A Self-Tuning Impedance-based Interaction Planner for Robotic Haptic Exploration

Yasuhiro Kato[1], Pietro Balatti[2] *Member, IEEE*, Juan M. Gandarias[2] *Member, IEEE*, Mattia Leonori[2], Toshiaki Tsuji[1] *Member, IEEE* and Arash Ajoudani[2] *Member, IEEE*

Abstract—This paper presents a novel interaction planning method that exploits impedance tuning techniques in response to environmental uncertainties and unpredictable conditions using haptic information only. The proposed algorithm plans the robot's trajectory based on the haptic interaction with the environment and adapts planning strategies as needed. Two approaches are considered: *Exploration* and *Bouncing* strategies. The *Exploration* strategy takes the actual motion of the robot into account in planning, while the *Bouncing* strategy exploits the forces and the motion vector of the robot. Moreover, self-tuning impedance is performed according to the planned trajectory to ensure compliant contact and low contact forces. In order to show the performance of the proposed methodology, two experiments with a torque-controller robotic arm are carried out. The first considers a maze exploration without obstacles, whereas the second includes obstacles. The proposed method performance is analyzed and compared against previously proposed solutions in both cases. Experimental results demonstrate that: i) the robot can successfully plan its trajectory autonomously in the most feasible direction according to the interaction with the environment, and ii) a compliant interaction with an unknown environment despite the uncertainties is achieved. Finally, a scalability demonstration is carried out to show the potential of the proposed method under multiple scenarios.

Index Terms—Compliance and Impedance Control, Integrated Planning and Control, Planning under Uncertainty.

I. INTRODUCTION

FLEXIBLE materials naturally deform when interacting with the surrounding environment. This behavior enables us to exploit environmental constraints, facilitating 'blind' explorations through physical interaction, e.g., insertion tasks [1], [2] or pipe inspection [3]. For instance, running a wire inside the cable raceways hidden in a wall becomes a very simple task, which can be accomplished by using a flexible stick and pushing it from an external tip. In other words, the elastic behavior allows solving the maze even without 'planning' the
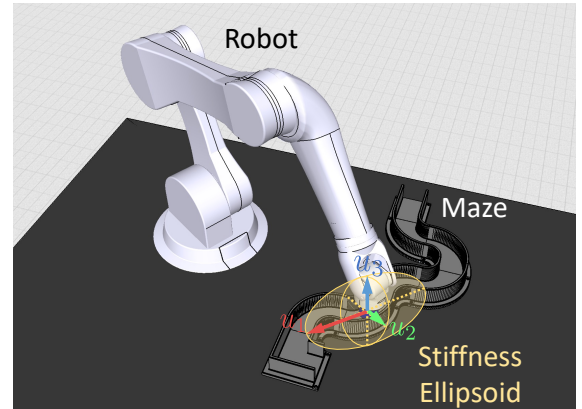
Fig. 1. Maze exploration concept: the robotic arm carries out an exploration task in an unknown and constrained environment, planning the trajectories while interacting with it. The impedance parameters are adaptively tuned online according to the direction of the motion: the robot is stiffer along the principal axis of the stiffness ellipsoid (u_1), and more compliant in the others (u_2, u_3). The arrows' length represent the Cartesian stiffness and damping values.

trajectory. Such elastic behavior (i.e., reacting and accommodating to changes in the environment) can be replicated by a robot in autonomous tasks despite existing uncertainties and missing of visual data.

Traditional solutions rely on trajectory planners to deal with uncertainties, avoiding unexpected contacts with obstacles such as RRT [4] or PRM [5]. These approaches need to build or estimate a highly accurate environment model. However, the more complex the environment, the more challenging is the modeling of the environment, which makes these solutions impractical. On the other hand, allowing contacts with the environment and planning based on haptic information enables the robot to perform better and more autonomously in complex environments. In this context, planners based on contact forces have been proposed [6], [7]. Although the environmental uncertainty is significantly reduced by considering contact, those planning methods are computationally expensive and require building/having a map of the environment. Trajectory-scaling [8] or admittance behavior [9] are computationally less expensive and can flexibly react to the external world. However, these solutions may not be suitable for exploration/path-finding tasks in which the environment can change significantly.

Regarding the problem of physical interaction with unstructured environments, impedance controllers [10] are known to be an effective way to regulate the interaction forces when a contact occurs. In particular, variable impedance control [11], [12] can yield soft contacts between the robot and the en-

vironment/human when necessary. This can be advantageous in human-collaborative scenarios, since the compliance introduced by the controller allows the person to modify the robot's trajectory [13]. On the other hand, in the impedance control scheme, high stiffness may be required to achieve high precision or interaction forces [14]–[16]. For instance, in a collaborative sawing task [17], the stiffness values were higher in the direction of sawing, to create cutting forces, and lower in others, to comply with the task constraints. Therefore, it is necessary to tune the impedance profiles of the robots based on the actual task, human, and environmental states. In such a way, robots can respond to modeling/environment uncertainties or unexpected changes [18], [19].

This paper proposes a novel interaction planner that shapes the trajectory and the direction of maximum stiffness without prior knowledge of the environment, only exploiting real-time haptic interaction, being naturally compliant to the environmental constraints. This method regulates the haptic interaction with the external world by an active impedance control scheme. One of the most significant benefits of our method is that it does not require prior knowledge/map of the environment. Furthermore, our interaction planner adapts two planning strategies based on the robot state. The *Exploration* strategy plans by considering the actual concurrent motion of the robot, while the *Bouncing* strategy employs the interaction forces and previous motion vector of the robot. On the other hand, the second strategy reacts when the robot gets stuck by the environmental constraints. The main components of the proposed approach are the novel interaction planner and our self-tuning impedance technique, previously presented in [20]. The interaction planner, which deploys the robot's motion and estimated forces, is responsible for planning the exploration trajectory by interacting with the environment. The self-tuning impedance method renders a stiffer profile in the direction of its primary motion while still being compliant along the other axes. This method allows the robot to adapt to environmental changes and avoid unnecessarily high interaction forces when in contact with rigid materials.

We validate the proposed approach with experiments on a real system using the Franka Emika Panda robotic arm. To do so, we built a modular maze to create different paths with rigid environmental constraints to evaluate the robot's performance during autonomous path generation and tracking. The proposed method was additionally compared to fully stiff and compliant cases using the same *Exploration* and *Bouncing* logics.

The paper is structured as follows: Sec. II and III present preliminaries and the proposed interaction planning strategy. Sec. IV discusses the experimental setup and the experiments. Finally, discussion and conclusion are addressed in Sec. V.

II. PRELIMINARY

A. Cartesian impedance controller

An active impedance control can be implemented through torque sensing and actuation [21], with the robot joint torques $\boldsymbol{\tau} \in \mathbb{R}^n$ calculated as:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_{\text{ext}}, \qquad (1)$$

$$\boldsymbol{\tau}_{\text{ext}} = \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{F}_c + \boldsymbol{\tau}_{\text{st}}, \qquad (2)$$

where n is the number of joints, $\boldsymbol{q} \in \mathbb{R}^n$ is the joint angles vector, $\boldsymbol{J} \in \mathbb{R}^{6 \times n}$ is the Jacobian matrix, $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ is the mass matrix, $\boldsymbol{C} \in \mathbb{R}^{n \times n}$ is the Coriolis and centrifugal matrix, and $\boldsymbol{g} \in \mathbb{R}^n$ is the gravitational vector. $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^n$ represents the external torques and $\boldsymbol{\tau}_{\text{st}} \in \mathbb{R}^n$ is the secondary task torques in the null-space of $\boldsymbol{J}$. The Cartesian forces $\boldsymbol{F}_c \in \mathbb{R}^6$ are calculated as:

$$\boldsymbol{F}_c = \boldsymbol{K}_c(\boldsymbol{x}_d - \boldsymbol{x}) + \boldsymbol{D}_c(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}), \qquad (3)$$

where $\boldsymbol{K}_c \in \mathbb{R}^{6 \times 6}$ and $\boldsymbol{D}_c \in \mathbb{R}^{6 \times 6}$ represent, respectively, the Cartesian stiffness and damping matrix. $\boldsymbol{x}_d$ and $\boldsymbol{x} \in \mathbb{R}^6$ are the Cartesian desired and actual poses, while $\dot{\boldsymbol{x}}_d$ and $\dot{\boldsymbol{x}} \in \mathbb{R}^6$ represent their corresponding velocity profiles.

B. Self-tuning impedance unit

Hereafter, we introduce the basic principles of the self-tuning impedance controller, originally presented in [20], that is going to be employed with the method proposed in this work. This adaptive controller allows the robot to accurately track the desired motion along the motion vector, but also allows flexibility along the other directions to make the robot adapt to external unintended disturbances (e.g., obstacles). Being more stiff in the principal direction of the movement, and more compliant in the others can be achieved by tuning the major axis of the Cartesian stiffness and damping ellipsoids in the direction of interaction, i.e., deriving the Cartesian stiffness and damping matrices (symmetric and positive definite) as follows:

$$\boldsymbol{K}_c = \boldsymbol{U}\boldsymbol{\Sigma}_k\boldsymbol{U}^T, \qquad (4)$$

$$\boldsymbol{D}_c = \boldsymbol{U}\boldsymbol{\Sigma}_d\boldsymbol{U}^T, \qquad (5)$$

where the diagonal matrix $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Sigma}_d$ are the desired stiffness and damping factors along the direction of the vector composing the orthonormal basis $\boldsymbol{U}$, whose first column represents the motion vector, calculated as $\boldsymbol{x}_{d,t} - \boldsymbol{x}_{d,t-1}$ and defined as u_1 in Fig. 1, and the remaining ones (i.e., u_2 and u_3) are shaped so as to compose an orthonormal basis. $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Sigma}_d$ are defined by:

$$\boldsymbol{\Sigma}_k = \text{diag}(k_{\max}, k_{\min}, k_{\min}), \qquad (6)$$

$$\boldsymbol{\Sigma}_d = \text{diag}(d_{\max}, d_{\min}, d_{\min}), \qquad (7)$$

where $k_{\max}$ is the maximum controllable stiffness and $d_{\max} = 2\zeta\sqrt{k_{\max}}$ [22] its corresponding damping value. Likewise, $k_{\min}$ and $d_{\min}$ are the minimum stiffness and damping coefficients.

III. METHOD

A. Interaction planner

In this study, we consider the case where the robot is not given any prior knowledge about the environment it needs to interact with. Additionally, no external sensors are used (e.g., camera or depth sensor), and the only information the system can rely on are represented by the robot's internal torque sensing, which is used to estimate the external interaction forces. Considering these conditions, in this work we propose an
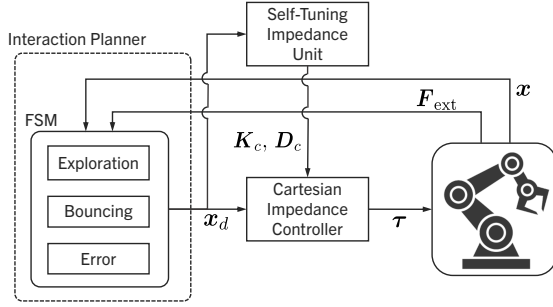
Fig. 2. System framework: The framework is composed of 3 modules, Cartesian impedance controller, self-tuning impedance unit, and interaction planner. The self-tuning impedance unit and interaction planner rely on sensing robot state, i.e., external forces and robot pose.

Fig. 3. Interaction planner *Exploration* state: (left) shows the phase when $\boldsymbol{F}_{\text{ext}}$ exceeds $F_{\text{th\_low}}$, and thus $\boldsymbol{r}_{\text{low}}$ is stored. (middle) shows the phase when $\boldsymbol{F}_{\text{ext}}$ exceeds $F_{\text{th\_high}}$, and thus where $\boldsymbol{r}_{\text{high}}$ is stored. (right) shows the new motion, $\hat{\boldsymbol{r}}_{\text{nd}}$, based on the displacement caused by the contacts with the environment.

Algorithm 1 Exploration algorithm

Input: $\boldsymbol{x}, \boldsymbol{F}_{\text{ext}}$
Output: $\boldsymbol{x}_d$
　　Control loop :
1: **if** ($F_{\text{th\_high}}$ is not detected) **then**
2: 　　**if** ($F_{\text{th\_low}}$ is not detected) **then**
3: 　　　　**if** ($\boldsymbol{F}_{\text{ext}} >= F_{\text{th\_low}}$) **then**
4: 　　　　　　$\boldsymbol{r}_{\text{low}} = \boldsymbol{x}$ {$F_{\text{th\_low}}$ is detected}
5: 　　　　**end if**
6: 　　**end if**
7: 　　**if** ($\boldsymbol{F}_{\text{ext}} >= F_{\text{th\_high}}$) **then**
8: 　　　　$\boldsymbol{r}_{\text{high}} = \boldsymbol{x}$ {$F_{\text{th\_high}}$ is detected}
9: 　　**end if**
10: **end if**
11: $\boldsymbol{r}_{\text{nd}} = \boldsymbol{r}_{\text{high}} - \boldsymbol{r}_{\text{low}}$
12: $\hat{\boldsymbol{r}}_{\text{nd}} = \dfrac{\boldsymbol{r}_{\text{nd}}}{\|\boldsymbol{r}_{\text{nd}}\|_2}$
13: $\Delta \boldsymbol{r}_d = \hat{\boldsymbol{r}}_{\text{nd}} * v_{\text{constE}} * \Delta T$
14: $\boldsymbol{x}_{d,t} = \boldsymbol{x}_{d,t-1} + \Delta \boldsymbol{r}_d$

---

interaction planner that is able to plan the desired trajectories based on the interaction with the environment. The proposed interaction planner mainly consists of two strategies, i.e., *Exploration* and *Bouncing*. The robot autonomously switches between the two, based on the sensed contact forces. This planner was inspired by [23], where planning trajectories are achieved by scaling the desired velocity based on contact forces. Such approach enables flexible trajectory planning without prior knowledge on the external environments. We have exploited and extended this algorithm for the presented *Bouncing* strategy. This strategy has a wider range of scaling factors $[-1...1]$, which allows us not only to avoid obstacles, but also to plan a trajectory towards the goal flexibly by considering interaction forces and motion vector.

1) Exploration: in this state, the robot is in free space or, if in contact with the environment, is still able to move along the desired motion (subject to the relative environmental perturbations). The desired direction of the motion is modified only when the contact with the environment reaches the values of certain force thresholds, meaning that the robot needs to deviate its trajectory to be able to explore the surrounding environment, as illustrated in Fig. 3.

To this end, the *Exploration* algorithm, whose pseudo-code is reported in Alg. 1, calculates the new desired pose $\boldsymbol{x}_d$ based on the actual pose $\boldsymbol{x}$ and on the sensed external forces $\boldsymbol{F}_{\text{ext}}$. The algorithm is in charge of detecting two poses, namely $\boldsymbol{r}_{\text{low}}$ and $\boldsymbol{r}_{\text{high}}$, that are then used to compute the new motion direction, $\hat{\boldsymbol{r}}_{\text{nd}}$ and its relative increment $\Delta \boldsymbol{r}_{\text{nd}}$. The above-mentioned poses are calculated by assigning the actual robot pose $\boldsymbol{x}$ in two different time frames, triggered by two external forces thresholds. The first, $\boldsymbol{r}_{\text{low}}$, is retrieved when the robot external forces go beyond a first threshold named $F_{\text{th\_low}}$. While the contact force exceeds $F_{\text{th\_low}}$, the algorithm checks if the external forces go beyond the second threshold, $F_{\text{th\_high}}$. These thresholds are set empirically. When this is the case, the actual robot pose is assigned to the second pose, $\boldsymbol{r}_{\text{high}}$. From these values, the new motion vector $\boldsymbol{r}_{\text{nd}}$ is computed, and then normalized as $\hat{\boldsymbol{r}}_{\text{nd}}$ so as to avoid significant incremental changes. Lastly, the increment to the previous desired pose, $\boldsymbol{x}_{d,t-1}$, is calculated as in line 13, where v_{constE} is the desired velocity, and ΔT represents control loop time.

2) Bouncing: this second strategy was implemented to substantially change the direction of the motion when the robot gets stuck by environme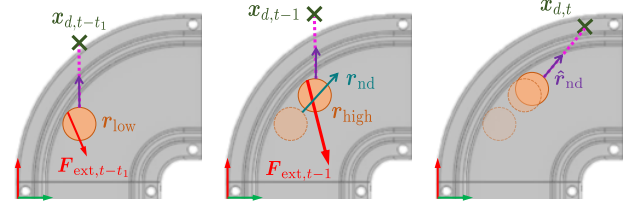ntal constraints. When the robot is in contact with the environment perpendicularly, the *Exploration algorithm* presented above is unable to find the movable direction because it relies on the displacement caused by two different contacts with the environment, in order to plan a new trajectory. As a representative example, we can consider an L-shaped constraint that has limited possible directions of movement, as illustrated in Fig. 4.

To address this issue, the new direction that needs to be computed cannot rely anymore on two consecutive poses, but the robot needs to be steered to another direction, i.e., bouncing from the contact point. To determine the bouncing direction, we consider the normalized external forces vector $\hat{\boldsymbol{F}}_{\text{ext}} \in \mathbb{R}^6$ when the robot gets stuck by environmental constraints, and the unit vector $\hat{\boldsymbol{r}}_{\text{trend}} \in \mathbb{R}^3$, that is the robot motion vector derived as $\boldsymbol{r}_{\text{trend}} = \boldsymbol{x}_t - \boldsymbol{x}_{t-m}$,

where m represents the number of time steps in the past that are considered. The larger this values is set, the bigger the robot's motion trend becomes. The desired velocity v_{constB}, is scaled by the scaling factor α, $\beta \in [-1...1]$ which are calculated as line 5 and 9 in Alg. 2, where $\phi(x) \in [-\pi...\pi]$ and $\phi(y) \in [-\pi...\pi]$ are angles between $\hat{\boldsymbol{F}}_{\text{ext}}(x)$ and $\hat{\boldsymbol{r}}_{\text{trend}}$, as well as $\hat{\boldsymbol{F}}_{\text{ext}}(y)$ and $\hat{\boldsymbol{r}}_{\text{trend}}$ respectively. The angle between external forces and motion vector is calculated by:

$$\phi = arccos \left(\frac{< \hat{\boldsymbol{F}}_{\text{ext}}, \hat{\boldsymbol{r}}_{\text{trend}} >}{\|\hat{\boldsymbol{F}}_{\text{ext}}\| \|\hat{\boldsymbol{r}}_{\text{trend}}\|} \right). \tag{8}$$
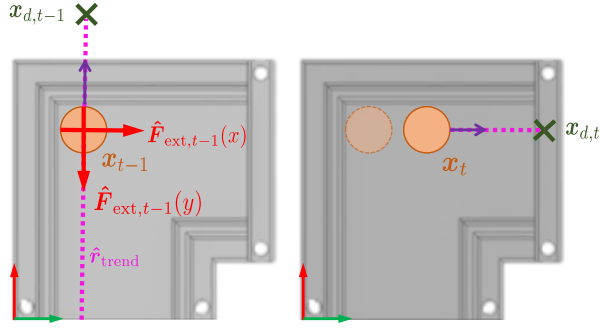
Fig. 4. Interaction planner *Bouncing* state: (left) shows the phase where a robot is trapped by environmental constraints. At this point, by considering the sensed external forces and the motion trend vector, the robot finds a new direction as shown in (right).

Algorithm 2 Bouncing algorithm

Input: $\boldsymbol{r}_{\text{trend}}, \boldsymbol{F}_{\text{ext}}$
Output: $\boldsymbol{x}_d$
  Initialization :
1: $\hat{\boldsymbol{F}}_{\text{ext}} = \dfrac{\boldsymbol{F}_{\text{ext}}}{\|\boldsymbol{F}_{\text{ext}}\|_2}$

2: $\hat{\boldsymbol{r}}_{\text{trend}} = \dfrac{\boldsymbol{r}_{\text{trend}}}{\|\boldsymbol{r}_{\text{trend}}\|_2}$

3: $\phi(x) = getAngleBetween(\hat{\boldsymbol{F}}_{\text{ext}}(x), \hat{\boldsymbol{r}}_{\text{trend}})$
4: $\phi(y) = getAngleBetween(\hat{\boldsymbol{F}}_{\text{ext}}(y), \hat{\boldsymbol{r}}_{\text{trend}})$
5: $\alpha = (1 - |cos(\phi(x))|)$
6: **if** $(\hat{\boldsymbol{F}}_{\text{ext}}(x) < 0.0)$ **then**
7:    $\alpha = -\alpha$
8: **end if**
9: $\beta = (1 - |cos(\phi(y))|)$
10: **if** $(\hat{\boldsymbol{F}}_{\text{ext}}(y) < 0.0)$ **then**
11:    $\beta = -\beta$
12: **end if**
  Control loop :
13: $\Delta r_d(x) = \alpha * v_{\text{constB}} * \Delta T$
14: $\Delta r_d(y) = \beta * v_{\text{constB}} * \Delta T$
15: $\boldsymbol{x}_{d,t} = \boldsymbol{x}_{d,t-1} + \Delta \boldsymbol{r}_d$

Furthermore, the scaling factor was designed to be larger when two vectors are perpendicular while it becomes smaller when two vectors are on the same axis.

B. Finite state machine

A Finite State Machine (FSM) is also implemented to enable the robot to change its strategy according to its state. The FSM consists of the following states: *Exploration*, *Bouncing*, and *Error*. The default robot behavior resides in the *Exploration* state, where the robot can navigate through the environment. On the other hand, the FSM shifts to the *Bouncing* state if the robot is considered to be trapped by the constraints. To switch between the two states, the following equation is introduced:

$$\Delta d = \|\boldsymbol{x} - \boldsymbol{x}_{t-\text{h}}\|_2 \qquad (9)$$

where Δd is the displacement and h is the sampling time. If $\Delta d < R_{\text{th}}$, where R_{th} is the displacement threshold, the FSM shifts into the *Bouncing* state to continue the exploration task.

After the interaction forces decrease, the FSM switches back to the *Exploration* state once again. Additionally, the *Error* state is implemented since we assume that the interaction between the robot and the rigid environment can generate large interaction forces. When these forces become excessively large, the exploration task gets aborted.

IV. EXPERIMENTS AND RESULTS

A. Experiment setup and description

The software architecture was developed with the robotics middleware Robot Operating System (ROS) using C++ as client library. The experimental setup included a Franka Emika Panda robotic arm, three different mazes placed in front of the robot, representing environmental constraints, and a peg (Diameter: 30mm - Length: 55mm) mounted at the robot end-effector. The maze parts and the peg were designed and 3D printed with rigid plastic material (PLA). The mazes were constructed from components that could be disassembled and were built combining linear, curved, and L-shaped parts.

In order to validate the presented method, we conducted several experiments, in which the robot had no prior knowledge about the environment it had to interact with. The first two experiments (Sec. IV-B) demonstrate the performances of the *interaction planner* with the adaptive impedance controller without and with cluttered conditions, i.e., causing interaction resistance. In the third experiment (Sec. IV-C), the robot performs the exploration task in other two mazes. The maze designs were different from the one in the first two experiments in order to show the scalability of the proposed method. A video of the experiments is available in the multimedia extension[1].

B. Maze exploration experiment

In the first experiment, the performance of the maze exploration task was evaluated in three scenarios: high impedance, low impedance and self-tuning impedance conditions. In the first two cases, the diagonal stiffness values were set to constant values on the three Cartesian axes, i.e., 1000 N/m for the high impedance scenario, and 300 N/m for the low impedance one. In the latter, the value was selected so as to let the robot follow the desired pose of the robot while still keeping a high level of compliance with the environment.

In the self-tuning impedance scenario, the robot's impedance controller achieved a Cartesian impedance profiles along the direction of the motion (xy plane). The maximum controllable stiffness along the primary motion axis was set to $k_{\text{max}} = 1000$ N/m, while the minimum controllable stiffness was set to $k_{\text{min}} = 300$ N/m. The values of both k_{max} and k_{min} were set to align with the high and low impedance scenarios for a fair comparison.

We follow the FSM states sequence to describe the experiments. After the robot is sent to the starting position, the *Exploration* state takes place. In the *Exploration* state, an interaction with the environment was expected to happen, and therefore the self-tuning impedance unit was enabled. As the

---

[1]The video can also be found at https://youtu.be/DDqosdN2274

robot came into contact with the environment, the interaction planner changed the desired pose by shaping the desired velocity v_{constE}, which was set to 0.04 m/s, when external forces exceeded the two forces thresholds set to $F_{\text{th\_low}} = 5$ N and $F_{\text{th\_high}} = 7$ N. Those forces threshold values were set with relatively high values, so as to mitigate the influence of the signal noises. Consequently, the impedance tuning allowed the robot to be rigid only in the direction of the primary motion and compliant in all the other directions, thus ensuring to make a compliant contact with the rigid environment which is the crucial information for our interaction based planner. At the very end of the maze, an L-shaped corner was set up and the robot was trapped in a state where it was unable to move. This is because the interaction planner in the *Exploration* state modifies the trajectory based on the displacement caused by the environment, and the robot was unable to detect a movable direction colliding perpendicularly to the surface of the environment. Here, following (9), with $R_{\text{th}} = 1$ mm and $h = 500$ ms, the FSM shifted to the *Bouncing* state, since $\Delta d < R_{\text{th}}$. In the *Bouncing* state, the robot determined the possible direction of motion by scaling v_{constB}, which was set to 0.05 m/s, based on two vectors i.e., $\hat{\boldsymbol{F}}_{\text{ext}}$ and $\hat{\boldsymbol{r}}_{\text{trend}}$. The sampling time steps to update $\hat{\boldsymbol{r}}_{\text{trend}}$ was set to $m = 2000$. Fig. 6 shows the experimental result of exploration task with the three impedance profiles described above. In all the scenarios, the proposed interaction planner enabled the robot to complete the exploration task based on the contacts with the environment. However, it was observed from the results that the interaction behavior appears different for each impedance profile. To compare the results among the different controllers, we also define the expected completion time (CT), assuming the optimal collision-free path exploration at $\dot{x}_d$, and its exploration distance (ED). For the maze configuration illustrated in Fig. 5 these two values are 26.0 s and 1.06 m, respectively.

Fig. 6 (left) shows the data obtained while maintaining high impedance profiles (1000 N/m). Although, this setup showed high desired pose tracking capability ($\Delta \boldsymbol{x}_{\text{max}} \approx 0.04$ m), where $\Delta \boldsymbol{x}$ is the tracking error, the high impedance profiles resulted in higher average interaction forces $F_{\text{avg}} \approx 23$ N due to the rigid contact with the environment. In terms of performances, the actual CT was equal to 31 s and ED 1.16 m.

Fig. 6 (middle) shows the plots of the trial with low impedance profiles (300 N/m). Despite this setup provides compliance, relatively higher $F_{\text{avg}} \approx 18$ N were observed due to the larger value of $\Delta \boldsymbol{x}_{\text{max}} \approx 0.10$ m. The possible cause of larger $\Delta \boldsymbol{x}$ is as follows. For low impedance profiles, the interaction forces gradually increased. Since the force thresholds used by the proposed interaction planner to change the trajectory was set to relatively high values, this allowed for a larger $\Delta \boldsymbol{x}$ before changing the trajectory. Another drawback given by the lower impedance profiles, was that the robot was unable to compensate for $\Delta \boldsymbol{x}$ due to its lower tracking ability for desired pose, thus highlighting a less desirable behavior. Moreover, the actual CT was 36 s and ED was 1.13 m.

Fig. 6 (right) depicts the data employing the self-tuning impedance method. $k_{\text{max}} = 1000$ N/m was set along the primary motion axis while $k_{\text{min}} = 300$ N/m was set in the

Fig. 5. Experimental setup: (top) is the maze path for the first experiment. (bottom) is the maze with a cluttered condition by screws for the second experiment.

other axis. Since the high impedance was maintained in the motion direction, the error was as small as the high impedance profiles. Furthermore, $F_{\text{avg}} \approx 11$ N were lower than the low impedance profiles because of the low impedance in non primary motion direction and the smaller $\Delta \boldsymbol{x}_{\text{max}}$ (≈ 0.04 m). Additionally, the actual CT was 32 s and ED was 1.13 m. CT is comparable to the stiff case, and 11% lower with respect to the compliant one. On the other hand, ED is comparable to the compliant case and 2.5% lower with respect to the stiff case. Taking into account the optimal values, the increase in CT is 23% while for ED is 7%. These considerations demonstrate the exploration efficiency of our algorithm despite our interaction planner seeks real-time haptic interaction/collision with unknown environments.

In the second experiment, the setup was modified by adding cluttered condition, to add soft resistance to the robot interaction. Possible applications of the proposed method, e.g., insertion tasks/pipe inspection, where some degree of resistance is expected to occur due to the cluttered conditions. It is desirable for a robot to be capable of performing the task regardless of such disturbances. As shown in Fig. 5, the robot performed an exploration task in the same maze as in the first experiment but with screws added along the path. As illustrated in Fig. 7, the experimental result showed that the task was completed only by self-tuning impedance scenario. Figs. 7 (left) and (middle) are the data with high and low impedance profiles, respectively. In high impedance scenario, the sum of interaction forces increased gradually and reached the maximum force threshold (60 N). Hence, the FSM transited to *Error* state which aborted the robot's task execution. In low impedance case, due to lower tracking capability, the robot's motion was disturbed by screws as shown in larger $\Delta \boldsymbol{x}_{\text{max}} \approx 0.07$ m. Approximately at $t = 8$ s, the interaction planner shifted to *Bouncing* behavior due to small actual pose change and generated high interaction forces. Eventually, the robot's task execution was aborted because of the high interaction forces (> 60 N). Fig. 7 (right) shows the robot's behavior in self-tuning impedance scenario. The stiffness ellipsoid design was robust to disturbances in the direction of motion, while being compliant in other directions, thus suppressing high interaction forces.
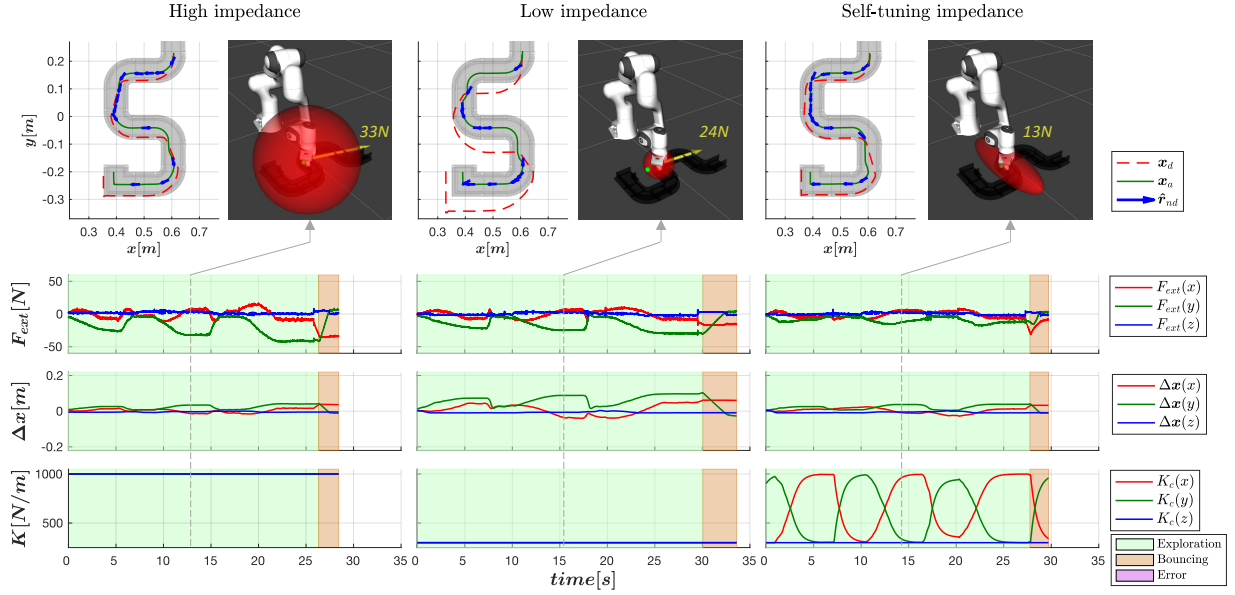
Fig. 6. Comparison of three impedance profiles during exploration task: the upper row shows the maze path and the robot's trajectory. The red dash line shows the desired trajectory while the green line plotted the actual trajectory. The blue arrow shows the motion vector of the robot. Additionally, snapshots of the real robot's behavior at a similar point in the maze are provided. The red marker visualizes stiffness ellipsoid while the yellow arrow shows the interaction forces. The green dot represents the desired pose of the robot. High impedance (left) shows high interaction forces compared to the other impedance profiles. Low impedance (middle) shows larger Δx. This is also visualized in the snapshots as the desired pose was far from the actual end-effector position. Self-tuning impedance (right) shows smaller Δx as in the case of high impedance and interestingly, interaction forces smaller with respect to the low impedance case. Moreover, the execution time of self-tuning impedance is comparable to high impedance while low impedance showed a slightly longer time.
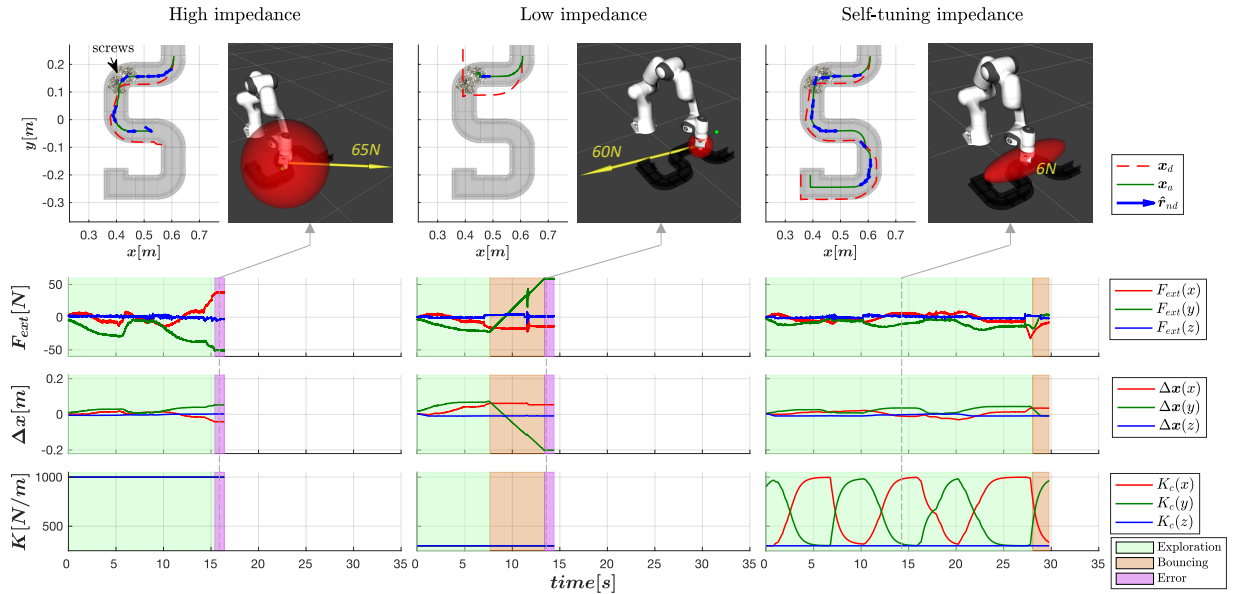

Fig. 7. Comparison of three impedance profiles during exploration task in cluttered condition: as depicted in the first row, the maze path was cluttered by screws. In high impedance (left) and low impedance (middle) scenarios, robot's motion was aborted due to high interaction forces (> 60 N). The self-tuning impedance scenario (right) showed its robustness as it completed the task execution. The interaction forces were significantly lower with self-tuning impedance compare to the other impedance profiles as visualized in the snapshots.

C. Algorithm scalability

In the third experiment, the robot performed an exploration task in two other mazes. The maze designs were different from the one in the first experiment in order to show the scalability of the proposed method. When a robot performs tasks in unstructured environments or pHRC scenarios, a robot's work space is expected to have a wide variety of environmental models. Therefore, it is desirable for a robot

to be capable of performing the task despite various changes of the environment model as the maze was reconstructed in multiple ways. As shown in Figs. 8 and 9, the results of this experiment showed that the proposed planner with self-tuning impedance method can successfully perform the exploration tasks regardless of its variance in the environmental model. In the left side of the figures, the maze paths and the robot's trajectory are plotted. It can be seen that the trajectory was
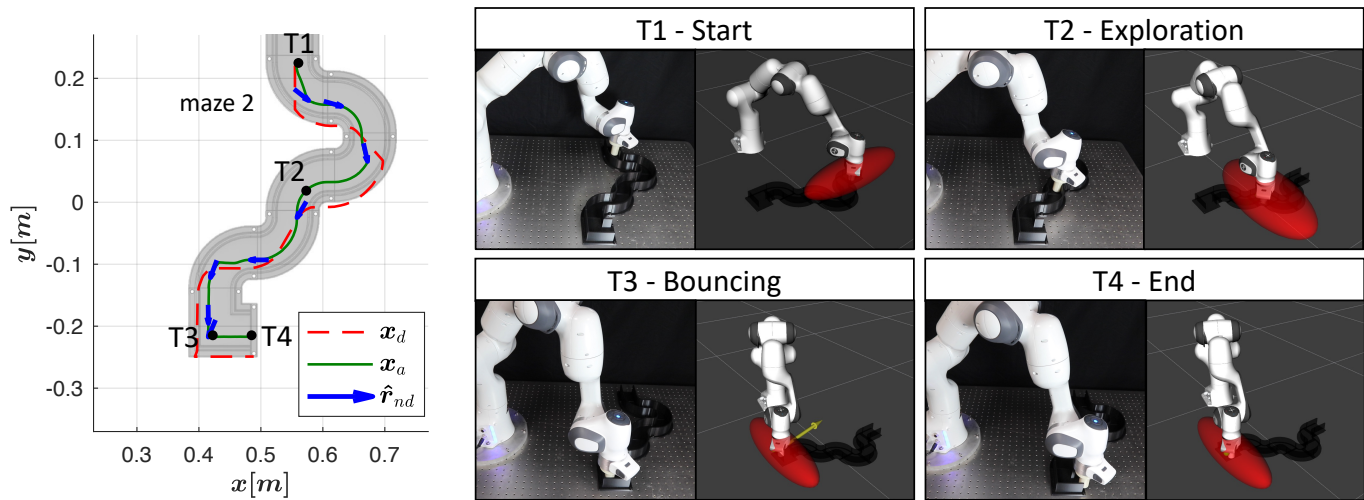
Fig. 8. To show the algorithm scalability with different constrained environments, the interaction planner has been tested with mazes with different features. Here, *maze 2* is constructed mainly by curved parts. As can be inferred by the *xy* plane plot (left), the interaction planner frequently changed the motion direction $\hat{r}_{nd}$, so as to adapt to the continuous changes of the maze path. On the right four different frames show the real robot and its 3D visualization enriched with impedance values (red ellipsoid) and interaction forces (yellow arrow).
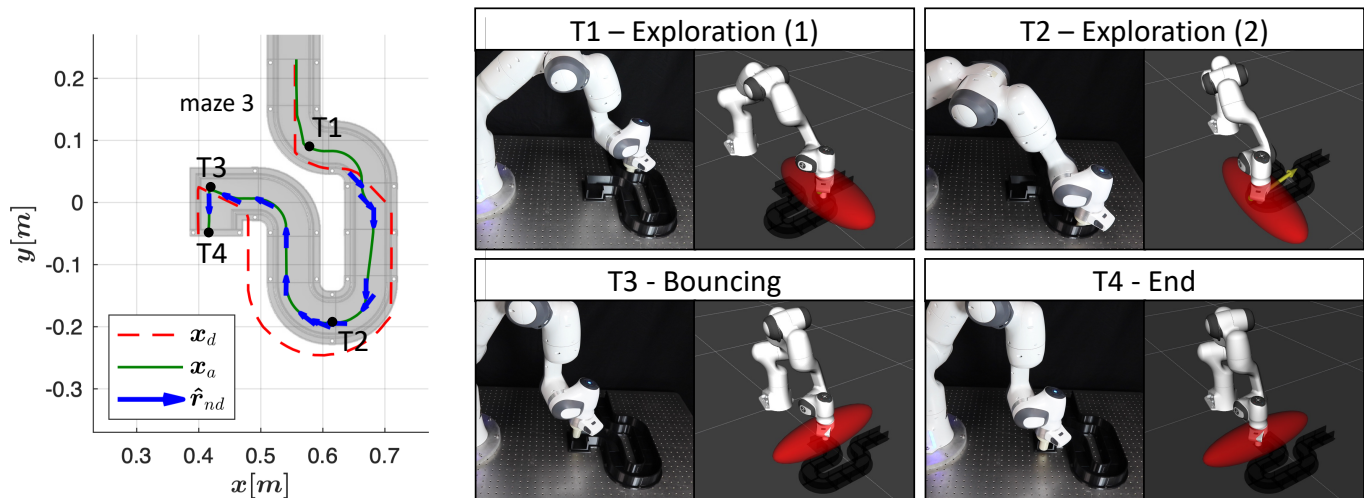

Fig. 9. Similarly as Fig. 8, *maze 3* shows another scenarios where the U-shape requires dynamic changes of the trajectory. Nevertheless, the presented interaction planner was able to successfully execute the exploration task.

adapted for each maze and the robot was following with $\Delta x_{max} \approx 0.04$ m in *maze 2* and $\Delta x_{max} \approx 0.06$ m in *maze 3*. The right part of the figure shows four frames of the real robot's behavior and its 3D visualization enriched with information about its desired pose planned by the interaction planner (green dot), and about the stiffness ellipsoid that was changing its shape/direction accordingly.

V. CONCLUSION AND DISCUSSION

This study presented a novel method to plan the robot's trajectory based on interaction with the environment and tune impedance profiles accordingly. The robot had to execute exploration tasks without prior knowledge of the environment. It performed the task within the FSM by changing its strategies according to robot's state. The proposed interaction planner mainly used *Exploration* and *Bouncing* strategies to perform trajectory planning. The experimental results have shown that the presented interaction planner can successfully plan

trajectories in feasible directions. In addition, a compliant interaction was ensured by tuning impedance despite unexpected changes in the environment. The forces thresholds of the interaction planner should be predefined at reasonable values. This part is not yet automated and was adjusted through preliminary trials. However, it was not in the focus of this study because we considered interacting only with rigid environment, and hence, the concern was mainly signal noises.

The strength of the proposed method relies in the adaptation capabilities to the environmental changes, not relying on learning techniques [7], [24] in both interaction planner and self-tuning impedance. In this study, the robot was required to execute exploration tasks without prior knowledge of the environment. Additionally, several variations of environment models were prepared as in the third experiment. In such a scenario, a robot needs continuous model adaptation in the learning scheme which may not be practical due to scarce data availability. In contrast, our proposed method completed ex-

ploration tasks in several maze paths only by interacting with the environment without the need for such model adaptation.

In addition, we were able to exploit the impedance design in the exploration tasks. The geometric ellipsoid of the self-tuning impedance method is similar to the one of human impedance design which elongates impedance only in the required directions [25]. The second experiment well demonstrated the advantage of our impedance design. In high impedance scenario, since the impedance was also high in the unnecessary axis, the robot generated large energy by interacting with rigid environments which resulted in aborting the task execution, while lower stability against disturbances was observed in low impedance scenario. In self-tuning impedance scenario, by maintaining the high impedance at the contact point, this strategy helped to stabilize the system against disturbances. The stability of the system can be analyzed by the tank-based system passivity observer as already demonstrated in other applications of variable impedance control [26]. Furthermore, the interaction planner and self-tuning impedance unit were closely related in a sense that impedance was tuned according to the planned trajectory. In this way, a compliant interaction was ensured during collisions. This behavior, planning trajectory and impedance simultaneously, is mimicking human behavior of contact tasks [27]. Therefore, our proposed framework has shown robustness for robotic autonomous exploration task in unknown environments. In future works, we will consider the extension of our proposed methodology to 3D mazes and explore its adaptation to bio-inspired haptic-based navigation approaches based on the behavior of vision-impaired insects and animals such as worms and moles.

References

[1] M. Klein, T. Lueth, A. Hein, M. Stien, O. Schermeier, S. Weber, H. Menneking, O. Schwerdtner, and J. Bier, "Robot-assisted insertion of craniofacial implants—clinical experience," in *International Congress Series*, vol. 1230, pp. 131–137, Elsevier, 2001.

[2] N. Chentanez, R. Alterovitz, D. Ritchie, L. Cho, K. K. Hauser, K. Goldberg, J. R. Shewchuk, and J. F. O'Brien, "Interactive simulation of surgical needle insertion and steering," in *ACM SIGGRAPH 2009 papers*, pp. 1–10, 2009.

[3] A. A. Nassiraei, Y. Kawamura, A. Ahrary, Y. Mikuriya, and K. Ishii, "Concept and design of a fully autonomous sewer pipe inspection mobile robot" kantaro"," in *Proceedings 2007 IEEE international conference on robotics and automation*, pp. 136–143, IEEE, 2007.

[4] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2, pp. 995–1001, IEEE, 2000.

[5] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[6] E. Páll, A. Sieverling, and O. Brock, "Contingent contact-based motion planning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6615–6621, IEEE, 2018.

[7] C. Yang, C. Zeng, C. Fang, W. He, and Z. Li, "A dmps-based framework for robot learning and generalization of humanlike variable impedance skills," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1193–1203, 2018.

[8] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3356–3363, IEEE, 2008.

[9] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1623–1630, IEEE, 2006.

[10] N. Hogan, "Impedance control: An approach to manipulation: Part i—theory," 1985.

[11] T. Tsumugiwa, R. Yokogawa, and K. Hara, "Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1, pp. 644–650, IEEE, 2002.

[12] R. Ikeura and H. Inooka, "Variable impedance control of a robot for cooperation with a human," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 3097–3102, IEEE, 1995.

[13] B. Nemec, N. Likar, A. Gams, and A. Ude, "Human robot cooperation with compliance adaptation along the motion trajectory," *Autonomous robots*, vol. 42, no. 5, pp. 1023–1035, 2018.

[14] A. Ajoudani, N. Tsagarakis, and A. Bicchi, "Tele-impedance: Teleoperation with impedance regulation using a body–machine interface," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1642–1656, 2012.

[15] F. J. Abu-Dakka, L. Rozo, and D. G. Caldwell, "Force-based variable impedance learning for robotic manipulation," *Robotics and Autonomous Systems*, vol. 109, pp. 156–167, 2018.

[16] A. Ajoudani, C. Fang, N. Tsagarakis, and A. Bicchi, "Reduced-complexity representation of the human arm active endpoint stiffness for supervisory control of remote manipulation," *The International Journal of Robotics Research*, vol. 37, no. 1, pp. 155–167, 2018.

[17] L. Peternel, N. Tsagarakis, and A. Ajoudani, "Towards multi-modal intention interfaces for human-robot co-manipulation," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 2663–2669, IEEE, 2016.

[18] E. Mobedi, N. Villa, W. Kim, and A. Ajoudani, "An adaptive control approach to robotic assembly with uncertainties in vision and dynamics," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 144–150, IEEE, 2020.

[19] E. Gribovskaya, A. Kheddar, and A. Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans," in *2011 IEEE International Conference on Robotics and Automation*, pp. 4326–4332, IEEE, 2011.

[20] P. Balatti, D. Kanoulas, N. Tsagarakis, and A. Ajoudani, "A method for autonomous robotic manipulation through exploratory interactions with uncertain environments," *Autonomous Robots*, vol. 44, no. 8, pp. 1395–1410, 2020.

[21] N. Hogan, "Stable execution of contact tasks using impedance control," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 1047–1054, IEEE, 1987.

[22] A. Albu-Schaffer, C. Ott, U. Frese, and G. Hirzinger, "Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms," in *2003 IEEE International conference on robotics and automation (Cat. No. 03CH37422)*, vol. 3, pp. 3704–3709, IEEE, 2003.

[23] S. Haddadin, H. Urbanek, S. Parusel, D. Burschka, J. Roßmann, A. Albu-Schäffer, and G. Hirzinger, "Real-time reactive motion generation based on variable attractor dynamics and shaped velocities," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3109–3116, IEEE, 2010.

[24] Z. Li, J. Liu, Z. Huang, Y. Peng, H. Pu, and L. Ding, "Adaptive impedance control of human–robot cooperation using reinforcement learning," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 8013–8022, 2017.

[25] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato, "The central nervous system stabilizes unstable dynamics by learning optimal impedance," *Nature*, vol. 414, no. 6862, pp. 446–449, 2001.

[26] Y. Wu, F. Zhao, T. Tao, and A. Ajoudani, "A framework for autonomous impedance regulation of robots based on imitation learning and optimal control," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 127–134, 2020.

[27] G. Ganesh, A. Albu-Schäffer, M. Haruno, M. Kawato, and E. Burdet, "Biomimetic motor behavior for simultaneous adaptation of force, impedance and trajectory in interaction tasks," in *2010 IEEE International Conference on Robotics and Automation*, pp. 2705–2711, IEEE, 2010.