Learning from Demonstration and Correction via Multiple Modalities for a Humanoid Robot

Brenna Argall* Aude Billard*

(*) Learning Algorithms and Systems Laboratory (LASA),
École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
E-mail: brennadee.argall@epfl.ch, aude.billard@epfl.ch

Abstract

This paper reports ongoing work that employs multiple demonstration modalities in order to accomplish motion control learning in a multi-staged policy adaptation process. A novel interface for providing tactile guidance to correct learned motion control behaviors is introduced. This interface extends our prior work by making use of a more sophisticated set of tactile sensors, developed by the ROBOSKIN consortium.

1 Introduction

The challenge of developing paradigms for robot motion control in increasingly complex environments has prompted the emergence of alternatives to more traditional approaches to control. One such alternative is to demonstrate a behavior to the robot, and from the resultant dataset use machine learning routines to derive a control policy. The modality used to transfer demonstration information to the learner is key under such a paradigm, since due to differences in correspondence for the actuation or sensing, the human teacher demonstrations may not directly map to the robot learner.

Correspondence issues are minimized when the learner records directly from its own sensors while under the control of the teacher. For example, under teleoperation the teacher remotely controls the robot platform (e.g. [12]), while under kinesthetic control the teacher touches the robot to guide its motion (e.g. [7]). Teleoperation requires an interface for the direct control of all degrees of freedom, while kinesthetic teaching requires a (passive or active) responsiveness to human touch, for example back-drivable motors or force-torque sensing in the joints. Whether or not the teacher receives any sensor feedback while executing a demonstration (e.g. haptic feedback when grasping an object) can also play a role in the quality of the examples recorded within the training dataset.

Despite the many gains of demonstration learning [2, 6], there are potential drawbacks. One is the aforementioned issue of teacher-learner correspondence. Others include deficiencies in the demonstration interface, suboptimality of the human demonstrator and dataset sparsity. Continuing to adapt a policy after demonstration, based on execution experience, can allow the learner to address potential drawbacks. For example, within demonstration-based paradigms execution experience is used to update reward-determined state values [10] and learned state transition models [1], while other approaches provide more demonstration data [8, 9]. Another practical technique is to reuse policies in multiples behaviors, for example by decomposing demonstrations into a library of motion primitives [5].

The feedback signals used for policy adaptation might be computed automatically from the environment [1, 5, 10] or come from further teacher instruction [7, 8, 9]. In the latter case, it can be beneficial for the teacher to use a different modality than the demonstration interface when providing execution feedback, if interface deficiences contributed to dataset limitations.

The feedback signals employed in our approach consist of policy corrections provided through a tactile interface located on the body of the robot. In addition to providing an alternate modality to the demonstration interface (teleoperation), we argue that tactile instruction is a form of information transfer already employed by humans when teaching other humans, and that tactile sensing furthermore can be of crucial importance for safe robot operation around humans [4].

The following section presents our touch-based feedback algorithm and its implementation on a high degree-of-freedom (DoF) humanoid. Section 3 introduces the tactile correction interface that makes use of an array of sensing nodes. Conclusions and a discussion of ongoing work are then provided in Section 4.

2 Approach and Implementation

Here our touch-based feedback algorithm is overviewed, along with the hardware of our humanoid validation platform and its tactile sensors.

2.1 Algorithm Overview

The *Tactile Policy Correction (TPC)* algorithm offers an approach for the adaptation of a demonstrated policy, using tactile feedback from a human teacher [3]. Corrections are provided in order to accomplish two goals (Fig. 1). The first goal is to *refine* a policy during execution, and thus to improve its performance based on execution experience. The second goal is to assist in policy *reuse*, by guiding an existing policy towards accomplishing a different task.

During the first phase of operation demonstration data, consisting of sequences of poses and their time components, is gathered via teleoperation. A policy is derived from the data using *GMM-GMR* [7], which first encodes demonstrations in a *Gaussian Mixture Model (GMM)* via *Expectation-Maximization (EM)* parameter estimation and then predicts a target pose through *Gaussian Mixture Regression (GMR)*.

In the second phase of operation, the robot executes with the learned policy as the teacher provides tactile corrections. Here corrections indicate relative translational and rotational adjustments to the end-effector pose, and thus to the policy predictions. The robot responds online and the resulting trajectory is treated as new training data, added to the demonstration set, and the policy rederived.

For our initial empirical validations of the TPC algorithm [3], the tactile correction interface consisted of *Ergonomic Touchpads* encircling the wrist of a manipulator arm, with validation on grasp positioning tasks. Comparisons to policies derived from solely teleoperation demonstration confirmed policy reuse to be an effective mechanism for transferring domain knowledge, and policy refinement to be more successful at improving performance. Moreover the different teaching modalities - namely, teleoperation demonstration and tactile correction via the touchpad interface - were found to be individually better suited for information transfer in different areas of the state space.

Feedback from the touchpad interface was however somewhat limited in comparison to more sophisticated tactile sensors, for example that provide an indication of contact force or finer spatial resolution. Furthermore, in practice a response lag during corrective repositioning necessitated pausing the policy execution while corrections were being given. Here we report on the development of a novel tactile interface, that makes use of a more sophisticated tactile sensor.

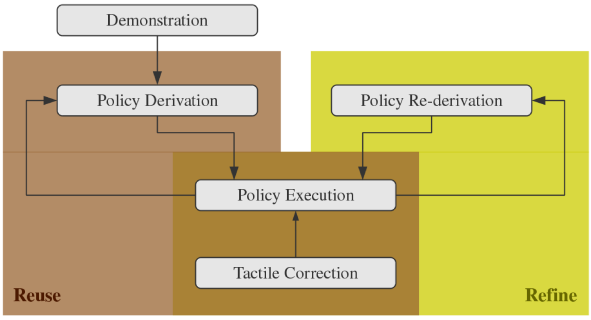

Figure 1: Flow overview of the Tactile Policy Correction algorithm for refinement and reuse.

2.2 Robot Hardware

Our robot platform and tactile interface consists of the 53-DoF iCub humanoid [13] and a sensor array skin developed under the ROBOSKIN consortium (`www.roboskin.eu`).

The first modality used for information transfer under our approach is teleoperation, which is non-trivial for the iCub arm as it requires simultaneous control of 7 degrees of freedom. Sensing units from the commercial $XSens$ joint recording system are placed on the human teacher's upper and lower arm, and back of the hand (Fig. 2, left). Each unit contains an accelerometer, gyroscope and inertial sensing unit, and provides orientation information that we translate into human joint angles and then map to the joint angles of the robot arm. Remote control is thus accomplished, during which the robot records from its own sensors.

The advantages of this teleoperation system is the ability to control many degrees of freedom simultaneously and cover large areas of the execution space quickly. The disadvantages however are that to achieve fine motor control with this system is difficult, and that no haptic feedback is provided to the human user.

The second modality used for information transfer under our approach is tactile correction, by having the robot actively respond to pose corrections indicated through the ROBOSKIN sensors (Fig. 2, right). Capacitive sensing nodes arranged on triangular taxels (12 nodes per taxel) and covered in silicone foam provide response signals that scale with contact force (sensor node output range: [0-255]). The forearm of the iCub arm is covered in 30 such taxels, with a total of 360 sensor channels. In our work the sensors are read and processed at a rate of $10Hz$. To realize a tactile correction, the outputs of these channels are mapped to small

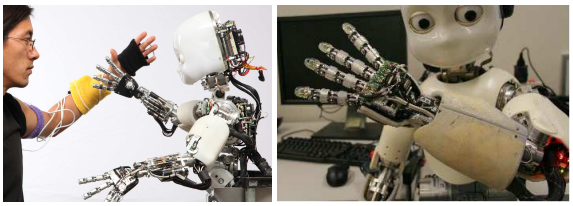

Figure 2: Teleoperation (left) and the ROBOSKIN sensor array (right), on the iCub robot.

changes in end-effector orientation or position (Sec. 3).

This correction mechanism is suitable for providing information at the level of fine motor control, and furthermore addresses potential correspondence issues in the demonstration interface by having the human directly touch the robot during execution.

3 Tactile Correction Interface

One challenge to developing a corrective movement mapping for the ROBOSKIN sensor channels is that the exact positions of the individual sensor nodes are not known. To interpret the direction of a touch, and infer from it an indicated correction, therefore is not immediate. Our approach addresses this challenge by learning a mapping from sensor readings to touch classification.

3.1 Interactive Data Collection

Learning begins by gathering a dataset of sensor readings while having a human touch the robot according to 8 movement classes. The classes are defined by constraints on end-effector motion as dictated by the robot kinematics: 6 for each direction within Cartesian space (forward, back, up, down, in, out), and 2 for rotation about the forearm axis (roll out, roll in). A null label is furthermore assigned to datapoints recorded prior to the initiation, and after the cessation, of a detected contact sequence. Note that, due to the location of the skin sensors on the forearm only, some of the labels (up, down, in, out) might be inferred from the contact normal and thus a single contact position, while others (front, back, roll out, roll in) require information from a sequence of contact positions in order to infer the direction of the correction.

The data gathering procedure is interactive. Namely, the robot moves for a short period of time in one of the movement class directions, and the human teacher is then prompted to touch the robot in a manner that s/he believes would have encouraged such a movement. The process is then repeated for each movement class. Multiple examples of presses or swipes to the arm (each of which consists of a sequence of datapoints) are gathered for each movement class; in practice the number of examples is set by the teacher and may vary between movement classes. Data gathering is relatively quick: for example under 5 minutes to gather 100 example swipes that result in $6,000$ datapoints. The raw data is then slightly preprocessed to remove sensor channels that were never activated during data gathering (e.g. 80 of 360 channels); this channel mask is then also applied at runtime. The result is a dataset of vectors $\boldsymbol{v} \in \mathbb{R}^N$, $N \leq 360$ of raw sensor values, each of which is associated with a single movement label.

3.2 Feature Computation

Features are computed from the sensor values prior to their mapping to a movement label. The feature computation aims to accomplish two goals: address spurious sensor readings and encode a notion of a swipe trajectory (Description 1). In particular, Step 1 addresses spurious sensor readings, by subtracting off an estimate on the sensor drift. Step 2 encodes a notion of the trajectory taken by a touch, by remembering past sensor values and thus effectively providing a snapshot of the trace across the sensor nodes. With such a mechanism, it is possible to encode a notion of the swipe sequence without knowing the physical positional relationship between sensor nodes.

A directed parameter search found classifier performance to depend heavily on the feature computation parameters, with classification accuracy ranging from $80.2 \pm 0.2\%$ to $95.6 \pm 0.7\%$ (10-folds cross validation, $h \in [5, 15]$, $\alpha_w \in [0, 1]$, $\alpha_v \in [0.0001, 1]$). Moreover, the feature computations were found to be essential to achieving good classification accuracy.

3.3 Prediction Bias

A classifier, able to map computed features to movement labels, is next learned from the training data. A Self Organizing Map (SOM) [11] is particularly well-suited to this learning problem, since it is able to map from a high-dimensional space to a low dimensional space while preserving a notion of the high dimensional topological relationships between points; and our sensor nodes, and therefore their readings, are indeed spatially correlated. The learning input is a (N-dimensional) vector of sensor features, and the output is a node within the (2-dimensional) SOM map structure and its associated label, which is used as the label for the prediction.

At run time, the SOM predictions additionally are post-processed to bias towards matching recent past and

Description 1 Feature Computation

Parameters :
  $\boldsymbol{v}^t \in \mathbb{R}^N$: current senor reading
  $\boldsymbol{\delta}^t \in \mathbb{R}^N$: adjustment for sensor drift ($\mathbf{0} \leftarrow \boldsymbol{\delta}^0$)
  $\boldsymbol{v}^t_{min} \in \mathbb{R}^N$: minimum over recent sensor readings
  $h \in \mathbb{R}$: drift window
  $\alpha_w \in [0, 1]$: blending factor on drift estimate
  $\alpha_v \in (0, 1]$: blending factor on past sensor values

Step 1: *Drift adjustment*
  $v^t_{j,min} = \min_{i \in [t-h,t]} v^i_j \in \boldsymbol{v}^i$
  $\boldsymbol{\delta}^t \leftarrow (1 - \alpha_w) \cdot \boldsymbol{\delta}^{t-1} + \alpha_w \cdot \boldsymbol{v}^t_{min},\ v^t_{j,min} \in \boldsymbol{v}^t_{min}, \forall j \in N$

Step 2: *Blend with prior readings*
  $\boldsymbol{v}^t \leftarrow (1 - \alpha_v) \cdot \boldsymbol{v}^{t-1} + \alpha_v \cdot \boldsymbol{v}^t$

Description 2 Prediction Bias

Parameters :
  $\boldsymbol{w}^t \in \mathbb{R}^9$: weights on movement classes ($\mathbf{1} \leftarrow \boldsymbol{w}^o$)
  $\alpha \in [0, 1]$: blending factor on weight update
  $\lambda \in [0, 1]$: trend strength
  $\ell \in \mathbb{N}!$: number of timesteps to build trend

Given prediction of class i at timestep t
Update weights on class labels:
  $w^t_j \leftarrow (1 - \alpha) \cdot w^{t-1}_j,\ \forall w^t_j \in \boldsymbol{w}^t, j \neq i$
  $w^t_i \leftarrow w^{t-1}_i + \alpha$

Given prediction $p^{t-\ell}$ of class j at timestep $t - \ell$
Bias the prediction to match a strong trend, if present:
  $a^t = \max_i w^t_i \in \boldsymbol{w}^t$
  $b^t = \max_i w^t_i \in \boldsymbol{w}^t, i \neq a^t$
  IF $(w^t_{a^t} - \lambda) > w^t_{b^t}$ AND $j \neq a^t$
    THEN $p^{t-\ell} \leftarrow a^t$

future predictions, based on the assumption that a corrective touch lasts multiple (typically on the order of tens of) timesteps when sampled at $10Hz$ (Description 2). The execution of the movement associated with a predicted label thus operates with a delay of ℓ timesteps, during which the SOM prediction might be changed to fit a recently observed trend.

4 Conclusions, Ongoing and Future Work

Different modalities for transferring information to a robot learner convey varying types of information, and therefore can be more or less appropriate for providing feedback regarding different aspects of a task. Since it is unlikely that a single modality will be superior in all measures, redundancy can be useful; thus lending importance to the ability to provide behavior information via multiple modalities.

Our prior work with the TPC algorithm indeed found adding the touchpad modality to be more effective than providing information via the demonstration modality alone. Here we have introduced a novel interface for providing tactile corrections, that employs information from the ROBOSKIN sensors and addresses the challenge of inferring spatial corrections from a large number of sensor channels whose relative and absolute positions on the robot body are unknown. In addition to effective information transfer, like the touchpad interface, we hypothesize that our interactive data gathering and map learning approach not only eliminates the need for accurate knowledge about the sensor locations, but will allow for customization to the human teacher that enables more effective correction-giving. The validation of this hypothesis is ongoing work.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement n^o [231500]-[ROBOSKIN].

References

[1] P. Abbeel and A. Y. Ng. Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of ICML*, 2005.

[2] B. Argall, S. Chernova, B. Browning, and M. Veloso. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5), 2009.

[3] B. Argall, E. Sauser, and A. Billard. Tactile guidance for policy adaptation. *Foundations and Trends in Robotics*, 1(2), 2010.

[4] B. D. Argall and A. G. Billard. A survey of tactile human-robot interactions. *Robotics and Autonomous Systems*, 58, 2010.

[5] D. C. Bentivegna. *Learning from Observation Using Primitives*. PhD thesis, College of Computing, Georgia Institute of Technology, Atlanta, GA, July 2004.

[6] A. Billard, S. Callinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, chapter 59. Springer, New York, NY, USA, 2008.

[7] S. Calinon and A. Billard. Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of HRI*, 2007.

[8] S. Chernova and M. Veloso. Learning equivalent action choices from demonstration. In *Proceedings of IROS*, 2008.

[9] D. H. Grollman and O. C. Jenkins. Dogged learning for robots. In *Proceedings of ICRA*, 2007.

[10] J. Kober and J. Peters. Learning motor primitives for robotics. In *Proceedings of ICRA '09*, 2009.

[11] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, and K. Torkkola. LVQ_PAK: The learning vector quantization program package. Technical Report A30, Helsinki University of Technology, 1996.

[12] J. Sweeney and R. Grupen. A model of shared grasp affordances from demonstration. In *Proceedings ofHumanoids*, 2007.

[13] N. Tsagarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, F. Becchi, L. Righetti, J. Santos-Victor, A. Ijspeert, M. Carrozza, and D. Caldwell. iCub: The design and realization of an open humanoid platform for cognitive and neuroscience research. *Advanced Robotics*, 21, 2007.