A method for autonomous robotic manipulation through exploratory interactions with uncertain environments

Pietro Balatti\textsuperscript{1,2} · Dimitrios Kanoulas\textsuperscript{3} · Nikos Tsagarakis\textsuperscript{4} · Arash Ajoudani\textsuperscript{1}

Received: 17 June 2019 / Accepted: 14 July 2020 / Published online: 3 August 2020
© The Author(s) 2020

Abstract
Expanding robot autonomy can deliver functional flexibility and enable fast deployment of robots in challenging and unstructured environments. In this direction, significant advances have been recently made in visual-perception driven autonomy, which is mainly due to the availability of rich sensory data-sets. However, current robots’ physical interaction autonomy levels still remain at a basic level. Towards providing a systematic approach to this problem, this paper presents a new context-aware and adaptive method that allows a robotic platform to interact with unknown environments. In particular, a multi-axes self-tuning impedance controller is introduced to regulate quasi-static parameters of the robot based on previous experience in interacting with similar environments and the real-time sensory data. The proposed method is also capable of differentiating internal and external disruptions, and responding to them accordingly and appropriately. An agricultural experiment with different deformable material is presented to validate robot interaction autonomy improvements, and the capability of the proposed methodology in detecting and responding to unexpected events (e.g., faults).

Keywords Robotic manipulation · Interaction autonomy · Impedance control · Adaptive control

1 Introduction
To respond to the rapidly increasing demand for high levels of flexibility in manufacturing and service applications, recent research has focused on endowing robots with the ability to react and adapt to their environments. From the one hand, robotic systems based on torque sensing and actuation or variable impedance mechanisms have been developed to make them compliant to their surroundings (Albu-Schaffer et al. 2003; Tsagarakis et al. 2016). On the software level instead, a great deal of attention has been devoted to the perception autonomy of the robots, to capture the effects of appearance and context (Kotseruba et al. 2016; Harbers et al. 2017). Although these two directions have seen significant advancements over the past decade, the bridging action, i.e., associating perception to interaction in an autonomous way, still remains in a non-satisfactory level. This fundamental shortcoming has limited the application of robots in out-of-the-cage application scenarios, making a framework to enhance their physical interaction autonomy a critical requirement.

Previous attempts to endow robots with adaptive interaction skills have pursued different directions. A well-known approach is based on learning from human demonstrations (Katz et al. 2014, 2008; Kronander and Billard 2014), which has shown promising results when sufficient training data is available. In fact, the high dependency of observation-based approaches to the quality of training data sets has
been a limiting factor. Moreover, while performing complex manipulation tasks, accurate sensory measurements related to physical interactions (e.g., forces and torques) may not be possible through wearable sensory systems, that is why most learning by demonstration techniques function on a kinematic level.

To provide a solution to this problem, analytical techniques have focused on the use of impedance control (Dietrich et al. 2016; Lee et al. 2014; Ajoudani et al. 2014b), force control for both fixed manipulators (Roveda et al. 2018; Righetti et al. 2014) and mobile base applications (Roveda 2018), or hybrid interaction controllers (Anderson and Spong 1988; Schindlbeck and Haddadin 2015). However, in the majority of cases, the control parameters are pre-selected by robot programmers based on their experience in carrying out analogous tasks. In such a way, the framework cannot adapt when task conditions change, hence, the full potential of such powerful control techniques cannot be exploited (Yang et al. 2011; Ferraguti et al. 2013).

To provide a solution to this shortcoming, adaptive learning techniques have been proposed. In (Xu et al. 2011), an adaptive impedance controller for upper-limb rehabilitation, based on evolutionary dynamic fuzzy neural network was proposed to regulate the impedance profile between the impaired limb and the robot. However, this method lacks versatility, since the algorithm is limited to a specific task. In a similar work (Gribovskaya et al. 2011), empirical constants had to be set, which reduced the flexibility of the framework. In addition, the desired impedance matrices were assumed to be diagonal, resulting in limited adaptability in selective Cartesian axes. More generic methods have been introduced by reduc- ing impedance control to position control (through high position loop gains) when there is no interaction (Li et al. 2012) with the aim to minimize the error between the desired and actual trajectories (He and Dong 2018). This significantly reduced the system’s ability to deliver a distinguished response to the desired (task) and undesired (e.g., collisions) interactions (see also Nemec et al. 2018).

To address these challenges and towards bridging the autonomy gap in the perception-reaction chain, we propose a novel manipulation framework that integrates various components to achieve a context-aware and adaptive robot physical interaction behavior.

The main component of our framework is a multi-axes self-tuning impedance controller, to tune the Cartesian stiffness and damping profiles (see Fig. 1) in any arbitrary direction, which coincide with the direction of interaction, based on the previous experience in interacting with similar environments and the real-time robot sensory data. As (Song et al. 2019) report in the survey and comparison of impedance control techniques on robotic manipulation, variable impedance controllers need to determine when and how to vary the impedance parameters. Autonomously carrying out this non-trivial task falls within the primary aims of this work.

To recognize and localize different materials in the robot workspace and associate their newly/previously identified characteristics to the robot interaction knowledge (i.e., impedance control and self-tuning gains), a visual perception module has been developed. This module, embedded in our multi-axis self-tuning controller, enables the robot to explore an environment, to identify its characteristics, and to effectively interact with it. This behavior is inspired by the human’s way of adapting to their surroundings, by constantly building internal models of the external world, while exploring and identifying it. When interacting with new or similar environments, the prior knowledge is used as a preparatory strategy, while keeping open the possibility of adaption, to update our internal knowledge (Kawato 1999). Another similarity between our method and human behavior is given by the default compliant behavior of the robot. In fact, when no interaction is expected, we tend to relax our muscles to comply with unexpected external disturbances (and to minimize energy consumption).

The organization of the individual components and their inter-connections are achieved through a Finite State Machine (FSM). The proposed FSM fuses the data received from the robot sensors and the vision module, to distinguish expected interactions from disturbances and faults, and accordingly, to self-tune control gains and trajectories in real-time. Note that, the presented FSM is connected to the main experiment we conducted to validate the proposed method. Nevertheless, the algorithm can be applied to many other applications by modifying its states, as it is described in a section dedicated to the algorithm scalability.

We prepared an agricultural experimental setup to demonstrate the potential of the presented methodology in one of the
most promising but somewhat disregarded fields in robotics. Two Franka Emika Panda robotic arms, one equipped with the Pisa/IIT SoftHand (Ajoudani et al. 2014a) and the other with a standard gripper were used to perform the experimental task.

The original idea of this work was presented in a short conference paper (Balatti et al. 2019). However, to advance the methodology and explain its full potential, in this article, we have added significant contributions w.r.t. to the original work. Here, we introduce a novel “Fault Detection” unit, able to recover from unexpected collisions with the boundary (for instance due to a fault of the perception unit). This can also be useful to detect substantial changes in the material properties, since these would be system malfunction symptoms in industrial production. Next, the performance of the multi-axis self-tuning impedance controller is thoroughly evaluated with several details. Additionally, we performed new experiments to compare our method to non-adaptive techniques, to highlight the full potential of the framework in interacting with uncertain environments. Furthermore, we conducted an analysis to guarantee the stability of the controller, since it is based on continuous variations of the impedance parameters. To illustrate the generality and scalability of the presented approach, we also included a new experiment showing the controller adaptation while grasping and handling a pallet jack handle.

2 Method

The main purpose of the presented framework is to enable robots to acquire an original set of skills to explore various environments, identify their characteristics and accordingly adapt to them, build the knowledge, and use it for future interactions. This methodology is based on the concept of self-adaptability, even after building the knowledge on task or environments. In this way, the robot has the ability to adapt by starting from a reasonable initial condition, even if the environmental conditions were subject to changes.

The required theoretical and technological components to build such a framework are integrated into five main modules, that are illustrated in Fig. 2: (1) a Cartesian impedance controller whose parameters can be tuned online, (2) a multi-axes self-tuning impedance unit to tune the aforementioned parameters when an interaction with the environment is predicted, (3) a trajectory planner to calculate the spatial points to be reached by the controller, (4) a visual perception module that locates the materials’ positions in the robot workspace, and (5) a Finite State Machine (FSM) that, based on the data provided by (4), triggers unit (2) and (3), being also responsible of detecting system faults.

2.1 Cartesian impedance controller

Cartesian impedance control techniques provide the ability to achieve any arbitrary quasi-static behavior at the robot end-effector (Schindlbek and Haddadin 2015; Ajoudani et al. 2017). This is however limited to the positive definiteness and symmetry of the impedance matrices, by considering robot torque boundaries and the number of degrees of freedom ($\geq 6$).

This control technique relies on torque sensing and actuation, with the vector of robot joint torques $\tau \in \mathbb{R}^n$ calculated as follows:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_{\text{ext}},$$  (1)
In the near future, robots will enter in several application scenarios to collaborate with humans, in environments designed and built to match their human counterparts’ needs, characterised by high uncertainty levels. To respond to the varying task conditions and uncertainly levels, we aim to develop a novel self-tuning impedance controller that is able to adapt its parameters when an interaction is expected. The adaptation is limited to the expected direction(s) of interaction/movement to avoid unnecessary stiffening/complying of the remaining axes (Fig. 3).

Since our method implies the adaptation of the impedance parameters only in the interactions that are expected, we need to distinguish when this is the case. To this end, we define a Boolean value, named Interaction expectancy \((I_e)\), that results from the Boolean logic rule \(I_e = I_m \land I_f\). The first value, \(I_m\), induced by the FSM, is True only in the states where an interaction with the environment is expected. The second one, \(I_f\), set by the visual perception module, is set to True when the tool attached to the end-effector is inside the material that has to be manipulated. The importance of this consideration has already been shown in our preliminary work (Balatti et al. 2018) and it will not be repeated here.

When no interaction is expected, and therefore \(I_e\) is False, the Cartesian stiffness matrix \(K_c\) is set to a default diagonal matrix with all the non-zero coefficients set to \(k_{\text{min}}\) to deliver a compliant behavior. That is because the base condition of the presented self-tuning impedance controller is to be soft in all Cartesian and redundant axes, unless an interaction is expected to occur. The impedance values to render softness, however, has to be chosen based on a trade-off between the position tracking accuracy (affected by the existence of unmodelled dynamics such as friction) and the force response, if an unexpected interaction occurs.

The damping matrix \(D_c\) is derived from \(K_c\) by:

\[
D_c = \Lambda \Lambda_c^{\ast}D_{\text{diag}}K_{\text{adj}}^{\ast} + K_{\text{adj}}^{\ast}D_{\text{diag}}\Lambda^{\ast},
\]

where \(D_{\text{diag}}\) is the diagonal matrix containing the damping factor \((\zeta = 0.7)\), \(K_{\text{adj}}^{\ast}K_{\text{adj}} = K_c\) and \(\Lambda, \Lambda^{\ast} = \Lambda\), where \(\Lambda\) is the desired end-effector mass matrix (Albu-Schaffer et al. 2003).

Instead, when an interaction is expected, being \(I_e\) True, the Cartesian stiffness matrix \(K_c\) and consequently the damping matrix \(D_c\) are subject to changes increasing (or decreasing) the impedance parameters only along the direction of the desired movement defined by:

\[
\tilde{P} = X_d,t - X_d,t-1,
\]

(which can also be calculated from \(\dot{X}_d\)) and keeping a compliant behavior, set to \(k_{\text{min}}\) and \(d_{\text{min}} = 2\zeta\sqrt{K_{\text{min}}}\) (Albu-Schaffer et al. 2003), along the other axes. To achieve this, the stiffness and damping matrices, as being symmetric and positive definite, can be expressed by:

\[
A = U\Sigma V^*,
\]
which is known by the Singular Value Decomposition (SVD). Such a decomposition enables us to project the desired stiffness and damping, calculated w.r.t. the reference frame of desired motion vector $\vec{P}$, onto the reference frame of the robot base. $U \in \mathbb{R}^3$ and $V \in \mathbb{R}^3$ are orthonormal bases, and $\Sigma \in \mathbb{R}^3$ is a diagonal matrix whose elements are the singular values of matrix $A$ sorted in decreasing order and representing the principal axes amplitudes of the resulting geometric ellipsoid. The columns of matrix $U$ form a set of orthonormal vectors, which can be regarded as basis vectors. In this work, the first column of $U$ represents the desired motion vector $\vec{P}$, while the second and the third ones are derived from the first in such a way they form an orthonormal basis. Since the Hermitian transpose $V^* \in \mathbb{R}^3$, and the resulting matrix $A$, that represents the impedance values, is positive definite, we have that:

\[
V^* = V^T, \quad (7)
\]

\[
V = U. \quad (8)
\]

Combining (6), (7) and (8), we can derive the stiffness and the damping matrix:

\[
K_\epsilon = U \Sigma_k U^T, \quad (9)
\]

\[
D_\epsilon = U \Sigma_d U^T, \quad (10)
\]

where the diagonal matrix $\Sigma_k$ and $\Sigma_d$ coefficients are respectively the desired stiffness and damping coefficients along the direction of the vectors composing the $U$ basis. They are diagonal matrices defined by:

\[
\Sigma_k = \text{diag}(k_{st}, k_{min}, k_{min}), \quad (11)
\]

\[
\Sigma_d = \text{diag}(d_{st}, d_{min}, d_{min}), \quad (12)
\]

where $k_{st}$ is the self-tuning stiffness coefficient to be set along the motion vector $\vec{P}$ and $d_{st}$ its correspondent damping element. $k_{st}$ is defined at every time $t$ as:

\[
k_{st,t} = k_{st,t-1} + \alpha \Delta P \Delta T, \quad (13)
\]

\[
\Delta P = |\hat{\Delta} X \cdot \hat{P}|, \quad (14)
\]

\[
\Delta X = X_d - X_a, \quad (15)
\]

where $\alpha$ is the update parameter, $\Delta P$ is the absolute value of the Cartesian error $\Delta X$ projected onto the direction of the motion vector $\hat{P}$, and normalized $\hat{P}$, where $\Delta T$ is the control loop sample time. It is important to notice that, $k_{st}$ is subject to changes only when $\Delta P$ is beyond a limit. To this end, we introduce a threshold, defined as $\Delta P^*$, since usually in impedance controlled robots it is hard to achieve a small error between the desired and actual position, unless the gains, i.e. stiffness and damping, are very high and it can be compared to position control. In this way we let a small margin of error, not to increase the impedance parameters when it is not required by the task, and to arrest its growth when a desired accuracy is achieved. Moreover, we want to avoid unnecessary adaptation which can be caused by unmodeled robot dynamics and small amount of friction at joints, and therefore the error can also not be related to the task.

In order to increase robot autonomy, the desired impedance parameters need to be adapted in a reasonably short time. This implies an accurate choice of $\alpha$. A high value of this parameter would lead to a rapid convergence in materials with high density. Nevertheless, choosing a high value for non-dense material will cause needless stiffening of the robot that must be avoided. Therefore, to obtain an average $\alpha$ value, as a trade-off between fast convergence and stiffening performance, we performed experiments on different materials such as soil, sand, rocks with different density, air and water.

To achieve this, for every material $m$, $\alpha_m$ was estimated and the average value was defined through the arithmetic mean of all the $n$ materials taken into account in the analysis:

\[
\alpha = \frac{1}{n} \sum_{m=1}^{n} \alpha_m. \quad (16)
\]

There are also situations in which the impedance parameters adaptation has to be carried out in the opposite way, i.e. decreasing them, and (13) cannot be applied. An example is given by the case where the tool attached to the robot end-effector exits the material, even still being inside the interaction expectancy area. We define $\Delta F_{ext,t}$ as the variation of the external forces, along the motion vector, detected at the robot end-effector at time $t$ w.r.t. the ones measured at time $t-1$:

\[
\Delta F_{ext,t} = (F_{ext,t} - F_{ext,t-1}) \cdot \hat{P}. \quad (17)
\]

In the aforementioned situations, $\Delta F_{ext,t}$ is positive and $k_{st}$ is defined at every time $t$ as:

\[
k_{st,t} = k_{st,t-1} - \beta \Delta F_{ext,t} \Delta T, \quad (18)
\]

where $\beta$ is given by $\alpha$ scaled by a factor of $10^{-2}$, to implement a similar rate of adaptation as in (13). To avoid unnecessary changes caused by negligible force sensing difference, the positiveness of $\Delta F_{ext,t}$ is defined considering a small $\epsilon$. A pseudo-code of the proposed method is presented in Algorithm 1.
Algorithm 1 Self-tuning impedance algorithm

Input: \( \mathbf{P} \)
Output: \( K_c, D_c \)

Initialization:
\[
\begin{align*}
\Sigma_k &= k_{\text{in}} I_{3 \times 3}, \quad \Sigma_d = d_{\text{in}} I_{3 \times 3} \\
k_{st,0} &= k_{\text{in}}, \quad d_{st,0} = d_{\text{in}}
\end{align*}
\]

Control loop:
\[
I_c = I_{\text{in}} \land I_f
\]

if \( (I_c \land \text{No fault detected}) \) then
\[
\mathbf{p} = \mathbf{P} - \frac{1}{\| \mathbf{P} \|_2} \\
\Delta P = |(X_d - X_a) \cdot \mathbf{p}|
\]

if \( (\Delta P \geq \Delta P_{1}) \) then
\[
k_{st} = k_{st,1} + \alpha \Delta P \Delta T
\]

else if \( (\Delta F_{\text{ext}} \geq \varepsilon) \) then
\[
k_{st} = k_{st,1} - \beta \Delta F_{\text{ext},1} \Delta T
\]

end if
\[
\begin{align*}
\Sigma_k &= k_{st} I_{3 \times 3} \\
\Sigma_d &= 2 \varepsilon \sqrt{k_{st}} \\
U &= \text{getOrtho normalBasis}(\mathbf{P}) \\
K_c &= U \Sigma_k U^T \\
D_c &= U \Sigma_d U^T
\end{align*}
\]

else
\[
\begin{align*}
K_c &= k_{\text{in}} I_{3 \times 3} \\
D_c &= d_{\text{in}} I_{3 \times 3}
\end{align*}
\]

end if

return \( K_c, D_c \)

2.3 Trajectory planner

The framework offers different kinds of motion planning trajectories. The Trajectory planner unit receives in input from the FSM (see Sect. 2.5) the target pose, the desired period to reach it, and the type of trajectory planner that can be selected among:

- point-to-point motion between two via points, i.e. starting from the actual one and reaching the desired one that is given as input. To achieve smoother trajectory profiles and prevent impulsive jerks, this kind of motion is implemented with fifth-order polynomial. In this way, velocity/acceleration initial and final values are set to zero.
- scooping motion, which reaches the target pose through a half circular motion on the vertical axis, replicating a scooping movement. This type of motion, designed with constant angular rate of rotation and constant speed, is helpful to collect materials when a scoop-like end-effector is attached to the robot flange.
- shaking motion, that is based on a rapid sinusoidal movement performed in place. By getting as input the shaking direction, this motion is needed to completely pour the materials in the pot during the “Task” state, without leaving any residuals on the spoon.

Note that, the design and implementation of the trajectories are not explained in details since they do not contribute to the novelty of this work, as they are well-known in robotics literature. However, a short introduction is useful to understand better the different phases explained in Sect. 2.5.

2.4 Visual perception module

Visual perception plays a key role in the Finite State Machine, providing information about the materials which are going to be manipulated by the robotic arm. The module splits in two sub-systems, using as input RGB-D data from a range sensor (ASUS Xtion PRO) placed in fixed position with respect to the arm and facing the materials. In the first sub-system, different types of materials are detected in the scene, using RGB data, and their three-dimensional (3D) surface convex hull polygon is calculated, using depth data (materials localization sub-module). In the second one, a peak point per each material is localized in the base robot frame, using the depth data (peaks localization sub-module).

Materials Localization To localize different materials in 3D, we use color-based region growing segmentation. The RGB-D sensor provides colored point clouds that are first transformed in the robot’s base frame (z-axis upwards and y-axis towards left). Point cloud filtering, such as pass-through and box-cropping, is applied to keep only those points of interest that are inside the working space of the robot. Keeping the structure of the point cloud in its original grid organization (i.e. instead of removing points, setting

![Fig. 4 The visual perception module. Upper left: the RGB-D point cloud, acquired from the range sensor. Upper right: each material vertices \( V_1, V_2, V_3, V_4 \) visualized in different colors. Down: the peak point \( p_i \) per each material \( i \) (Color figure online)]
them to NaNs) makes the method more efficient. Then, region

growing is applied based on color, in order to classify similar
points in clusters. This is a two-steps algorithm. In the first
step, points are sorted based on their local curvature. Regions
that grow from points with minimum curvature (more flat
surfaces) reduce the number of segments. Two neighboring
points are consider to be part of the same material if their
color is similar to each other. The process continues for a
seed’s neighboring point, until no further neighbors can be
classified to the same segment. In the second step, clusters
with similar average colors or small size are merged. In Fig. 4,
the result of this process is visualized. To localize the convex
hull polygon of each material, we identify the extreme points
in the xy-plane (i.e. the 2.5D bounding box), which results
to vertices \( V_1, V_2, V_3, V_4 \) (Fig. 4), stored and passed to the
Finite State Machine.

**Peaks Localization** Identifying each material’s peak point
is straightforward after having localized the materials them-
selves. The peak point \( p_i \), for a material \( i \), is the point
with the maximum z-value. Similarly, each material’s center is the
average of the encapsulating polygon’s vertices.

### 2.5 Finite state machine

To increase the autonomy of the system, we designed a Finite
State Machine that is responsible of managing the transitions
between the different phases of the framework (see Fig. 2).
This unit gets as input the data sent by the Visual perception
module, and by processing them, defines the target poses that
are sent to the Trajectory planner. Moreover, this module is in
charge of determining if an interaction with the environment
is to happen, in order to activate the Self-tuning impedance
unit.

The FSM is formed by four states. The “Workspace
definition” state is responsible of acquiring the knowledge
regarding the environment where the robot will operate in
the next steps. To do so, it gets as input the vertices of the
polygons that shape the materials within the robot workspace.
These data are received from the “Materials localization” unit
and stored in an appropriate data structure. In order to iden-
tify the \( k_{st} \) parameter for every material, the FSM switches
to the “Exploration” state. At this step, the robot end-effector
grasps a stick-like tool, and reaches in a compliant way the
material placed in the leftmost part of the workspace, pre-
viously identified by the Visual Perception Module. After
having dunked into the matter, the Self-tuning impedance unit
is triggered: both the boolean value \( I_m \) and \( I_f \) have been set
to True, since a contact with the environment is expected and
the end-effector tool is inside the material. Then, the robot
follows a point-to-point motion towards the polygon cen-
ter while adapting the impedance parameters as described in
Sect. 2.2. The framework stores the resulting \( k_{st} \) in the
data structure, associating it to the corresponding material.

After that, the tool is pulled out from the substance and the
impedance parameter identification process is repeated for all
the materials in the workspace, until the rightmost material
has been analyzed. Notice that, it is not necessary to contin-
uously track the tool pose to check if it is inside the material
and therefore to activate \( I_m \). In fact, during the “Workspace
definition” state, the Visual Perception Module localizes and
defines the areas of interaction, and from here on, through
the robot forward kinematics, we can identify whether a tool
is located within that interaction expectancy area or not. This
is possible since the tool is attached to the robot end-effector.
In such a way, we can define whether \( I_m \) has to be set to True
or False at every time step.

To enhance the robustness of the system to unexpected
events, we designed a “Fault Detection” sub-unit, within the
“Exploration” state, that is triggered in case of a collision
with the boundary (possibly due to a perception unit fault). If
the sensed external forces projected along the motion vector
experience an abrupt increase, the robot ends its motion and
goes back to its homing position. The external forces trend
w.r.t the robot displacement, is estimated through a linear
regression algorithm, computed for every \( n \) samples. When
the linear regression slope \( m \) goes beyond a threshold set to
\( m_{fault} \), the fault is triggered.

Next, in the “Materials distribution” state, the vision unit
detects the highest point for every matter through the “Peaks
localization” unit. The FSM receives these points and asso-
ciates them to the relative material. During the final state,
named “Task”, the robot needs to scoop some material and
pour it in a pot held by another robot. The scooping trajec-
tories are designed to start in the Euclidean points identified as
peak points to ensure that some material is found in that part
of the container. While holding a scooping tool, as a scoop or
a small shovel, the robot starts to carry out the task by reach-
ing the first scheduled material with the default compliance
\( k_{min} \) set in all the Cartesian axes. When it dunks inside the
matter in order to scoop some of it, and therefore activating
the interaction expectancy value, the impedance parameter
are adapted in the direction of the motion setting the relative
\( k_{st} \) for every material. This value is retrieved from the data
structure, where it was stored during the learning phase in the
“Exploration” state. Like this, since the very beginning of the
task, the robot does not lag behind and can execute the task in
a more precise manner. When the scooping motion, that goes
from the highest point towards the polygon center, is over,
the material is poured in a pot and the process starts over
with the next material, as scheduled by the task sequence.
There are cases in which the substances’ viscous properties
are subject to changes over time. This can be given either
because of the material intrinsic properties or due to exter-
nal circumstances. That is why, even in the “Task” state, the
impedance parameters are exposed to changeability. Never-
theless, we decided to set a maximum value \( k_{st, max} \) that can
be reached by $k_{st}$ in this state to avoid significant variations w.r.t. the one computed in the “Exploration” state:

$$k_{st,max,m} = k_{st,exploration,m} \ast (1 + p)$$

$$0 \leq p \leq 0.5 \ \forall m,$$

(19)

where $k_{st,exploration,m}$ is the value stored in the “Exploration” state for material $m$, and $p$ is the percentage of variation that can be set. If, in the “Task” state, this value is exceeded, the “Fault Detection” sub-unit is triggered, and the robot goes back to its homing position. This method can be useful for detecting both a collision with the boundary (for instance due to a fault of the perception unit) and a substantial change in the material properties, since in industrial production these would be symptoms of system malfunction or material inconsistency.

3 Experimental setup

The framework software architecture is developed with the robotics middleware Robot Operating System (ROS) using C++ as client library. The modules illustrated in Sect. 2 are implemented as ROS nodes, and they communicate through the ROS topics depicted in Fig. 2 by means of the publisher/subscriber design pattern.

The experimental setup (Fig. 5) includes two Franka Emika Panda robotic arms. The main robot carries out all the steps of the described method, and the other one is used as a support to the first one, providing the pot where the materials are poured after the scooping task. The presented architecture relies upon a tailored version of franka_ros metapackage, the ROS integration for Franka Emika research robots. This package integrates libfranka, Franka Emika’s open source C++ interface, into ROS and ROS Control. This interface communicates with the robot through the Franka Control Interface (FCI), that provides the current robot status and enables its direct control with an external workstation PC. The communication is realized via an Ethernet cable in real-time, and the communication rate is 1 kHz.

An underactuated robotic hand, i.e., the Pisa/IIT Soft-Hand (Ajoudani et al. 2014a), is used as end-effector. We installed a pole in front of the robot arm, and we mounted on top an ASUS Xtion Pro RGB-D sensor to provide the perception data. The camera calibration is conducted w.r.t. the robot base frame.

4 Experiments

We conducted experiments within an agricultural robotics scenario, in order to validate the presented method. The setup included a container, placed between the camera pole and the robot, where three different materials were placed. To demonstrate distinct behaviors in the impedance parameters self-tuning, we considered materials with substantial difference among their viscoelastic properties, and their large-scale use in agriculture: seeds, soil and expanded clay. A video of the experiment is available in the multimedia extension.

We follow the FSM states sequence to describe the experiments. The materials’ container had a rectangular shape, and the materials inside were equally separated into three rectangular areas. Therefore, in the “Workspace definition” state, the FSM receives from the “Materials localization” perception unit the 12 Euclidean points delimiting the areas of the three materials. Accordingly, it updates the column of the data structure proposed in Table 1 related to Polygon vertices. Afterwards, the robot grasps a metal stick 27 cm long, to carry out the next phase, i.e. the “Exploration” state. Based on (16) and on the values reported in Table 2, $\alpha$ was set to 20,000. To ensure a good level of compliance in case of unpredicted collisions, the value of $k_{min}$ was set to 500 N/m. During this state, the robot gets to the leftmost material, formed by seeds (material 1), dunking the metal stick into it. An interaction with the environment is expected to happen, and therefore the $I_e$ value is activated, as shown at $t = 1.5$ s in the fourth plot of Fig. 6. Consequently, the Self-tuning impedance unit is enabled. While keeping the tool immersed in the material, the robot performs an 18 cm long movement along the x axis. Since $\Delta P$ goes beyond the threshold $\Delta P_t$ set to 1 cm, $k_{st}$ increases following (13) as shown in the first plot of Fig. 6. As a consequence, the Cartesian stiffness along the direction of interaction is adapted. In this case the movement direction is performed only on x axis. With the increase of the impedance values, we can notice that $\Delta P$, that represents the Cartesian error along the motion vector, gets reduced and goes below the threshold $\Delta P_t$. The maximum value reached by $k_{st}$ in this case is equal to 1100 N/m, and it gets associated to the relative material as reported in the right column of Table 1. To complete the “Exploration” state, the robot repeats the same described procedure for the other two materials. As expected,
the soil (material 2) turns out to be the stiffest material, with \( k_{\text{st}} \) reaching a value of 1650 N/m, and the expanded clay (material 3) is in between the other two, i.e. 1330 N/m. In the third plot of Fig. 6, we can notice how these values are tuned.

After the identification of the impedance parameters, the FSM transits to the “Material distribution” state. The starting point of the scooping trajectories are detected by the “Peaks localization” unit of the perception module, and stored in the relative column of Table 1.

Then, the robotic hand grasps a scooping tool in order to carry out the “Task” state, subdivided in four substates. The robot scoops and pours in a plant pot, provided by the second robotic arm, the three materials following this sequence: soil (a), plant seeds (b), other soil (c), and expanded clay (d). These four substates are depicted in the plots of Fig. 7 and Fig. 8. In the latter, the green triangles represent the highest point of each substance provided by the “Peaks localization” perception module, and stored in the default compliant value, i.e. \( k_{\text{st}} \) gets reduced by the sudden growth in the arrows length and color intensity. In case \( \Delta P \) keeps its value below the threshold, it means that the viscoelastic properties of the material did not change, and so there is no need for further adaptation. When the scooping is over, but still being inside the container, \( k_{\text{st}} \) gets reduced according to (18), as can be seen the last part of the scooping. Notice that, negligible variations could lead to unnecessary changes, so we designed a moving average window to calculate \( \Delta F_{\text{ext}} \). In the last part of the depicted motion, the robot exits the interaction expectancy area, and \( k_{\text{st}} \) is restored to its default compliant value, i.e. \( k_{\text{min}} \).

To show that the impedance self-tuning would occur also in case of viscoelastic properties changes, we decided to pour some water in the soil between the “Exploration” and the “Task” states. This adaptation is visible when the scooping tool enters the soil during “Task” (a), and it is caused by \( \Delta F_{\text{ext}} \) exceeding the threshold \( \Delta P_t \) as shown in the third plot of Fig. 7 at \( t = 6.3 \) s when the Self-tuning impedance unit is activated again. The value of \( k_{\text{st}} \) for material 2 gets increased from 1650 N/m to 1750 N/m. This is highlighted by the difference between the first and the other arrows inside the leftmost container in Fig. 8.

In Fig. 9 we show how the tuning of the Cartesian stiffness is achieved only in the directions of movement \( \vec{P} \), when the tool is inside the materials in two of the “Task” state

### Table 1

The data structure containing the data coming from the Self-tuning impedance unit, relative to the stiffness \( k_{\text{st}} \) (rounded to integers), and from the Visual perception module, relative to the polygon vertices of the materials and their peak points.

| Material                  | Polygon vertices (x,y,z) | Peak point (x,y,z) | \( k_{\text{st}} \)[N/m] |
|---------------------------|--------------------------|-------------------|--------------------------|
|                           | V1 (0.66,0.30,0.10)      |                   |                           |
|                           | V2 (0.36,0.30,0.11)      |                   |                           |
|                           | V3 (0.36,0.09,0.09)      |                   |                           |
|                           | V4 (0.66,0.08,0.09)      |                   |                           |
|                           |                         |                   | 1100                      |
|                           |                         |                   |                           |
|                           | V1 (0.66,0.09,0.08)      |                   |                           |
|                           | V2 (0.36,0.09,0.09)      |                   |                           |
|                           | V3 (-0.16,0.10)          |                   |                           |
|                           | V4 (-0.16,0.10)          |                   |                           |
|                           |                         |                   | 1650                      |
|                           |                         |                   |                           |
|                           | V1 (0.66,-0.14,0.10)     |                   |                           |
|                           | V2 (0.36,-0.14,0.10)     |                   |                           |
|                           | V3 (-0.36,0.13)          |                   |                           |
|                           | V4 (-0.36,0.11)          |                   |                           |
|                           |                         |                   | 1330                      |

### Table 2

To derive the value of \( \alpha \), different materials have been taken into account. The resulting average, computed through (16), has been set to 20,000.
Fig. 7 “Task” state: starting from the impedance values already tuned in the “Exploration” state, the framework allows the robot not to lag behind, so that $\Delta P$ remains below the $\Delta P^*$ threshold. Only in “Task” (a), $\Delta P$ goes beyond the threshold, since the material viscoelastic properties have been intentionally changed.

Fig. 8 A lateral view of the four “Task” substates. $k_{st}$ is projected onto the direction of motion and here represented by means of red arrows, whose intensity grows with their value (Color figure online).

Fig. 9 “Task” state: the tuning of the Cartesian stiffness is achieved only in the directions of movement $\vec{P}$. The vectorial sum of the three diagonal components is always equal to $k_{st}$. In “Task” (a), $k_{st}$ of material 2 is adapted at $t=6.5$ s, since the soil viscoelastic properties were changed as explained above. We notice that the sum of the three Cartesian stiffness diagonal components is always equal to $k_{st}$. In “Task” (d), we see the adaptation also on $K_{c}(y)$.

To show that the framework reliability has been increased by means of the two “Fault Detection” sub-units, we repeated the experiment simulating a fault in the perception unit by changing the pose of the box containing the materials both during the “Exploration” state and the “Task” state execution. Like this, following the desired trajectory, the tool grasped by the robot end-effector collides with one of the container sides.

Figure 10 shows an execution of the “Exploration” state performed to retrieve the first material $k_{st}$. As can be noticed in the third subplot, performing a linear regression (red solid curve) on the measured external forces data (blue scat-
...the "Exploration" state (performed without faults), the robot stops the current motion and goes back to its homing configuration in a compliant way.

Fig. 10 “Fault Detection” during “Exploration” state: if the external forces linear regression slope $m$ goes beyond the threshold $m_{\text{fault}}$, the robot stops the current motion and goes back to its homing configuration in a compliant way.

![Fig. 10](image)

The first column represents the data acquired while keeping always a high level of the impedance parameters, i.e. 1100 N/m (see bottom plot), that is the value reached by $k_{\text{st}}$ for material 1 during the “Exploration” state. In this case, although the tracking of the error $\Delta P$ does not exceed excessively the imposed threshold $\Delta P_1$, we measured interaction forces on z axis, represented by the blue line on the top plot, reached a quite high value ($\approx 13$ N). Therefore, this approach could lead to a system failure caused by a tool/robot damage. Notice that, with higher obstacle curvatures, the external forces measurements could scale quite rapidly easily leading to more substantial failures.

The second column shows the plots of the trial with lower impedance parameters, set at 500 N/m, as no interaction was ever expected. The robot is able to better comply with the external environment, as highlighted by lower interaction forces acting on the direction of the movement, i.e. $F_{\text{ext}}$, and the Cartesian error projected onto the direction of the movement, i.e. $\Delta P$, under the different conditions of the impedance parameters.

Fig. 11 “Fault Detection” during “Task” state: if $k_{\text{st}}$ goes beyond the $k_{\text{st, max}}$ threshold, the execution halts and the robot goes back to its initial configuration in a compliant way.

![Fig. 11](image)

The second column shows the plots of the trial with lower impedance parameters, set at 500 N/m, as no interaction was ever expected. The robot is able to better comply with the external environment, as highlighted by lower interaction forces acting on the direction of the movement, i.e. $F_{\text{ext}}$, and the Cartesian error projected onto the direction of the movement, i.e. $\Delta P$, under the different conditions of the impedance parameters.

Lastly, the third column depicts the data logged applying the method presented in this work. Stiffness and damping are updated on-line, based on the interaction expectancy and on the direction of the movement $\hat{P}$. The external interaction forces are further reduced, reaching at maximum $\approx 8$ N. $\Delta P$ is significantly less w.r.t. to the case with low impedance.

In Fig. 13, we report the setup used in this experiment enhanced with the reference trajectory (blue curve) and the measured path (red curve) logged during the last trial, when the “Self-tuning impedance unit” was enabled.
Fig. 12 The plots show three different executions of the “Task” state with an object placed inside material 1 (setup shown in Fig. 13): at the left without applying the presented method and always keeping high impedance values, in the center as the previous case but with low impedance parameters, and at the right the trial with the Self-tuning impedance unit enabled. In the latter case, there are less external interaction forces w.r.t. the first case, and the error in the direction of the movement $\Delta P$ is substantially less w.r.t. the second trial.

Fig. 13 To simulate an uncertain environment, a piece of wood has been inserted inside material 1. The curves in the picture represent the desired trajectory (blue line) and the actual trajectory (red line) (Color figure online).

5 Tank-based system passivity observer

Since the presented controller is based on continuous variations of the impedance parameters, we must demonstrate that the passivity of the system, and so its stability, is guaranteed. Following the approach presented in (Ferraguti et al. 2013), we implemented a tank-based approach to monitor the system stability. Formally, the model of the robot in the task space is given by:

$$
\Lambda(x)\ddot{X} + (\mu(x, \dot{x}) + D_d)\dot{X} + K_{\text{const}}\Delta X - \omega \dot{x}_t = F_{\text{ext}}.
$$

$$
\dot{x}_t = \sigma_{x_t} (\Delta X^T D_d \Delta X) - \omega^T \Delta X.
$$

(20)
The self-tuning impedance algorithm applied to the handle pulling phase of a standard industrial pallet jack. When an interaction is expected and $\Delta P_t$ is beyond a threshold, the impedance parameters are tuned along the motion vector $\hat{P}$ (a). On the right, the initial and final configuration assumed by the cobot are depicted, with a sketch representing the trajectory followed by the robot to pull the handle down (b).

where the desired stiffness is equal to $K_d(t) = K_{\text{const}} + K'(t)$, being $K_{\text{const}} \in \mathbb{R}^{6 \times 6}$ the constant stiffness term and $K' \in \mathbb{R}^{6 \times 6}$ the time-varying stiffness. $\Lambda(x) \in \mathbb{R}^{6 \times 6}$ and $\mu(x, \dot{x}) \in \mathbb{R}^{6 \times 6}$ are the Cartesian inertia and Coriolis/centrifugal matrices respectively. The scalar $x_t \in \mathbb{R}$ is the state associated with the tank and the tank energy $T \in \mathbb{R}^+$ and $T = \frac{1}{2} x_t^2$. $\sigma \in \mathbb{R}$ and $\omega \in \mathbb{R}^6$ respectively are

$$
\sigma = \begin{cases} 
1 & \text{if } T(x_t) \leq \bar{T}_u \\
0 & \text{otherwise}
\end{cases} \quad (21)
$$

$$
\omega = \begin{cases} 
-K'(t)x_t & \text{if } T(x_t) \geq \bar{T}_l \\
0 & \text{otherwise}
\end{cases} \quad (22)
$$

where, $\bar{T}_u \in \mathbb{R}^+$ is a suitable, application dependent, upper bound that avoid excessive energy storing in the tank while $\bar{T}_l \in \mathbb{R}^+$ is a lower bound below which energy cannot be extracted by the tank for avoiding singularities in (20) and thus the time-varying stiffness $K'(t)$ will be removed. For a detailed analysis of the system passivity, please refer to (Ferraguti et al. 2013).

Figure 14 shows the stability analysis performed during the entire duration of the experiment, i.e. including the “Exploration” and the “Task” states, when no faults occurred.

As we can see from the bottom subplot, the tank energy was above the lower bound ($\bar{T}_l = 0.5 J$) during all the phases, which means that the full stiffness including the constant (set to the compliant value, 500 N) and varying parts can be realised without loss of stability.

### 6 Algorithm scalability

The self-tuning impedance algorithm has extensively demonstrated its validity in the previous sections. However, its applicability is not limited to the experimental setup described so far. In fact, by changing the FSM states illustrated in Sect. 2.5, various applications can gain advantages thanks to the proposed methodology.

An example is given by pick and place tasks, where the robot keeps a compliant profile on all the Cartesian axes before picking and after placing, while adjust the impedance parameters while carrying the task. Preliminary results, although tuning the parameters only along Cartesian axes, are presented in (Balatti et al. 2018) for a debris removal task.

Another application field is represented by the logistics sector, where cobots are expected to automatize repetitive and physically demanding works, as transporting airport...
mobile stairways, heavy industrial carts, and warehouses pallet jacks. To this end, hereafter we show a demonstration of the self-tuning impedance algorithm applied to the pulling task of a standard industrial pallet jack. We define a new FSM, that includes a “Handle pull down” state where $I_m$ gets triggered. On the other hand, vision comes into play triggering $I_v$ whenever the end-effector is within the area around the pallet jack handle (similarly as in (Balatti et al. 2018)), which is recognized through a standard template matching perception algorithm. The “Handle pull down” state is performed with a circular trajectory as depicted in the lower part of Fig. 15b. In this sketch the red arrows represent the online stiffness parameters adaption as in Fig. 8. In the plots of Fig. 15a, it is represented the tuning of $k_{st}$ along the motion vector, triggered only when $I_v$ is set to True and $Δ P$ is beyond the threshold $Δ P_r$, set to 3 cm. The top right part of Fig. 15 represents the initial and final configuration of the MOCA robot (Wu et al. 2019) used in this experiment. A video showing the pallet jack handle pulling results is part of the multimedia extension.

7 Conclusion and discussion

This paper presented a novel framework to enhance robot adaptability in unknown and unstructured environments. The system relied on a self-tuning impedance controller, able to regulate the impedance parameters only on the direction of the motion vector, and activated just when an interaction with the external environment was predicted. It additionally included a visual perception module that improved the situation-awareness of the robot, localizing the surrounding materials and their peak points. Notice that the material detection process (i.e. color-based region growing segmentation) could be replaced with other RGB-based methods, such as deep learning, to improve the time complexity. Given that the visual results were accurate and fast enough for our application, we leave this as future work. To detect faults, we also presented a novel unit capable of recovering from unexpected collisions with the boundary. A Finite State Machine was introduced to manage the transitions between the system states. We experimentally validated the presented framework in an agricultural task. Although we are working towards making the method autonomous, it still needs an offline tuning of some parameters, such as $α$ and $β$, whose online adaptation will be the focus of our future work. Nevertheless, using non-optimal $α$ and $β$ values does not imply failure in task execution, but rather introduces some variations to the convergence speed of the self-tuning control gains. An analysis of the stability of the system is reported, since the algorithm includes time-varying tuning of the impedance parameters.

Acknowledgements Open access funding provided by Istituto Italiano di Tecnologia within the CRUI-CARE Agreement. This work was supported by the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 871237 (SOPHIA). The authors would like to thank Yuqiang Wu for his contributions to the stability analysis.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Ajoudani, A., Godfrey, S. B., Bianchi, M., Catalano, M. G., Grioli, G., Tsagarakis, N., et al. (2014a). Exploring teleimpedance and tactile feedback for intuitive control of the pisa/fit softhand. IEEE Transactions on Haptics, 7(2), 203–215.

Ajoudani, A., Lee, J., Rocchi, A., Ferrati, M., Hoffman, E. M., Settimi, A., Caldwell, D. G., Bicchi, A., & Tsagarakis, N. G. (2014b). A manipulation framework for compliant humanoid coman: Application to a valve turning task. In 14th IEEE-RAS international conference on Humanoid Robots (Humanoids), pp 664–670.

Ajoudani, A., Tsagarakis, N. G., & Bicchi, A. (2017). Choosing poses for force and stiffness control. IEEE Transactions on Robotics, 33(6), 1483–1490.

Albu-Schaffer, A., Ott, C., Frese, U., & Hirzinger, G. (2003). Cartesian impedance control of redundant robots: recent results with the dlr-light-weight-arms. IEEE international conference on robotics and automation (ICRA), 3704–3709.

Anderson, R. J., & Spong, M. W. (1988). Hybrid impedance control of robotic manipulators. IEEE Journal on Robotics and Automation, 4(5), 549–556.

Balatti, P., Kanoulas, D., Rigano, G. F., Muratore, L., Tsagarakis, N. G., & Ajoudani, A. (2018). A self-tuning impedance controller for autonomous robotic manipulation. In 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 5885–5891.

Balatti, P., Kanoulas, D., Tsagarakis, N. G., & Ajoudani, A. (2019). Towards robot interaction autonomy: Explore, identify, and interact. IEEE international conference on robotics and automation (ICRA), 9523–9529.

Dietrich, A., Busmann, K., Petit, F., Kotyczka, P., Ott, C., Lohmann, B., et al. (2016). Whole-body impedance control of wheeled mobile manipulators. Autonomous Robots, 40(3), 505–517.

Ferraguti, F., Secchi, C., & Fantuzzi, C. (2013). A tank-based approach to impedance control with variable stiffness. IEEE international conference on robotics and automation (ICRA), 4948–4953.

Gribovskaya, E., Kheddar, A., & Billard, A. (2011). Motion learning and adaptive impedance for robot control during physical interaction with humans. IEEE international conference on robotics and automation (ICRA), 4326–4332.

Harbers, M., Peeters, M. M., & Neerincx, M. A. (2017). Perceived autonomy of robots: effects of appearance and context. In M. I. Aldinhas Ferreira, J. Silva Sequeira, M. O. Tokhi, E. K. Endre, &
G.S. Virk (Eds.), *A world with robots: International conference on robot ethics: ICORE 2015* (pp. 19–33). Cham: Springer International Publishing.

He, W., & Dong, Y. (2018). Adaptive fuzzy neural network control for a constrained robot using impedance learning. *IEEE Transactions on Neural Networks and Learning Systems, 29*(4), 1174–1186.

Katz, D., Pyuro, Y., & Brock, O. (2008). Learning to manipulate articulated objects in unstructured environments using a grounded relational representation. In O. Brock, J. Trinkle, & F. Ramos (Eds.), *Robotics: Science and systems IV*. Zurich, Switzerland: The MIT Press.

Katz, D., Venkatraman, A., Kazemi, M., Bagnell, J. A., & Stentz, A. (2014). Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. *Autonomous Robots, 37*(4), 369–382.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology, 9*(6), 718–727.

Kotseruba, I., Gonzalez, O.J.A., & Tsotsos, J.K. (2016). A review of 40 years of cognitive architecture research: Focus on perception, attention, learning and applications, pp 1–74. arXiv:1610.08602

Kronander, K., & Billard, A. (2014). Learning compliant manipulation through kinesthetic and tactile human-robot interaction. *IEEE Transactions on Haptics, 7*(3), 367–380.

Lee, J., Ajoudani, A., Hoffman, E.M., Rocchi, A., Settimi, A., Ferrati, M., Bicchi, A., Tsagarakis, N.G., & Caldwell, D.G. (2014). Upper-body impedance control with variable stiffness for a door opening task. In *2014 4th IEEE-RAAS international conference on Humanoid Robots (Humanoids)*, pp 713–719.

Li, Y., Sam Ge, S., & Yang, C. (2012). Learning impedance control for physical robot-environment interaction. *International Journal of Control, 85*(2), 182–193.

Nemec, B., Likar, N., Gams, A., & Ude, A. (2018). Human robot cooperation with compliance adaptation along the motion trajectory. *Autonomous Robots, 42*(5), 1023–1035.

Righetti, L., Kalakrishnan, M., Pastor, P., Binney, J., Kelly, J., Voorhies, R. C., et al. (2014). An autonomous manipulation system based on force control and optimization. *Autonomous Robots, 36* (1–2), 11–30.

Roveda, L. (2018). Adaptive interaction controller for compliant robot base applications. *IEEE Access, 7*, 6553–6561.

Roveda, L., Iannacci, N., & Tosatti, L. M. (2018). Discrete-time formulation for optimal impact control in interaction tasks. *Journal of Intelligent & Robotic Systems, 90*(3–4), 407–417.

Schindlbeck, C., & Haddadin, S. (2015). Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks. *IEEE international conference on robotics and automation (ICRA)*, 440–447.

Song, P., Yu, Y., & Zhang, X. (2019). A tutorial survey and comparison of impedance control on robotic manipulation. *Robotica, 37*(5), 801–836.

Tsagarakis, N. G., Caldwell, D. G., Negrello, F., Choi, W., Baccelliere, L., Loc, V., et al. (2016). Walk-man: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics, 34*(7), 1225–1259.

Wu, Y., Balatti, P., Lorenzini, M., Zhao, F., Kim, W., & Ajoudani, A. (2019). A teleoperation interface for loco-manipulation control of mobile collaborative robotic assistant. *IEEE Robotics and Automation Letters, 4*(4), 3593–3600.

Xu, G., Song, A., & Li, H. (2011). Adaptive impedance control for upper-limb rehabilitation robot using evolutionary dynamic recurrent fuzzy neural network. *Journal of Intelligent & Robotic Systems, 62*(3–4), 501–525.

Yang, C., Ganesh, G., Haddadin, S., Parusel, S., Albou-Schaeffer, A., & Burdet, E. (2011). Human-like adaptation of force and impedance in stable and unstable interactions. *IEEE Transactions on Robotics, 27*(5), 918–930.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Pietro Balatti is a PhD candidate at the Human-Robot Interfaces and physical Interaction Lab, Istituto Italiano di Tecnologia, and at the Department of Information Engineering, University of Pisa, Italy. He received the BD and the MD, cum laude, in computer science and engineering from Politecnico di Milano, Italy, in 2014 and 2016, respectively. He was member of team “CoAware” that won the KUKA Innovation Award in April 2018, who was responsible of the robot control framework. His main research interests are focused in the enhancement of robot interaction autonomy, within the fields of adaptive impedance control and physical human-robot interaction.

Dimitrios Kanoulas is currently a Lecturer (Assistant Professor) in Robotics and Computation at the University College London (UCL), Department of Computer Science, working on robotic perception and learning for multi-DoF robotic platforms, such as humanoids, quadrupeds, and mobile manipulators. Between 2014–2019 he was a senior post-doctoral researcher at the Humanoids and Human-Centered Mechatronics Lab at the Istituto Italiano di Tecnologia (IIT), working with Dr. Nikos Tsagarakis and Prof. Darwin Caldwell on the WALK-MAN, CENTAuro, and CogIMon European projects. He was the perception team leader during the DARPA Robotics Challenge 2015. In August 2014, he completed his PhD at the Geometric and Physical Computing Lab in the College of Computer and Information Science at Northeastern University, advised by Prof. Marsette Vona.

Nikos Tsagarakis received his D.Eng. degree in Electrical and Computer Science Engineering in 1995 from the Polytechnic School of Aristotle University, Greece, his MSc degree in Control Engineering in 1997 and his PhD in Robotics from the University of Salford, UK, in 2000. He is Tenured Senior Scientist at IIT with overall responsibility for Humanoid design & Human-Centered Mechatronics development. Since 2013, he is also serving as a Visiting Professor at the Centre for Robotics Research (CORE), Department of Informatics, King’s College University, London, UK.
Arash Ajoudani is a tenured senior scientist at the Italian Institute of Technology (IIT), where he leads the Human-Robot Interfaces and physical Interaction (HRI2) laboratory. He received his PhD degree in Robotics and Automation from University of Pisa and IIT in July 2014. His PhD thesis was a finalist for the Georges Giralt PhD award 2015 - best European PhD thesis in robotics. He is a recipient of the European Research Council (ERC) starting grant 2019. He was a winner of the Amazon Research Awards 2019, the winner of the Solution Award 2019 (Premio Innovazione Robotica at MECSPE2019), the winner of the KUKA Innovation Award 2018, the winner of the Werob best poster award 2018, a finalist for the best conference paper award at Humanoids 2018, a finalist for the best interactive paper award at Humanoids 2016, a finalist for the best oral presentation award at Automatica (SIDRA) 2014, the winner of the best student paper award and a finalist for the best conference paper award at ROBIO 2013, and a finalist for the best manipulation paper award at ICRA 2012. He is the coordinator of the Horizon-2020 project SOPHIA with a consortium of 12 partners from the leading European research and industrial organisations. He has contributed to several successful European projects (H2020 and FP7) such as WALKMAN, WearHap, SOMA, and SoftPro. He is the author of the book “Transferring Human Impedance Regulation Skills to Robots” in the Springer Tracts in Advanced Robotics (STAR), and several publications in journals, international conferences, and book chapters. He is currently serving as the executive manager of the IEEE-RAS Young Reviewers’ Program (YRP), and as chair and representative of the IEEE-RAS Young Professionals Committee. He has been serving as a member of scientific advisory committee and as an associate editor for several international journals and conferences such as IEEE RAL, Biorob, ICORR, etc. His main research interests are in physical human-robot interaction and cooperation, robotic manipulation, robust and adaptive control, assistive robotics, and tele-robotics.