This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Hazara, Murtaza; Kyrki, Ville

Reinforcement learning for improving imitated in-contact skills

Published in:
16th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2016

DOI:
[10.1109/HUMANOIDS.2016.7803277](https://doi.org/10.1109/HUMANOIDS.2016.7803277)

Published: 02/01/2017

Document Version
Peer reviewed version

Please cite the original version:
Hazara, M., & Kyrki, V. (2017). Reinforcement learning for improving imitated in-contact skills. In *16th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2016* (pp. 194-201). [7803277] (IEEE-RAS International Conference on Humanoid Robots). IEEE. <https://doi.org/10.1109/HUMANOIDS.2016.7803277>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Reinforcement learning for improving imitated in-contact skills

Murtaza Hazara and Ville Kyrki

Abstract—Although motor primitives (MPs) for trajectory-based skills have been studied extensively, much less attention has been devoted to studying in-contact tasks. With robots becoming more commonplace, it is both economical and convenient to have a mechanism for learning an in-contact task from demonstration. However, transferring an in-contact skill such as wood planing from a human to a robot is significantly more challenging than transferring a trajectory-based skill; it requires a simultaneous control of both pose and force. Furthermore, some in-contact tasks have extremely complex contact environments. We present a framework for imitating an in-contact skill from a human demonstration and automatically enhancing the imitated force profile using a policy search method. The framework encodes both the demonstrated trajectory and the normal contact force using Dynamic Movement Primitives (DMPs). In experiments, we utilize Policy Improvement with Path Integral (PI²) algorithm for updating the imitated force policy. Our results demonstrate the effectiveness of this approach in improving the performance of a wood planing task. After only two update rounds, all the updated policies have outperformed the imitated policy at a significance level of $P < 0.001$.

I. INTRODUCTION

Humans excel over robots in using tools to accomplish tasks involving complex contact interactions. Humans have learned these skills by watching other humans performing them or by being guided by other humans. With robots becoming more commonplace, it is both economical and convenient if they can also learn new skills from human demonstration. Physical skills require controlling motion skilfully either in space or in contact with a surface. If human motions have underlying regularities, we can apply machine learning for learning those skills. There is ample evidence that animal motions consist of regular patterns both in the position [1] and force [2] domain. These underlying regularities are referred to as motor primitives (MPs).

Although trajectory-based skills [3], [4], [5] can be mastered by using a pure position control strategy, imitating many industrial and assistive skills require controlling the contact forces as well. For instance, both force and position need to be controlled in tasks such as wood-working, ironing, door opening, lifting, and cleaning a white-board. We refer to them as in-contact tasks. Recently, researchers have shown interest in studying in-contact skills. In-contact tasks which have been learned from demonstration include cleaning a vertical surface [6], controlling stiffness [7], ball-in-box [8], pouring drink [8], box pulling [9], flipping

Fig. 1. KUKA LBR 4+ planing a wooden board

task [9], stapling [10], and grasping small objects [11]. However, all these studies aim at learning a policy merely to reproduce the demonstrated force.

Mastering craftsmanship is like learning to ride a bicycle. In addition to mimicking a master's performance, one needs to practice and learn from his mistakes and successes. Similarly, a robot can improve an imitated skill through trials and errors. For example, a robot carpenter needs several trials to remove more chips or to cut more smoothly. Although automatic self-improvement of an imitated in-contact skill can be a crucial step toward independent robot explorer with significant economical benefits, only one study has considered this aspect. In fact, this problem has been considered only in very simple applications such as opening a door or picking up small objects [12].

In this paper, we study how to improve an imitated wood planing task which exhibits an extremely complex dynamic interaction; every cut will generate a new environment, which does not allow for generalization [13], [14], [15] or planning [16] of the planing task. Hence, we apply the trial and error framework of reinforcement learning (RL) for improving the performance of the imitated planing skill. Furthermore, we provide an intuitive mechanism for simultaneous recording of both position and force [17]. In addition, we apply DMPs [18] to encode a policy for a demonstrated wood planing task, and optimize the parameters of the DMPs using Policy Improvement with Path Integral (PI²) [19]. No other existing framework would be able to learn the optimal force profile for this task (see section II).

The main contribution of this paper is developing a

*This work was supported by Academy of Finland, decisions 264239 and 271394. M. Hazara and V. Kyrki are with School of Electrical Engineering, Aalto University, Finland murtaza.hazara@aalto.fi, ville.kyrki@aalto.fi

framework (section IV) for imitating an in-contact skill with subsequent self-improving of the imitated in-contact skill using a policy search method. This framework has been effective in improving the performance of an imitated planing task (section V), indicating that motor primitives can convey both position and force modalities even in a skill with a very complex contact interaction such as wood planing.

II. RELATED WORK

In some applications, a mere reproduction of the imitated skill does not achieve the objective of the task. For instance, reenacting an imitated peg-in-hole skill might not lead to a successful insertion of the peg into the hole because of the inaccuracies in the measured location or forces. As a result, the imitated skill need to be modified. Modification of an imitated in-contact skill can be implemented for three main reasons: adaptation using Iterative Learning Control (ILC), generalization, and improvement of the task performance using Reinforcement Learning.

Iterative learning control (ILC) has been exploited mainly for adapting the trajectory of in-contact tasks. Adaptation is achieved in an iterative manner, where the previous tracking error information is used in the next replication of the same trajectory for refining the feedforward compensation signal [20]. ILC has been applied for minimizing force tracking error in [21], [22] and in a peg-in-hole task [23]. However, the effectiveness of these approaches is limited to in-contact tasks with linearizable and stable contact interaction. Secondly, improvement of a demonstrated force/torque profile is not considered in ILC.

A demonstrated force profile can be generalized to new contact locations using different approaches. A model-free probabilistic movement primitives (ProMPs) provides a variable stiffness controller for grasping under uncertain locations [13]. In addition, differential calculus generalizes the applied force to a new path in a simple sculpting task [14]. Furthermore, the mean of trajectories with varying feedback gain is extracted using object centric warping of the demonstrated trajectories for tightening knots, flattening towels, and erasing a white board [15]. Using this approach, a robot will be stiff in every direction and throughout the wood planing task since there is little variance in the demonstrated trajectories. Moreover, the complex and non-homogeneous wood environment does not allow for a generalizable policy. Above all, these studies do not involve in self-improvement of an imitated task.

To our best knowledge, [12] is the only related paper optimizing two imitated in-contact skills: opening a door and picking up a pen. They proposed a policy for encoding a discretized trajectory. The parameters of this policy represents the state of a discretized system. Although such policy can encode complex trajectories, the number of parameters grow with the length of trajectories. Moreover, the system would behave as a low-pass filter attenuating fast variations in the desired force profile. Besides that, a

human reaction time is 200 millisecond, and yet he can manipulate an in-contact task dexterously. Hence, we select DMPs as the policy encoding for both the kinematic and force profile resulting in a smooth trajectory. In addition, they have initialized the force/torque profile to zero and learned the desired force profile through RL. Although this exploration strategy is safe for simple in-contact tasks such as opening a door or picking up a pen, it is not safe for a planing task to start exploration from an initially zero force profile. If the robot is compliant in the orthogonal direction (wood surface normal direction), it would go up and down as a result of forces in opposite directions (noisy exploration). Otherwise, chips might not be removed as the longitudinal force is much bigger than the orthogonal force; thus, exploration policies cannot be compared with respect to the chips weight, and learning would not be possible. Furthermore, human demonstration is very informative; a human knows how to correlate the magnitude of the force with the speed and orientation of the tool. Therefore, we have provided an intuitive mechanism for recording the demonstrated force profile simultaneously with the pose of the robot. Additionally, We propose a framework for improving an in-contact task by exploring the demonstrated force profile. We evaluate our framework in improving a planing task regarding the weight of the removed chips.

III. BACKGROUND

In this section, we briefly introduce the policy encoding using a DMP and policy search RL using PI^2 , which will be the two main components of our self-improving LfD framework.

A. Dynamic Movement Primitives

DMPs provide a policy encoding based on a set of differential equations. This policy maps the states of a one-dimensional system to actions resulting in a smooth trajectory, and it is made of two parts: a canonical system and a transform system.

$$\dot{z} = -\tau \alpha_z z \quad (1)$$

The canonical system (1) is a first order system representing the phase of the movement with state z ; it resembles an adjustable clock driving the transform system (2). $\tau = \frac{1}{T}$ denotes the time constant where T is the duration of the demonstrated motion.

$$\frac{1}{\tau} \ddot{x} = \alpha_x (\beta_x (g - x_t) - \dot{x}_t) + \mathbf{g}^T \theta \quad (2)$$

The transform system (2) consists of a simple linear dynamical system acting like a spring damper perturbed by a non-linear component ($\mathbf{g}^T \theta$). x denotes the state of a one-dimensional system, and g represents the goal. The linear system is critically damped by setting the gains as $\alpha_x = \frac{1}{4} \beta_x$. The scaling factor A is set to $g - x_0$ in order to allow for the spatial scaling of the resulting trajectory.

$$[\mathbf{g}]_n = \frac{\psi^n(z)z}{\sum_{n=1}^N \psi^n(z)}(g - x_0) \quad (3)$$

The non-linear component (3) determines the shape of the motion whose contribution decays exponentially (by incorporating the phase variable z), thus allowing for the linear part of the transform system to converge to the goal g . This component is a mixture of basis functions ψ multiplied with a weight vector (shape parameters) θ and the phase variable z . The shape parameters θ can be estimated from a human demonstration using Linear Weighted Regression [18].

$$\psi^n(z) = \exp(-h_n(z - c_n)^2) \quad (4)$$

Normally, a Gaussian kernel (4) is selected as the basis function. The centres of kernels (c_n) are usually equispaced in time spanning the whole demonstrated trajectory. It is also a common practice to choose the same temporal width ($h_n = \frac{2}{3}|c_n - c_{n-1}|$) for all kernels. Check [18] for more details about DMP.

B. Policy Improvement with Path Integral

DMPs yield a parametric policy encoding allowing for a smooth replication of a human demonstrated trajectory. An imitated policy can be further optimized using a policy search method. Such a model-free RL method updates the parameters in an iterative process of exploration and evaluation. Furthermore, the optimization process using PI² is described in Algorithm 1. This process consists of three steps: exploration, evaluation, and updating.

$$\frac{1}{\tau} \ddot{x} = \alpha_x(\beta_x(g - x_t) - \dot{x}_t) + \mathbf{g}^T(\boldsymbol{\theta} + \boldsymbol{\epsilon}_t) \quad (5)$$

In the exploration step, K noisy roll-outs are generated. Each noisy trajectory is generated by adding a white Gaussian noise $\boldsymbol{\epsilon}$ to the policy parameters using (5). This noise allows for exploration in the policy parameters space. The noise $\boldsymbol{\epsilon}_t$ is sampled from a Gaussian distribution with zero mean and with covariance $\boldsymbol{\Sigma}^\theta$. The covariance $\boldsymbol{\Sigma}^\theta$ of this Gaussian noise needs to be determined in advance of the optimization process and with respect to the application.

In the evaluation step, the probability $P(\tau_{i,k})$ of every trajectory at each time step is computed using (8). This step involves the computation of the cost-to-go at every time-step using (7). The cost-to-go ($S(\tau_{i,k})$) indicates how good a trajectory starting at t_i and ending at t_N is; the lower this value, the better the trajectory is. The terminal cost $\phi_{t_N,k}$ and the state-dependent immediate cost $q_{t_j,k}$ need to be defined according to the application. Then, the probability of the trajectory is computed by exponentiating the cost-to-go. In this case, trajectory probability is inversely proportional to its cost. λ regulates the probability of each trajectory by distinguishing its cost from the costs of other trajectories. Furthermore, λ can be eliminated using (11) if $S(\tau_{i,k})$ is positive at every time step. We have

Algorithm 1 PI² algorithm for updating DMPs Policy Parameters

Input: The imitated policy with parameter $\boldsymbol{\theta}^{init}$, the exploration Variance $\boldsymbol{\Sigma}^\theta$, the cost function regulator λ , and DMP kernels w_j

Output: Improved policy with parameters $\boldsymbol{\theta}^{improved}$

Initialisation :

$\boldsymbol{\theta} = \boldsymbol{\theta}^{init}$

K =the number of exploration iterations

KL =Number of DMP kernels

1: **repeat**

2: Draw K random samples $\boldsymbol{\epsilon}_k \sim \mathcal{N}(0, \boldsymbol{\Sigma}^\theta)$

3: (Exploration Process): Create K roll-outs of the system $\tau_1 \dots \tau_K$ from the same starting state x_0 but using noisy parameters $\boldsymbol{\theta} + \boldsymbol{\epsilon}_{t,k}$

Evaluation Process:

4: **for** $k = 1$ to K **do**

5: **for** $i = 1$ to N **do**

6: Calculate the projection matrix using (12)

$$\mathbf{M}_{ti} = \frac{\mathbf{R}^{-1} \mathbf{g}_{t_i} \mathbf{g}_{t_i}^T}{\mathbf{g}_{t_i}^T \mathbf{R}^{-1} \mathbf{g}_{t_i}} \quad (6)$$

7: Calculate the cost-to-go at every time-step

$$S(\tau_{i,k}) = \phi_{t_N,k} + \sum_{j=i}^{N-1} q_{t_j,k} + \frac{1}{2} \sum_{j=i}^{N-1} (\boldsymbol{\theta} + \mathbf{M}_{t_j} \boldsymbol{\epsilon}_{t_j,k})^T \mathbf{R} (\boldsymbol{\theta} + \mathbf{M}_{t_j} \boldsymbol{\epsilon}_{t_j,k}) \quad (7)$$

8: Compute the probability of each roll-out:

$$P(\tau_{i,k}) = \frac{e^{-\frac{1}{\lambda} S(\tau_{i,k})}}{\sum_{j=1}^K e^{-\frac{1}{\lambda} S(\tau_{i,j})}} \quad (8)$$

9: **end for**

10: **end for**

11: **for** $i = 1$ to N **do**

12: Compute the correction terms at every time step

$$\delta \boldsymbol{\theta}_{t_i} = \sum_{k=1}^K P(\tau_{i,k}) \mathbf{M}_{t_i,k} \boldsymbol{\epsilon}_{t_i,k} \quad (9)$$

13: **end for**

14: **for** $j = 1$ to KL **do**

15: Average the correction terms for every kernel

$$[\delta \boldsymbol{\theta}]_j = \frac{\sum_{i=0}^{N-1} (N-i) w_{j,t_i} [\delta \boldsymbol{\theta}_{t_i}]_j}{\sum_{i=0}^{N-1} (N-i) w_{j,t_i}} \quad (10)$$

16: **end for**

17: update the policy parameter $\boldsymbol{\theta} = \boldsymbol{\theta} + \delta \boldsymbol{\theta}$

18: Create a noiseless roll-out $R = \phi_{t_N} + \sum_{i=0}^{N-1} r_t$

19: **until** The cost function R converges

20: **return** $\boldsymbol{\theta}^{improved} = \boldsymbol{\theta}$

chosen $c = 12$ in our experiments. The variable c controls

Fig. 2. Multi-DoF Self-improving LfD framework for in-contact tasks

the discrimination of the trajectories respecting their costs. The larger c is, the larger the discrimination will be.

$$e^{-\frac{1}{\lambda}} = \exp(-c \frac{S(\tau_i) - \min S(\tau_i)}{\max S(\tau_i) - \min S(\tau_i)}) \quad (11)$$

In the update step, the correction terms at each time instant are calculated using (9); In this equation, the exploration random samples are weighted by the probability of each trajectory; thus, random samples leading to a lower cost trajectory will gain more weight. Then, these correction terms are averaged over all time instants yielding the kernel correction terms using (10). The earlier correction terms will gain more weight by incorporating $(N - i)$. The policy parameter vector is updated by adding the correction term.

Finally, a noiseless roll-out is generated by executing the policy with the updated parameters set. The cost of this noiseless roll-out is calculated. The whole process of exploration, evaluation and updating is repeated if the cost does not converge. Otherwise, the convergence of this cost signals the detection of a locally optimal policy parameter.

IV. THE AUTOMATIC SELF-IMPROVEMENT LfD FRAMEWORK

In section III, we have reviewed how to represent the policy of a one-dimensional system and how to improve its policy using PI² algorithm. However, a robot consists of multiple degrees of freedom (DoF). Thus, a multi-DoF framework is needed for imitating a skill using such multi-DoF systems and for improving the performance of the imitated skill.

In this paper, the state of the robot is defined in an eight dimensional space $\mathbf{X} = \{x, y, z, q_x, q_y, q_z, q_w, f_z\}$, where $\mathbf{X}_{pos} = \{x, y, z\}$ represents the position of the robot end effector, $\mathbf{X}_{quat} = \{q_x, q_y, q_z, q_w\}$ formulates its orientation using the quaternion notation, and f_z denotes the Cartesian force along the contact normal.

The multi-DoF self-improving LfD framework is depicted in Fig 2; this framework is divided into four major subcomponents: imitation, DMP executor, RL optimizer, and a Cartesian impedance controller.

In the imitation component, every single dimension of the aforementioned 8-dimensional state (each DoF) is modelled by a separate policy. Furthermore, the transform systems (2) of all these 8 policies (MPs) are driven by the same canonical system (1). In this way, the longitudinal position and the orthogonal force are effectively coordinated avoiding the hand plane's stall during the reenactment. In addition, each policy is formulated and parameterized using a one-dimensional DMP. The parameter set θ of this policy is learned using Linear Weighted Regression and sent to the DMP executor.

In the DMP executor component, the state of each single degree of freedom is computed by the numerical integration of its corresponding transform system with the associated learned policy parameters θ . Besides that, the same canonical system (1) drives all the transform systems, thus ensuring the stability of the system. In this way, the desired kinematic state of the robot ($\mathbf{X}^{des}$) is computed including the desired Cartesian position $\{x^{des}, y^{des}, z^{des}\}$ and the desired orientation $\{q_x^{des}, q_y^{des}, q_z^{des}, q_w^{des}\}$. Additionally, numerical integration of the force transform system yields the desired normal force. Forces in other directions are set to zero: $\{f_x^{des} = 0, f_y^{des} = 0\}$. Both the desired kinematic state $\mathbf{X}^{des}$ and the desired Cartesian force $\mathbf{F}^{des} = \{f_x^{des} = 0, f_y^{des} = 0, f_z^{des}\}$ are sent to the impedance controller.

$$\tau_{Cmd} = \mathbf{J}^T (\mathbf{k}_c (\mathbf{X}^{des} - \mathbf{X}^{msr}) + \mathbf{D}(\mathbf{d}_c) + \mathbf{F}^{des}) + \tau_{dynamics} \quad (12)$$

In the impedance control (12), a virtual spring-damper system $(\mathbf{k}_c (\mathbf{X}^{des} - \mathbf{X}^{msr}) + \mathbf{D}(\mathbf{d}_c) + \mathbf{F}^{des})$ is realized in the Cartesian space, where $\mathbf{k}_c$ denotes its stiffness, and $\mathbf{d}_c$ denotes the damping. The virtual spring $(\mathbf{k}_c (\mathbf{X}^{des} - \mathbf{X}^{msr}))$ and the damping $\mathbf{D}(\mathbf{d}_c)$ terms are added to the desired Cartesian force $\mathbf{F}^{des}$ which is coming from force transform system. This addition results in a force in the Cartesian space. The resulting Cartesian force is transformed into joint torques when multiplied by the the transposed Jacobian $\mathbf{J}^T$. Besides that, the resulting desired joint torques are added to the dynamics model $\tau_{dynamic}$ yielding in the command torques τ_{Cmd} . The command torques are realized by the joint actuators, thus generating a trajectory.

The RL optimization component consists of three sub-processes: exploration, evaluation, and parameter updating. Exploration is achieved by adding white Gaussian noise (5) to policy parameters of the normal force in the Cartesian space. In other words, we are exploring in the force

domain instead of exploring in the kinematic domain. In addition, multiple (K) trajectories are required for updating the parameters. As a result, K sets of random samples are drawn from the Gaussian distribution. The Gaussian distribution is zero-mean and the variance Σ is determined according to the application. Each set of K noisy policy parameters is sent to the Cartesian force transform system separately. Numerical integration of the Cartesian force transform system with these noisy policy parameters $\theta + \epsilon_k$ results in a force trajectory which is different from the demonstrated one. Meanwhile, the noiseless Cartesian position $\{x, y, z\}$ and quaternions $\{q_x, q_y, q_z, q_w\}$ are generated by incorporating the noiseless translation $\{\theta^x, \theta^y, \theta^z\}$ and noiseless orientation policy parameters $\{\theta^{q_x}, \theta^{q_y}, \theta^{q_z}, \theta^{q_w}\}$. Thus, the kinematic state of the system is reproduced, while exploring the force. This process is repeated K times (trials); each trial is achieved with the same Cartesian and Quaternion parameters but with a different normal force policy parameters. Hence, the demonstrated kinematic state is reproduced, but demonstrated force profile is explored.

The cost of these K exploratory trajectories are evaluated using (7), and their probabilities are calculated using (8). These probabilities are used in updating the parameters. In the update step, the correction terms are computed at every time step using (9); then, they are averaged over the demonstrated trajectory time resulting in the correction term for every kernel using (10). Furthermore, the convergence of the force policy parameters can be checked by running N noiseless trials with the updated policy parameter $\theta^{improved}$. In a nutshell, the kinematic state is imitated and reproduced, while the force profile is imitated, explored, and improved.

V. EXPERIMENT

This section specifies the system which we have utilized as a testbed for our self-improving LfD framework. Next, we describe the wood planing as an application for testing how effective our framework is in learning a complex in-contact task and improving it. In addition, we specify how we designed a system for several planing experiments.

A. System

Our testbed system consists of a light weight robot (LWR) arm (KUKA LBR 4+), a force/torque (F/T) sensor, and a hand plane. Our framework requires a multi-DOF arm manipulator providing:

- A programmable active compliance
- Gravity compensation
- A Cartesian impedance controller
- A fast sampling rate (at least 100 Hz)

The LWR arm was fixed vertically on a table (see Fig. 3). Furthermore, we clamped a plank to the side of the table for securing a board which is going to be planed. In addition, we fixed a knob and a handle on a thick aluminium plate and attached it to the robot end effector. Besides that, we mounted a 6-axis ATI mini 45

Fig. 3. Parts of the hardware system

Force/Torque (F/T) sensor and a hand plane on the plate; we placed the F/T sensor between the aluminium plate and the hand plane. This order of mounting the tools enables us to record only the contact forces. Additionally, it allows for a simultaneous recording of both position and force profile. This intuitive demonstration mechanism would make it convenient for transferring an in-contact task from a human to a robot.

B. The wood planing task

We selected wood planing for testing the effectiveness of our framework. We employed a high quality Rali hand plane for planing wood. Initially, the depth of the blade needs to be adjusted regulating the depth of the cut.

After fine-tuning the hand plane, side of a board was planed along the grain while passively guiding the robot in a gravity compensation mode. The planing task was demonstrated using the kinesthetic teaching mechanism; a human demonstrator grabs the manipulator, guide it in the space, and when plane starts to contact the surface of the wooden board, a downward force needs to be exerted on the knob initially. When the plane is well on the board, the knob and the handle should be both pushed equally. As the plane passes of the beginning of the board, all the force must be exerted on the handle. This way of planing would prevent the emergence of a convex surface on the board.

Wood planing exhibits a complex environment. For example, Fig. 4 shows the measured force resulting from two reenactments of the same imitated planing policy. Although the policy parameters are the same, the measured force do not follow the same pattern. Furthermore, wood planing requires a simultaneous control of both pose and force. Moreover, a model of the chips weight as a function of the exerted force is not known.

C. Design

The imitated policy was reenacted using 14 kernels (shape parameters) with gain parameters set as $\alpha_x = 500$ and $\beta_x = 4\alpha_x$, thus making it a critically damped system. Furthermore, the stiffness of the impedance controller (12) was modified as $\mathbf{k}_c = (k_x = 4000\text{N/m}, k_y = 4000\text{N/m}, k_z =$

Fig. 4. Complex wood environment

0N/m), thus generating a longitudinally stiff while orthogonally compliant controller.

A wooden board needs to be cut aggressively initially in order to level its surface quickly; in this case, the more chips shaved, the better. Hence, a reenacted policy was evaluated by the weight of the chips; however, it is not enough to assess the performance of a policy using one reenactment because of the complex dynamic interaction between the plane and the wood surface; every cut creates a new environment in which the reactive force profile can be significantly different; hence, one cannot predict a force in a certain state with sufficient accuracy. Moreover, moisture in the boards and a plane slipping over the surface of the wood increases the uncertainty of the environment. In general, the environment in a wood planing task is greatly more stochastic than previously studied tasks such as ball-in-a-cup, opening a door, and cleaning a whiteboard. Thus we assess the performance of an imitated policy using 30 reenactments.

$$\Sigma^\theta = (\alpha_\epsilon \times \sigma^2) \mathbf{I} \quad (13)$$

$$\sigma^2 = \text{var}(\theta_{f_z}^{\text{imitated}}) \quad (14)$$

$$R(\tau_i) = \phi_{t_N} + \int_{t_i}^{t_N} \left(\frac{1}{2} \theta_t^T \mathbf{R} \theta_t \right) \quad (15)$$

$$\phi_{t_N} = \exp(-2 \times w) \quad (16)$$

The imitated policy was optimized using two update iterations. In the first update round, the diagonal elements of the exploration noise covariance were initially set to $\alpha_\epsilon = 40\%$ of the variance of the imitated policy parameters for the normal force ($\theta_{f_z}^{\text{imitated}}$). This initial variance was determined experimentally. The exploration noise parameter was decreased to $\alpha_\epsilon = 30\%$ in the second update round. Furthermore, we utilized a state-independent cost function (15) (negative exponentiation of weight of the chips w (16))

Fig. 5. Different wooden boards planed

for evaluating the performance of the exploration policies. Additionally, $N = 10$ roll-outs were generated by executing the policy with the updated parameters set θ^{improved} for assessing the performance of the updated policy.

VI. RESULTS

In this section, we verify the effectiveness of our self-improvement LfD framework on planing four different wooden boards (see Fig. 5) along the grain. Board A and C are two different Douglas Fir boards, while board B and D are two different kinds of pine boards. Furthermore, we demonstrate the resulting chips weight using box plots. In the following figures, the blue box represents the chips weight cut by the imitated policy; a black box denotes the chips weight removed by the exploration policies, while the green box demonstrates the chips weight shaved off by an updated policy. We have tested three null hypothesis using the Mann-Whitney U test:

- 1) Under the first null hypothesis, the weights of the chips removed by the imitated policy and the first updated policy are drawn from the same distribution.
- 2) The second null hypothesis states that the weights of chips cut by the reenacted imitated policy and the second updated policy are distributed equally.
- 3) The third null hypothesis explains that weights of the chips resulting from the reenactment of the first updated policy has the same distribution as the weights of the chips shaved off by the second updated policy.

Fig. 6 shows the result of planing experiment on board A. Although the imitated policy was reenacted with a fixed parameters set θ^{init} , it removed chips with varying weights. Furthermore, some planing trials (marked by red crosses) resulted in a very different chips weight indicating the complexity of the wood surface environment. Yet, the reenactment of the updated policies led to a significant increase in the chips weight (see TABLE I). Additionally, all the aforementioned three null hypotheses were rejected at a significance level of $P < 0.001$, indicating an improvement in the performance of the updated policies. Besides that, another planing experiment was performed on board C. All the three null hypotheses were rejected (see TABLE I), indicating the effectiveness of our approach in improving the in-contact task of wood planing on board C.

The result of planing experiment on board B is demonstrated in Fig 7. There is a strong evidence against the first

Fig. 6. Chips weight resulting from planing board A; from left to right: IP(imitated policy); EXP1(exploration policies of the first round), UP1(the first updated policy), EXP2(exploration policies of the second round), and UP2(the second updated policy) respectively.

Fig. 8. Chips weight resulting from planing board D; from left to right: IP(imitated policy); EXP1(exploration policies of the first round), UP1(the first updated policy), EXP2(exploration policies of the second round), and UP2(the second updated policy) respectively.

Fig. 7. Chips weight resulting from planing board B; from left to right: IP(imitated policy); EXP1(exploration policies of the first round), UP1(the first updated policy), EXP2(exploration policies of the second round), and UP2(the second updated policy) respectively.

and second null hypotheses with the $P < 0.001$, indicating a positive shift in the mean of the chips weights from the imitated policy to the first and second updated policies. However, the third hypothesis could not be rejected as the p-value was 0.1409. In other words, the first updated policy was performing substantially well that does not leave place for further improvement.

Fig. 8 shows the result of planing experiment on board D. This board was the easiest one to plane, and it requires much less force than the other boards because of its thin width. Furthermore, the imitated policy led to cutting chips with an average weight of 0.20867. Although the majority of the chips removed by the first updated policy

had a consistent weight with an average of 0.212, the first null hypothesis was accepted with a p-value of 0.8722, indicating no positive shift in the chips weight from the imitated policy to the first updated one. This is mainly because the blade depth and the imitated demonstration were perfectly set, thus leaving no place for significant improvement. On the other hand, both the second and third null hypotheses were rejected at a significance level of $P < 0.001$, indicating an improvement in the performance of the second updated policy.

VII. CONCLUSION

Our results shows the effectiveness of our framework in imitating a planing task and improving the imitated skill on planing four different boards. After only two update rounds, a policy was learned which showed statistically significant improvement over the initial imitated policy in removing more chips on all different boards, while a single update round was not sufficient. This work can be improved by considering a more detailed cost function including state-dependent metrics such as the immediate acceleration and magnitude of the normal force. In this case, an improved skill requiring less effort can be found. Furthermore, we determined that a planing task can be imitated using only 14 kernels (shape parameters) and that a diagonal covariance matrix with diagonal elements set to 40% of the variance of the imitated policy parameters was sufficient for an initial exploration, but how to control the exploration rate is a subject of further studies. Besides that, this paper considered only normal forces while some other applications are likely to benefit from policies including forces in other directions.

TABLE I
SUMMARY OF THE PLANING EXPERIMENTAL RESULTS

wooden board	dimension (width*length)	chips mean weight (gr) by the Imitated policy = μ_0	mean weight of the first updated policy = μ_1	mean weight of the second updated policy = μ_2	p-value for $\mu_0 < \mu_1$ (significance =1%)	p-value for $\mu_0 < \mu_2$ (significance =1%)	p-value for $\mu_1 < \mu_2$ (significance =1%)
Board A	58cm * 26cm	0.87	0.91	0.96	8.32E-005	2.08E-006	8.11E-005
Board B	38cm * 20cm	0.25	0.31	0.30	1.88E-006	1.89E-006	0.1409
Board C	58cm * 25cm	0.41	0.43	0.47	0.0541	1.74E-004	0.0026
Board D	49cm * 10cm	0.21	0.21	0.22	0.8722	3.37E-004	4.40E-004

REFERENCES

- [1] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.
- [2] F. A. Mussa-Ivaldi, S. F. Giszter, and E. Bizzi, "Linear combinations of primitives in vertebrate motor control," *Proceedings of the National Academy of Sciences*, vol. 91, no. 16, pp. 7534–7538, 1994.
- [3] J. Kober, E. Oztop, and J. Peters, "Reinforcement learning to adjust robot movements to new situations," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, p. 2650, 2011.
- [4] J. R. Peters, *Machine learning of motor skills for robotics*. PhD thesis, University of Southern California, 2007.
- [5] P. Kormushev, S. Calinon, R. Saegusa, and G. Metta, "Learning the skill of archery by a humanoid robot icub," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pp. 417–423, IEEE, 2010.
- [6] P. Kormushev, D. N. Nenchev, S. Calinon, and D. G. Caldwell, "Upper-body kinesthetic teaching of a free-standing humanoid robot," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3970–3975, IEEE, 2011.
- [7] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human-robot interaction," *Haptics, IEEE Transactions on*, vol. 7, no. 3, pp. 367–380, 2014.
- [8] L. Rozo, P. Jiménez, and C. Torras, "A robot learning from demonstration framework to perform force-based manipulation tasks," *Intelligent service robotics*, vol. 6, no. 1, pp. 33–51, 2013.
- [9] J. Kober, M. Gienger, and J. J. Steil, "Learning movement primitives for force interaction tasks," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 3192–3199, IEEE, 2015.
- [10] L. Righetti, M. Kalakrishnan, P. Pastor, J. Binney, J. Kelly, R. C. Voorhies, G. S. Sukhatme, and S. Schaal, "An autonomous manipulation system based on force control and optimization," *Autonomous Robots*, vol. 36, no. 1–2, pp. 11–30, 2014.
- [11] M. Kazemi, J.-S. Valois, J. A. Bagnell, and N. Pollard, "Human-inspired force compliant grasping primitives," *Autonomous Robots*, vol. 37, no. 2, pp. 209–225, 2014.
- [12] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, "Learning force control policies for compliant manipulation," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 4639–4644, IEEE, 2011.
- [13] A. Paraschos, E. Rueckert, J. Peters, and G. Neumann, "Model-free probabilistic movement primitives for physical interaction," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 2860–2866, IEEE, 2015.
- [14] V. Koropouli, S. Hirche, and D. Lee, "Learning and generalizing force control policies for sculpting," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 1493–1498, IEEE, 2012.
- [15] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 177–184, IEEE, 2015.
- [16] J.-C. Latombe, *Robot motion planning*, vol. 124. Springer Science & Business Media, 2012.
- [17] A. Montebelli, F. Steinmetz, and V. Kyrki, "On handing down our tools to robots: Single-phase kinesthetic teaching for dynamic in-contact tasks," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 5628–5634, IEEE, 2015.
- [18] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2, pp. 1398–1403, IEEE, 2002.
- [19] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *The Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.
- [20] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, 2006.
- [21] B. Nemec, A. Gams, M. Denisa, and A. Ude, "Human-robot cooperation through force adaptation using dynamic motion primitives and iterative learning," in *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, pp. 1439–1444, IEEE, 2014.
- [22] B. Nemec, T. Petrič, and A. Ude, "Force adaptation with recursive regression iterative learning controller," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 2835–2841, IEEE, 2015.
- [23] B. Nemec, F. J. Abu-Dakka, B. Ridge, A. Ude, J. A. Jorgensen, T. R. Savarimuthu, J. Jouffroy, H. G. Petersen, and N. Kruger, "Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile," in *Advanced Robotics (ICAR), 2013 16th International Conference on*, pp. 1–7, IEEE, 2013.