This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2748236, IEEE Transactions on Industrial Informatics

Iterative Learning Procedure with Reinforcement for High-Accuracy Force Tracking in Robotized Tasks

Loris Roveda, *Member, IEEE,* Giacomo Pallucca, Nicola Pedrocchi, *Member, IEEE,* Francesco Braghin, *Member, IEEE,* and Lorenzo Molinari Tosatti, *Member, IEEE*

***Abstract*—The paper focuses on industrial interaction robotics tasks, investigating a control approach involving multiples learning levels for training the manipulator to execute a repetitive (partially) changeable task, accurately controlling the interaction. Based on compliance control, the proposed approach consists in two main control levels: i) iterative friction learning compensation controller with reinforcement and ii) iterative force-tracking learning controller with reinforcement. The learning algorithms relies on the iterative learning and reinforcement learning procedures to automatize the controllers parameters tuning. The proposed procedure has been applied to an automotive industrial assembly task. A standard industrial UR 10 Universal Robot has been used, equipped by a compliant pneumatic gripper and a force/torque sensor at the robot end-effector.**

***Index Terms*—Learning Procedures, Dynamics Compensation, Interaction Control, Impedance Control.**

I. Introduction

Nowadays, light-weitght robotics applications are increasingly devoted to improve automation and flexibility in high-accuracy industrial autonomous interaction tasks [1]. Moreover, many markets demands are commonly requiring customizable products with common features [2]. To achieve high production performance, the robotic system has to automatically adapt its behavior to (partially) new products/environments. In fact, it is not reasonable for a human operator to manually set the robot behaviour for each target task, *i.e.*, a directly teaching the manipulator. Target industrial interaction applications with high-accuracy requirements [3] involve i) an interaction controller with ii) robot dynamics compensation and iii) a force tacking control. While interaction controllers can be implemented by standard compliance controllers [4], robot dynamics compensation and force tacking control require an analysis of both the specific robot modeling and interaction properties in the target executed task. Specifically, robot dynamics compensation is mostly affected by the identification of the joints friction [5] (function of the robot hardware, loads on joints, temperature, etc.). Force tacking control is mostly affected by the interacting environment compliance that affects the stability and the control performance of the global system *controlled robot - interacting environment* [6]. Both the controllers tuning can be automatized by applying the machine learning techniques, avoiding complex estimation/modeling and/or a direct human manual controllers tuning.

A. Learning Procedures in Industrial Robotics Applications

To adapt the manipulator behavior to a new application, learning algorithms are increasingly applied [7]. Three basic classes of learning paradigms can be identified: supervised learning, unsupervised learning and reinforcement learning [8]. Supervised learning is performed under the supervision of an external teacher [9]. Unsupervised learning is performed in a self-organized manner in that no external teacher or critic is required to guide synaptic changes in the network [10]. Reinforcement learning involves the use of a critic that evolves through a trial-and-error process [11]. Considering interaction tasks and the related complexity, the reinforcement learning procedures represent a particularly suitable solution to implement adaptive controllers capable to tune the robot behavior in unknown scenarios (*i.e.*, without a specific modeling), defining simple policies to achieve a target task goal (*i.e.*, force tracking capabilities with force overshoots avoidance).

B. Friction Modeling and Compensation & Learning

Many efforts have been made on the friction modeling (also considering temperature effects) [12], identification and compensation [13] to improve the controlled robot performance [14]. However, such common procedure takes a lot of time to execute experiments and for the data analysis. Moreover, considering adaptive controllers [15] to be robust to friction parameters uncertainties, the robot control bandwidth has to be limited. Learning procedure has been also used to identify and compensate for friction effects [16]. However, such methods are very complex and not easy to implement in a real application. In fact, they have only been applied in simulation scenarios or on 2 degrees of freedom systems by performing simplified tasks, not related to real complex industrial scenarios.

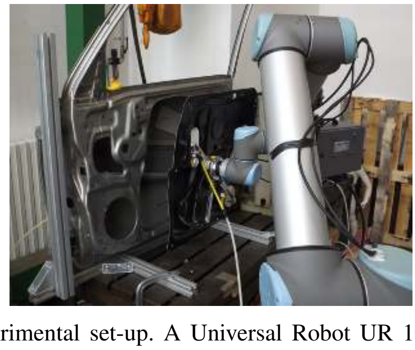


Fig. 1: Experimental set-up. A Universal Robot UR 10 involved in the assembly of the target automotive component.

L. Roveda (loris.roveda@itia.cnr.it), G. Pallucca, N. Pedrocchi and L. M. Tosatti are with the Institute of Industrial Technologies and Automation (ITIA) of National Research Council (CNR), via Corti, 21 - 20133 Milan, Italy

G. Pallucca and F. Braghin are Politecnico di Milano, Department of Mechanical Engineering, via La Masa 1, 20156 Milan, Italy

C. Interaction Control & Learning

Considering compliance controllers, impedance control [4] is particularly effective in order to interact with compliant environments. Learning procedures have also been developed in order to achieve a compliant behavior of the controlled manipulator without any identified model [17]. Learning solutions have also been designed in order to physically teach a task to the robot [18] or to cooperate with humans during interaction tasks [19]. To perform a tracking of a target force based on the impedance control and preserving the properties of the impedance behavior, many works have been presented, adapting the impedance control parameters based on the interaction force. Such works can be divided in two main families: class (a) force-position tracking impedance controllers and class (b) variable impedance controllers [20]. Both the control class (a) and (b) require the definition of a specific interaction model in order to achieve high force-tracking performance, being very related to a target interaction scenario. Some works are also applying learning approaches to establish a stable contact during interaction tasks execution [21], [22], [23], [24]. However, the most of these approaches are only using such learning approaches to define a compliant behavior of the manipulator, without taking into account force-tracking capabilites and the important problem of force overshoots is not considered (critical issue when the robot is manipulating a fragile-delicate component). Additionally, many works are only proposing simulation results (proposed approaches are too difficult to be implemented on real robots).

D. Paper Contribution

From state-of-the-art methods analysis no approaches for the automation of (partially) unknown real complex industrial interaction tasks (combining learning methodologies at both low and high control level) for the optimization of the robot behaviour have been found. The main issue is related to the force overshoots avoidance (critical for interaction tasks). In order to automatize the execution of a repetitive (partially) changeable interaction task, the proposed compliance control based approach consists in two main control levels: an i) iterative friction learning algorithm with reinforcement and ii) an iterative force tracking learning algorithm with reinforcement. Following the approach described in [21], the main idea of this paper is to combine iterative learning controllers [25] with the reinforcement learning procedures. Therefore, a policy that evolves through a trial-and-error process is defined for both the learning procedures, guiding the evolution of each control parameter. More in details, the iterative friction learning algorithm allows to improve the dynamics compensation low level control. The main purpose of the method is to define a friction parameter learning procedure for manipulators involved in a real cluttered industrial environment, in which the motion of the robot is limited in a small region of its working space. In fact, lightweight manipulators might be involved in many different applications (*e.g.*, with changeable payload) and might be moved in different locations inside the plant with different limitations on their working space. Therefore, the definition of a task-dependent friction parameters learning procedure (firstly performed in free-space, and then executed at each repetition of the task to indirectly take into account modifications of the friction parameters related to any change affecting the task execution - *e.g.*, joint temperature) in relation to a specific Cartesian task allows to overcome the definition of general optimized identification joint trajectories. The iterative force tracking learning algorithm allows to improve the force-tracking high level control. The proposed method is defined to iteratively learn the proportional force-tracking control gain and the Cartesian damping gain to achieve an overdamped dynamics of the coupled controlled robot - environment system without any estimation of the environment parameters. The proposed procedure has been applied to an industrial assembly task (compliant automotive component assembly, Figure 1). A standard industrial UR 10 Universal Robot has been used, equipped by a force/torque sensor at the robot end-effector. Experimental results show the obtained performance in both the proposed approaches, improving the friction compensation (with respect to the local identification described in Section VI) and guaranteeing force overshoots free tracking capabilities.

II. Robot Modeling and Cartesian Impedance Control

To design and implement the proposed learning procedure on the UR 10 Universal Robot manipulator, the global dynamics identification and compensation is needed, together with the Cartesian impedance control. Many works have covered the robot dynamic parameters identification procedure [26]. Having the following manipulator dynamics [27]:

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{h}_{f,q}(\dot{\mathbf{q}}) = \boldsymbol{\tau} - \mathbf{J}(\mathbf{q})^T \mathbf{h}_{ext} \quad (1)$$

where, $\mathbf{B}(\mathbf{q})$ is the robot inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is the robot Coriolis vector, $\mathbf{g}(\mathbf{q})$ is the robot gravitational vector, $\mathbf{h}_{f,q}(\dot{\mathbf{q}})$ is the robot joint friction vector, $\mathbf{q}$ is the robot joint position vector, $\mathbf{J}(\mathbf{q})$ is the robot Jacobian matrix, and $\mathbf{h}_{ext}$ is the robot external force/torque vector, $\boldsymbol{\tau}$ is the robot joint torque vector, the parameters to be identified are the link masses (l parameters, where l is the number of robot links) and centers of mass ($3 \times l$ parameters) parameters, the inertia parameters ($9 \times l$ parameters) and the friction parameters ($m \times l$ parameters, where m is the number of friction parameters from the friction model) - See Section VI-A.

Based on the identified dynamics in (1) it is possible to design the Cartesian impedance control [27] (as described in Section VI-B), obtaining an equivalent mass-spring-damper dynamic behavior of the controlled manipulator:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} + \mathbf{K}\Delta\mathbf{x} = \mathbf{h}_{ext} \quad (2)$$

where $\mathbf{M}$, $\mathbf{D}$, $\mathbf{K}$ are the impedance mass, stiffness and damping matrices composed by both the translational and rotational parts, and $\Delta\mathbf{x}$, $\dot{\mathbf{x}}$, $\ddot{\mathbf{x}}$ are, respectively, the Cartesian position, velocity and acceleration of the controlled manipulator composed by both the translational and rotational parts. On the basis of the developed controller, the learning algorithm for the friction compensation optimization and for the force-tracking optimization can be designed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2748236, IEEE Transactions on Industrial Informatics

III. Friction Parameters Iterative Learning Procedure with Reinforcement

A. Iterative Learning Procedure with Reinforcement Design

In the following, an iterative learning controller with reinforcement is defined. The policy guiding the whole procedure is described, relying on the robot states (the Cartesian positions/orientations) and on a target reward function to track for procedure improvements in the task execution, allowing to iterative learn the friction parameters $\mathbf{F}_p$ independently for each joint, on the basis of a defined friction model $\mathbf{h}_{f,q}(\dot{\mathbf{q}})$.

In order to define the target policy for the friction parameters learning, the error function $\mathbf{E}$ is introduced. Such error function is based on the calculation of the Cartesian position tracking error (obtained by comparing the target action execution $\mathbf{x}^d$ with the target system state $\mathbf{x}$):

$$\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}^d \tag{3}$$

(3) is used to evaluate the equivalent Cartesian friction force/torque matrix $\mathbf{h}_{f,c}$. In fact, the friction deforms the robot Cartesian trajectory during the task execution equivalently to the external force/torque vector:

$$\mathbf{h}_{f,c} = -\mathbf{K}\Delta\mathbf{x} \tag{4}$$

The Cartesian position tracking error (3) is multiplied by the impedance control stiffness matrix $\mathbf{K}$, returning an estimation of the friction action in the Cartesian space. To evaluate the friction torque to be compensated at the joint side, Cartesian friction force/torque vector $\mathbf{h}_{f,c}$ can be projected in the joint space through the Jacobian matrix $\mathbf{J}(\mathbf{q})$. The error matrix $\mathbf{E}$ can be expressed as:

$$\mathbf{E} = \mathbf{J}^T(\mathbf{q})\,\mathbf{h}_{f,c} \tag{5}$$

The error matrix $\mathbf{E}$ has size $[l,T]$ (where l is the number of robot joints, T is the total samples of a target reinforcement learning procedure iteration). The error matrix $\mathbf{E}$ is calculated at each iteration k (and overwritten) of the proposed learning procedure. The definition of the error matrix $\mathbf{E}$ in (5) allows a faster update of the friction parameters related to the most excited joints, based on the target task directions. Unexcited task direction results in a joint contribution to the friction parameters update that $\longrightarrow 0$.

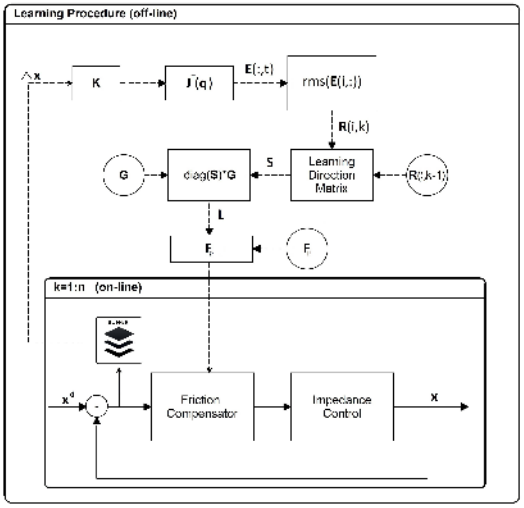


Fig. 2: Friction learning schema is detailed, highlighting the use of the updated friction parameters in the control law.

Such estimation of the friction torques at the joint level can be used to define the reinforcement signal $\mathbf{R}$ as the root mean square (rms) of (5). In such a way, each joint friction parameters are independently updated through the reinforcement learning iterations based on:

$$\mathbf{R}(i,k) := rms\,(\mathbf{E}(i,:)) = \sqrt{\frac{1}{T}\sum_{t=0}^{T}\mathbf{E}(i,:)^2} \tag{6}$$

Reinforcement signal $\mathbf{R}$ has size $[l,n]$ (where n is the maximum number of iterations of the learning procedure). The learning direction matrix $\mathbf{S}$ for the actual learning iteration is evaluated as follows:

$$\mathbf{S} := sign\,(\mathbf{R}(i,k) - \mathbf{R}(i,k-1)) \tag{7}$$

The learning direction matrix $\mathbf{S}$ has size $[l,1]$. The learning direction matrix is calculated at each iteration k (and overwritten) of the proposed learning procedure. Therefore, the learning matrix $\mathbf{L}$ is updated, taking into account the learning gain matrix $\mathbf{G}$:

$$\mathbf{L} := diag\,(\mathbf{R}(:,k))\,diag\,(\mathbf{S})\,\mathbf{G} \tag{8}$$

The learning matrix $\mathbf{L}$ has size $[l,m]$ (where m is the number of friction parameters for each joint, depending on the target friction model). The learning matrix is calculated at each iteration k (and overwritten) of the proposed learning procedure. Friction parameters $\mathbf{F}_p$ can, therefore, be updated at each iteration (and overwritten) on the basis of the following policy:

$$\begin{cases} if\ k=0\ \mathbf{F}_p = \mathbf{F}_p^0 \\ if\ k>0\ \mathbf{F}_p \leftarrow \mathbf{F}_p - \mathbf{L} \end{cases} \tag{9}$$

Friction parameters $\mathbf{F}_p$ has size $[l,m]$. $\mathbf{F}_p^0$ are the initialized friction parameters at iteration $k=0$. It has to be underline that in the proposed formulation the friction parameters $\mathbf{F}_p$ on the left side of the second equation are the updated parameters at the k iteration on the basis of the previously calculated parameters $\mathbf{F}_p$ on the right side at iteration $k-1$. Updated friction parameters are used in (1) to calculate $\mathbf{h}_{f,q}(\dot{\mathbf{q}})$ in each k iteration of the target task execution. Such procedure allows to test the updated friction parameters and to calculate the next update parameters set.

A global reward function $\mathbf{P}$ can be defined in order to evaluate the global improvements during the learning procedure execution for each iteration:

$$\mathbf{P}(k) := \frac{1}{l}\sum_{i=1}^{l}\mathbf{R}(i,k) \tag{10}$$

Figure (2) synthesizes the described learning procedure for the friction parameters update. The internal on-line loop is active during the task execution and it allows to collect the required data at each time step t (*i.e.*, the Cartesian position tracking error $\Delta\mathbf{x}$ and the robot Jacobian $\mathbf{J}$) in order to calculate the error function $\mathbf{E}$. The external off-line loop allows to evaluate the defined policy in (9) and the reward function $\mathbf{P}$ guiding the evolution of the learned friction parameters (and, consequently, of the friction compensation). The procedure succeeds when the global reward function is lower than the target tolerance ρ. The procedure stops if the maximum number of iterations n is reached. If instabilities or divergence arise in the k iteration (*i.e.*, overcompensated friction), learning gain matrix $\mathbf{G}$ is reduced by a factor λ and the iteration $k+1$ will use $\mathbf{F}_p = \lambda\mathbf{F}_p$ (where $\mathbf{F}_p$ on the right side are the friction parameters calculated in the $k-1$ iteration). In such a way, a divergence of the procedure is stopped and the algorithm is able to continue. An additional condition is included in order to stop the procedure after a number σ of consecutive iterations without any improving in the friction compensation with respect to the minimum calculated value of $\mathbf{P}(k)$. The optimal set of friction parameters (*i.e.*, minimizing (10)) will be selected at the end of the procedure.

B. Iterative Learning Procedure with Reinforcement Convergence and Stability

Since the proposed algorithm is not based on classical reinforcement learning paradigms, its convergence and stability can be shown as in the following. Considering the robot dynamics in (1), considering that no external forces are applied to the manipulator (*i.e.*, $\mathbf{h}_{ext} = \mathbf{0}$, and considering that the gravity and Coriolis terms ($\mathbf{g}(\mathbf{q})$ and $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$) are perfectly compensated by the control torque, equation (1) can be written as follows:

$$\ddot{\mathbf{q}} = \mathbf{B}(\mathbf{q})^{-1}\left(\mathbf{B}(\mathbf{q})\tau_{imp} + \bar{\mathbf{h}}_{f,q}(\dot{\mathbf{q}}) - \mathbf{h}_{f,q}(\dot{\mathbf{q}})\right) \tag{11}$$

where τ_{imp} results from the impedance control (VI-B) and $\bar{\mathbf{h}}_{f,q}(\dot{\mathbf{q}})$ is related to the friction compensation torque on the basis of the learned friction parameters in (9) at the k iteration of the learning process.

Considering the exponential friction model characterized by four parameters for each joint (where $\boldsymbol{\eta} = [\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \boldsymbol{\eta}_3, \boldsymbol{\eta}_4]$ is the real friction parameters matrix with size $[l,m]$, same model used in the experimental results Section V-A), $\bar{\mathbf{h}}_{f,q}(\dot{\mathbf{q}})$ at iteration k can be written as follows:

$$\begin{aligned}\bar{\mathbf{h}}_{f,q}(\dot{\mathbf{q}}) &= diag(\mathbf{F}_{p1})\dot{\mathbf{q}} \\ &+ sign(\dot{\mathbf{q}})\left(diag(\mathbf{F}_{p2}) - diag(\mathbf{F}_{p3})\right)e^{-diag(\mathbf{F}_{p4})\dot{\mathbf{q}}}\end{aligned} \tag{12}$$

where $\mathbf{F}_p = \mathbf{F}_p^0 + \sum^{\bar{k}} diag\left(\mathbf{S}\right)\mathbf{G} = [\mathbf{F}_{p1}, \mathbf{F}_{p2}, \mathbf{F}_{p3}, \mathbf{F}_{p4}]$ is the learned friction parameters matrix at $k = \bar{k}$.
Substituting (12) in (11):

$$\begin{aligned}\ddot{\mathbf{q}} = \mathbf{B}(\mathbf{q})^{-1}(&\mathbf{B}(\mathbf{q})\tau_{imp} + \Delta\mathbf{F}_{p,1}\dot{\mathbf{q}} \\ &+ sign(\dot{\mathbf{q}})\left(\Delta\mathbf{F}_{p,2} - \Delta\mathbf{F}_{p,3}\right)e^{-\Delta\mathbf{F}_{p,4}\dot{\mathbf{q}}})\end{aligned} \tag{13}$$

where $\Delta\mathbf{F}_{p,1} = diag(\mathbf{F}_{p1}) - diag(\boldsymbol{\eta}_1)$, $\Delta\mathbf{F}_{p,2} = diag(\mathbf{F}_{p2}) - diag(\boldsymbol{\eta}_2)$, $\Delta\mathbf{F}_{p,3} = diag(\mathbf{F}_{p3}) - diag(\boldsymbol{\eta}_3)$, $\Delta\mathbf{F}_{p,4} = diag(\mathbf{F}_{p4}) - diag(\boldsymbol{\eta}_4)$.
By imposing that the acceleration $|\ddot{\mathbf{q}}| < \ddot{\mathbf{q}}^{max}$ is limited (the control action is stable, *i.e.*, the estimation of the friction parameters is converging), it is clear from (13) that the initialization of the friction $\mathbf{F}_p^0$ and the learning gain matrix $\mathbf{G}$ are affecting the convergence of the proposed algorithm. While $\mathbf{F}_p^0 = 0$ can be imposed, the selection of the learning gain matrix $\mathbf{G}$ is crucial in order to define the convergence and the velocity of the evolution of the learned friction parameters. Therefore, the proposed solution for arising instabilities/divergence limiting $\mathbf{G}$ allows to select reasonable values to guarantee a required sensibility of the method during the learning task. Guidelines for the definition of such learning gain matrix are givenn in Section V-A. It has to be underlined that the stability of the method is valid for any friction model.

Remark 1. While the described methods takes as an input a Cartesian target trajectory, the methodology projects the Cartesian quantities in the joint space. Therefore, the method is valid for any Cartesian trajectory required by a target task.

IV. Iterative Learning Procedure with Reinforcement Applied to Force-Tracking Impedance Control Tuning

As described in [28], the *impedance controlled robot - environment* interaction can be modeled as follows:

$$\mathbf{M}\ddot{\mathbf{x}} + (\mathbf{D} + \mathbf{D}_e)\,\dot{\mathbf{x}} + (\mathbf{K} + \mathbf{K}_e)\,\mathbf{x} = \mathbf{K}\mathbf{x}^d \tag{14}$$

where $\mathbf{D}_e$ is the interacting environment damping, and $\mathbf{K}_e$ is the interacting environment stiffness. Such equation defines the dynamics of the coupled controlled robot - interacting environment and it is used to develop the proposed control strategy for the force-tracking application.
Since the Cartesian impedance control decouples the dynamics of the controlled manipulator and under the assumption that only small rotations are allowed in the task execution (as in many industrial tasks, such as polishing, assembly, disassembly tasks), it is possible to consider only one translational DoF to design the reinforcement learning based procedure. The algorithm can be extended to all the cartesian translational DoFs without loss of generality.
The impedance control set-point x^d can be on-line calculated to track a target force f^d. The simplest impedance control set-point definition results in:

$$x^d = x_e^0 + G_p K^{-1} f^d - G_d K^{-1}\dot{x} \tag{15}$$

x_e^0 is the equilibrium position of the target environment.
The force-tracking control gain G_p allows to perform the force tracking task in the presence of compliance in the interacting environment. In fact, imposing $G_p = 1$ and substituting (15) in (14), the steady state robot position results in:

$$x = \frac{f^d}{K + K_e}$$

Considering an infinite rigid equivalent interacting elastic system (*i.e.*, $K_e \to 0$) and $x_e^0 = 0$ the measured force at the robot end-effector $f = -K(x - x^d) \rightarrow f^d$ since $x \rightarrow 0$, while considering a compliant equivalent interacting elastic system (*i.e.*, $K_e = \overline{K}_e$) the measured force at the robot end-effector $f = -K(x - x^d) = f^d - \dfrac{K f^d}{K + \overline{K}_e} < f^d$.

The additional damping control gain G_d allows to increase the coupled system damping, achieving the critically-dampend/overdamped behavior. To avoid any force overshoot during the task execution, the additional damping gain G_d parameter has to be properly shaped, satisfying (see [20]):

$$(D + D_e + G_d)^2 - 4M(K + K_e) \geq 0 \tag{16}$$

Therefore, to avoid any force overshoot during the task execution while tracking a target force to achieve a target goal, the force-tracking control gain G_p has be properly imposed to compensate for any elasticity of the interacting environment and the additional damping gain G_d has to be properly shaped to obtain a critically-damped/overdamped coupled system (14). To face such problem in a (partially) unknown scenario, the learning procedure on the control gain G_p and on the additional damping gain G_d parameters is described in the following.

1) Force-Tracking Control Gain G_p Iterative Learning Procedure with Reinforcement: To avoid any estimation of the environment parameters $\{K_e, D_e\}$ (particularly critical in complex industrial scenarios such as the EuRoC Project, challenge 1, benchmarking task [29]) an automatic tuning of the control gain G_p allows to proper track the target force f^d.

The proposed iterative learning controller with reinforcement is defined to adapt the force-tracking control gain G_p. The policy guiding the whole procedure is described, relying on the robot states (the measured Cartesian interaction force at the steady state f_s) and on a target reward function to track for procedure improvements in the task execution. The reinforcement signal R_{G_p} can be defined as:

$$R_{G_p} = \frac{f^d}{f_s} - 1 \tag{17}$$

The force tracking control gain G_p can be, therefore, updated for each iteration through the following policy on the basis of the reinforcement signal R_{G_p}:

$$G_p = G_p + G_{G_p} R_{G_p} \tag{18}$$

where G_{G_p} is the reinforcement gain to speedup the learning procedure. Therefore, the force-tracking control gain G_p can be iteratively updated to track the target interaction force. A global reward function P_{G_p} can be defined in order to evaluate the global improvements during the learning procedure execution for each iteration:

$$P_{G_p}(k) := (f^d - f_s)/f^d \tag{19}$$

The force-tracking control gain G_p is not affecting the stability of the system (see (14), (15)) since it has constant value during the task execution.

2) Damping D Parameter Iterative Learning Procedure with Reinforcement: Since no estimates of the environment parameters $\{K_e, D_e\}$ are available to analytically impose the additional damping gain G_d in (16), such control gain has to be iteratively learned. The additional damping gain G_d can be written as:

$$G_d = 2\xi M \sqrt{K/M} \tag{20}$$

where ξ is the additional damping ratio. On-line calculating ξ_c (coupled damping ratio of (14)) by the logarithmic decrement methodology [30] (measuring consecutive force picks f_p^1, f_p^2):

$$\xi_c = \frac{\delta}{\sqrt{\delta^2 + 4\pi^2}} \tag{21}$$

where $\delta = \log\left(f_p^1/f_p^2\right)$, the proposed iterative learning controller with reinforcement is defined to adapt the additional damping gain G_d. The policy guiding the update of the additional damping ratio ξ is described, relying on the robot states (the measured Cartesian interaction force) and on a target reward function to track for procedure improvements in the task execution. A reinforcement signal R_ξ can be defined as:

$$R_\xi = 1 - \xi_c \tag{22}$$

The additional damping ratio ξ can be, therefore, updated for each iteration (to calculate the additional damping gain G_d as in (20)) through the following policy on the basis of the reinforcement signal R_ξ:

$$\xi = \xi + G_\xi R_\xi \tag{23}$$

where G_ξ is the reinforcement gain to speedup the learning procedure. A global reward function P_ξ can be defined in order to evaluate the global improvements during the learning procedure execution for each iteration:

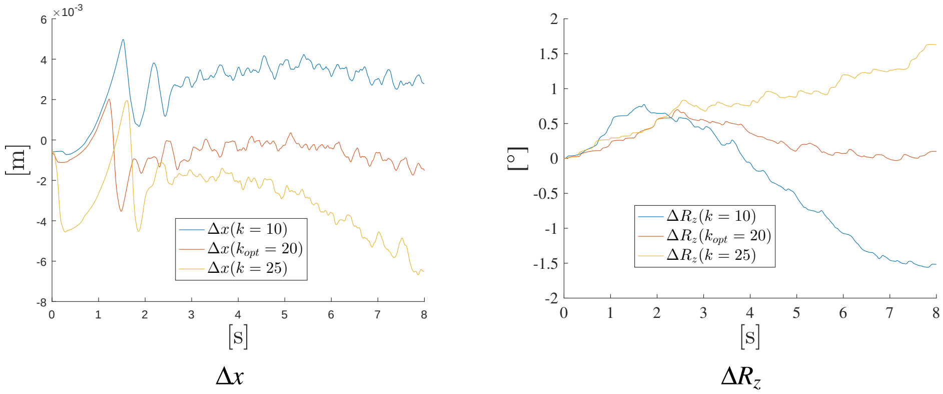
$$P_\xi(k) := (f^d - f_{max})/f^d \tag{24}$$

where f_{max} is the maximum measured force during the task execution. The additional damping gain G_d is not affecting the stability of the system (see (14), (15)) since it has constant value during the task execution and it only increases the closed-loop damping.

Remark 2. Since the impedance control decouples the Cartesian DoFs, the method can be applied independently to each interaction direction/multi-contact scenario.

Remark 3. Exit strategies from the proposed learning procedure can be defined. A target force tracking reward function value $P_{G_p}^t = \pm 0.1$ is defined (*i.e.*, the admissible difference between equilibrium force and target force is 10%) and a target damping reward function value $P_\xi^t = 0.05$ is defined (*i.e.*, the admissible difference between maximum force and target force is 5%). As soon as the reward functions (19), (24) satisfy the exit conditions the learning procedure is concluded. Achived performance are related to $P_{G_p}^t = \pm 0.05$ and $P_\xi^t = 0.025$ (limited by the control frequency and hardware properties). Such performance are however comparable with state-of-the-art methods in terms of bandwidth, tracking error, steady state oscillations (amplitude $< 0.5[N]$) [20], [31].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2748236, IEEE Transactions on Industrial Informatics



Δx

ΔR_z

Fig. 3: The x position tracking error Δx and the z rotation tracking error ΔR_z are shown for iterations $k = [10, 20, 25]$.

V. Experimental Results

Experimental validations of the proposed methods have been carried out using as a test platform a Universal Robot UR10. Its controller allows to torque control the manipulator and to record the required data (robot joint and Cartesian positions and velocities, robot Jacobian) with a control/sampling frequency of 125 Hz. The equipped force sensor allows to record the interaction forces/torques.

A. Friction Compensation Experiment

The selected friction model has been the exponential model. Such friction model is characterized by four parameters for each joint (*i.e.*, $m = 4$, with parameters $\boldsymbol{\eta}_1$, $\boldsymbol{\eta}_2$, $\boldsymbol{\eta}_3$, $\boldsymbol{\eta}_4$). The friction torque acting on the i^{th} joint can be expressed as:

$$h_{f,q}(\dot{q}_i) = \eta_{1,i}\dot{q}_i + sign(\dot{q}_i)\,(\eta_{2,i} - \eta_{3,i})e^{-(\eta_{4,i}|\dot{q}_i|)} \qquad (25)$$

A free-motion task along the Cartesian direction x has been selected as a target task. Such task is representative of an insertion task to be performed in impedance control. The friction learning has to be performed without any external force/torque acting on the manipulator. Once the learning procedure is completed, the friction can be compensated in the interaction task to improve the controller performance.

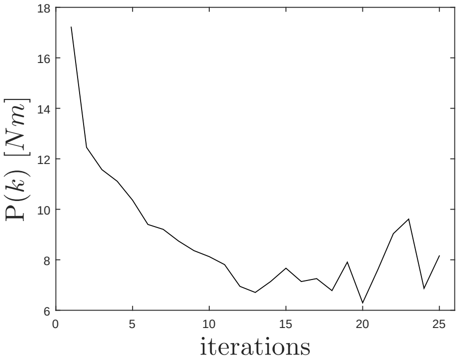


Fig. 4: Reward function P through the learning procedure. Optimal iteration $k_{opt} = 20$.

Only the x component of $\mathbf{x}^d$ is therefore defined as a cycloidal function, keeping other components constant:

$$x^d = x^0 + h\left(c - \frac{\sin(2\pi c)}{2\pi}\right)$$

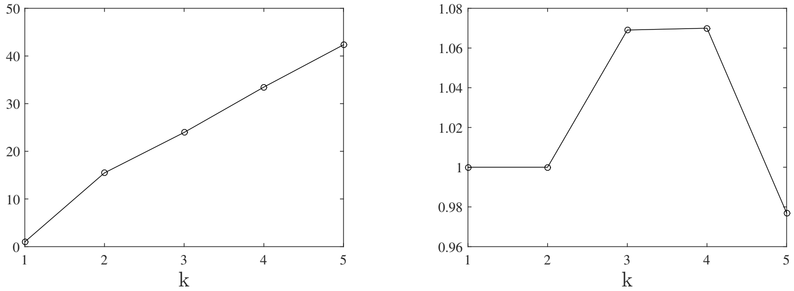
with h the length of the target motion and $c = 0 : 0.001 : 1$. Mass matrix has been set as $\mathbf{M} = diag([5, 5, 5, 5, 5, 5])$, stiffness matrix as $\mathbf{K} = diag([5000, 5000, 5000, 250, 250, 250])$, and damping ratio vector as $\boldsymbol{\xi} = [0.7, 0.7, 0.7, 0.7, 0.7, 0.7]$. The diagonal element $\mathbf{D}(i,i)$ of the damping matrix is calculated as $\mathbf{D}(i,i) = 2\boldsymbol{\xi}(i)\mathbf{M}(i,i)\sqrt{\mathbf{K}(i,i)/\mathbf{M}(i,i)}$.
The learning gain $\mathbf{G}$ can be imposed by measuring the reinforcement signal $\mathbf{R}$ in (6) at iteration $k = 0$ by imposing $\mathbf{F}_p^0 = \mathbf{0}$. The following definition of $\mathbf{G}$ can be used:

$$\mathbf{G} = diag(\mathbf{R}(:,0))^{-1}\overline{\mathbf{F}}_p\kappa_f$$

where $\overline{\mathbf{F}}_p$ can be a rough estimation of the friction parameters from literature and κ_f is a coefficient experimentally defined in the range $[0.01, 0.1]$ allowing to have a good estimation resolution during the learning procedure. The reducing factor can be imposed as $\lambda = 0.5$ in order to increase the estimation resolution when instabilities or divergences arise. First iteration of the performed task is executed imposing $\mathbf{F}_p$ on the basis of results in Table IV. Subsequent iterations adapt friction parameters matrix $\mathbf{F}_p$ based on (9). Considering the selected task direction x, two Cartesian DoFs are mostly excited: translation x and rotation R_z.
Figure 3 shows the x position error Δx and the z rotation error ΔR_z during the task execution, considering different iterations of the learning procedure. Such data have been recorded as

TABLE I: Learned optimal friction parameters.

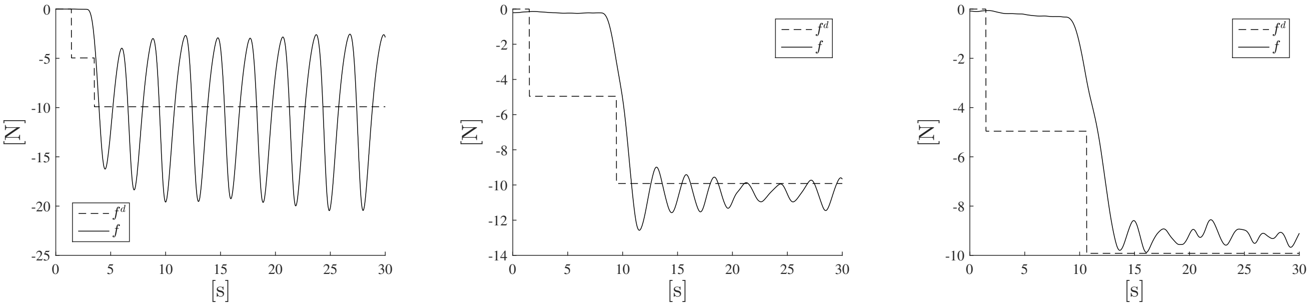
	Optimal Set of Friction Parameters			
	$\mathbf{F}_{p1j}[Ns/m]$	$\mathbf{F}_{p2j}[Ns/m]$	$\mathbf{F}_{p3j}[Ns/m]$	$\mathbf{F}_{p4j}[s/rad]$
joint 1	37.40	11.90	9.91	$3.9755 \cdot 10^6$
joint 2	157.49	31.28	26.52	$7.3053 \cdot 10^6$
joint 3	25.98	8.14	6.02	$5.6843 \cdot 10^6$
joint 4	1.95	2.99	1.85	$4.2840 \cdot 10^6$
joint 5	8.35	10.12	4.16	$4.5838 \cdot 10^6$
joint 6	3.49	4.38	1.00	$4.1280 \cdot 10^6$



ξ

G_p

Fig. 5: Learned force-tracking control gain and impedance control damping ratio through iterations.



$k = 1$

$k = 3$

$k = 5$

Fig. 6: Interaction force through iterations. The learning procedure is capable to adapt the control gains to achieve the target performance.

described in Section III-A, Figure 2, by the on-line internal loop. The minimum error iteration is highlighted. The same behavior is shown for the remaining DoFs. Considering iteration $k_{opt} = 20$, the obtained tracking error is comparable with the state-of-the-art compliance controllers with friction compensation [27]. In fact, the defined compliance results in a deviation from the target trajectory during the motion execution. The advantage of the proposed solution is related to the possibility to use a user-defined friction model and locally perform the optimization of its parameters while executing a target Cartesian task. Moreover, the implementation of the proposed method is easily applicable to a real n-DoFs manipulator as shown in the proposed experiment. Figure 4 shows the evolution of the reward function in (10). Table I details the learned optimal friction parameters.

Remark 4. As shown in Figure 4, the reward function P reaches its minimum at iteration $k = 20$. The following increasing of the reward function P (as described in Section III-B) is due to the overcompensation of the friction. The algorithm modifies the learning gain $\mathbf{G}$ in order to correct the friction parameters estimation. Due to the defined stop criteria, the methods ends the learning of the friction parameters at iteration $k = 25$.

B. Force-Tracking Experiment

The learning methods have been applied to an automotive industrial assembly task (Figure 1). The assembly trajectory is the same specified in Section V-A (main assembly direction along x axis). The impedance control parameters have been imposed as follows: $\mathbf{M} = diag([10, 10, 10, 10, 10, 10])$, $\mathbf{K} = diag([100, 100, 100, 10, 10, 10])$. The learning gain for the force-tracking control gain has been imposed equal to $G_{G_p,x} = 1$ (to only use the reinforcement signal R_{G_p} to guide the learning) and the learning gain for the damping ratio has been imposed equal to $G_{\xi,x} = 10$ (to speedup the learning process), while the force-tracking control gain and the damping ratio have been initialized as follows: $G_{p,x}(1) = 1$ (*i.e.*, as described in Section IV, considering $K_e \rightarrow 0$ the target force f^d can be perfectly tracked - $f \rightarrow f^d$), $\xi_x(1) = 1$ (the robot behavior is imposed to be critically damped).

In general, the learning gain $G_{G_p,x}$ for the force-tracking control gain can be imposed ≤ 1 in order to increase the estimation resolution during the learning process. From experimental validation, the suggested range for such parameter is $[0.1, 1]$. The learning gain for the damping ratio $G_{\xi,x}$ can be imposed on the basis of the estimation of the target interacting environment damping ratio $\overline{\xi}_e$. In fact, on the basis of the environment properties (*e.g.*, material), an estimation can be performed from literature. The learning gain for the damping ratio can be calculated as $G_{\xi,x} = \overline{\xi}_e \kappa_\xi$, where κ_ξ is a coefficient experimentally defined in the range $[1, 20]$.

The task consists in two phases: firstly, the robot finds the assembly surface (detecting a force $f_x > 5N$, then the assembly task is performed (with $f_x^d = 10N$). In the second phase, the learning procedures are performed.

Figure 5 shows the learned force-tracking control gain $G_{p,x}$

and the impedance control damping ratio ξ_x through iterations execution. While the force-tracking control gain $G_{p,x}$ modifies its value based on the locally properties of the equivalent interacting elastic system and based on the estimated environment equilibrium position x_e^0, the damping ratio ξ_x increases through the iterations until no force overshoots are shown in the task execution. Figure 6 shows the obtained interaction force during the learning procedure execution, highlighting the obtained target performance at iteration $k = 5$. Iteration $k = 1$ shows a low-damped behavior of the closed-loop system (14) with the initial value $\xi_x(1) = 1$ on the basis of (16). On the basis of Remark 3, the experimental training of the control parameters is concluded as soon as the reward functions satisfy the defined exit strategies. As shown in Figure 6, iteration $k = 5$ satisfies the defined exit strategies. It is possible to modify the exit strategies parameters in order to improve the obtained performance. It has to be underline that the resulting steady-state oscillations are related to the defined experimental set-up: since the high mass of the target assembly location (*i.e.*, the car door), the backlash in its mounting and the flexibility of the manipulated component, the resulting interaction dynamics is highly non-linear and under-damped. The defined approach allows to compensate for such non-linear under-damped dynamics, having a stable steady-state interaction.

VI. Conclusions

In this paper, a control approach involving multiples learning levels for training the manipulator to execute a repetitive (partially) changeable task, while accurately controlling the interaction force, has been presented. Based on compliance control, the proposed approach consists in two main control levels: i) iterative friction learning compensation controller and ii) iterative force-tracking learning controller. The proposed control procedure has been applied to an automotive industrial assembly task. Experimental results show the validation of the performance for both the proposed learning procedures, improving the friction compensation (with respect to the local identification described in Section VI) and guaranteeing force overshoots free tracking capabilities. Ongoing work is devoted to include a compliant robot mounting. Its dynamics will affect the target task dynamics and the interaction at the robot end-effector. Moreover, external force sensor will be replaced by the estimation of the interaction forces through the robot motor currents measurements.

Acknowledgements

The authors would like to thank T. Dinon (ITIA-CNR) for expertise, setup and experimental support.

The work has been developed within the project EURECA, funded from H2020 CleanSky 2 under grant agreement n. 738039.

Appendix

A. Set-up: Modeling and Identification

Considering the Universal Robot UR 10 manipulator, the number of links $l = 6$. From the CAD it is possible to obtain the inertia and mass parameters of the links (Table II). It has to be underline that the CAD of the manipulator only gives an estimation of its mass and inertia parameters, since such CAD model is not precise (as in many practical cases). While the mass values have been used to initialize the estimation, the inertia parameters have been imposed to the manipulator link inertia to reduce estimation variables.

TABLE II: Estimated link mass and inertia.

	Link					
	1	2	3	4	5	6
mass $[kg]$	7.78	10	2,2	1.96	1.96	0.2
inertia x $\left[kgm^2\right]$	0.006	0.2	0.09	0.001	0.001	0.0001
inertia y $\left[kgm^2\right]$	0.005	0.2	0.09	0.0009	0.0007	0.0002
inertia z $\left[kgm^2\right]$	0.006	0.02	0.004	0.0007	0.0009	0.0001

Adopting a friction model characterized by two parameters for each joint (*i.e.*, $m = 2$, with parameters $\boldsymbol{\alpha_1}$, $\boldsymbol{\alpha_2}$), the friction torque acting on the i^{th} joint, can be expressed as:

$$F(\dot{q}_i) = \alpha_{1,i}\,\dot{q}_i + sign(\dot{q}_i)\alpha_{2,i} \tag{26}$$

A simple friction model has been considered to reduce the number of parameters to be estimated. A more complex friction model has been considered in the proposed learning approach (Section III-A) to improve the control performance. In such a way, the effectiveness of the proposed approach has been highlighted, showing the capabilities of the developed procedure to deal with any friction models.

The number of parameters to be identified is 36. The following joint trajectories (and congruent velocities and accelerations) have been imposed for the identification purpuses [32]:

$$\mathbf{q}(t) = \mathbf{q}^0 + \mathbf{A}\cos(2\pi\mathbf{f}_1\,(1 + \frac{1}{20}\cos(2\pi\mathbf{f}_2\,t))t + \boldsymbol{\phi}) \tag{27}$$

where vector $\mathbf{q}^0$ defines the starting position of the manipulator, vector $\mathbf{A}$ defines the amplitude of the motion, vector $\mathbf{f}_1$ and $\mathbf{f}_2$ define the movement frequency, vector $\boldsymbol{\phi}$ defines the (random) phase and t is the time instant of the trajectory execution. The parameters used for the identification procedure are shown in Table III. By measuring joint currents $\mathbf{i}_{msr}$ (to calculate measured joint torques $\boldsymbol{\tau}_{msr} = k_t\,\mathbf{i}_{msr}$, with $k_t = 12.7$ $[Nm/A]$ experimentally determinated), the joint positions $\mathbf{q}(t)$, velocities $\dot{\mathbf{q}}(t)$ and accelerations $\ddot{\mathbf{q}}(t)$ and imposing the input as described by (27) to the dynamic model (1) to estimate the torques $\boldsymbol{\tau}_{est}$, it has been possible to perform a non-linear optimization (based on Matlab function *lsqnonlin()* [33]) on the error function:

$$\boldsymbol{\tau}_e(t) = \boldsymbol{\tau}_{msr}(t) - \boldsymbol{\tau}_{est}(t) \tag{28}$$

The non-linear optimization has resulted in the local estimated link masses ($[kg]$), link centers of mass ($[m]$), friction parameters ($\overline{\boldsymbol{\alpha}}_1$ $\left[\frac{Nm}{radian/s}\right]$, $\overline{\boldsymbol{\alpha}}_2$ $[Nm]$) as shown in Table IV. Considering the link mass parameters and comparing the estimated values with respect to the CAD values, a significant

TABLE III: Identification joint trajectories parameters.

	Joint					
	1	2	3	4	5	6
$\mathbf{f}_1$ $[Hz]$	0.05	0.075	0.08	0.1	0.11	0.12
$\mathbf{f}_2$ $[Hz]$	0.0167	0.0167	0.0167	0.0167	0.0167	0.0167
$\mathbf{A}$ $[rad]$	$\pi/6$	$\pi/9$	$\pi/6$	π	$\pi 5/9$	π

TABLE IV: Estimated mass, centers of mass and friction parameters.

	joint					
	1	2	3	4	5	6
mass	2.8	11.801	4.974	0	0	1.610
x	0.00008	0.2914	−0.0733	0	0	0.0085
y	0.00244	0.0609	0.0384	0.110949	0.0018	0.0289
z	−0.037	0	0	0.01634	0.11099	0
$\bar{\boldsymbol{\alpha}}_1$	79.63	56.73	32.92	8.22	15.54	12.07
$\bar{\boldsymbol{\alpha}}_2$	21.26	17.61	9.97	2.59	2.76	1.51

difference arises. Such difference is related to both the facts that the CAD model is not accurate and the optimization procedure allows to obtain an equivalent set of parameters describing the robot dynamics that can differ from the real robot parameters. A validation of the estimated parameters have been performed using the joint trajectory defined in (27) with the parameters in Table V (different parameters with respect to the identification parameters in Table III, *i.e.*, resulting in a different robot motion and sub-workspace exploration). The torque error characteristics have been analyzed evaluating the root mean square (rms) of the torque error:

$$\mathbf{rms}_{e_\tau} := \sqrt{\frac{1}{T}\sum_{t=0}^{T}\left(\boldsymbol{\tau}_{msr}(t)-\boldsymbol{\tau}_{est}(t)\right)^2} \tag{29}$$

Obtained results are shown in Table VI. The obtained root mean square of the torque error for each joint is comparable with the state-of-the-art identification methods [26].

The estimated link mass parameters have been used in the experimental tests in both Sections V-A and V-B, resulting in an adequate controlled robot dynamics in comparison with the state-of-the-art compliance controllers [27].

The effect of the inertia values has been verified in simulation, imposing a 4 times inertia values and the validation trajectory with parameters defined in Table V. The simulation has resulted in a negligible torque variations for the UR 10 manipulator ($< 0.2Nm$), allowing to neglect inertia parameters in the estimation of the dynamics.

B. Impedance Control Loop Design

As described in [27], an impedance controller with dynamics compensation can be design based on (1), defining the robot joint torque vector $\boldsymbol{\tau}$ as:

$$\boldsymbol{\tau} = \mathbf{B}(\mathbf{q})\boldsymbol{\gamma}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})+\mathbf{g}(\mathbf{q})+\mathbf{h}_{f,q}(\dot{\mathbf{q}}) \tag{30}$$

TABLE V: Validation joint trajectories parameters.

	Joint					
	1	2	3	4	5	6
$\mathbf{f}_1$ $[Hz]$	0.04	0.065	0.07	0.09	0.1	0.1
$\mathbf{f}_2$ $[Hz]$	0.015	0.015	0.015	0.015	0.015	0.015
$\mathbf{A}$ $[rad]$	$\pi/5$	$\pi/7$	$\pi/5$	$\pi 2/3$	$\pi/2$	$\pi 2/3$

$\boldsymbol{\gamma}$ is the impedance control law. It can be written as:

$$\boldsymbol{\gamma} = \mathbf{J}(\mathbf{q})^{-1}\left(\ddot{\mathbf{x}} - \dot{\mathbf{J}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\right) \tag{31}$$

where the target acceleration $\ddot{\mathbf{x}} = [\ddot{\mathbf{p}};\ddot{\boldsymbol{\varphi}}_{cd}]$ derives from the target translational impedance behavior $\ddot{\mathbf{p}}$, and the target rotational impedance behavior $\ddot{\boldsymbol{\varphi}}_{cd}$ described by the intrinsic Euler angles representation:

$$\begin{aligned} \ddot{\mathbf{p}} &= \mathbf{M}_t^{-1}\left(-\mathbf{D}_t\dot{\mathbf{p}} - \mathbf{K}_t\Delta\mathbf{p} + \mathbf{f}\right) \\ \ddot{\boldsymbol{\varphi}}_{cd} &= \mathbf{M}_r^{-1}\left(-\mathbf{D}_r\dot{\boldsymbol{\varphi}}_{cd} - \mathbf{K}_r\boldsymbol{\varphi}_{cd} + \mathbf{T}^T(\boldsymbol{\varphi}_{cd})\boldsymbol{\mu}^d\right) \end{aligned} \tag{32}$$

Considering the translational part of the impedance control, $\mathbf{M}_t$ is the target mass matrix, $\mathbf{D}_t$ is the target damping matrix, $\mathbf{K}_t$ is the target stiffness matrix, $\mathbf{f}$ is the external forces vector. $\mathbf{p}$ is the actual Cartesian positions vector, while $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}^d$, where $\mathbf{p}^d$ is the target positions vector. Considering the rotational part of the impedance control, $\mathbf{M}_r$ is the target inertia matrix, $\mathbf{D}_r$ is the target damping matrix, $\mathbf{K}_r$ is the target stiffness matrix. $\boldsymbol{\varphi}_{cd}$ is the set of Euler angles extracted from $\mathbf{R}_c^d = \mathbf{R}_d^T\mathbf{R}_c$, describing the mutual orientation between the compliant frame (at the end-effector) and the target frame. $\boldsymbol{\mu}^d$ is the external torques vector referred to the target frame. Matrix $\mathbf{T}(\boldsymbol{\varphi}_{cd})$ defines the transformation from Euler angles derivatives to angular velocities $\boldsymbol{\omega} = \mathbf{T}(\boldsymbol{\varphi}_{cd})\dot{\boldsymbol{\varphi}}_{cd}$ [27].

Substituting (32), (31), and (30) in (1), under the hypothesis that the manipulator dynamics is known, the controlled robot dynamics results in:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} + \mathbf{K}\Delta\mathbf{x} = \mathbf{h}_{ext}$$

where $\mathbf{M}$, $\mathbf{D}$, $\mathbf{K}$ are the impedance matrices composed by both the translational and rotational parts, $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}^d = [\Delta\mathbf{p};\boldsymbol{\varphi}_{cd}]$, and $\mathbf{h}_{ext} = [\mathbf{f},\mathbf{T}^T(\boldsymbol{\varphi}_{cd})\boldsymbol{\mu}^d]$ considering the external forces/torque acting on the robot (*i.e.*, related to the interaction with the surrounding environment).

References

[1] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson, “Robotic assembly of emergency stop buttons,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ Int Conf on.* IEEE, 2013, pp. 2081–2081.

TABLE VI: Root mean square (rms) of the torque error.

	Joint					
	1	2	3	4	5	6
$[Nm]$	1.2867	2.2024	1.7046	1.5207	1.5280	1.6764

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2748236, IEEE Transactions on Industrial Informatics

[2] N. Morelli, "Design in new industrial contexts: shifting design paradigms and methodologies," in *EAD 06. 6th EAD Conference*, 2005.

[3] L. Roveda, N. Iannacci, F. Vicentini, N. Pedrocchi, F. Braghin, and L. M. Tosatti, "Optimal impedance force-tracking control design with impact formulation for interaction tasks," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 130–136, 2016.

[4] N. Hogan, "Impedance control: An approach to manipulation," in *American Control Conference, 1984*, june 1984, pp. 304–313.

[5] M. Jin, S. H. Kang, and P. H. Chang, "Robust compliant motion control of robot with nonlinear friction using time-delay estimation," *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 1, pp. 258–269, 2008.

[6] E. Colgate and N. Hogan, "An analysis of contact instability in terms of passive physical equivalents," in *Robotics and Automation (ICRA), 1989 IEEE Int Conf on*. IEEE, 1989, pp. 404–409.

[7] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *The Int Journal of Robotics Research*, vol. 30, no. 7, pp. 820–833, 2011.

[8] E. Alpaydin, *Introduction to machine learning*. MIT press, 2014.

[9] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.

[10] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning," in *The elements of statistical learning*. Springer, 2009, pp. 485–585.

[11] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[12] A. Visioli and G. Legnani, "On the trajectory tracking control of industrial scara robot manipulators," *Industrial Electronics, IEEE Transactions on*, vol. 49, no. 1, pp. 224–232, 2002.

[13] M. Gautier, W. Khalil, and P. Restrepo, "Identification of the dynamic parameters of a closed loop robot," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 3. IEEE, 1995, pp. 3045–3050.

[14] N. Mallon, N. Van de Wouw, D. Putra, and H. Nijmeijer, "Friction compensation in a controlled one-link robot using a reduced-order observer," *Control Systems Technology, IEEE Transactions on*, vol. 14, no. 2, pp. 374–383, 2006.

[15] S. I. Han, K. S. Lee, M. G. Park, and J. M. Lee, "Robust adaptive deadzone and friction compensation of robot manipulator using rwcmac network," *Journal of mechanical science and technology*, vol. 25, no. 6, pp. 1583–1594, 2011.

[16] R.-J. Wai and R. Muthusamy, "Design of fuzzy-neural-network-inherited backstepping control for robot manipulator including actuator dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 709–722, 2014.

[17] Y. Li and S. S. Ge, "Impedance learning for robots interacting with unknown environments," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1422–1432, 2014.

[18] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning physical collaborative robot behaviors from human demonstrations," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 513–527, 2016.

[19] L. Rozo, S. Calinon, D. Caldwell, P. Jiménez, and C. Torras, "Learning collaborative impedance-based robot behaviors," *parameters*, vol. 1, no. 1, p. 1, 2013.

[20] L. Roveda, N. Pedrocchi, and L. M. Tosatti, "Exploiting impedance shaping approaches to overcome force overshoots in delicate interaction tasks," *International Journal of Advanced Robotic Systems*, vol. 13, no. 5, 2016.

[21] C.-P. Kuan and K.-y. Young, "Reinforcement learning and robust control for robot compliance tasks," *Journal of Intelligent and Robotic Systems*, vol. 23, no. 2-4, pp. 165–182, 1998.

[22] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 156–163.

[23] J. Navarro-Gonzalez, I. Lopez-Juarez, K. Ordaz-Hernández, and R. Rios-Cabrera, "On-line incremental learning for unknown conditions during assembly operations with industrial robots," *Evolving Systems*, vol. 2, no. 6, pp. 101–114, 2015.

[24] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 3, pp. 334–344, 2016.

[25] Y. Wang, F. Gao, and F. J. Doyle, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.

[26] N. Pedrocchi, E. Villagrossi, F. Vicentini, and L. Molinari Tosatti, "On robot dynamic model identification through sub-workspace evolved trajectories for optimal torque estimation," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2370–2376.

[27] B. Siciliano and L. Villani, *Robot Force Control*, 1st ed. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

[28] L. Roveda, F. Vicentini, and L. Molinari Tosatti, "Deformation-tracking impedance control in interaction with uncertain environments," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ Int Conf on*. IEEE, 2013, pp. 1992–1997.

[29] Website, "EuRoC Project homepage," http://www.euroc-project.eu/, 2016, accessed: 2016-04-07.

[30] R. Brincker, C. Ventura, and P. Andersen, "Damping estimation by frequency domain decomposition," *IMAC XIX, Kissimmee, USA*, vol. 9, p. 72, 2001.

[31] T. Kim, H. S. Kim, and J. Kim, "Position-based impedance control for force tracking of a wall-cleaning unit," *International Journal of Precision Engineering and Manufacturing*, vol. 17, no. 3, pp. 323–329, 2016.

[32] J. Swevers, W. Verdonck, and J. D. Schutter, "Dynamic model identification for industrial robots," *Control Systems, IEEE*, vol. 27, no. 5, pp. 58–71, 2007.

[33] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

Eng. Ph.D. Loris Roveda Eng. Ph.D. Loris Roveda holds an M.Sc. (2011) and PhD (2015) in Mechanical Engineering at Politecnico di Milano. He is a researcher at ITIA-CNR (involved in many projects, *e.g.*, CleanSky 2 EURECA).

Eng. Giacomo Pallucca Eng. Giacomo Pallucca holds a M.Sc. (2016) in Mechanical Engineering at Politecnico di Milano. He is a research fellow at ITIA-CNR.

Eng. Ph.D. Nicola Pedrocchi Eng. Ph.D. Nicola Pedrocchi holds an M.Sc. (2004) and PhD (2008) in Mechanical Engineering from the University of Brescia. He is a permanent researcher at ITIA-CNR. He has been the referee for many European Projects (*e.g.*, ROBOFOOT, FLEXICAST).

Prof. Francesco Braghin Prof. Francesco Braghin holds a M.Sc. in Mechanical Engineering (1997) and PhD in Applied Mechanics (2001) at Politecnico di Milano. In 2001 he became Researcher and in 2011 Associate Professor at the Department of Mechanical Engineering of Politecnico di Milano. Since 2015 he is Full Professor in Applied Mechanics in the same department.

Eng. Ph.D. Lorenzo Molinari Tosatti Eng. Ph.D. Lorenzo Molinari Tosatti is the Head of the IRAS Division in CNR-ITIA. Since 2001 is a senior researcher at ITIA-CNR. Coordinator of Cluster project "Factory of the Future".