2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

Enhancing the Performance of Adaptive Iterative Learning Control with Reinforcement Learning

Bojan Nemec[1], Mihael Simonič[2], Nejc Likar[1], and Aleš Ude[1†]

Abstract— **In this study we propose a new method to enhance the performance of iterative learning control (ILC). We focus on robotic tasks dealing with adaptation to the unknown or partially known environment, where the robot has to learn the environment geometry in order to perform the desired task with the given reference forces and torques. The initial motion trajectories are obtained by kinesthetic teaching, whereas the required forces and torques are prescribed by the task. We are interested in incremental learning, which assures smooth and safe operation, aiming at handling of delicate, fragile objects, such as objects made of glass. In order to achieve these goals we propose a new adaptive ILC scheme, where the adaptation is supervised by reinforcement learning. We also show how to apply ILC to orientational motion, taking into account the curved geometry of SO(3). The performance of the proposed algorithm is verified on a bi-manual glass wiping task.**

I. Introduction

The ability to learn from previous experience is among the most important properties of a truly autonomous robot. A subclass of robot learning is sensorimotor learning, where the aim is to obtain completely new or to improve previously learned skills and adapt them to new situations given by environmental changes. Our approach to sensorimotor adaptation is based on techniques from machine/robot learning and robot control.

Robot learning is closely related to adaptive control and reinforcement learning as well as to the developmental robotics, which considers the problem of autonomous lifelong acquisition of repertoires of skills [1]. Traditional robot control methods assume that exact a priori models are available. Although remarkable results can be obtained in this way, model-based control can be very sensitive to inaccurately or not at all modeled system dynamics [2]. This is particularly evident for robots operating in human environments, where compliant (low gain) control is usually required in order to assure the safety of humans, the robot's environment and the robot itself. A promising solution to that problem is the learning of models, which can be generally divided into two subclasses: parametric and non-parametric model learning. Parametric learning is generally based on classical models, where the goal is to learn accurate model parameters. Here, we face two problems: a) a large number of parameters have to be estimated and b) it is hard to obtain general parametric models of human-robot-environment interaction [3]. Non-parametric models do not suffer from such limitations. By applying statistical learning methods, the control signal that assures the desired performance can be estimated from previously acquired data [3], [4], [5]. However, a huge collection of data is often required, which limits the practical applicability of such methods.

On the other hand, many practical problems can be successfully solved with simple learning schemes, such as iterative learning control (ILC). The key idea is to use the repetitive system dynamics to compensate for control errors. Due to its simplicity, effectiveness and robustness, ILC is becoming increasingly popular in the robotics community [6], [7]. However, although ILC is intrinsically robust to the variation of learning parameters, careful parameter tuning is still required. In order to overcome this problem, various adaptive ILC (AILC) algorithms were proposed in the literature [8], [9]. Roughly, they can be divided into two subclasses: a) ILC with adaptation of the feedback in the current iteration loop, and b) ILC with adaptation of the learning mechanism (also referred to as adaptation of the previous cycle learning). Note that combinations of both approaches are also possible [10].

In order to prove the learning and closed-loop stability of AILC, a number assumptions and limitations have to be considered. Some of these limitations cannot always be fulfilled in practice, especially for algorithms based on adaptation of the current iteration feedback. This might be the reason why the majority of the proposed adaptive ILC algorithms were verified only in simulation and just a few of them in real robot experiments.

A. Motivation

We are interested in tasks where a robot needs to handle fragile objects. Such operations take place in many different tasks, e. g. preparation of biological specimens, pipetting, tissue preparation, assembly of small electronic components, kitchen tasks such as wiping, etc. Although modern robots can operate with high speed and precision in highly structured environments, they are often not able to execute the desired task if an exact model of the environment is not available. In such cases we either have to learn the necessary task models and proceed with classical model-based control or we can directly learn the appropriate, possibly even optimal control policy. In our research we focus on the direct learning and adaptation of control policies that are

[†]The research leading to these results has received funding from the EU Horizon 2020 Programme grant no. 680431, ReconCell, and from the Slovenian Research Agency under grant agreement no. J2-7360.

[1]The authors are with Humanoid and Cognitive Robotics Lab, Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia (bojan.nemec@ijs.si, nejc.likar@ijs.si, ales.ude@ijs.si)

[2]The author is a student at Wilhelm Schickard Institute for Informatics, University of Tuebingen, Germany (mihael.simonic@student.uni-tuebingen.de)

needed to accomplish the given tasks. Our approach is to learn such control policies incrementally. We ensure safe and efficient robot performance before the optimal control policy has been learnt by exploiting compliance of modern robotic systems such as KUKA LWR 4. To achieve these goals we propose a new adaptive ILC algorithm, which is based on a gain switching technique proposed in [9]. The initial robot policy is obtained by kinesthetic teaching, but such policies are rarely optimal and have to be improved. The proposed adaptive ILC algorithm is combined with state of the art reinforcement learning approach PI^2, which enhances its performance and improves the robustness of the resulting control policy. Moreover, it provides stable learning also with sub-optimal learning parameters. One of the most important properties of the proposed algorithm is that in contrast to other adaptive iterative learning control methods, the desired compliance does not change during the adaptation process. It is thus applicable to robots interacting with humans and/or handling fragile objects.

The rest of the paper is organized as follows. In Section 2 we introduce the proposed adaptive iterative learning control algorithm and discuss the stability of the proposed scheme. In Section 3 we describe the main result of the paper, which is the synergy of ILC and reinforcement learning (RL) algorithms. The performance of the proposed algorithm is validated in Section 4, where we apply the proposed framework to the bi-manual glass wiping task. We conclude with a discussion of benefits of the proposed approach.

II. Adaptive Iterative Learning Controller

Most of adaptive iterative learning controllers (AILC) adjust their parameters after the completion of each learning cycle. As previously explained, they can be divided into two categories: a) ILC with adaptation of the feedback in the current iteration loop, and b) ILC with adaptation of the learning mechanism. These two approaches are also referred to as current cycle AILC and previous cycle AILC, respectively [11]. The proposed approach belongs to the second class of algorithms. The main motivation for us was to assure compliant behavior of the robot with low feedback gains, which cannot be achieved with current cycle AILC. Another reason is that the feedback stability in current cycle is always assured (providing that we properly designed the feedback control), since the feedback does not change through iteration. The only concern was thus to assure the learning stability.

We assume that the dynamics of n degrees of freedom robot interacting with the environment modeled as

$$\boldsymbol{\tau} = \mathbf{H}\ddot{\boldsymbol{\theta}} + \mathbf{h} + \mathbf{J}^T \begin{bmatrix} \mathbf{f} \\ \boldsymbol{\gamma} \end{bmatrix} \tag{1}$$

where $\boldsymbol{\tau} \in \mathbb{R}^n$ is a vector of joint torques, $\mathbf{H} \in \mathbb{R}^{n\times n}$ is a symmetric, positive definite inertia matrix, $\mathbf{h} \in \mathbb{R}^n$ contains nonlinear terms due to the centrifugal, Coriolis, friction and gravity forces, $\mathbf{J} \in \mathbb{R}^{6\times n}$ is the robot Jacobian, and $\mathbf{f}, \boldsymbol{\gamma} \in \mathbb{R}^3$ are the external contact forces and torques acting on the robot's end-effector. $\boldsymbol{\theta} \in \mathbb{R}^n$ denotes the joint angles. The relationship between joint space and Cartesian space accelerations is given by the following formula [12]

$$\ddot{\boldsymbol{\theta}} = \mathbf{J}_{\mathbf{H}}^{+}\left(\begin{bmatrix} \ddot{\mathbf{p}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} - \dot{\mathbf{J}}\dot{\boldsymbol{\theta}}\right) + \mathbf{N}\boldsymbol{\xi}, \tag{2}$$

where $\mathbf{J}_{\mathbf{H}}^{+} = \mathbf{H}^{-1}\mathbf{J}^{\mathrm{T}}(\mathbf{J}\mathbf{H}^{-1}\mathbf{J}^{\mathrm{T}})^{+} = \mathbf{H}^{-1/2}(\mathbf{J}\mathbf{H}^{-1/2})^{+}$ denotes the inertia weighted pseudo-inverse of $\mathbf{J}$. $\boldsymbol{\xi} \in \mathbb{R}^n$ is an arbitrary vector that determines the null space motion and $\mathbf{N} \in \mathbb{R}^{n\times n}$ is the projection matrix onto the null space of inertia weighted Jacobian. It can be computed as $\mathbf{N} = \mathbf{I} - \mathbf{J}_{\mathbf{H}}^{+}\mathbf{J}$. By inserting (2) into (1) we obtain a general form control law for a redundant robot [13]

$$\boldsymbol{\tau}_c = \mathbf{H}\mathbf{J}_{\mathbf{H}}^{+}\left(\begin{bmatrix} \ddot{\mathbf{p}}_c \\ \dot{\boldsymbol{\omega}}_c \end{bmatrix} - \dot{\mathbf{J}}\dot{\boldsymbol{\theta}}\right) + \mathbf{H}\mathbf{N}\boldsymbol{\xi}_c + \mathbf{h} + \mathbf{J}^T \begin{bmatrix} \mathbf{f} \\ \boldsymbol{\gamma} \end{bmatrix}. \tag{3}$$

Here parameters $\ddot{\mathbf{p}}_c$, $\dot{\boldsymbol{\omega}}_c$, and $\boldsymbol{\xi}_c$ are used as control inputs that should be set in such way that the task space tracking error is minimized. The first term in Eq. (3) is the task controller, the second term is the null space controller and the third and the fourth term compensate for the non-linear robot dynamics and external forces, respectively. According to the well known impedance control law [14] we choose the task command inputs $\ddot{\mathbf{p}}_c$, $\dot{\boldsymbol{\omega}}_c$ as

$$\ddot{\mathbf{p}}_c = \ddot{\mathbf{p}}_d + \mathbf{M}_p^{-1}(\mathbf{D}_p\dot{\mathbf{e}}_p + \mathbf{K}_p\mathbf{e}_p - \mathbf{f}), \tag{4}$$

$$\dot{\boldsymbol{\omega}}_c = \dot{\boldsymbol{\omega}}_d + \mathbf{M}_q^{-1}(\mathbf{D}_q\mathbf{e}_\omega + \mathbf{K}_q\mathbf{e}_q - \boldsymbol{\gamma}), \tag{5}$$

where subscript $_d$ denotes the desired values and variables without the subscript are the current values as received from the robot. $\mathbf{M}_p$, $\mathbf{M}_q$, $\mathbf{D}_p$, $\mathbf{D}_q$, $\mathbf{K}_p$ and $\mathbf{K}_q$ are positive definite, diagonal mass and inertia matrices, positional and rotational damping matrices, positional and rotational stiffness matrices, respectively. By subtracting the measured forces and torques in Eq. (4) and (5) and adding them back to (3) we preserve compliance and obtain decoupled behavior of the system. The position and orientation tracking errors are defined as $\mathbf{e}_p = \mathbf{p}_d - \mathbf{p}$, $\mathbf{e}_\omega = \boldsymbol{\omega}_d - \boldsymbol{\omega}$, and $\mathbf{e}_q = 2\log(\mathbf{q}_d * \overline{\mathbf{q}})$. We use unit quaternions $\mathbf{q} = (v, \mathbf{u}) \in \mathbb{R}^4$ to represent orientations and quaternion logarithm to compute the quaternion difference $\mathbf{e}_q$ $\log : \mathbf{S} \mapsto \mathbb{R}^3$ [15], [16], which is defined as

$$\log(\mathbf{q}) = \log(v, \mathbf{u}) = \begin{cases} \arccos(v)\dfrac{\mathbf{u}}{\|\mathbf{u}\|}, & \mathbf{u} \neq 0 \\ [0,0,0]^{\mathrm{T}}, & \text{otherwise} \end{cases}, \tag{6}$$

where $\mathbf{S}$ is a unit sphere in $\mathbb{R}^4$. Its inverse, i. e. the exponential map $\exp : \mathbb{R}^3 \mapsto \mathbf{S}$, is defined as

$$\exp(\mathbf{r}) = \begin{cases} \left(\cos\left(\|\mathbf{r}\|\right), \sin\left(\|\mathbf{r}\|\right)\dfrac{\mathbf{r}}{\|\mathbf{r}\|}\right), & \mathbf{r} \neq 0 \\ \left(1, [0,0,0]^{\mathrm{T}}\right), & \text{otherwise} \end{cases}. \tag{7}$$

In Eq. (4) and (5) $\mathbf{M}_p$, $\mathbf{M}_q$, $\mathbf{D}_p$, $\mathbf{D}_q$, $\mathbf{K}_p$ and $\mathbf{K}_q$ are positive definite, diagonal mass and inertia matrices, positional and rotational damping matrices, positional and rotational stiffness matrices, respectively.

We control the robot by inserting command accelerations (4) and (5) into (3). Taking into account Eq. (2), the target dynamics of the system is described by a set of 6 decoupled equations in the form

$$\mathbf{M}_p\ddot{\mathbf{e}}_p + \mathbf{D}_p\dot{\mathbf{e}}_p + \mathbf{K}_p\mathbf{e}_p = \mathbf{f}, \tag{8}$$

$$\mathbf{M}_q\dot{\mathbf{e}}_\omega + \mathbf{D}_q\mathbf{e}_\omega + \mathbf{K}_q\mathbf{e}_q = \boldsymbol{\gamma}. \tag{9}$$

In our real robot implementation of the impedance control, the desired acceleration and velocity terms are not present in $\dot{\mathbf{e}}_p$, $\ddot{\mathbf{e}}_p$, $\mathbf{e}_\omega$, and $\dot{\mathbf{e}}_\omega$ that are used in impedance equations (4) and (5). The purpose of this simplification is to guarantee the passivity of the system when the end effector is in contact with the environment [17]. We further simplify the control scheme (4) – (5) by selecting the desired mass and inertia matrices as identity, as it is common in impedance control [18].

By properly selecting the null space command input $\boldsymbol{\xi}_c$, one can control the null space motion of the robot. Khatib [19] suggested to calculate it as follows

$$\boldsymbol{\xi}_c = -\mathbf{K}_n\dot{\boldsymbol{\theta}}, \tag{10}$$

which results in an energy dissipation controller.

A. AILC within the feedforward adaptation loop

As explained in the introduction, our goal is to control contact forces. Force control can be accomplished in admittance mode by modifying the reference positions and orientations [20], i. e. we can adjust the contact forces and torques by controlling the commanded positions and orientations. We apply the standard ILC algorithm [6] to implement the adaptation of the commanded positions

$$\mathbf{p}_{d,l}(k) = \mathbf{p}_0(k) + \Delta\mathbf{p}_l(k) + \mathbf{C}_p\mathbf{e}_{f,l}(k), \tag{11}$$

$$\Delta\mathbf{p}_l(k) = \mathcal{Q}(\Delta\mathbf{p}_{l-1}(k) + \mathbf{L}_{p,l}\mathbf{e}_{f,l-1}(k)), \tag{12}$$

where force tracking errors are defined as $\mathbf{e}_f(k) = \mathbf{f}_d(k) - \mathbf{f}(k)$ with $\mathbf{f}_d(k)$ and $\mathbf{f}(k)$ being the desired and actual forces, respectively, while $\mathbf{p}_0(k)$ are the positions on the initial positional reference trajectory. For orientations we propose a modification of the standard ILC to account for nonlinear structure of the rotation space

$$\mathbf{q}_{d,l}(k) = \exp\left(\frac{1}{2}\left(\mathbf{C}_q\mathbf{e}_{\gamma,l}(k) + \Delta\mathbf{q}_l(k)\right)\right) * \mathbf{q}_0(k), \tag{13}$$

$$\Delta\mathbf{q}_l(k) = \mathcal{Q}(\Delta\mathbf{q}_{l-1}(k) + \mathbf{L}_{q,l}\mathbf{e}_{\gamma,l-1}(k+1)), \tag{14}$$

where torque tracking errors are defined as $\mathbf{e}_\gamma(k) = \boldsymbol{\gamma}_d(k) - \boldsymbol{\gamma}(k)$ with $\boldsymbol{\gamma}_d(k)$ and $\boldsymbol{\gamma}(k)$ being the desired and actual torques, respectively, while $\mathbf{q}_0(k)$ are the orientations on the initial orientational reference trajectory. Index k, $1 \leq k \leq T$ denotes the time sample, $T = \tau/\Delta t$ is the number of samples, τ the duration of the trajectory, and Δt the sampling time. Subscript l denotes the learning cycle. $\mathbf{C}_p \in \mathbb{R}^{3\times3}$, $\mathbf{C}_q \in \mathbb{R}^{3\times3}$, $\mathbf{L}_p \in \mathbb{R}^{3\times3}$ and $\mathbf{L}_q \in \mathbb{R}^{3\times3}$ are diagonal, positive definite gain matrices and $\mathcal{Q}$ is a discrete implementation of a low pass filter. Note that the learned $\Delta\mathbf{p} \in \mathbb{R}^3$ and $\Delta\mathbf{q} \in \mathbb{R}^3$ are translational and rotational displacements, which are initialized to 0 at the beginning of learning. The aim of ILC is to compute $\Delta\mathbf{p}$ and $\Delta\mathbf{q}$ so that the modified

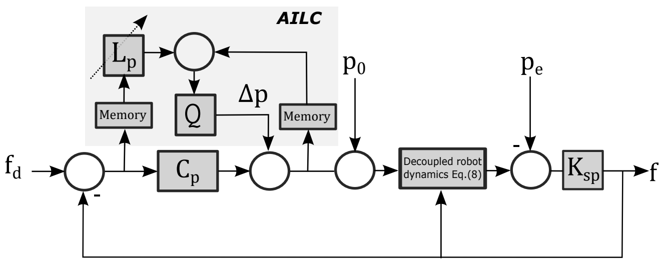

Fig. 1. Block diagram of the position part of the force based adaptation scheme. The arrow through $\mathbf{L}_p$ block indicates that $\mathbf{L}_p$ is not constant but changes with every learning cycle.

trajectories $\mathbf{p}_d = \Delta\mathbf{p} + \mathbf{p}_0$, $\mathbf{q}_d = \exp\left(\frac{1}{2}\Delta\mathbf{q}\right) * \mathbf{q}_0$ track the desired forces and torques.

Eqs. (11) – (14) define an ILC algorithm without adjustment. To achieve faster convergence, we propose to adapt the learning gains as follows

$$\mathbf{L}_{p,l} = \beta_l\mathbf{L}_{p,l-1}, \tag{15}$$

$$\mathbf{L}_{q,l} = \beta_l\mathbf{L}_{q,l-1},, \tag{16}$$

where $\beta_l \geq 1$. This adaptation rule is motivated by the work of [21].

Similarly to [22], the proposed algorithm is a combination of traditional impedance control law and admittance based iterative force adaptation with increasing gains. The role of learning is to compensate for the incompletely modeled robot and environment dynamics and to iteratively adjust contact forces by learning new reference positions. The main difference to [21] is that with our formulation the adaptation acts in the feedforward loop instead in the feedback loop. With this choice we can assure low feedback gains necessary for the intrinsically compliant behavior. On the other hand, the iterative adaptation of gains in the feedback loop as proposed in [21] was designed for free motion cases. It eventually results in a high-gain feedback control, which does not assure compliant behavior when interacting with the environment.

The overall control scheme is shown in Fig. 1. For the sake of simplicity, only the position part of the force adaptation scheme is shown in this diagram.

B. Stability of AILC within the feedforward adaptation loop

To prove stability, we describe the environment with a stiffness model. According to the stiffness model, contact forces and torques in steady state can be computed as $\mathbf{f} = \mathbf{K}_{sp}(\mathbf{p} - \mathbf{p}_e)$ and $\boldsymbol{\gamma} = 2\mathbf{K}_{sq}\log(\mathbf{q} * \overline{\mathbf{q}}_e)$, where $\mathbf{p}_e$ and $\mathbf{q}_e$ respectively denote the position and orientation of the contact point in the environment and $\mathbf{K}_{sp}, \mathbf{K}_{sq} \in \mathbb{R}^{3\times3}$ are the positive definite diagonal positional and rotational stiffness matrices.

We assume that the underlying impedance controller is properly designed to assure stable operation. Therefore, our concern is to examine how learning affects the stability of the system. For the sake of simplicity, learning stability will be examined only for adaptation of the positional part of the trajectory. Learning stability of ILC can be evaluated in time domain applying framework of time-lifted system [6] or in frequency domain [23], [11]. Although both approaches lead to the similar results, frequency-domain analysis gives better insight on how to design stable learning.

Starting point of the frequency-domain analysis is to express the dynamics of the position controlled robot in s domain. We describe the response of the decoupled robot with the transfer function $\mathbf{G}(s)$, obtained applying Laplace transform $\mathcal{L}$ to (8) and setting $\ddot{\mathbf{p}}_d = \dot{\mathbf{p}}_d = 0$, yielding $\mathbf{P}(s) = \frac{\mathbf{K}_p\mathbf{p}_d - \mathbf{F}(s)}{s^2+\mathbf{D}_p s+\mathbf{K}_p} = \mathbf{G}(s)(\mathbf{K}_p\mathbf{P}_d(s) - \mathbf{F}(s))$. Here, we will use the notation that uppercase letters denote $\mathcal{L}$ transform of the corresponding time dependent signal denoted with low letter. $\mathbf{P}(s)$ is thus $\mathcal{L}$ transform of $\mathbf{p}(k)$ and $\mathbf{P}_\delta(s)$ is $\mathcal{L}$ transform of $\delta(k)$, which is a signal that denotes upper bound of all disturbances induced by non-modeled robot dynamics and environment. The robot position controlled with impedance control law derived from (8) is

$$\mathbf{P}(s) = \mathbf{G}(s)(\mathbf{K}_p\mathbf{P}_d(s) - \mathbf{F}(s)) + \mathbf{P}_\delta(s). \tag{17}$$

To simplify the notation, we will omit explicit dependence on s in transfer functions and in $\mathcal{L}$ transform of the signals in further stability analyses. According to the Fig. 1, the system output is the force at the TCP of the robot, which is modeled assuming known environment stiffness $\mathbf{K}_{sp}$ as

$$\begin{aligned}\mathbf{F}_l &= \mathbf{K}_{sp}(\mathbf{K}_p\mathbf{G}\mathbf{P}_l - \mathbf{P}_e - \mathbf{P}_\delta - \mathbf{G}\mathbf{F}_l), \quad (18)\\ \mathbf{F}_l &= \frac{\mathbf{K}_{sp}}{\mathbf{I}+\mathbf{K}_{sp}\mathbf{G}}(\mathbf{K}_p\mathbf{G}\mathbf{P}_l - \mathbf{P}_e - \mathbf{P}_\delta)\\ &= \mathbf{K}^*(\mathbf{G}^*\mathbf{P}_l - \mathbf{P}_e - \mathbf{P}_\delta),\end{aligned}$$

where $\mathbf{P}_e$ denotes the environment contact positions. New transfer functions $\mathbf{K}^*$ and $\mathbf{G}^*$ are introduced to simplify the notation. According to (11,12), $\mathcal{L}$ transform of the error function $\mathbf{E}_l$, position update function $\mathbf{P}_l$ and learned offset function $\Delta\mathbf{P}$ are

$$\begin{aligned}\mathbf{E}_l &= \mathbf{F}_d - \mathbf{F}_l, &(19)\\ \mathbf{P}_l &= \mathbf{P}_0 + \Delta\mathbf{P}_l + \mathbf{C}_p\mathbf{E}_l, &(20)\\ \Delta\mathbf{P}_l &= \mathbf{Q}(\Delta\mathbf{P}_{l-1} + \mathbf{L}_p\mathbf{E}_{l-1}), &(21)\end{aligned}$$

where $\mathbf{Q}$ is $\mathcal{L}$ transform of $\mathcal{Q}$. Now, let express the error $\mathbf{E}_l$ as a function of the error in the previous learning cycle $\mathbf{E}_{l-1}$,

$$\begin{aligned}\mathbf{E}_l &= \mathbf{F}_d - \mathbf{F}_l \quad (22)\\ &= \mathbf{F}_d - \mathbf{K}^*(\mathbf{G}^*\mathbf{P}_l - \mathbf{P}_e - \mathbf{P}_\delta)\\ &= \mathbf{F}_d - \mathbf{K}^*(\mathbf{G}^*(\mathbf{P}_0 + \Delta\mathbf{P}_l + \mathbf{C}_p\mathbf{E}_l) - \mathbf{P}_e - \mathbf{P}_\delta)\\ &= \mathbf{F}_d - \mathbf{K}^*(\mathbf{G}^*(\mathbf{P}_0 + \mathbf{Q}(\Delta\mathbf{P}_{l\text{-}1} + \mathbf{L}_p\mathbf{E}_{l\text{-}1}) + \mathbf{C}_p\mathbf{E}_l)\\ &\quad -\mathbf{P}_e - \mathbf{P}_\delta)\\ &= \mathbf{Q}(\mathbf{F}_d - \mathbf{K}^*(\mathbf{G}^*(\mathbf{P}_0 + \Delta\mathbf{P}_{l\text{-}1} + \mathbf{C}_p\mathbf{E}_{l\text{-}1} - \mathbf{C}_p\mathbf{E}_{l\text{-}1} +\\ &\quad \mathbf{L}_p\mathbf{E}_{l\text{-}1}) - \mathbf{P}_e - \mathbf{P}_\delta)) - \mathbf{K}^*\mathbf{G}^*\mathbf{C}_p\mathbf{E}_l +\\ &\quad (\mathbf{I} - \mathbf{Q})(\mathbf{F}_d - \mathbf{K}^*(\mathbf{G}^*\mathbf{P}_0 - \mathbf{P}_e - \mathbf{P}_\delta))\\ &= \mathbf{Q}(\mathbf{F}_d - \mathbf{F}_{l\text{-}1} - \mathbf{K}^*\mathbf{G}^*(\mathbf{L}_p - \mathbf{C}_p)\mathbf{E}_{l\text{-}1}) - \mathbf{K}^*\mathbf{G}^*\mathbf{C}_p\mathbf{E}_l\\ &\quad +(\mathbf{I} - \mathbf{Q})(\mathbf{F}_d - \mathbf{K}^*(\mathbf{G}^*\mathbf{P}_0 - \mathbf{P}_e - \mathbf{P}_\delta))\\ &= \mathbf{Q}(\mathbf{I} - \mathbf{K}^*\mathbf{G}^*(\mathbf{L}_p - \mathbf{C}_p))\mathbf{E}_{l\text{-}1} - \mathbf{K}^*\mathbf{G}^*\mathbf{C}_p\mathbf{E}_l +\\ &\quad (\mathbf{I} - \mathbf{Q})(\mathbf{F}_d - \mathbf{K}^*(\mathbf{G}^*\mathbf{P}_0 - \mathbf{P}_e - \mathbf{P}_\delta)).\end{aligned}$$

In the above equation we added and subtracted the term $\mathbf{Q}(\mathbf{F}_d - \mathbf{K}^*(\mathbf{G}^*(\mathbf{P}_0 + \mathbf{C}_p\mathbf{E}_{l\text{-}1}) - \mathbf{P}_e - \mathbf{P}_\delta))$ and used (18) – (21). Rearranging (22) we obtain

$$\begin{aligned}\frac{\mathbf{E}_l}{\mathbf{E}_{l\text{-}1}} &= \frac{\mathbf{Q}(\mathbf{I} - \mathbf{K}^*\mathbf{G}^*(\mathbf{L}_p - \mathbf{C}_p))}{\mathbf{I} + \mathbf{K}^*\mathbf{G}^*\mathbf{C}_p} \quad (23)\\ &+ \frac{\mathbf{I} - \mathbf{Q}}{\mathbf{I} + \mathbf{K}^*\mathbf{G}^*\mathbf{C}_p}\frac{\mathbf{F}_d - \mathbf{K}^*(\mathbf{G}^*\mathbf{P}_0 - \mathbf{P}_e - \mathbf{P}_\delta)}{\mathbf{E}_{l-1}}\end{aligned}$$

Asymptotic stability is assured iff $\frac{\mathbf{E}_l}{\mathbf{E}_{l\text{-}1}} < 1 \; \forall \, l$.

Inserting again the s dependence into transfer functions and signals and substituting $s = j\bar{\omega}$ in (23), where $\bar{\omega}$ is the frequency, the condition for asymptotic stability becomes [24]

$$\begin{aligned}&\frac{\mathbf{Q}(j\bar{\omega})(\mathbf{I} - \mathbf{K}^*(j\bar{\omega})\mathbf{G}^*(j\bar{\omega})(\mathbf{L}_p - \mathbf{C}_p))}{\mathbf{I} + \mathbf{K}^*(j\bar{\omega})\mathbf{G}^*(j\bar{\omega})\mathbf{C}_p} +\\ &\quad \frac{\mathbf{I} - \mathbf{Q}(j\bar{\omega})}{\mathbf{I} + \mathbf{K}^*\mathbf{G}^*(j\bar{\omega})\mathbf{C}_p}\epsilon \; < \; 1, \forall \, \bar{\omega}, \quad (24)\end{aligned}$$

where

$$\epsilon = \frac{\mathbf{F}_d(j\bar{\omega}) - \mathbf{K}^*(j\bar{\omega})(\mathbf{G}^*(j\bar{\omega})\mathbf{P}_0(j\bar{\omega}) - \mathbf{P}_e(j\bar{\omega}) - \mathbf{P}_\delta(j\bar{\omega}))}{\mathbf{E}_{l-1}(j\bar{\omega})}.$$

Given the known transfer function $\mathbf{G}(s)$ and estimated environment stiffness $\mathbf{K}_{sp}$ we have to design such admittance control gains $\mathbf{C}_p$, learning function $\mathbf{Q}(s)$ and gains $\mathbf{L}_p$, that the learning error $\mathbf{E}(s)$ asymptotically decays to 0 when $l \to \infty$, Note that the nominator of the term ϵ is small bounded number, since $\mathbf{K}^*(\mathbf{G}^*\mathbf{P}_0 - \mathbf{P}_e) \approx \mathbf{F}_d$. It depends only on the desired force $\mathbf{F}_d$, initial demonstrated trajectory $\mathbf{P}_0$, environment $\mathbf{P}_e$ and upper bound of disturbances $\mathbf{P}_\delta$ and thus does not change during the adaptation. Therefore, ϵ increases when the error $\mathbf{E}$ decreases. Consequently, zero learning error can be guaranteed only with the choice $\mathbf{Q}(s) = \mathbf{I}$. On the other hand, it is generally very hard to fulfill condition (24) with this choice. In most cases $\mathbf{Q}(s)$ in the form of a low pass filter will assure the stability, but increase learning error [7]. Therefore, the design of $\mathbf{Q}(s)$ is a trade off between the robustness, stability and performance of the learning algorithm.

Let us analyze the stability of the AILC with gain adaptation (refereed as switching in the original approach) given with (15) using Bode diagram. The Bode plot in Fig. 2 was generated with the following parameters: $\mathbf{K}_p = 500\,\mathbf{I}\,N/m$, $\mathbf{D}_p = 2\sqrt(\mathbf{K}_p)$, $\mathbf{K}_{sp} = \mathbf{K}_p$, $\mathbf{C}_p = \mathbf{K}_p^{-1}$, $\mathbf{L}_{p,0} = \mathbf{K}_p/2$, $\beta = 1.5$. Initial gain $\mathbf{L}_{p,0}$ was increased in 10 steps according to (15). Initially, increasing $\mathbf{L}$ stabilizes the learning and improves the learning speed, since Bode gain stays below 0 dB for all frequencies (see Fig. 2). However, if we further increase $\mathbf{L}$, Bode gain crosses 0 dB margin at low frequencies and learning becomes unstable. The controller thus requires careful parameters tunning for stable operation. On the other hand, if we want truly autonomous robots, we can not rely on exact parameter tuning, which vary from case to case. We need a mechanism, that will constantly monitor the adaptation and autonomously choose proper adaptation gains. A good candidate for this mission are reinforcement learning algorithms.

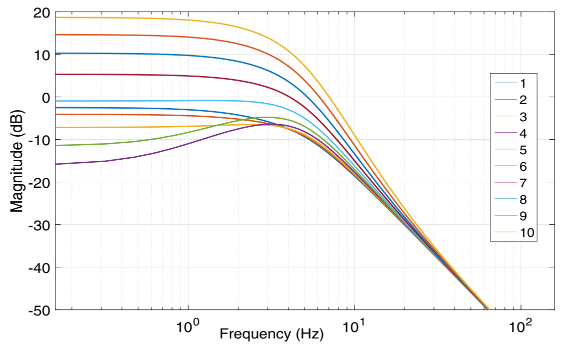

Fig. 2. Bode plot for increasing $\mathbf{L}$ in subsequent learning cycles. Legend numbers show learning cycles. We can see that Bode gain drops at initial learning cycles, but starts to increase in subsequent repetitions and crosses learning stability margin at 0 dB after 6 cycles.

III. Enhancing AILC with Reinforcement Learning

Reinforcement Learning (RL) is often used in robotics for solving problems where exact models are not available. It enables a robot to autonomously find an optimal policy by direct trial-and-error exploration within its environment [25]. One of the major problems in autonomous exploration is a huge search space, which is determined by the degrees of freedom of controlled variables and underlying policy representations. The probabilistic policy improvement RL algorithms like PI^2 [26] and PoWER [27] can scale to complex learning systems and minimize the number of tuning parameters. However, due to a high dimensionality of the parameter space, the adaptation speed is still low, which limits the applicability of RL for robot control. The main reason for the low adaptation speed is random search, which is usually applied in RL and is needed to ensure convergence. This gave raise to the idea to replace the random search with a more focused search process, e. g. an AILC. This way we can join the high convergence rate of the AILC algorithm with the ability of RL algorithm to discard those parts of the learning process where the robot performed badly and persist searching within those parameters where the performance was good. Note, however, that the proposed approach can be used only for problems that can be addressed by ILC. There are problems that can be solved by RL but not with ILC. e. g. when the cost function is not directly related to the control signal.

The proposed approach can resolve the potential learning instabilities of AILC. Such instabilities, unlike feedback instabilities, do not result in a sudden "explosion" of the control signal. They rather manifest through a slow raise of the tracking error, which usually results in an increased oscillation around the reference value of the control signal. Such oscillations are caused by a badly learned feedforward signal. In this paper we propose to monitor the learning cycles and optimize the feedforward ILC signal with respect to a given criterion. RL algorithm is a perfect candidate to implement this optimization process.

Although there are many similarities between ILC and RL [28], there are differences in how they act. While ILC acts directly on signals, RL acts on states or parameters that describe the control signals. Therefore, in order to integrate ILC and RL algorithms, it is necessary to parametrize the time varying ILC parameters. A suitable choice for this parametrization are radial basis function (RBF). Our approach aims at modifying the feedforward control signals $\Delta\mathbf{p}(k)$, $\Delta\mathbf{q}(k)$ and learning gains $\mathbf{L}_p$, $\mathbf{L}_q$, which are initially set by AILC algorithm (11) – (16). Note that learning gains in AILC formulation do not depend on time, since they change only after each learning cycle. Here we will encode them as time variable signals $\mathbf{L}_p(k)$ and $\mathbf{L}_q(k)$ in order to let the RL algorithm to optimize their values in each time sample. Namely, in original AILC formulation, feedforward control signals $\Delta\mathbf{p}(k)$, $\Delta\mathbf{q}(k)$ depend only on the past tracking errors and fixed learning gains. By introducing time variability of the learning gains, one can better shape the feedforward control signals.

The signals described above are thus encoded as a weighted sum of M Gaussian kernels for each parameter dimension

$$\begin{aligned}
\Delta p_j(k) &= \frac{\sum_{i=1}^M w_{p,i,j}\Psi_i(k\Delta t)}{\sum_{i=1}^M \Psi_i(k\Delta t)},\ j=1,2,3, && (25)\\
\Delta q_j(k) &= \frac{\sum_{i=1}^M w_{q,i,j}\Psi_i(k\Delta t)}{\sum_{i=1}^M \Psi_i(k\Delta t)},\ j=1,2,3, && (26)\\
L_{p,j}(k) &= \frac{\sum_{i=1}^M w_{pL,i,j}\Psi_i(k\Delta t)}{\sum_{i=1}^M \Psi_i(k\Delta t)},\ j=1,2,3, && (27)\\
L_{q,j}(k) &= \frac{\sum_{i=1}^M w_{qL,i,j}\Psi_i(k\Delta t)}{\sum_{i=1}^M \Psi_i(k\Delta t)},\ j=1,2,3, && (28)
\end{aligned}$$

where

$$\Psi_i(t) = \exp\left(-h_i\,(t-c_i)^2\right) \tag{29}$$

Here Δt is the sampling time, c_i are the centers of radial basis functions distributed along the time evolution of the trajectory, and $h_i > 0$ their widths. Weights $w_{p/q/L_p/L_q,i,j}$ determine the shape of the corresponding signal.

To calculate $w_{p,i,j}$ from $\Delta p_j(k), 1 \leq k \leq T$, we need to solve

$$\mathbf{B}\mathbf{w}_{p,j} = \mathbf{u}, \tag{30}$$

with

$$\mathbf{w}_{p,j} = \begin{bmatrix} w_{p,1,j} \\ \vdots \\ w_{p,M,j} \end{bmatrix}, \mathbf{u} = \begin{bmatrix} \Delta p_j(1) \\ \vdots \\ \Delta p_j(T) \end{bmatrix}, \tag{31}$$

and the system matrix $\mathbf{B} \in \mathbb{R}^{T\times M}$ defined as

$$\mathbf{B} = \begin{bmatrix} \dfrac{\psi_1(\Delta t)}{\sum_{j=1}^M \psi_j(\Delta t)} & \cdots & \dfrac{\psi_M(\Delta t)}{\sum_{j=1}^M \psi_j(\Delta t)} \\ \vdots & \vdots & \vdots \\ \dfrac{\psi_1(T\Delta t)}{\sum_{j=1}^M \psi_j(T\Delta t)} & \cdots & \dfrac{\psi_M(T\Delta t)}{\sum_{j=1}^M \psi_j(T\Delta t)} \end{bmatrix}. \tag{32}$$

The weights $\mathbf{w}_{q,j}$, $\mathbf{w}_{p_L,j}$ and $\mathbf{w}_{q_L,j}$ are calculated in the same way.

In order to apply RL, a cost function has to be defined [25]. For the learning of control signals, both the immediate and terminal costs are important. A straightforward choice for the immediate cost might be the norm of the tracking error at the current time. However, a simple norm of the tracking error might not be appropriate to detect unstable learning. There are many ways how to detect instable behavior, such as passivity observers [29], stability observers [30], instability indexes [31], etc. In this work we propose to detect learning instabilities by observing oscillations of the force error signal. The Bode plot depicted in Fig. 2 shows that the stability margin is violated at low frequencies with improper learning parameter settings. Consequently, learning instability will be manifested as low frequency oscillations, which can be evaluated by calculation of the signal envelope using Hilbert transform [32]. The proposed cost function combines this criteria with the norm of force and torque tracking errors. Immediate cost can be thus calculated as

$$c_i(k) = \|\mathbf{e}_f(k)\| + \|\mathbf{e}_q(k)\| + \|\mathcal{H}(\mathbf{e}_f(k))\| + \|\mathcal{H}(\mathbf{e}_q(k))\| \tag{33}$$

where $\mathcal{H}$ denotes the Hilbert transform. Terminal cost c_t is application dependent and therefore cannot be defined independently of the task. It usually describes how well the robot accomplished the desired task.

After each execution of the task (rollout), new weights for $\Delta\mathbf{p}$, $\Delta\mathbf{q}$ $\mathbf{L}_p$, and $\mathbf{L}_q$ are calculated as

$$\mathbf{W}^* = \Upsilon(\pi, \mathbf{W}, \mathbf{c}_i, \mathbf{c}_t) \tag{34}$$

where π denotes the set of all signals $\Delta\mathbf{p}(t)$, $\Delta\mathbf{q}(t)$, $\mathbf{L}_p(t)$, $\mathbf{L}_q(t)$, $\mathbf{f}(t)$, $\boldsymbol{\gamma}(t)$, $\mathbf{W}$ combines all weights $\mathbf{w}_{p,j}, \mathbf{w}_{q,j}, \mathbf{w}_{p_L,j}, \mathbf{w}_{q_L,j},\ j = 1,2,3$, and $\mathbf{c}_i$, $\mathbf{c}_t$ denote all intermediate and terminal costs obtained during exploration. Υ denotes the reinforcement learning algorithm, e. g. PI^2 [26]. A good description of PI^2 can be found in [33]. New control signals $\Delta\mathbf{p}^*(k)$, $\Delta\mathbf{q}^*(k)$ and ILC parameters $\mathbf{L}_p^*(k)$, $\mathbf{L}_q^*(k)$ are calculated from optimized weights $\mathbf{W}^*$ using Eqs. (25) – (28). In order to speed up learning and reject the unsuccessful attempts, the input data to (34) are reordered after each learning cycle using importance sampling [25]. The entire learning procedure is summarized in Algorithm 1.

Note that encoding of the signals with RBFs smooths the original signals. As shown in [34], the output signals generated by RBFs can be approximated with signals passed though a second order low pass filter, whose cutoff frequency depends on the number of kernel function M and widths of kernel functions h_i. Therefore it is not necessary to explicitly implement the low pass filter Q in (12), (14), which is otherwise needed to improve the learning stability at higher frequencies [7], [34]. The entire control scheme of this new algorithm, referred to as AILC-RL, is outlined in Fig. 3.

IV. Experimental evaluation

The proposed approach was experimentally evaluated on a challenging task of bi-manual glass wiping. This task is naturally described in cylindrical coordinate system with coordinates $[r\ \varphi\ z]^T$. The initial trajectory for this task was obtained by human demonstration applying kinesthetic guiding for cylindrically shaped glasses. After that, we replaced the glass with a rectangular, cone-shaped glass (see Fig. 4). The goal was to improve and adapt the demonstrated motion by assuring perfect tracking of force-torque profiles after learning. Experimental setup consisted of two 7 degrees of freedom KUKA LWR 4 robot arms, mounted on torso with one rotational degree of freedom. The right and left robot arm held the glass and the sponge with the Barret hand, as shown in Fig. 4.

For bi-manual robot control we applied coordinated task-space framework as presented in [35]. Within this framework, a bi-manual task is decoupled into relative and absolute coordinates [36], which can be controlled independently. In this task only relative coordinates are important for the performance of the task, as it is generally unimportant where the robot holds the glass during wiping. Consequently, this task was characterized by a large null space with 8 redundant degrees of freedom. The relative coordinates were controlled with impedance control law (3), (4). The initial parameters

Algorithm 1: trajectory adaptation scheme

```
Input: reference trajectory and force-torque profiles
       $\mathbf{p}_0(k), \mathbf{q}_0(k), \mathbf{f}_d(k), \boldsymbol{\gamma}_d(k),\ k = 1, \ldots, T$
       initialize control gains $\mathbf{K}_p$, $\mathbf{K}_q$, $\mathbf{D}_p$, $\mathbf{D}_q$ and
       dynamic and kinematic parameters $\mathbf{H}$, $\mathbf{h}$, $\mathbf{J}$
       initialize ILC gains $\mathbf{L}_{p,0}$, $\mathbf{L}_{q,0}$, $\mathbf{C}_p$, and $\mathbf{C}_q$
       initialize compensation signals $\Delta\mathbf{p} = \Delta\mathbf{q} = 0$
Output: optimal control signal $\Delta\mathbf{p}(k)$, $\Delta\mathbf{q}(k)$ and ILC
        gains $\mathbf{L}_p(k), \mathbf{L}_q(k)$
1 for $l = 1, \ldots,$ max learning cycles do
2     execute one step of AILC (11) – (16)
3     collect intermediate costs $\mathbf{c}_i$ and terminal cost $c_t$
4     calculate weights $\mathbf{w}_p$, $\mathbf{w}_q, \mathbf{w}_{pL}$ and $\mathbf{w}_{qL}$ from
        $\Delta\mathbf{p}_l(k)$, $\Delta\mathbf{q}_l(k)$, $\mathbf{L}_{p,l}(k)$ and $\mathbf{L}_{q,l}(k)$ using
        regression (30)
5     save data in the importance sampler
6     estimate new optimized weights using RL (34)
7     using (25) – (28) calculate $\Delta\mathbf{p}_{l+1}(k)$, $\Delta\mathbf{q}_{l+1}(k)$,
        $\mathbf{L}_{p,l+1}$, and $\mathbf{L}_{q,l+1}$ from optimized weights $\mathbf{W}^*$
```

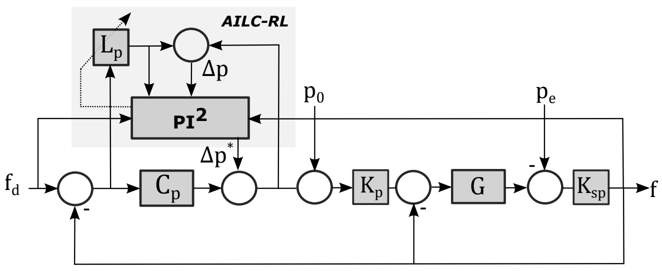

Fig. 3. Block diagram of position part of AILC-RL control scheme. The arrow through $\mathbf{L}_p$ block indicates that $\mathbf{L}_p$ is not constant but changes with every time sample according to the estimated value from PI^2 algorithm.

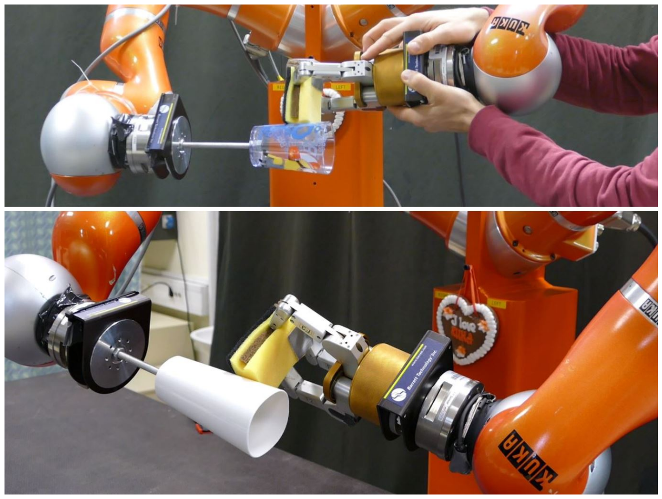
Fig. 4. The image above shows the demonstration of a glass wiping policy. Bi-manual task adaptation to a glass with different geometry using AILC-RL is shown in the image below. Note that we substituted round shaped glass with the rectangular shaped one in order to show the adaptation capabilities of the proposed approach.

were set as follows: $\mathbf{K}_p = 200\,\mathbf{I}\,\mathrm{N/m}$, $\mathbf{K}_q = 100\,\mathbf{I}\,\mathrm{Nm/rd}$, $\mathbf{C}_p = \mathbf{K}_p^{-1}$, $\mathbf{C}_q = \mathbf{K}_q^{-1}$, $\mathbf{L}_p = \mathbf{C}_p/2$, $\mathbf{L}_q = \mathbf{C}_q/2$, $\beta = 1.35$, $\mathbf{f}_d = \mathrm{diag}(5,0,0)\,\mathrm{N}$, $\boldsymbol{\gamma}_d = \mathbf{0}\,\mathrm{Nm}$. The terminal cost was defined as the norm of the tracking errors

$$c_t = \sum_{k=1}^{T} \|\mathbf{e}_f(k)\| + \|\mathbf{e}_q(k)\|) \tag{35}$$

In our learning experiment 20 learning cycles were executed. Fig. 5 shows radial forces f_r measured by a wrist mounted ATI force-torque sensor, radial force error e_{f_r}, radial compensation signal Δp_r, the learned ILC gain for the radial dimension L_{p_r} and the intermediate c_i and terminal costs $\mathbf{c}_t$ during the subsequent adaptation cycles. For clarity, results are shown for radial cylindrical relative coordinate (r) only, as most of adaptation happens in this direction.

We compared the performance of the proposed AILC-RL with AILC controller with the same settings. The result of comparison is shown in terminal cost bar of Fig. 5. Note that the initial convergence of AILC is comparable to AILC-RL, in some cases AILC even converges slightly faster. This occurs because PI^2 algorithm combines all past roll outs to predict new values, which slows down the initial convergence. However, after 8 iterations AILC starts to diverge, whereas AILC-RL still improves the cost and overall performance. Note also that time dependent learning gains $L_{p,r}$ converge to steady state values after 10 learning cycles.

V. Conclusions

In this paper we proposed a new algorithm for adaptation of control policies using AILC and RL based on force feedback. The resulting AILC-RL algorithm effectively combines fast convergence of ILC and robustness of RL. We also showed how ILC can be applied for the learning of orientation displacements represented by unit quaternions. In order to increase the exploration ability of standard ILC, we applied AILC with gain adaptation policy. Perhaps the most important feature of the proposed algorithm is that it always results in a stable policy, even though the original AILC algorithm can exhibit learning instabilities. This property was obtained by adding RL-based optimization layer to the algorithm, which constantly monitors and optimizes feedforward compensations signals, AILC gains, and tracking errors. Consequently, the proposed learning algorithm does not require careful parameter tuning to assure stable performance. Here we note that the applied RL algorithm generates the new and improved policies based on search provided by AILC.

Our goal was to assure compliant behavior during adaptation. For this reason we applied impedance controller with admittance-based force feedback. The proposed learning approach was validated in a bi-manual glass wiping experiment.

References

[1] S. Thrun and T. Mitchell, "Lifelong robot learning," *Robotics and Autonomous Systems*, vol. 15, no. 1-2, 1995.

[2] J. Peters, K. Mülling, J. Kober, D. Nguyen-Tuong, and O. Krömer, *Towards Motor Skill Learning for Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 469–482.

[3] D. Nguyen-Tuong and J. Peters, "Model Learning for Robot Control: A Survey," *Cognitive Processing*, vol. 12, no. 4, pp. 319–340, 2011.

[4] S. Schaal, C. Atkeson, and S. Vijayakumar, "Scalable techniques from nonparametric statistics for real time robot learning," *Applied Intelligence*, vol. 17, no. 1, pp. 49–60, 2002.

[5] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 800–815, 2010.

[6] 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.

[7] M. Norrlöf and S. Gunnarsson, "Experimental comparison of some classical iterative learning control algorithms," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 636–641, 2002.

[8] A. Tayebi, "Adaptive iterative learning control for robot manipulators," *Automatica*, vol. 40, no. 7, pp. 1195–1203, 2004.

[9] P. R. Ouyang, B. a. Petz, and F. F. Xi, "Iterative Learning Control With Switching Gain Feedback for Nonlinear Systems," *Journal of Computational and Nonlinear Dynamics*, vol. 6, no. 1, 2011.

[10] M. Norrlöf, "An adaptive iterative learning control algorithm with experiments on an industrial robot," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 245–251, 2002.

[11] J.-X. Xu, S. K. Panda, and T. H. Lee, *Real-time Iterative Learning Control*. London, UK: Springer, 2009.

[12] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational Space Control: A Theoretical and Empirical Comparison," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[13] B. Nemec, L. Žlajpah, and D. Omrčen, "Comparison of null-space and minimal null-space control algorithms," *Robotica*, vol. 25, no. 5, pp. 511–520, 2007.

[14] N. Hogan and S. P. Buerger, "Impedance and Interaction Control," *Robotics and automation handbook*, pp. 19.1–19.23, 2005.

[15] A. Ude, "Filtering in a unit quaternion space for model-based object tracking," *Robotics and Autonomous Systems*, vol. 28, no. 2-3, pp. 163–172, 1999.

[16] A. Ude, B. Nemec, T. Petrič, and J. Morimoto, "Orientation in Cartesian space dynamic movement primitives," in *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014, pp. 2997–3004.

[17] S. Chiaverini, B. Siciliano, and L. Villani, "A survey of robot interaction control schemes with experimental comparison," *IEEE/ASME Transactions on Mechatronics*, vol. 4, no. 3, pp. 273–285, 1999.

[18] S. Jung, T. Hsia, and R. Bonitz, "Force Tracking Impedance Control of Robot Manipulators Under Unknown Environment," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 474–483, 2004.

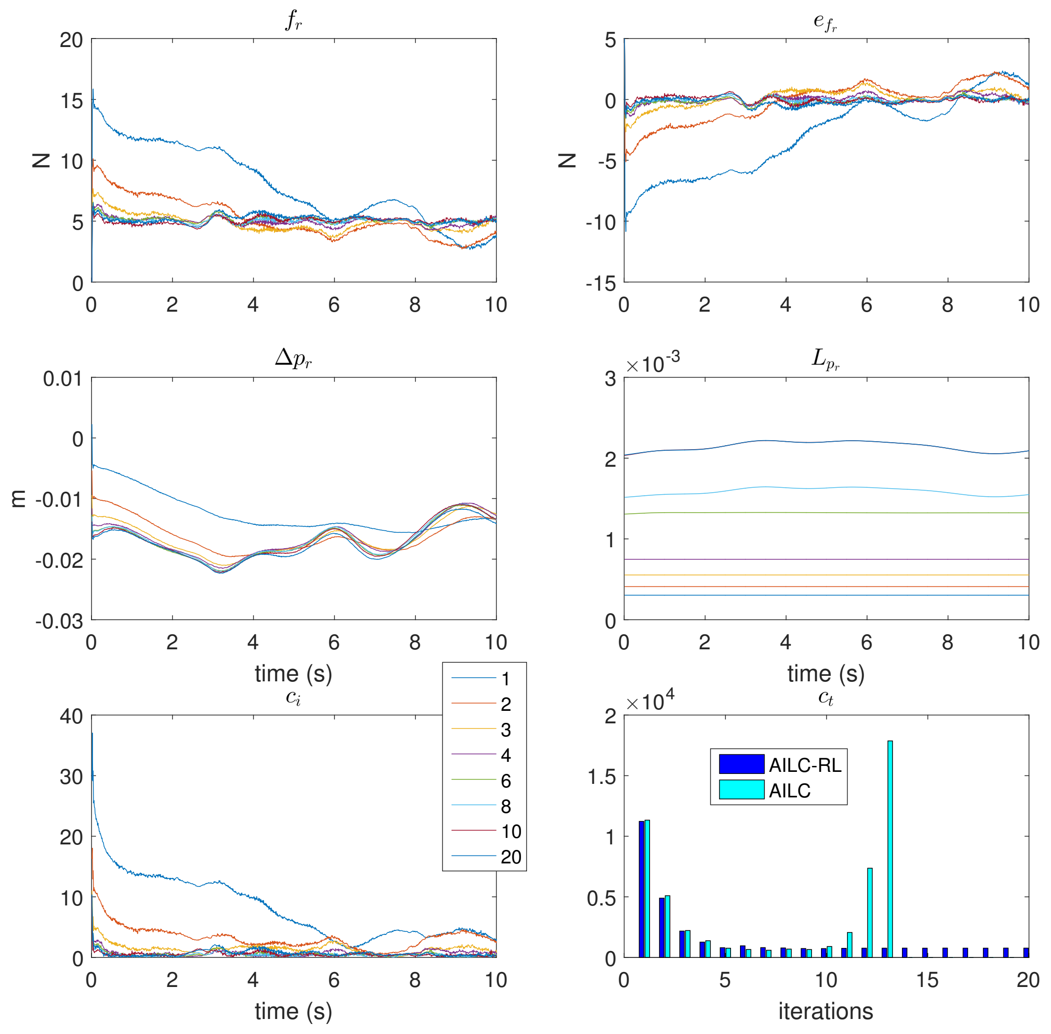

Fig. 5. Learning results during glass wiping. Legend denotes adaptation cycles. Results are shown for radial positional (r) relative coordinate.

[19] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 43–53, 1987.

[20] L. Villani and J. De Schutter, "Force control," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer, 2008, pp. 161–185.

[21] P. R. Ouyang, W. J. Zhang, and M. M. Gupta, "An adaptive switching learning control method for trajectory tracking of robot manipulators," *Mechatronics*, vol. 16, no. 1, pp. 51–61, 2006.

[22] F. J. Abu-Dakka, B. Nemec, J. A. Jørgensen, T. R. Savarimuthu, N. Krüger, and A. Ude, "Adaptation of manipulation skills in physical contact with the environment to reference force profiles," *Autonomous Robots*, vol. 39, no. 2, pp. 199–217, 2015.

[23] K. L. Moore, Y. Chen, and H.-S. Ahn, "Iterative Learning Control: A Tutorial and Big Picture View," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 2352–2357.

[24] A. De Luca, G. Paesano, and G. Ulivi, "A frequency-domain approach to learning control: implementation for a robot manipulator," *IEEE Transactions on Industrial Electronics*, vol. 39, no. 1, pp. 1–10, 1992.

[25] J. Kober, J. a. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

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

[27] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Machine Learning*, vol. 84, no. 1-2, pp. 171–203, 2010.

[28] H.-S. Ahn, "Reinforcement learning and iterative learning control: Similarity and difference," in *IEEE International Conference on Mechatronics and Information Technology*, 2009, pp. 422–424.

[29] B. Hannaford and J. H. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, 2002.

[30] D. Ryu, J. B. Song, J. Choi, S. Kang, and M. Kim, "Frequency domain stability observer and active damping control for stable haptic interaction," in *IEEE International Conference on Robotics and Automation (ICRA)*, Rome, Italy, 2007, pp. 105–110.

[31] F. Dimeas and N. Aspragathos, "Online Stability in Human-Robot Cooperation with Admittance Control," *IEEE Transactions on Haptics*, vol. 9, no. 2, pp. 267–278, 2016.

[32] F. Hlawatsch and F. Auger, *Time-Frequency Analysis*. John Wiley & Sons, 2010.

[33] F. Stulp and O. Sigaud, "Path Integral Policy Improvement with Covariance Matrix Adaptation," in *29th International Conference on Machine Learning (ICML)*, 2012, pp. 281–288.

[34] B. Nemec, T. Petrič, and A. Ude, "Force adaptation with recursive regression iterative learning controller," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015, pp. 2835–2841.

[35] B. Nemec, N. Likar, A. Gams, and A. Ude, "Bimanual human robot cooperation with adaptive stiffness control," in *16th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Cancun, Mexico*, 2016, pp. 607–613.

[36] F. Caccavale, P. Chiacchio, and S. Chiaverini, "Task-space regulation of cooperative manipulators," *Automatica*, vol. 36, no. 6, pp. 879–887, 2000.