mu}_{joint}^{new} = \boldsymbol{\mu}_{joint}^{old} + \frac{\left(\boldsymbol{x}^T - \boldsymbol{\mu}_{joint}^{old}\right)}{M} \tag{18}$$

and the covariance matrix $\boldsymbol{\Sigma}_{joint}$ with

$$\boldsymbol{S}^{new}(i,j) = \boldsymbol{S}^{old}(i,j) + \left(\frac{M-1}{M}\right)\left(\boldsymbol{x}(i) - \boldsymbol{\mu}_{joint}^{old}(i)\right)\left(\boldsymbol{x}(j) - \boldsymbol{\mu}_{joint}^{old}(j)\right), \quad (19)$$

$$\boldsymbol{\Sigma}_{joint}^{new}(i,j) = \frac{\boldsymbol{S}^{new}(i,j)}{M-1}, \quad (20)$$

where $\boldsymbol{S} = (M-1)\boldsymbol{\Sigma}$ is an auxiliary matrix.

Afterwards, the whole procedure is repeated for a new context. Algorithm 1 summarizes the proposed method.

Algorithm 1 Incremental Imitation Learning of Context-Dependent Motor Skills

1: Initialize $\boldsymbol{\mu}_{joint}$ and $\boldsymbol{\Sigma}_{joint}$ with a few demonstrations or with predefined values
2: **for** each new $\boldsymbol{c}$
3: compute $\boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{c}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{c}}$ (Eqs. 12 and 13)
4: compute $\boldsymbol{\mu}_{\boldsymbol{\tau}|\boldsymbol{c}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\tau}|\boldsymbol{c}}$ (Eqs. 16 and 17)
5: $\boldsymbol{\tau}_D = \boldsymbol{\mu}_{\boldsymbol{\tau}|\boldsymbol{c}}$
6: **refinement loop** (until human decides to stop)
7: robot tracks $\boldsymbol{\tau}_D$ with feedback controller ($\boldsymbol{\tau}_R$ and $\boldsymbol{u}_R$ are recorded)
8: robot executes $\boldsymbol{u}_R$ in feedforward and human is allowed to disturb the trajectory ($\boldsymbol{\tau}_{HR}$ is recorded)
9: $\boldsymbol{\tau}_D^{new} = \boldsymbol{\tau}_D^{old} + \alpha(\boldsymbol{\tau}_{HR} - \boldsymbol{\tau}_R)$
10: **end**
11: $\boldsymbol{w}_M = \left(\boldsymbol{\Psi}^T\boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Psi}^T\boldsymbol{\tau}_D$
12: update $\boldsymbol{\mu}_{joint}$ and $\boldsymbol{\Sigma}_{joint}$ (Eqs. 18, 19 and 20)
13: **end**

IV. Experiments

This section presents two experimental evaluations of the proposed algorithm. First, a simple 2D problem is presented. In this problem, the human can incrementally teach trajectories to the learning system, which can infer what to do in the face of new contexts. Afterwards, experiments are described that show the applicability of our method for teaching motor skills to a real robot.

In all the experiments described in this paper, the number N of Gaussian basis functions, introduced in Section III-B.1, was 20. This value was determined empirically by increasing the number of basis until the trajectories could be approximated to a desired level of detail.

A. 2D Problem

We designed a simple 2D problem in order to facilitate the understanding of the algorithm and the visualization of results. In this problem, a particle moves with constant velocity in the x-direction. Initially, its velocity in the y-direction is zero. The user can introduce an acceleration in the y-direction by pressing the keys "up" or "down" on the keyboard. The objective is to teach the system a trajectory that passes through two via points. The y-coordinates of each via point constitute in this case the context, which is represented by the vector $[y_1, y_2]^T$, where y_1 is the y-coordinate of the via point located at $x = 2$ and y_2 is the y-coordinate of the via point located at $x = 3.6$.

After being trained for a number of different configurations of the via points, the system should be able to infer the right trajectory to pass through the via points both in the configurations that it has already experienced as well as in a previously unseen configuration. The human can iteratively apply disturbances to the trajectory of the particle. This way, the human can correct the trajectory initially inferred by the system for a given context. See Fig. 3 for an illustration of this 2D problem.

In this problem, $\boldsymbol{u}_R$ is a time-indexed sequence of forces in y direction, while the trajectories $\boldsymbol{\tau}_R$, $\boldsymbol{\tau}_{HR}$ and $\boldsymbol{\tau}_D$ are time-indexed sequences of (x, y) positions.

Fig. 4 shows the prior and the posterior probability distributions over trajectories after the human had trained the system for two contexts. In this simple problem, after experiencing only two examples, the system could already infer trajectories that passed close to the via points. The progress in the generalization ability of the system is depicted in Figs. 5 and 6. Fig. 5 shows the prior and the posterior over trajectories without any training, after the human had trained the system for one context and so on until the system had previously observed eight different contexts. Fig. 6 shows the mean squared error

$$MSE = \frac{1}{2}\left(\left(y(2) - y_1\right)^2 + \left(y(3.6) - y_2\right)^2\right) \quad (21)$$

of the trajectories executed by the particle when observing each context for the first time. Here $y(2)$ and $y(3.6)$ are the y-coordinates of the particle when $x = 2$ and $x = 3.6$ respectively.

The MSE for the first context is 100, because the trajectory is 10 units apart from each via point. After the human has taught the trajectory for two different contexts through the refinement loop, the MSE of the trajectories executed by the particle for previously unseen contexts already drops to almost zero.

B. Real Robot Experiments

We tested our algorithm in experiments with the BioRob [2], a 4-DoF elastically actuated robot arm. It is a biologically inspired robot whose cables and springs roughly mimic the functionality of antagonistic pairs of muscles. The BioRob is light and compliant, nevertheless challenging to model and control[1].

In these experiments, the objective of the human was to teach the robot how to reach a ball positioned on steps of a ladder as shown in Fig. 1. We defined six different positions for the ball (see Fig. 7). We chose to represent each position by two values, expressing the two different heights with respect to the ground and the three different left-to-right positions. This choice was motivated by the fact that the

[1]The robot is controlled by a proportional-derivative (PD) controller with feedforward terms to compensate for gravity and friction/stiction. In these experiments, only three DoFs are actually being used.

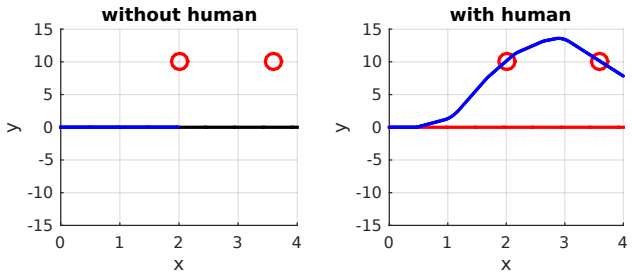


Fig. 3: 2D problem. **Left plot**: the red circles represent two via points through which the trajectory should pass; the black line represents the trajectory being currently tracked by the particle; the blue line represents the trajectory executed by the particle until the current instant. **Right plot**: the red circles represent the same two desired via points; the red line represents the trajectory that the particle would execute without any human interference; the blue curve represents the trajectory executed by the particle with human interference.

relations between the values assumed by the context variables are important, not their absolute values. In this setup, there was no camera detecting the ball. The values representing the different contexts were manually provided to the robot.

In our experiments with the BioRob, $\boldsymbol{u}_R$ is a time-indexed sequence of torques, while the trajectories $\boldsymbol{\tau}_R$, $\boldsymbol{\tau}_{HR}$ and $\boldsymbol{\tau}_D$ are time-indexed sequences of joint angles. By working with trajectories in joint space, we avoid any possible problems related to kinematic redundancy that would need to be addressed if our trajectories were sequences of Cartesian end effector positions.

In order to initialize the prior probability distribution $p(\boldsymbol{w}, \boldsymbol{c})$ over weight vectors and contexts, the refinement loop depicted in Fig. 2 was executed for four different positions of the ball: "top left" represented by the vector $[1, 1]^T$, "top right" represented by $[1, 3]^T$, "down left" represented by $[2, 1]^T$ and "down right" represented by $[2, 3]^T$.

After building the prior, the for loop described in Algorithm 1 was executed. The robot computed the most probable trajectory $\boldsymbol{\tau}$ for the context "top middle" represented by $[1, 2]^T$. The human improved the trajectory of the robot through the refinement loop until the robot could successfully reach the ball at the "top middle" position. After the human decided to quit the refinement loop, the joint probability distribution $p(\boldsymbol{w}, \boldsymbol{c})$ over weights and contexts was updated according to (18), (19) and (20). Then the robot used its updated prior to compute a trajectory for the context "down middle" represented by $[2, 2]^T$.

Having built the prior based on trajectories that reach the ball at each of the four corner positions, the robot was able to compute trajectories to reach or pass close to the ball at the two previously untrained middle positions. Using our refinement loop, the human could correct the trajectories inferred by the robot. In a second pass of the algorithm over all the six positions, the robot was able to reach the ball every time without needing any further refinement from the human. Please refer to our accompanying video for a recording of

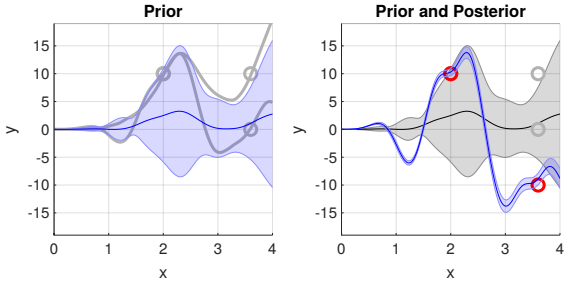


Fig. 4: Prior and posterior over trajectories for the 2D problem after the human had trained the system for two contexts. **Left plot**: the light gray curves represent the trajectories executed by the particle after refinement provided by the human for two different configurations of the via points, represented by the light gray circles; the blue curve represents the prior mean and the blue shade represents two standard deviations of the prior. **Right plot**: the configuration of the via points represented by the red circles is different from the configurations that have been observed so far by the system, which are represented by the light gray circles; the black curve represents the prior mean and the gray shade represents two standard deviations of the prior (the prior in this plot is the same as in the left plot); the blue curve represents the posterior mean and the blue shade represents two standard deviations of the posterior.

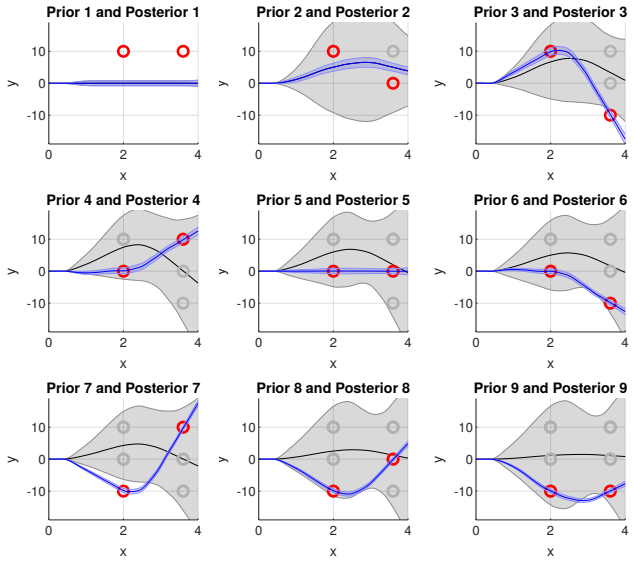


Fig. 5: Prior and posterior (with mean and two standard deviations) over trajectories given contexts without any training (Prior 1 and Posterior 1), after training for one context (Prior 2 and Posterior 2) and so on until the system had previously observed eight different contexts (Prior 9 and Posterior 9). Only the two first contexts needed refinement from the human. After that, the system was already able to infer trajectories intercepting the via points.

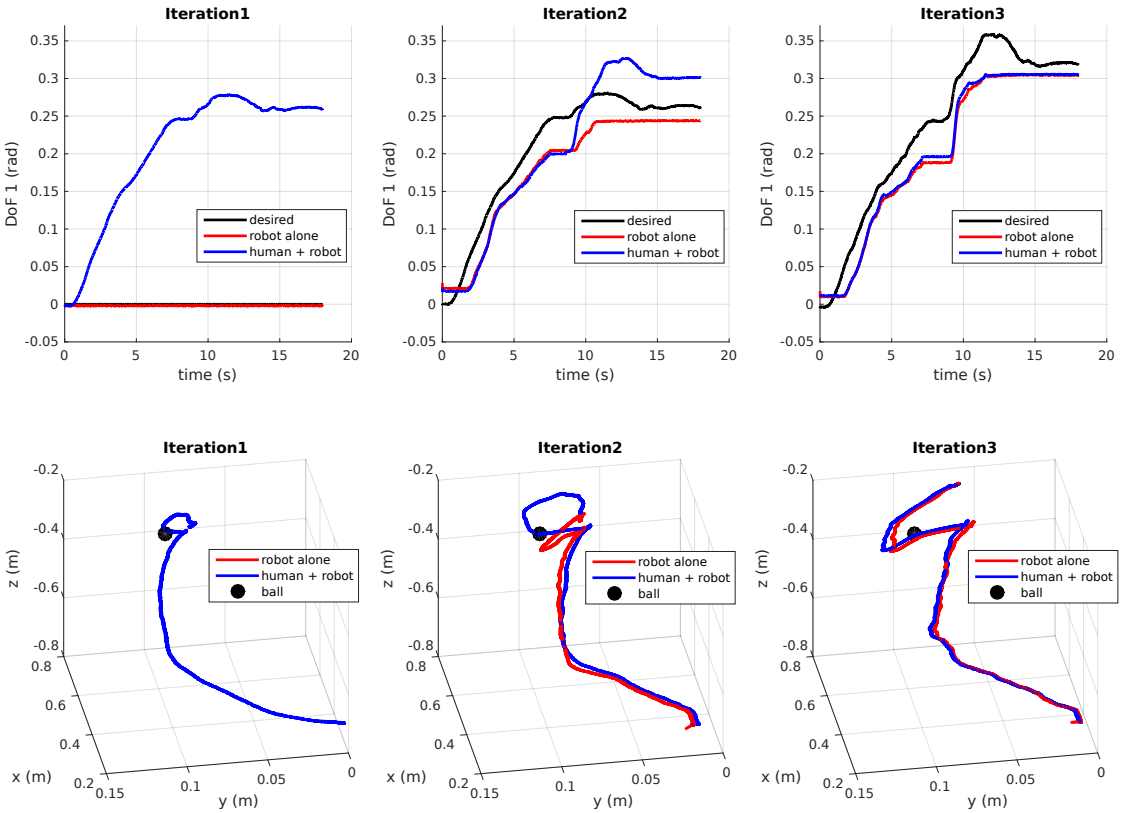


Fig. 8: Three iterations of the refinement loop. **Upper row**: Trajectories in joint space of the 1^{st} DoF of the BioRob. **Lower row**: The correspondent end effector trajectories in Cartesian space.

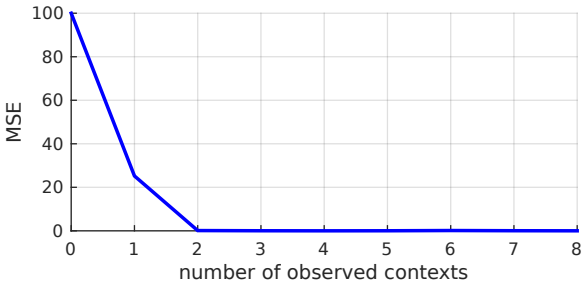


Fig. 6: Mean squared error (MSE) of trajectory executed by the particle when the system is observing each context for the first time.

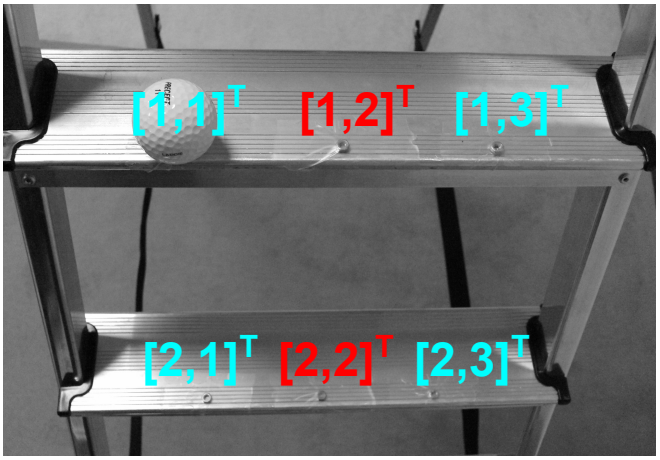


Fig. 7: Ball positions in an experiment involving the real robot. The positions represented by the vectors written in blue are the contexts that were used to initialize the prior probability distribution $p(\boldsymbol{w}, \boldsymbol{c})$ over weight vectors and contexts. Having built this prior, the robot infers trajectories to reach the ball at the positions represented by the vectors written in red.

the full experiment.

Fig. 8 shows three iterations of the refinement loop to teach the robot how to reach the ball at the position "top left" when the robot had no previous experience. The figure shows the trajectories in joint space of the 1^{st} DoF of the BioRob and the corresponding end effector trajectories in Cartesian space. In the first iteration, the robot arm simply hung down. Then, the human demonstrated through kinesthetic teaching how to reach the ball. In the second iteration, the robot tried to track the new desired trajectory. Since the controller of the robot is not able to track a desired trajectory accurately, the robot only passed close to the ball, but still

did not reach it (see the middle plot in the lower row). The human refined this trajectory further by interfering in the movement of the robot. In the third iteration, the robot was able to reach the ball alone and the human did not give any further feedback. The small differences between the red curve (trajectory generated by feedback tracking) and the blue curve (trajectory generated by reproducing the sequence of torques generated by the feedback tracking controller) are, in the third iteration, due to small errors in the repeatability of the robot.

We also performed experiments in which the human taught a robot how to putt in a minigolf-like task. In this case, there was only one context. In the phases in which the human was allowed to disturb the trajectory, the human kept his hands close to the robot and applied forces to its end effector when he judged necessary. After a few iterations, the desired trajectory $\boldsymbol{\tau}_D$ tracked by the feedback controller was such that it led to the robot being able to sink the ball. This experiment was recorded and is also included in the accompanying video.

V. Conclusion and Future Work

This paper presented an algorithm to allow humans to incrementally teach robots context-dependent motor skills. This algorithm is particularly relevant when the robot cannot track desired trajectories accurately or when trajectories initially computed by the robot given a new context do not solve the task at hand. In those cases, our algorithm offers the human an intuitive way of refining the trajectories of the robot. Moreover the refined trajectories and the new contexts are used to update the probability distribution used by the robot to compute trajectories given contexts. A 2D problem and experiments involving a 4-DoF elastically actuated robot arm demonstrate the effectiveness of the proposed algorithm.

As it is, the presented method for the human to teach the robot is not suitable when the robot moves very fast, is too heavy or manipulates a dangerous object. For those cases, an alternative way for the human to introduce changes in the trajectory of the robot would be necessary. Instead of physically interacting with the robot while it moves, the human could for instance use a graphical interface to change the trajectory or use teleoperation.

In this work, we modeled the joint probability distribution $p(\boldsymbol{w}, \boldsymbol{c})$ over weights and context variables as a single Gaussian. This model entails that $\boldsymbol{w}$ and $\boldsymbol{c}$ are linearly correlated, which is a reasonable assumption for the simple tasks we have dealt with so far. On the other hand, in a task such as playing table tennis, in which the robot would have to execute forehand and backhand strokes, this assumption would probably not hold. In order to deal with those cases, an alternative would be to model $p(\boldsymbol{w}, \boldsymbol{c})$ as a Gaussian Mixture Model.

The contexts addressed so far in our work have been only in the form of via points. Other possible contexts could be the weight or the size of objects manipulated by the robot, goal position of an object thrown by the robot, positions and orientations of objects in the workspace, etc. For the simple contexts evaluated so far, the human could teach the robot in a few iterations how to solve the task at hand and the generalization capabilities of the algorithm also helped reducing the amount of human intervention needed. Further evaluations shall be made in order to determine if the human can successfully teach skills with other types of context as well.

VI. Acknowledgments

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

References

[1] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[2] T. Lens and O. von Stryk, "Design and dynamics model of a lightweight series elastic tendon-driven robot arm," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 4512–4518.

[3] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *International Journal of Robotics Research*, vol. 31, no. 3, pp. 330–345, 2012.

[4] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2007, pp. 255–262.

[5] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Autonomous Robots*, vol. 31, no. 2-3, pp. 115–131, 2011.

[6] A. Jain, B. Wojcik, T. Joachims, and A. Saxena, "Learning trajectory preferences for manipulators via iterative improvement," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 575–583.

[7] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz, "Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2012, pp. 391–398.

[8] B. D. Argall, E. L. Sauser, and A. G. Billard, "Tactile guidance for policy refinement and reuse," in *Proceedings of the International Conference on Development and Learning (ICDL)*. IEEE, 2010, pp. 7–12.

[9] E. L. Sauser, B. D. Argall, G. Metta, and A. G. Billard, "Iterative learning of grasp adaptation through human corrections," *Robotics and Autonomous Systems*, vol. 60, no. 1, pp. 55–71, 2012.

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

[11] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 610–616.

[12] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[13] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, "Reinforcement learning to adjust parametrized motor primitives to new situations," *Autonomous Robots*, vol. 33, no. 4, pp. 361–379, 2012.

[14] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.

[15] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[16] B. Welford, "Note on a method for calculating corrected sums of squares and products," *Technometrics*, vol. 4, no. 3, pp. 419–420, 1962.