Optimized Residual Action for Interaction Control with Learned Environments

Luca Puricelli [1], Alessandro Pozzi [1], Vincenzo Petrone [2], Enrico Ferrentino [1], Pasquale Chiacchio [1], Francesco Braghin [1], and Loris Roveda [1]

[1]Affiliation not available
[2]University of Salerno

October 30, 2023

Posted on 24 Jan 2020 — CC-BY 4.0 — https://doi.org/10.36227/techrxiv.21905433.v1 — e-Prints posted on TechRxiv are preliminary reports that are not peer reviewed. They should not b...

Abstract

Robotic tasks featuring interaction with other bodies are increasingly required in industrial contexts. The manipulators need to interact with the environment in a compliant way to avoid damage, but, at the same time, are often required to accurately track a reference force. To this aim, interaction controllers are typically employed, but they either need human tinkering for parameter tuning or precise modeling of the environment the robot will interact with.

The former is a time-consuming procedure, while the latter is necessarily affected by approximations, which often lead to failure during the actual application. Both these aspects are problematic if it were often necessary to change the contact environment.Current research is concentrating on devising high-performance force controllers that are simple to tune and quick to adapt to changing environments.

Along this line, this work proposes a novel control strategy, that we term **ORACLE** (**O**ptimized **R**esidual **A**ction for interaction **C**ontrol with **L**earned **E**nvironments). It exploits an ensemble of neural networks to estimate the force generated by the robot-environment interaction. This estimate is input to an optimal residual action controller that locally corrects the main action, output of a base force controller, which guarantees stability.

The ORACLE strategy has been implemented and tested in the MuJoCo dynamic simulator and in a real-case scenario, both foreseeing a Franka Emika Panda robot used as a test platform. A reduction in terms of force tracking error is achieved by deploying the proposed strategy, with a short setup time.

Optimized Residual Action for Interaction Control with Learned Environments

Luca Puricelli*, Alessandro Pozzi*, Vincenzo Petrone, Enrico Ferrentino, Pasquale Chiacchio, Francesco Braghin, Loris Roveda

***Abstract*—Robotic tasks featuring interaction with other bodies are increasingly required in industrial contexts. The manipulators need to interact with the environment in a compliant way to avoid damage, but, at the same time, are often required to accurately track a reference force. To this aim, interaction controllers are typically employed, but they either need human tinkering for parameter tuning or precise modeling of the environment the robot will interact with. The former is a time-consuming procedure, while the latter is necessarily affected by approximations, which often lead to failure during the actual application. Both these aspects are problematic if it were often necessary to change the contact environment. Current research is concentrating on devising high-performance force controllers that are simple to tune and quick to adapt to changing environments. Along this line, this work proposes a novel control strategy, that we term ORACLE (Optimized Residual Action for interaction Control with Learned Environments). It exploits an ensemble of neural networks to estimate the force generated by the robot-environment interaction. This estimate is input to an optimal residual action controller that locally corrects the main action, output of a base force controller, which guarantees stability. The ORACLE strategy has been implemented and tested in the MuJoCo dynamic simulator and in a real-case scenario, both foreseeing a Franka Emika Panda robot used as a test platform. A reduction in terms of force tracking error is achieved by deploying the proposed strategy, with a short setup time.**

***Index Terms*—physical robot-environment interaction, artificial neural networks, optimized interaction control, impedance control**

I. Introduction

A. Context

Manipulators are nowadays massively employed in a broad range of tasks. From factories to aerospace [1], from warehouses to hospitals, robotic systems are increasingly deployed in various unstructured environments, occasionally foreseeing the presence of humans operating the manipulator [2]. In such scenarios, robots are demanded to interact with the surrounding workspace while guaranteeing safety and stability [3], [4], which translates into controlling the force the manipulator exerts on the environment, so as to avoid unwanted behaviors during its task execution [5].

To that purpose, interaction controllers have been largely studied in the last decades. Starting from the pioneering work in [6], the typical strategies can be divided into two categories, i.e. impedance control [7] and direct force control [8]. While the former aims at delivering a compliant motion to the robot imposing mass-spring-damper dynamics at its end-effector (EE), the goal of the latter is to actually reach a desired force; hence, it could be considered, in principle, more suitable for applications in which tracking a reference force profile is a task requirement.

B. Related works

In literature, impedance control is preferred to direct force control for two reasons. First, its generalized approach to control the manipulator's free-space motion and interaction behavior simultaneously allows for avoiding separating position- and force-controlled dimensions, which is not adequate in general [7]. Second, recent research is focused on developing advanced techniques that allow for overcoming the main disadvantage of impedance control, i.e. the impracticability to track a force reference. In fact, as demonstrated in [9], this objective can be achieved only if an accurate description of the environment is available; since such a reliable model is typically hard to devise [10], it is of unquestionable importance for the maximization of the performance of robot-environment interaction tasks to deliver force-tracking capabilities to impedance controllers. Indeed, our work contributes to the development of novel strategies in this context.

Reference generation approaches [11] compute a setpoint position to track such that the resulting expected force is the reference one. They have been recently improved with iterative learning methods: the work in [12] compensates for unmodeled dynamics in the reference computation, while a passivity-based iterative learning approach is proposed in [13] to balance compliance and stability. Alternatively, the setpoint for the impedance controller can be computed by a PI direct force controller [5].

Recently, AI-based methods have been proposed; in particular, Reinforcement Learning (RL) is increasingly applied to real-life scenarios for its ability to cope with unstructured environments [14]. In [15], a *residual* action, added to the output of a classic impedance controller, is learned via RL.

This work was partially supported by the project Robolutionary, funded from Hasler Stiftung.

Luca Puricelli, Alessandro Pozzi, and Francesco Braghin are with Department of Mechanical Engineering, Politecnico di Milano, Milano 20133, Italy (**Luca Puricelli and Alessandro Pozzi are co-first authors.*) (email: luca.puricelli@mail.polimi.it; alessandro11.pozzi@mail.polimi.it; francesco.braghin@polimi.it).

Vincenzo Petrone, Enrico Ferrentino, and Pasquale Chiacchio are with Department of Computer Engineering, Electrical Engineering and Applied Mathematics (DIEM), University of Salerno, 84084 Fisciano, Italy (e-mail: vipetrone@unisa.it; eferrentino@unisa.it; pchiacchio@unisa.it).

Loris Roveda is with Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), Università della Svizzera Italiana (USI), Lugano 6962, Switzerland (email: loris.roveda@idsia.ch).

The problem of adapting the damping and stiffness terms is faced in [3], where Adversarial Inverse Reinforcement Learning (AIRL) is exploited to infer, starting from human "expert" demonstrations, the objective function against which the variable terms are optimized. Recent research has shown how Neural Networks (NNs) can be exploited to learn interaction dynamics [16], whether the contacts happens with a working environment [17] or a human being [18]. Estimating this dynamics can be exploited for the computation of optimal impedance gains for a safe and compliant interaction [19].

In this scope, the methodology proposed in this article takes advantage of AI techniques to develop an optimization-based control law. In literature, NNs learning the unknown interaction dynamics are mainly exploited for optimizing the robot compliance [17], [18], whereas our objective is delivering accurate force tracking. As regards the optimization of robot-environment interaction tasks, the state-of-art strategies suppose that an "expert" system is employed beforehand to register a policy, from which an optimal control configuration is devised [3], [15], with the policy being strongly task-dependent. This aspect limits the generalization capabilities of such approaches.

We propose a framework capable of autonomously learning the robot-environment interaction dynamics, without inferring a policy from expert demonstrations. Being our system autonomous, and only configured by the human at startup, it does not depend on the specific task the robot has to perform.

C. Contribution

This work proposes a control strategy that enhances the capabilities of a state-of-art impedance controller, improving its force tracking performances. We developed an ORACLE (Optimized Residual Action for interaction Control with Learned Environments), able to predict the forces generated by the robot-environment interaction. Based on this prediction, a residual action is computed with an optimization approach, so as to minimize the force tracking error along a trajectory. Working on the same lines as [15], the residual is added to the output of the base impedance controller to achieve optimal performance.

Our ORACLE exploits a model of the environment embedding the input-output relationship between the robot's pose and velocity and the corresponding force exerted on the environment in such a configuration. This model is implemented by a simple feed-forward NN (FFNN).

The contribution in this paper is therefore twofold: the control algorithm that implements the computation of the ORACLE's optimal residual action; the NN modeling the interaction with our testing environment. Some design choices of the latter include its structural properties and the composition of the dataset used for its training. Similar NN models have been previously used for physical human-robot interaction [18]. Compared to them, we provide a straightforward setup that allows for autonomous learning of the interaction between the robot and an environment, which is of particular interest for the manipulators' deployment in industrial scenarios.

The ORACLE we propose is tested in a custom MuJoCo virtual environment [20] and in a laboratory setup on a real robot. In both cases, the robotic platform is the Franka Emika's Panda robot. In order to stress the improvement with respect to the state-of-art techniques, in both scenarios the results obtained employing the ORACLE control strategy are compared to the ones achieved by controlling the robot with a properly tuned impedance controller, which is already provided with an outer force control loop to guarantee basic force tracking performance. We demonstrate that, with the same control configuration, adding the ORACLE action, we can further reduce the force tracking error and mitigate other undesired behaviors, i.e. overshoot and slow transient.

D. Outline

The remainder of this article is structured as follows. Section II presents the novel ORACLE control strategy and provides details on the NN ensemble training, detailing guidelines on how to compose the dataset and a procedure to train the neural network and optimize its hyper-parameters. In Section III, we first present our validation task, together with all the details of the NN, in terms of architectural design, hyperparameters, and training set; then, we discuss simulation and experimental results obtained employing ORACLE. The paper is concluded in Section IV, summarizing the achievements attained in this work and providing directions for possible future works.

II. Methodology

The ORACLE is an AI-based enhancement of an impedance controller. As its name suggests, its output is a residual action, in that it corrects the setpoint computed by a base controller, with the goal of tracking a desired force reference. The residual is computed by solving an optimization problem, in which the prediction of the resulting robot-environment interaction force is exploited to minimize the force tracking error. The interaction dynamics is learned in an off-line phase, performed previously to the ORACLE's operational time, in which the training of a NN allows to capture the relationship between the EE's pose and velocity and the interaction force.

The main requirements the ORACLE aims to satisfy are: (i) to deliver a safe interaction with an unknown environment; (ii) to exert a reference force, which is established by the task settings and constraints; (iii) to allow the user to set up the system with ease, reducing the time necessary to bring it to a fully operational stage.

The ORACLE relies on a low-level Cartesian impedance controller, which allows considering the robot-environment ensemble as a decoupled mass-spring-damper system in the Cartesian space. Such a base controller is indeed the one whose force-tracking performances will be increased by the ORACLE. The whole system is therefore composed of the following elements:

- a low-level impedance controller provided with basic force-tracking capabilities;
- an optimization-based algorithm, in which the optimal residual action is computed;
- a dataset used to train the ORACLE, necessary for learning the robot-environment interaction dynamics;

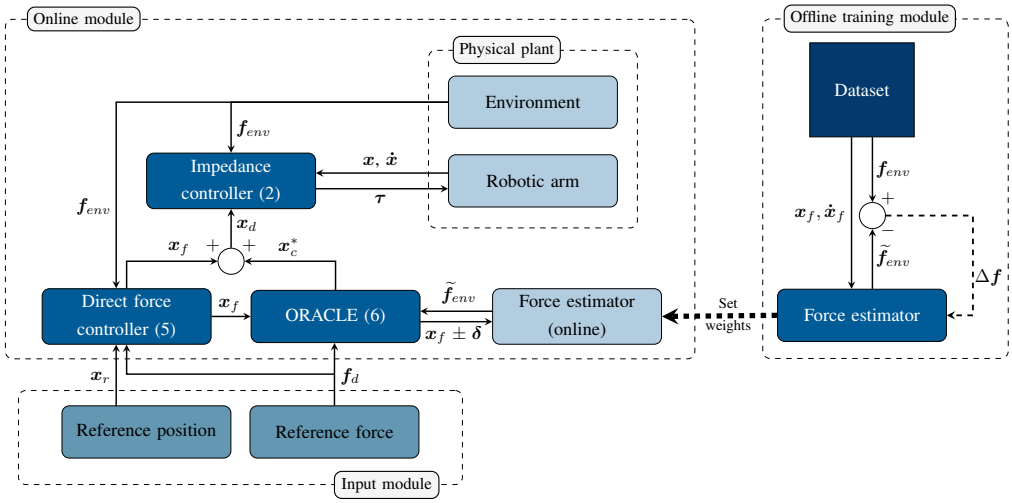

Fig. 1. Functioning scheme of the entire control system, comprised of the force-impedance controller, the ORACLE, the neural network training module, and the physical plant, i.e. the robot in contact with the environment

- a FFNN, whose structure and design are optimized so as to achieve a fast and accurate inference during the operational stage.

Each of the aforementioned elements, together with their communication interfaces, is described in the next sections. The high-level overview of the ORACLE control scheme is depicted in Fig. 1.

A. Base controller

A low-level impedance controller is used to control the manipulator, whose reference position is first generated by a Proportional-Integral (PI) direct force controller. Their ensemble, the *base controller*, is therefore termed *force-impedance controller*; such a controller's action is enhanced by the ORACLE's residual action.

1) Impedance controller: The low-level impedance controller imposes the desired interaction dynamics on the robot-environment physical plant. Consider the dynamic model of a manipulator with n Degrees of Freedom (DOFs) performing an m-dimensional task [5], with $m \leq 6 \leq n$:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}+\boldsymbol{\tau}_f(\dot{\boldsymbol{q}})+\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_{ctrl}-\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_{env}, \quad (1)$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the matrix accounting for the centrifugal and Coriolis effects, $\boldsymbol{\tau}_f(\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times 1}$ accounts for viscous and static friction, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^{n\times 1}$ represents the torque exerted on the links by gravity, $\boldsymbol{\tau}_{ctrl} \in \mathbb{R}^{n\times 1}$ indicates the impedance control action, $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m\times n}$ is the geometric Jacobian, $\boldsymbol{f}_{env} \in \mathbb{R}^{m\times 1}$ is the vector of wrenches exerted on the environment by the EE. The vectors $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \mathbb{R}^{n\times 1}$ are the joint positions, velocities, and accelerations, respectively.

As regards the EE wrench $\boldsymbol{f}_{env}$, it is worth specifying that, in most applications, it is measured through a force/torque sensor mounted at the manipulator's flange, but it could also be estimated from the robot state without additional sensing [5], [21].

The expression of Cartesian impedance control with robot dynamics compensation can be written as [22]:

$$\boldsymbol{\tau}_{ctrl} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_{ctrl} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}), \quad (2)$$

with

$$\begin{aligned}\boldsymbol{f}_{ctrl} =& \boldsymbol{M}_x(\boldsymbol{q})\ddot{\boldsymbol{x}}_d - \boldsymbol{f}_{env}+ \\ & \boldsymbol{M}_x(\boldsymbol{q})\boldsymbol{M}_{imp}^{-1}\left(\boldsymbol{f}_{env} - \boldsymbol{K}_{imp}\Delta\boldsymbol{x} - \boldsymbol{D}_{imp}\boldsymbol{\Delta}\dot{\boldsymbol{x}}\right),\end{aligned} \quad (3)$$

and

$$\boldsymbol{M}_x(\boldsymbol{q}) \triangleq \left(\boldsymbol{J}^T(\boldsymbol{q})\right)^{\dagger}\boldsymbol{M}(\boldsymbol{q})\left(\boldsymbol{J}(\boldsymbol{q})\right)^{\dagger}, \quad (4)$$

where $\boldsymbol{M}_x(\boldsymbol{q}) \in \mathbb{R}^{m\times m}$ is the task-space inertia matrix, and $\Delta\boldsymbol{x} = \boldsymbol{x}_d - \boldsymbol{x}$ is the error between the setpoint $\boldsymbol{x}_d$ and the robot's actual EE pose $\boldsymbol{x}$. For instance, if $m = 6$, $\boldsymbol{x} = (x, y, z, \phi, \theta, \psi)^T$ describes the full robot pose, where x, y and z represent the EE position and ϕ, θ and ψ are a minimal representation of its orientation. $\Delta\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}$ represents the position error derivative. $\boldsymbol{K}_{imp}, \boldsymbol{D}_{imp}, \boldsymbol{M}_{imp} \in \mathbb{R}^{m\times m}$ are the impedance control parameters, namely the stiffness, damping, and mass matrices, respectively. The $(\cdot)^{\dagger}$ notation indicates the pseudo-inverse of a matrix.

2) Direct force controller: Since the impedance controller only passively controls the interaction forces, without the possibility of tracking a reference force, when this is a requirement, a direct force control loop can be closed along the directions of interest, i.e., the contact directions. The control law of the PI controller is:

$$\boldsymbol{x}_f = \boldsymbol{x}_r + \boldsymbol{\Gamma}\left(\boldsymbol{K}_P\Delta\boldsymbol{f} + \boldsymbol{K}_I\int_t \Delta\boldsymbol{f}dt\right), \quad (5)$$

where, assuming the common case in which $m = 6$, $\boldsymbol{\Gamma} = \text{diag}\left(\gamma_x, \gamma_y, \gamma_z, \gamma_\phi, \gamma_\theta, \gamma_\psi\right)$ is the task specification matrix [8], with $\gamma_i = 1$ if the $i-$th direction is subject to force control, 0 otherwise. $\boldsymbol{x}_f = (x_f, y_f, z_f, \phi_f, \theta_f, \psi_f)^T$ is the force controller output, $\boldsymbol{x}_r$ is the reference pose generated by the trajectory generation module, $\boldsymbol{K}_P$ and $\boldsymbol{K}_I$ are the force controller's proportional and integral gains, and $\Delta\boldsymbol{f} = \boldsymbol{f}_d - \boldsymbol{f}_{env}$ is the difference between the reference wrench $\boldsymbol{f}_d$ and the actual wrench $\boldsymbol{f}_{env}$. In other words, by tracking the pose $\boldsymbol{x}_f$, the robot exerts $\boldsymbol{f}_d$ on the environment, along the direction on which $\boldsymbol{f}_d$ is specified, otherwise $\boldsymbol{x}_r$ is tracked.

3) Controller tuning: In order to analyze the ORACLE action, we first provide the configuration of the base controller

such that it is already capable to track a desired force reference. The impedance tuning is achieved by the rules reported in [5], so as to ensure a medium-soft interaction with the environment and a bandwidth of 2.5 Hz. As regards the PI direct force controller, its gains are tuned by computing the transfer function of the entire system and selecting the pole adequately. It must ensure stability of the control loop, null steady-state error and approximate tracking of sine waves at frequencies close to the controller's bandwidth.

It is worth specifying that, regardless of the technique used to tune the force-impedance controller, the ORACLE's goal is to improve its force-tracking capability. Furthermore, our ORACLE can complement, in principle, any impedance control action, hence not restricted to the controller employed in our setup.

B. ORACLE algorithm

The idea behind the ORACLE strategy is straightforward. At each control cycle, a control action $\boldsymbol{x}_f$ is proposed by the base force controller, according to (5), so as to track a desired force reference $\boldsymbol{f}_d$. Then, the ORACLE predicts the force $\widetilde{\boldsymbol{f}}_{env}$ the robot will exert on the environment for each pose in a (discretized) neighborhood of radius $\boldsymbol{\delta} > 0$ centered in $\boldsymbol{x}_f$. Finally, it outputs the residual action $\boldsymbol{x}_c^*$ such that it minimizes the predicted force tracking error $|\boldsymbol{f}_d - \widetilde{\boldsymbol{f}}_{env}|$.

The prediction function $\widetilde{\boldsymbol{f}}_{env}(\boldsymbol{x}_f, \dot{\boldsymbol{x}}_f)$ is implemented by a NN ensemble, which we term *force estimator*. As regards the whole system architecture, $\boldsymbol{x}_c^*$ is eventually summed up to $\boldsymbol{x}_f$, as visible in the control scheme of Fig. 1.

The ORACLE action can be summarized with the following optimization problem, running at each control step k:

$$\boldsymbol{x}_c^*(k) = \underset{\boldsymbol{x}_c \in [\boldsymbol{x}_f - \boldsymbol{\delta}; \boldsymbol{x}_f + \boldsymbol{\delta}]}{\arg\min} \mathcal{L}(\boldsymbol{x}_c), \tag{6}$$

with

$$\mathcal{L}(\boldsymbol{x}_c) = |\boldsymbol{f}_d(k) - \widetilde{\boldsymbol{f}}_{env}(\boldsymbol{x}_c, \dot{\boldsymbol{x}}_c)| + \Omega(k) \tag{7}$$

and

$$\Omega(k) = \|\boldsymbol{x}_c\|^2_{\boldsymbol{\alpha}} + |\boldsymbol{x}_c - \boldsymbol{x}_c^*(k-1)|_{\boldsymbol{\beta}}, \tag{8}$$

where $\mathcal{L}(\boldsymbol{x}_c)$ is the cost function to minimize, whose first term $|\boldsymbol{f}_d - \widetilde{\boldsymbol{f}}_{env}|$ is the (expected) force tracking error, while $\Omega(k)$ is a regularizer, contributing to smooth out large and rapid changes of $\boldsymbol{x}_c$. The first term, $\|\boldsymbol{x}_c\|^2_{\boldsymbol{\alpha}}$, penalizes heavy actions, so as to avoid deep penetrations of the EE in the environment. The second term, $|\boldsymbol{x}_c - \boldsymbol{x}_c^*(k-1)|_{\boldsymbol{\beta}}$, prevents fast variations between subsequent actions. In (7), with $|\cdot|$ we indicate the Manhattan norm; in (8),

$$\|\boldsymbol{x}_c\|^2_{\boldsymbol{\alpha}} \triangleq \sum_i \alpha_i x_{c_i}^2 \tag{9}$$

is the squared Euclidean norm, weighted according to the parameter vector $\boldsymbol{\alpha}$, and

$$|\boldsymbol{x}_c - \boldsymbol{x}_c^*(k-1)|_{\boldsymbol{\beta}} \triangleq \sum_i \beta_i |x_{c_i} - x_{c_i}^*(k-1)| \tag{10}$$

is the Manhattan norm, weighted according to the parameter vector $\boldsymbol{\beta}$. Both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are parameters to be chosen by the user.

Algorithm 1 ORACLE algorithm

Input: $\boldsymbol{x}_f$ from (5), $\boldsymbol{f}_d$
Output: $\boldsymbol{x}_c^*$

1: Set $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$
2: Set weights of the pre-trained force estimator neural network ensemble
3: Build a neighborhood $\boldsymbol{B}_{\boldsymbol{\delta}}(\boldsymbol{x_f}) = [\boldsymbol{x}_f - \boldsymbol{\delta}; \boldsymbol{x}_f + \boldsymbol{\delta}]$
4: **for** $\boldsymbol{x}_c \in \boldsymbol{B}_{\boldsymbol{\delta}}(\boldsymbol{x}_f)$ **do**
5: Infer the predicted force from the force estimator $\widetilde{\boldsymbol{f}}_{env}(\boldsymbol{x}_c, \dot{\boldsymbol{x}}_c)$
6: **end for**
7: Call the ORACLE by solving the optimization problem in (6) minimizing (7): $\boldsymbol{x}_c^* = \arg\min_{\boldsymbol{x}_c \in \boldsymbol{B}_{\boldsymbol{\delta}}(\boldsymbol{x}_f)} \mathcal{L}(\boldsymbol{x}_c)$

After the contact is established, both the force controller and the ORACLE strategy are activated in parallel. The algorithm implemented by the ORACLE, which executes at every control cycle, in search of the optimal corrective action, is summarized in Algorithm 1.

C. Dataset generation

As mentioned in Section II-B, the NN implementing the force estimator $\widetilde{\boldsymbol{f}}_{env}(\boldsymbol{x}_f, \dot{\boldsymbol{x}}_f)$ is trained prior to its actual employment. The training phase is done by collecting and processing contact data, from which the NN ensemble learns.

This phase is carried out by providing reference forces to the PI force controller, once the manipulator EE comes into contact with the environment. The particular shape of the reference force depends on randomly chosen parameters, configured according to user-defined constraints.

Randomizing the trajectories' parameters excludes biasing the training set, increasing the generalization of the data and the exploration of the state space. For instance, typical parameters are: for constants, the reference value; for ramps, the initial and final values; for sine waves, the amplitude and the frequency. The ranges in which the trajectory parameters are randomized are set by the user.

The final dataset should be the result of a comprehensive exploration of the state space, meaning that the ranges of interest should not present any data gaps. To achieve this goal, we considered the following guidelines.

Remark 1: The majority of the commanded force references are sine waves, as they are suitable for exploring both the position and, more importantly, the velocity space, difficult to explore otherwise.

Remark 2: It is advisable to exaggerate the amplitude and frequency of the reference force sine waves, with respect to the target ones. They will not be perfectly followed by the force-impedance base controller but, the EE is forced to reach higher velocities.

Remark 3: In general, the cleaner the data the better the NN learns; therefore it is advisable to perform the training phase moving only along the contact directions, minimizing outliers due to friction or environment discontinuities.

Remark 4: The distributions of $\boldsymbol{x}_f$ and $\boldsymbol{f}_{env}$ should be as uniform as possible for a proper exploration of the state

space along the contact direction. An approximate Gaussian distribution for $\dot{\boldsymbol{x}}_f$ is acceptable: a uniform distribution is not achievable for this quantity, as it is not a controlled variable in (5).

Remark 5: The data should be normalized and, in simulation, zero-mean Gaussian noise should be added to $\boldsymbol{f}_{env}$ so as to increase the NN's generalization capability [16], [23] and resemble the real-case scenario, in which wrench measures are inherently affected by measurement noise (or process noise, in case they are estimated [5]).

D. Neural network optimization and training

Hyperparameters are crucial when dealing with NNs. For example, depth and width (i.e., the number of layers and the number of neurons, respectively), heavily impact the inference time, whereas the learning rate and the weight optimizer (e.g. Stochastic Gradient Descent or Adam), are fundamental for correct training of the network.

Our ORACLE embeds the function $\widetilde{\boldsymbol{f}}_{env}(\boldsymbol{x}_f, \dot{\boldsymbol{x}}_f)$ with an ensemble of NNs, which is capable of capturing aleatoric and epistemic uncertainties [23], inherent in each learning-based control task. The architecture adopted in [16], [23] is applicable to industrial robotic tasks only by changing the NN design, i.e. properly tuning its hyperparameters.

In particular, a trade-off on the number of NN composing the ensemble, and the dimensions of the NNs themselves, must be considered. The motivations behind the trade-off are (i) the NN must be complex enough to fully capture the interaction dynamics complexity, yielding an accurate approximation; (ii) the NN must be simple enough to guarantee an inference time comparable with industrial robots' typical cycle times (i.e. in range $[1; 10]$ ms).

The solution to this trade-off depends on the task configuration and specifications, i.e. on its complexity and performance to guarantee. To this aim, we use Optuna [24] to optimize the possible configurations of hyperparameters. This framework selects the best hyperparameters (i.e. number of ensemble NNs, number of layers, number of neurons, weight optimizer, learning rate) by choosing the ones optimizing a certain performance index which, in case of reference tracking, is the MSE.

III. Experimental validation

In order to verify the proposed control strategy, we first deploy it in simulation, then on a real robot. In addition, to ensure robustness, the strategy is tested, in both scenarios, against three different values of environment stiffness. The video in [25] shows the execution of a trajectory under test when employing the ORACLE in the real scenario. The repository hosted on GitHub allows interested users to replicate our simulated scenario: https://github.com/unisa-acg/oracle-force-optimizer.

A. Task setup and materials

The ORACLE is tested on a 7-DOF arm, i.e., Franka Emika's Panda, whose virtual model is included in MuJoCo [20] to perform tests in a simulated environment. Through a low friction coefficient, the simulation has the aim of reproducing a plausible real case, therefore avoiding oscillations and stick-slip effects during the interaction. The task is moving the EE along a planar trajectory, with fixed orientation, and exerting a force along the surface normal (EE z axis), i.e. $\boldsymbol{\Gamma} = \mathrm{diag}\,(0, 0, 1, 0, 0, 0)^T$. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in (8) are specified accordingly (along z).

In both scenarios, the contact plane is a table, on which the robot tracks the same force reference, consisting of three subsequent sections: a ramp from 25 N to 35 N, a constant at 35 N, and a sine wave of amplitude 8 N and frequency 1.5 Hz, whose mean value linearly decreases down to 15 N. The force reference is plotted, in red, in Fig 3.

Algorithm 1 has been coded in Python 3, with the optimization problem (6) solved with CVXPY [26]; it interfaces with the other modules via ROS communication mechanisms.

B. Training Setup

The ORACLE exploits an ensemble of FFNNs, whose hyperparameters are selected by exploiting the Optuna framework [24], as discussed in Section II-D. The NN is trained with custom chosen trajectories, properly designed in accordance to the principles of Section II-C.

In particular, the dataset is composed by 10 trajectories, lasting 10 seconds each. The force profiles are selected among constant, ramp and sine wave, with probability 0.9, 0.05, 0.05 respectively (see Remark 1). Constant values are chosen in range $[3; 50]$ N; ramp endpoints are chosen in the same range, independently. As regards the sine waves, their amplitude is chosen in range $[5; 15]$ N, and their frequency is in range $[0.25; 2]$ Hz (see Remark 2); the centre of the wave changes over time according to a ramp, configured as already discussed. All the parameters are randomly chosen, as mentioned in Section II-C.

We recall that, in accordance to Remark 3, the trajectories are commanded so as to track the force profile at a single contact point; this choice favors a cleaner learning of the interaction dynamics over the contact direction.

As regards the NN design, the optimized configuration of the estimator, which guarantees an improved inference time without sacrificing prediction accuracy, foresees 3 hidden layers, 200 neurons per layer, $1\mathrm{e}{-3}$ as learning rate and Adam as weight optimizer. The inference time is reduced from 9 ms (using the same configuration as in [23]) to 3 ms (after optimization), measured on a system with Intel Core i5-7300HQ CPU, Nvidia GTX-1050Ti (Mobile) GPU and Ubuntu 20.04.

C. Simulation results

In simulation, the virtual robot is spawned and controlled through the Robosuite framework [27], so as to mimic the real environment of Fig. 2. The robot is commanded via the control scheme of Fig. 1, whose control loop executes at 100 Hz, only limited by the NN inference time.

The manipulator, during the execution of a trajectory, slowly approaches the table, whose stiffness is set, depending on the

TABLE I
CONTROL PARAMETERS ALONG THE z-AXIS IN SIMULATED ENVIRONMENTS

K_{env} [N/m]	M_{imp} [kg]	ξ	K_{imp} [N/m]	K_P [m/N]	K_I [s m/N]
5000	3.2	1.42	5000	0.001	0.01
12500	8	1.42	12000	0.0008	0.008
20000	10	1.42	17000	0.0006	0.006

TABLE II
MSE IN SIMULATED ENVIRONMENTS, PERFORMING FORCE CONTROL OPTIMIZATION ALONG z-AXIS. Ω INDICATES REGULARIZERS

K_{env} [N/m]	MSE [N^2] Base controller	ORACLE	ORACLE + Ω
5000	7.34	4.12	2.66
12500	8.16	3.21	1.55
20000	10.6	3.66	1.8

TABLE III
REGULARIZERS IN SIMULATED ENVIRONMENTS

K_{env} [N/m]	α [N/m^2]	β [N/m]
5000	50000	300
12500	40000	200
20000	30000	1000

test case, to the values $K_{env_z} \in \{5000, 12500, 20000\}$ N/m. The friction coefficient between the table and the EE has been set low, i.e. 0.2, for multiple reasons. First, the compensation of friction effects can interfere with the ORACLE strategy and it is not in the scope of this paper. Moreover, in MuJoCo, high friction coefficients introduce a severe stick-slip effect, which could heavily reduce the quality of the data being collected during the pre-training phase.

Following the principles and the requirements of Section II-A, the force-impedance controller is tuned on the three possible environment stiffness values. The resulting gains are listed in Table I, which, for the sake of brevity, only reports the gains for the z direction, which is of interest for the force tracking. One can notice that the impedance mass M_{imp} value increases with environment stiffness. This is due to the desired dynamics of the impedance controller, imposed by tuning. With higher values of environment stiffness, a high value of M_{imp} is required for a stable behavior of the controller.

The simulation results are collected in Table II: the ORACLE strategy applied to the base force controller yields improvements in all the environments. The force tracking Mean Squared Error $\text{MSE} = (f_{d_z} - f_{env_z})^2$ is computed on the reference trajectory described in Section III-A.

By activating the regularizers α_z and β_z in (8), the MSE further reduces, since their effect is to dampen excessive peaks and limit overshoots. The regularizer values used in the simulation scenario are reported in Table III. On average, we reduced the MSE by a factor of 4.64.

D. Experimental results

In the real scenario, thin blocks, of three different materials of different stiffness, are placed onto a larger and stiffer thin layer, sliding over the table, dragged by the robot during motion. This configuration allows reducing the aforementioned problems and guarantees closer coherence with the simulation experiments. The experimental setup is shown in Fig. 2.

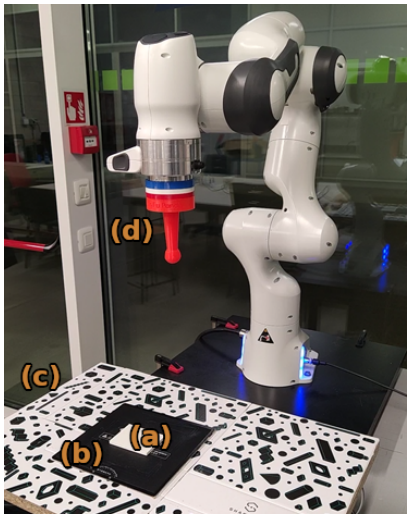

Fig. 2. Real scenario testing bench. It features a Franka Emika's Panda robot coming into contact with different environments with variable stiffness, which are represented by the white thin block (a). The different thin blocks are rigidly attached to a larger surface (b) that slides over the plane (c). This particular configuration reduces stick-slip effects at the interaction. The 3D-printed spherical end-effector (d) provides a small interaction surface with the material, making it possible to model the contact with the environment as a spring in the exploration phase.

The environment stiffness used in the three setups is $K_{env_z} \in \{5000, 10500, 23000\}$ N/m. Although they are very similar to the ones in the simulation, they are not exactly the same due to the difficulty of finding materials of specific stiffness.

The base controller is tuned according to the parameters of Table IV. The impedance controller is coded in C++, following the ROS control paradigm, and communicates with the robot via the Franka Communication Interface. This results in having the low-level impedance controller running at 1 kHz, to comply with the manipulator's standard, and the high-level control loop (i.e., the ORACLE algorithm including the NN ensemble) running at 100 Hz. The latter is low enough for the NN ensemble to output a prediction and high enough for force control to be effective.

The real world experiments confirm an increase in the performance on the force profile under test, plotted in Fig. 3. Fig. 3a reports the performances of the base force-impedance controller, which is characterized by poor step response and sine wave tracking, as better visible from the magnification in Fig. 4a. With the employment of the ORACLE, without the regularizers, the performances improve, even though an overshoot of 5 N is registered, as visible from Fig. 3b, but quickly recovered. The plot of Fig. 3c shows the complete elimination of the overshoot, obtained by tuning of the regularizers.

The tracking performances achieved by the base controller, shown in Fig. 4a, are compared to the force controller enhanced by the ORACLE with regularizers, in Fig. 4b. It is immediate to notice that the desired profile is followed accurately, as the actual sine shape strongly resembles the reference one, despite of a little delay that will be discussed later.

The improvements delivered by the ORACLE strategy are further confirmed by the force tracking MSE, computed about

TABLE IV
CONTROL PARAMETERS ALONG THE z-AXIS IN REAL ENVIRONMENTS

K_{env} [N/m]	M_{imp} [kg]	ξ	K_{imp} [N/m]	K_P [m/N]	K_I [s m/N]
5000	3.5	1.42	5000	0.0005	0.009
10500	10	1.42	10500	0.001	0.008
23000	10	1.51	18000	0.0001	0.005

TABLE V
MSE IN REAL ENVIRONMENTS, PERFORMING FORCE CONTROL OPTIMIZATION ALONG z-AXIS. Ω INDICATES REGULARIZERS

K_{env} [N/m]	MSE [N^2]		
	Base controller	ORACLE	ORACLE + Ω
5000	9.62	4.71	3.55
10500	8.03	3.77	3.10
23000	7.68	3.43	2.84

TABLE VI
REGULARIZERS IN REAL ENVIRONMENTS

K_{env} [N/m]	α [N/m^2]	β [N/m]
5000	100000	1000
10500	100000	1000
23000	20000	200

the test trajectory, reported in Table V. In comparison to the simulation, in general, the MSE is larger in all the cases, but the ORACLE deployment still outperforms the base force controller in all the tested environments. The regularizers adopted in the real scenario are listed in Table VI. On average, we reduced the MSE by a factor of 2.67.

About the control action delay of 0.1 s, its cause can be found in the closure of the control loop, and in the NN ensemble, in particular. The inference time is affected by the amplitude $\boldsymbol{\delta}$ of the pose neighborhood. If reduced, or chosen not symmetrically with respect to $\boldsymbol{x}_f$, it can shorten the inference time. Deploying the NN ensemble on a GPU is also expected to contribute to the same goal.

Although the delay is a limitation of the ORACLE, a remark is worthwhile. The real Panda's EE forces are not measured, but estimated with an internal algorithm; the estimation error is in the order of 1.5 N (registered during preliminary experiments with the robot). Due to the delay, the exerted force trend results shifted in time compared to the desired one (as shown in Fig. 4b), resulting in a MSE of, on average, 2.83 N^2, which is comparable with the estimation error, thus proving the effectiveness of the strategy in predicting the interaction forces and correcting the control action, despite of the delay. Nevertheless, if an accurate synchronization between position tracking and force tracking is required by the task specifications, then the delay must be reduced, e.g. by following the aforementioned suggestions.

IV. CONCLUSIONS

A. Discussion

In this work, we introduced ORACLE, a novel control strategy for achieving accurate force-tracking in robot-environment interaction tasks. By exploiting a force estimator based on neural networks, the ORACLE enhances the performances of a base interaction controller, improving its capabilities of

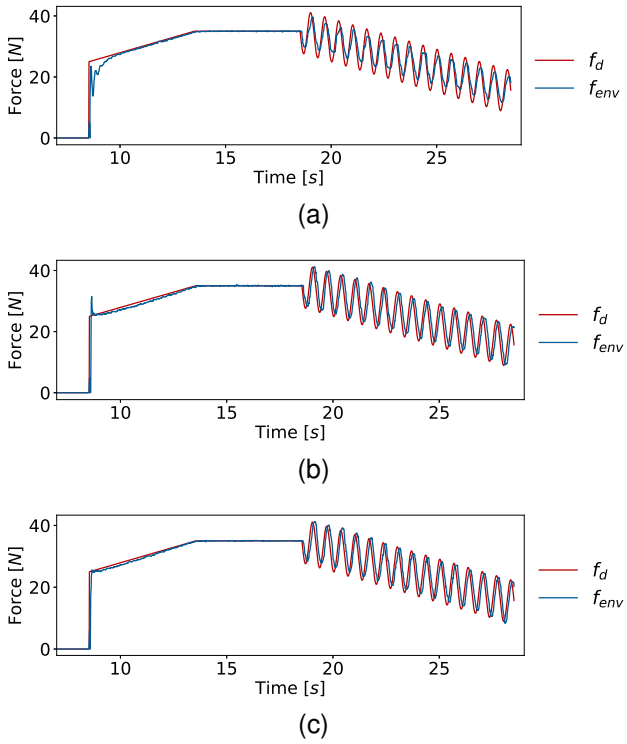

Fig. 3. Overview of force tracking for medium stiffness environment $K_{env_z} = 10500\,\mathrm{N/m}$ in the real scenario, with three different controllers: (a) base force-impedance controller; (b) base force-impedance controller enhanced with ORACLE without regularizers; and (c) base force-impedance controller enhanced with ORACLE and the regularizers, which reduces overshoots and smoothens possible spikes.

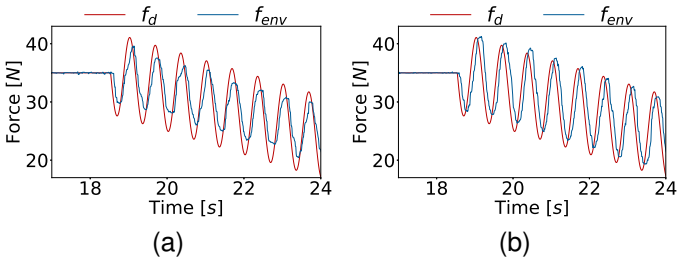

Fig. 4. Zoom on an area of the sine curve at 1.5 Hz, (a) of Fig. 3a and (b) of Fig. 3c. The presence of the regularizers does not affect the force trend. A small delay is introduced by ORACLE, whose motivation is detailed in Sec. III-D.

tracking a desired force profile when performing a contact task in an unknown environment.

We developed a straightforward yet effective optimization algorithm that corrects the output of a manually tuned impedance controller by predicting the optimal setpoint to track, with the goal to minimize the tracking error with respect to a reference force trajectory.

The ORACLE prediction is performed through an ensemble of neural networks, which serves as an estimator of the interaction forces that the manipulator exchanges with the environment. We highlighted how a fast and accurate contact force estimator can be retrieved with a simple NN design. Furthermore, we reported on the precautions that one should take to collect a suitable dataset, capturing the robot-environment interaction dynamics.

The novel proposed strategy was tested in simulation and real scenarios. In both cases, the experiments proved that the

proposed approach is capable of boosting the performance, in terms of force tracking MSE, of the base force controller in three different environment configurations, thus assessing its robustness with respect to variable stiffness.

Further enhanced with regularizers, the ORACLE can limit excessive peaks and overshoot, while reducing the rising time in step responses. Hence, we demonstrated that our control strategy can effectively tackle abrupt changes in the force reference, outperforming a state-of-art interaction controller we adopted for comparison. The resulting performance improves by a factor of 4.64 and 2.67 in simulation and real-case scenarios, respectively.

B. Future works

We envisage several improvements of the present work. Starting from the dataset collection, the velocity domain of the commanded action $\dot{z}_f$ is not explored thoroughly, which negatively affects the ORACLE's prediction accuracy. As this is due to the fact that $\dot{\boldsymbol{x}}_f$ is not a controlled variable, a possible solution for solving this problem is taking into account $\dot{\boldsymbol{x}}_f$ in the control law.

Another limitation is represented by the delay introduced by the ORACLE addition. Although the main cause of this delay has been identified, more work is needed to confirm our hypotheses and propose a way forward.

One last improvement left for future developments is including information about the tangential velocities, which could affect prediction and tracking of the normal forces. It would be interesting to analyze whether these terms could be captured by a NN ensemble similar to the one adopted in this work.

Moreover, having detailed a procedure to capture the robot-environment interaction dynamics permits the inclusion of this knowledge in a trajectory planning scenario, such as [10]. Providing a more complete understanding of the contact dynamics in the planning phase could result in a better trajectory construction, improving its reliability and feasibility prior to its execution.

REFERENCES

[1] F. Aghili, "Robust Impedance-Matching of Manipulators Interacting With Uncertain Environments: Application to Task Verification of the Space Station's Dexterous Manipulator," *IEEE/ASME Trans. Mechatron.*, vol. 24, no. 4, pp. 1565–1576, Aug. 2019.

[2] M. J. Kim, W. Lee, J. Y. Choi, G. Chung, K.-L. Han, I. S. Choi, C. Ott, and W. K. Chung, "A Passivity-Based Nonlinear Admittance Control With Application to Powered Upper-Limb Control Under Unknown Environmental Interactions," *IEEE/ASME Trans. Mechatron.*, vol. 24, no. 4, pp. 1473–1484, Aug. 2019.

[3] X. Zhang, L. Sun, Z. Kuang, and M. Tomizuka, "Learning Variable Impedance Control via Inverse Reinforcement Learning for Force-Related Tasks," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2225–2232, Apr. 2021.

[4] A. Pozzi, L. Puricelli, M. Rossoni, E. Spadoni, M. Carulli, M. Bordegoni, and G. Colombo, "Context-Aware Industrial Robot Testing: Low-Cost Virtual Prototyping Environment," in *Proc. ASME 42nd Comput. Inform. Eng. Conf.* American Society of Mechanical Engineers, Aug. 2022.

[5] L. Roveda and D. Piga, "Sensorless environment stiffness and interaction force estimation for impedance control tuning in robotized interaction tasks," *Auton. Robots*, vol. 45, no. 3, pp. 371–388, Mar. 2021.

[6] M. T. Mason, "Compliance and Force Control for Computer Controlled Manipulators," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 11, no. 6, pp. 418–432, Jun. 1981.

[7] N. Hogan, "Impedance Control: An Approach to Manipulation," in *Proc. Amer. Contrl. Conf.*, San Diego, CA, USA, Jul. 1984, pp. 304–313.

[8] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 43–53, Feb. 1987.

[9] S. Jung, T. C. S. Hsia, and R. G. Bonitz, "Force Tracking Impedance Control of Robot Manipulators Under Unknown Environment," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 3, pp. 474–483, May 2004.

[10] V. Petrone, E. Ferrentino, and P. Chiacchio, "Time-Optimal Trajectory Planning With Interaction With the Environment," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10 399–10 405, Oct. 2022.

[11] H. Seraji and R. D. Colbaugh, "Force Tracking in Impedance Control," *Int. J. Robot. Res.*, vol. 16, no. 1, pp. 97–117, Feb. 1997.

[12] X. Liang, H. Zhao, X. Li, and H. Ding, "Force tracking impedance control with unknown environment via an iterative learning algorithm," in *Proc. IEEE 3rd Int. Conf. Adv. Robot. Mechatron.*, Singapore, Singapore, Jul. 2018, pp. 158–164.

[13] H. Huang, Y. Guo, G. Yang, J. Chu, X. Chen, Z. Li, and C. Yang, "Robust Passivity-Based Dynamical Systems for Compliant Motion Adaptation," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 6, pp. 4819–4828, Dec. 2022.

[14] Í. Elguea-Aguinaco, A. Serrano-Muñoz, D. Chrysostomou, I. Inziarte-Hidalgo, S. Bøgh, and N. Arana-Arexolaleiba, "A review on reinforcement learning for contact-rich robotic manipulation tasks," *Robot. Comput.-Integr. Manuf.*, vol. 81, p. 102517, Jun. 2023.

[15] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual Reinforcement Learning for Robot Control," in *Proc. IEEE Int. Conf. Robot. Autom.*, Montreal, QC, Canada, May 2019, pp. 6023–6029.

[16] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning," in *Proc. IEEE Int. Conf. Robot. Autom.*, Brisbane, QLD, Australia, May 2018.

[17] H. Huang, C. Yang, and C. L. Philip Chen, "Optimal Robot–Environment Interaction Under Broad Fuzzy Neural Adaptive Control," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3824–3835, Jul. 2021.

[18] L. Roveda, A. Testa, A. A. Shahid, F. Braghin, and D. Piga, "Q-Learning-based model predictive variable impedance control for physical human-robot collaboration," *Artif. Intell.*, vol. 312, Nov. 2022, art. no. 103771.

[19] X. Liu, S. S. Ge, F. Zhao, and X. Mei, "Optimized Interaction Control for Robot Manipulator Interacting With Flexible Environment," *IEEE/ASME Trans. Mechatron.*, vol. 26, no. 6, pp. 2888–2898, Dec. 2021.

[20] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE Int. Conf. Intell. Robots Syst.* Vilamoura-Algarve, Portugal: IEEE, Oct. 2012.

[21] G. Peng, C. L. Philip Chen, W. He, and C. Yang, "Neural-Learning-Based Force Sensorless Admittance Control for Robots With Input Deadzone," *IEEE Trans. Ind. Electron.*, vol. 68, no. 6, pp. 5184–5196, Jun. 2021.

[22] A. Formenti, G. Bucca, A. A. Shahid, D. Piga, and L. Roveda, "Improved impedance/admittance switching controller for the interaction with a variable stiffness environment," *Complex Eng. Syst.*, vol. 2, no. 3, p. 12, Jul. 2022.

[23] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 31, Dec. 2018, pp. 4754–4765.

[24] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Anchorage, AK, USA, Jul. 2019, pp. 2623–2631.

[25] L. Puricelli, A. Pozzi, V. Petrone, E. Ferrentino, P. Chiacchio, F. Braghin, and L. Roveda, "Optimized Residual Action for Interaction Control with Learned Environments," Jan. 2023, youtube. [Online]. Available: https://www.youtube.com/watch?v=4VjteFCNIRU

[26] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, Apr. 2016.

[27] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, "robosuite: A Modular Simulation Framework and Benchmark for Robot Learning," in *arXiv preprint*, Sep. 2020.