A Self-tuning Impedance Controller for Autonomous Robotic Manipulation

Pietro Balatti[1,3], Dimitrios Kanoulas[2], Giuseppe F. Rigano[2], Luca Muratore[2,4], Nikos G. Tsagarakis[2], and Arash Ajoudani[1]

***Abstract*— Complex interactions with unstructured environments require the application of appropriate restoring forces in response to the imposed displacements. Impedance control techniques provide effective solutions to achieve this, however, their quasi-static performance is highly dependent on the choice of parameters, i.e. stiffness and damping. In most cases, such parameters are previously selected by robot programmers to achieve a desired response, which limits the adaptation capability of robots to varying task conditions. To improve the generality of interaction planning through task-dependent regulation of the parameters, this paper introduces a novel self-regulating impedance controller. The regulation of the parameters is achieved based on the robot's local sensory data, and on an interaction expectancy value. This value combines the interaction values from the robot state machine and visual feedback, to authorize the autonomous tuning of the impedance parameters in selective Cartesian axes. The effectiveness of the proposed method is validated experimentally in a debris removal task.**

I. INTRODUCTION

During the past decades, both natural catastrophes caused by the Earth's natural processes (e.g., earthquakes, tsunami and their consequences) and man-made disasters (e.g. Chernobyl nuclear accident) have highlighted the need for effective and efficient robotic systems that can be quickly deployable after the disaster, to assist in tasks too hazardous for humans to perform.

To prevent and respond to such disasters, some common manipulation tasks have been studied during the past years. The DARPA Robotics Challenge (DRC) [1] sought to address this problem by investigating tasks such as removing debris blocking an entryway, using a cutting tool, and turning a valve near a leaking pipe. The presence of high uncertainty levels in such environments and tasks urges the development of autonomous robotic behaviors that are triggered by the rich local sensory systems. Such systems must be capable of distinguishing expected interactions from disturbances, to be able to react accordingly and appropriately.

A set of autonomous skills can be developed based on the observations from human demonstrations [2]–[5], however, their performance is highly dependent on the richness of the training data sets. In addition, current wearable sensory systems still do not provide effective solutions for the measurement of contact forces while performing complex manipulation tasks, that is why most learning by demonstration techniques function on a kinematic level.

The analytical solutions to this problem have focused on the use of impedance control [6]–[8], force control [9], or hybrid interaction controllers [10], [11] to address the uncertainty levels. Nevertheless, in most cases, the choices of control parameters are carried out by the robot programmers, based on the knowledge or experience from executing similar tasks. Such presumptions will limit the applicability of interaction controllers with fixed parameters to varying tasks or task conditions (e.g. different valve friction, debris weight, etc.), reducing the flexibility and adaptation capacity of robots to unexpected events [12], [13]. Several adaptive learning techniques have been proposed to address this issue (e.g. see [14], [15]), however, the underlying policies for the modulation of contact forces were limited to a certain class of tasks, and responded similarly to all interactions coming from the external world (e.g., external disturbances, human applied forces, etc.).

To address these challenges, we propose a novel manipulation framework that enables autonomous regulation of the contact forces in selective Cartesian axes through self-tuning of the impedance parameters in quasi-static conditions. The tuning of the parameters in those axes are performed based on the local sensory feedback, that includes interaction forces, robot desired and measured trajectories, and vision.

With the aim to improve the situation-awareness of the robot to distinguish expected interactions from external disturbances, an *interaction expectancy* value is generated. The computation of this value is based on two Boolean values. The first is generated from an autonomous Finite State Machine (FSM) that reveals if an external interaction can occur in certain phases of robot movements or manipulation. The second value is generated by the vision system, by fitting an interaction field to the object or environment to distinguish the areas in the external world that expect interactions with robot end-effectors. A logical AND operator of the two values creates the interaction expectancy value. It is important to note here that, the second value created by the interaction field will generate True values when an object is grasped and being moved in the space by the robot hand (since the interaction field will move with the object). However, if no other external interaction is expected, the FSM will return False values. Like this, in case of accidental collisions, the self-tuning impedance will not react in those Cartesian directions, and remain to be compliant.

We perform proof-of-concept experiments for a debris

[1]HRII Lab and [2]HHCM Lab, Department of Advanced Robotics, Istituto Italiano di Tecnologia, Genoa, Italy. pietro.balatti@iit.it

[3]Dept. of Information Engineering, University of Pisa, Italy.

[4]School of Electrical and Electronic Engineering, The University of Manchester, M13 9PL, UK.

removal task using a KUKA LBR IIWA 14 robot to demonstrate the potential of the proposed methodology. Starting from a compliant end-effector, the robot will adjust the Cartesian stiffness and damping in directions which are necessary to perform the task, e.g. in the direction of gravity due to the deviations from the desired trajectory caused by the payload. Other unexpected disturbances will not lead to the tuning of the impedance parameters, and will be responded by a soft contact relying on the robot Cartesian and nullspace compliant behavior.

II. METHOD

The proposed framework can be subdivided into four main units: (1) a self-tuning Cartesian impedance controller that enables the tuning of the stiffness and damping parameters in certain Cartesian axes, (2) an object localization method that provides the exact pose from where the object needs to be grasped, (3) an interaction expectancy module that provides the possibility of an expected physical interaction between the robot and the object/environment, and (4) a sensor fusion of force and visual localization from step (2) to effectively execute the manipulation task. The aforementioned parts are regulated by means of a Finite State Machine (FSM) in charge of assembling all the components of the system.

A. Self-tuning Cartesian impedance controller

The ability to autonomously tune impedance parameters of the robot interaction controller can help to address varying uncertainty levels of a manipulation task, and contribute to improved performances. Preliminary work in this direction investigated the adaptation of impedance parameters based on human observations [12] and energy tank-based optimization [13], to improve the physical interaction performance based on the varying task conditions.

However, one of the biggest challenges in this context is to make such techniques work properly in unstructured environments, by distinguishing expected interactions from external disturbances. This consideration will enable adaptation of the impedance parameters only to the interactions that are expected, while achieving a compliant behavior in response to the external disturbances to avoid the generation of unnecessary high interaction forces (e.g. collisions with the environment).

To this end, in this paper, a novel self-tuning impedance controller is developed that functions based on an interaction expectancy value. This value is calculated from the FSM and visual feedback interaction values (see sections below), to enable autonomous tuning of impedance parameters when an external interaction is expected. Furthermore, the tuning of the parameters are achieved in selective axes of the Cartesian space based on end-point sensory data, to avoid unnecessary stiffening/complying of the remaining axes. The block scheme of the proposed self-tuning impedance controller is provided in Fig. 1.

The robot is torque controlled and the vector of robot joint torques $\boldsymbol{\tau} \in \mathbb{R}^n$ is generated as follows:

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

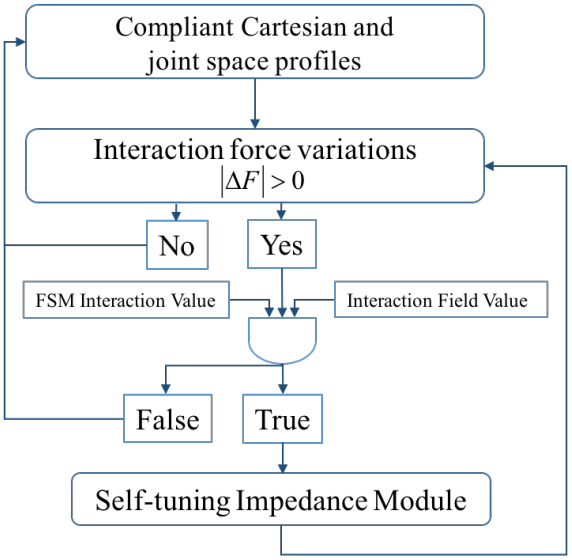

Fig. 1: Block scheme of the proposed self-tuning impedance controller.

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

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 robot arm 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, $\boldsymbol{g} \in \mathbb{R}^n$ is the gravity vector and $\boldsymbol{\tau}_{ext}$ is the external torque vector. $\boldsymbol{F}_c$ represents the forces vector in the Cartesian space and $\boldsymbol{\tau_{st}}$ the second task torques projected onto the null-space of $\boldsymbol{J}$.

Forces $\boldsymbol{F}_c \in \mathbb{R}^6$ are calculated as follows:

$$\boldsymbol{F}_c = \boldsymbol{K}_c(\boldsymbol{X}_d - \boldsymbol{X}_a) + \boldsymbol{D}_c(\dot{\boldsymbol{X}}_d - \dot{\boldsymbol{X}}_a) \tag{3}$$

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 $\boldsymbol{X}_a \in \mathbb{R}^6$ the Cartesian desired and actual position, $\dot{\boldsymbol{X}}_d$ and $\dot{\boldsymbol{X}}_d \in \mathbb{R}^6$ their corresponding velocity profiles. The Cartesian desired position and velocity are generated with a fifth-order polynomial trajectory.

The first rule of the proposed robot self-tuning impedance controller is to achieve compliant Cartesian and nullspace behaviors. To this end, the default $\boldsymbol{K}_c$ and $\boldsymbol{K}_n$ matrices are chosen to achieve a good tracking performance in the presence of joint friction, while providing a compliant response in case of accidental collisions. $\boldsymbol{K}_n$ is a constant nullspace stiffness matrix used in the secondary task of the self-tuning Cartesian impedance controller. A stiffness-consistent nullspace projection [16] was used to calculate $\boldsymbol{\tau}_{st}$[1].

When an expected interaction with the environment is detected by the interaction expectancy module (see Sec. II-C), the Cartesian stiffness matrix $\boldsymbol{K}_c$ and consequently

[1]Note that, in this work, self-tuning of the impedance parameters are only carried out in selective Cartesian axes, and the nullspace stiffness is kept as default. This is due to the assumption that most manipulation tasks are executed at robot endpoint.

the damping matrix $\boldsymbol{D}_c$ are subject to changes as follows, depending on $\boldsymbol{\Delta F}_{ext,t}$, i.e. the variation of the external forces detected by the robot at time t w.r.t. the ones measured in the previous control loop

$$\boldsymbol{\Delta F}_{ext,t} = \boldsymbol{F}_{ext,t} - \boldsymbol{F}_{ext,t-1}. \tag{4}$$

First case: if $\boldsymbol{\Delta F}_{ext,t} < 0$, the stiffness needs to be increased on a certain Cartesian axis to ensure the movement is completed in that particular direction in a precise manner. For instance, when a heavy object is grasped, the loading effect will introduce deviations from the desired trajectory. In this scenario the stiffness is regulated following equations:

$$\boldsymbol{\Delta X}_t = |\boldsymbol{X_d} - \boldsymbol{X_a}| \tag{5}$$

$$\boldsymbol{K}_{c,t} = \boldsymbol{K}_{c,t-1} + \boldsymbol{\alpha \Delta X}_t \boldsymbol{\Delta T} \tag{6}$$

$$\text{subject to} \quad -\boldsymbol{f}_{max} < \boldsymbol{F}_{ext,t} < \boldsymbol{f}_{max}$$

$$\text{subject to} \quad \boldsymbol{K}_{c,min} < \boldsymbol{K}_{c,t} < \boldsymbol{K}_{c,max}$$

where $\boldsymbol{\alpha}$ is a coefficient to be set, $\boldsymbol{\Delta T}$ the control loop sample time, $\boldsymbol{f}_{max}$ the maximum allowed interaction force level, $\boldsymbol{K}_{c,min}$ the minimum Cartesian stiffness value and $\boldsymbol{K}_{c,max}$ the maximum Cartesian stiffness value.

Second case: if $\boldsymbol{\Delta F}_{ext,t} > 0$, the stiffness needs to be decreased on a certain Cartesian axis, switching back to compliant mode, e.g., when an object is laid down on the ground before being released, the robot needs to adapt to the ground forces to avoid the generation of high interaction forces between the object and the environment. The stiffness parameters are decreased according to the following law:

$$\boldsymbol{K}_{c,t} = \boldsymbol{K}_{c,t-1} - \boldsymbol{\beta \Delta F}_{ext,t} \boldsymbol{\Delta T} \tag{7}$$

$$\text{subject to} \quad \boldsymbol{\Delta X}_t < \boldsymbol{\delta x}_{max}$$

$$\text{subject to} \quad \boldsymbol{K}_{c,min} < \boldsymbol{K}_{c,t} < \boldsymbol{K}_{c,max}$$

where β is a coefficient to be set, $\boldsymbol{\Delta F}_{ext,t}$ is defined as in (4) and $\boldsymbol{\delta x}_{max}$ represents the maximum allowed displacement error from the desired trajectory, i.e. $\boldsymbol{\Delta X}_t$.

The damping matrix $\boldsymbol{D}$ is tuned online based on the resulting $\boldsymbol{K}_{c,t}$

$$\boldsymbol{D} = \boldsymbol{\Lambda}_* \boldsymbol{D}_{diag} \boldsymbol{K}_{adj*} + \boldsymbol{K}_{adj*} \boldsymbol{D}_{diag} \boldsymbol{\Lambda}_*, \tag{8}$$

where $\boldsymbol{D}_{diag}$ is the diagonal matrix containing the damping factors ($\zeta = 0.7$), $\boldsymbol{K}_{adj*} \boldsymbol{K}_{adj*} = \boldsymbol{K}_c$ and $\boldsymbol{\Lambda}_* \boldsymbol{\Lambda}_* = \boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ is the desired end-effector mass matrix [17].

B. Object Localization and Grasping Pose Extraction

The second module involves the visual-based method of localizing the object to be manipulated in the environment and extracting its grasping pose frame, noted as ($\hat{\mathbf{x}}_g$,$\hat{\mathbf{y}}_g$,$\hat{\mathbf{z}}_g$). In this paper, we will focus on objects that are stick-like and lie in front of a planar segment, such as a door or wall. As an input to this method, we will consider point cloud data coming from an RGB-D sensor at $30Hz$. Given the input point cloud, we first segment the dominant plane using the RANdom SAmple Consensus (RANSAC) method [18].

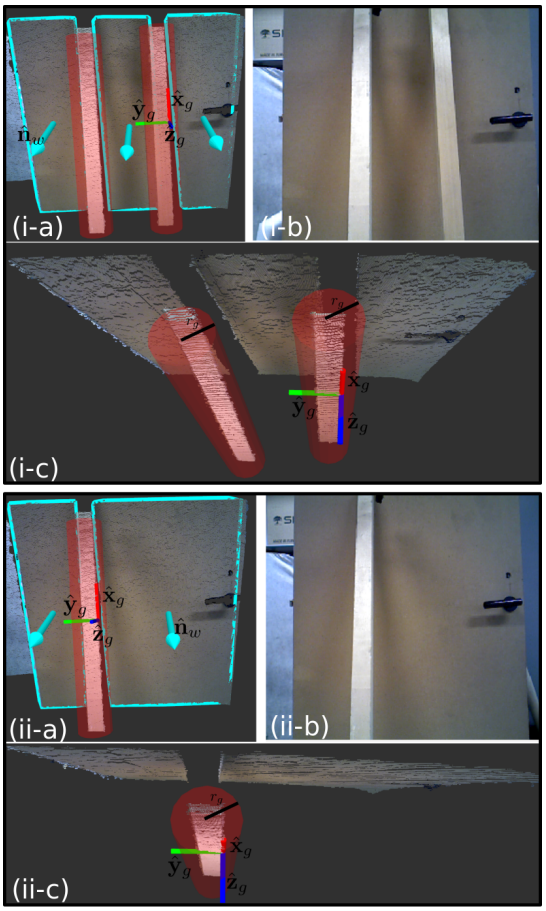

Fig. 2: (i-a) The point cloud of two debris on a door, with the door normal vector $\hat{\mathbf{n}}_w$ (cyan), the grasp axes of the right-most debris ($\hat{\mathbf{x}}_g$, $\hat{\mathbf{y}}_g$, $\hat{\mathbf{z}}_g$), and the interaction field cylinder around each debris in red. (i-b) The RGB image of the scene. (i-c) A top-view of the point cloud with the grasp axes and the interaction field cylinder with radius r_g. Similarly, for one debris in (ii).

In this way, we extract the normal of the plane $\hat{\mathbf{n}}_w$ (in cyan color in Fig. 2) and we segment all the points above the plane to different Euclidean clusters [19], [20]. Each point cloud cluster represents a potentially different object in the scene. For each one of the clusters, we fit a 3D line using again RANSAC. Those that respect some length constraints, i.e., longer but narrower than the hand's palm size, are considered to be stick-like objects, potentially good for grasping. In this way, we extract the fitted 3D line's axis as the grasping frame's $\hat{\mathbf{x}}_g$-axis (in red color in Fig. 2). We construct the remaining axes of the grasping frame in the following way. The $\hat{\mathbf{y}}_g$-axis is the cross product between the plane normal vector $\hat{\mathbf{n}}_w$ and the $\hat{\mathbf{x}}_g$-axis, i.e., $\hat{\mathbf{y}}_g = \hat{\mathbf{n}}_w \times \hat{\mathbf{x}}_g$. Finally, the $\hat{\mathbf{z}}_g$-axis is the cross product of the other two, i.e., $\hat{\mathbf{z}}_g = \hat{\mathbf{x}}_g \times \hat{\mathbf{y}}_g$. In this way, we generate the grasping frame to be matched to the hand frame during grasping, in a way that the closing fingers can encapsulate the object and the hand is perpendicular to the flat surface (wall/door), so that we avoid as much as possible early collisions with the object during the approaching phase. The position of the grasping frame has a fixed height with respect to the robot base frame and is at the edge of the object. We pick the right edges for right-

handed grasps and left edges, otherwise. To do that, for each object point cloud cluster, we extract the point cloud nearest neighborhood at the fixed height from the robot's base and we select as grasping position the right/left-most point.

Another important information that can be extracted from vision is a cylinder around the object main axis, that represents the interaction field between the hand and the object. This information serves as input to the interaction expectancy module (Sec. II-C). The cylinder is extracted along the fitted 3D line axis for each object, with radius r_g that equals the maximum size of the hand fingers (appears as red cylinder in Fig. 2) and its length is coming from the extreme 3D points of the object point cloud clusters that were extracted after the plane segmentation phase.

C. Interaction Expectancy Module

The third unit includes a method responsible of deciding if a physical interaction between the robot and the object/environment is to happen or not. There are some areas in which this interaction is expected and so the self-tuning Cartesian impedance module has to be activated and some other areas in which no interaction is predicted and the robot should keep the same or default level of compliance. Two Boolean values are used to decide if an external interaction is expected and if the interaction expectancy value has to be triggered. The first one, the FSM interaction value, is given by the FSM and is activated only in those states in which an interaction is expected, e.g., while picking an object and placing it down on the ground. The second one, the interaction field value, is given by the vision module and it is True if the end-effector is located inside the interaction field, or False, otherwise. These two values are correlated by the following Boolean logic rule:

$$\boldsymbol{I_e} = \boldsymbol{I_m} \ \wedge \ \boldsymbol{I_f} \tag{9}$$

where $\boldsymbol{I_e}$ is the interaction expectancy value, $\boldsymbol{I_m}$ is the FSM interaction value, $\boldsymbol{I_f}$ is the interaction fields value and $\wedge$ is the logical AND operator.

D. Sensor Fusion of Force and Visual Localization

In disaster scenarios, since the environment is uncertain, the manipulation workspace may not be always bright and clear, for instance because of illuminations, shadows, or even fire smoke. Therefore, a unit to assist the vision module and to enhance the robustness of the object pose estimation has been developed. After reaching the pose estimated by the perception module, the external forces are sensed and translated from the joint to the Cartesian space as in (10) where $\boldsymbol{J^{+T}}$ represents the transpose of Jacobian pseudoinverse matrix and $\boldsymbol{\tau_{q,ext}}$ the joint external torques vector. If $\boldsymbol{\Delta F}_{ext} \neq 0$ on axis i, a contact with the object has been established on that axis and there is no need of a further movement in that direction. On the contrary, if $\boldsymbol{\Delta F}_{ext} = 0$ on axis i, no contact is detected and a movement forward on that axis is performed until $\boldsymbol{\Delta F}_{ext} \neq 0$, ensuring that the end-effector is ready to grasp the object.

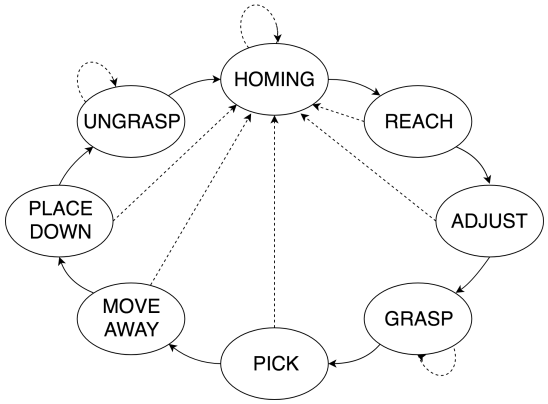

Fig. 3: The system is controlled by a Finite State Machine. Each state represents a motion primitive.

$$\boldsymbol{F_{c,ext}} = \boldsymbol{J^{+T}} \cdot \boldsymbol{\tau_{q,ext}} \tag{10}$$

Performing end-effector movements against the object to sense the mentioned forces could lead to an accidental fall of the object, if it is placed in a precarious position, but since these adjustments are made with a very high compliance, the end-effector will not displace the object from its original pose. Note that the fusion of data coming from visual localization and external forces sensing is done sequentially.

E. Finite State Machine

The described units are linked together by means of a Finite State Machine, depicted in Fig. 3, that allows the framework to be completely autonomous. Each state represents a motion primitive while the outgoing arrows represent the possible feedback that can be given in response to the performed subtask outcome. Solid arrows stand for successful actions while dashed arrows are directed to safe states in case a fail in the task occurs.

The FSM initial state is identified by the "Homing" state, that is also the safe exit state for the primitives involving arm movements. In this recovery state, the robot end-effector is in the center of the workspace. The natural control flow leads firstly to a "Reach" state, where the end-effector approaches the object depending on the position and the orientation provided by the object localization module. During this phase, the robot enters in the interaction field and the self-tuning impedance controller is enabled.

To double-check if the pose given by the vision object localization unit is accurate, the FSM switches to the "Adjust" state where the sensor fusion unit can move the end-effector forward on the axes where no forces are detected. Once the end-effector palm makes contact with the object surface, the "Grasp" primitive closes the robot hand and allows the object grasping. Consequently, the object has to be taken away in a safe space and released. To accomplish this goal, four other primitives were designed. In the first place, the object is brought in front of the robot ("Pick" state) and then moved away at its side ("Move Away" state). Before being released in the "Ungrasp" state, we want to make sure that the object falls down in the space where it does not hinder future robot operations. Therefore, in the "Place Down" state, the object

upper-end is first rotated in a way that it points away from the robot and the intended future operational space, and then placed down until it makes contact with the ground. In this state, the self-tuning impedance controller is enabled again, so that when the object makes contact with the ground, the robot becomes compliant since $\boldsymbol{\Delta F}_{ext,t}$ will be positive.

III. ROBOT INTERFACES

The software architecture of the robot relies upon XBotCore [21] (*Cross-Bot-Core*) – a recently developed open-source[2] and light-weight Real-Time (RT) platform for robotics, designed to be both an RT robot control framework and a software middleware. XBotCore satisfies RT requirements, ensuring 1 kHz hard RT control loop even in complex multi-DOF systems, moreover it provides a simple and easy-to-use middleware Application Programming Interface (API), for both RT and non-RT external control frameworks. This API is completely flexible with respect to the framework a user wants to utilize. It is also possible to reuse the code written using XBotCore API with different robots (cross-robot feature) thanks to the Robot Hardware Abstraction Layer (R-HAL) introduced in [22].

The R-HAL permits to seamlessly program and control any robotic platform providing a relatively uniform abstraction that hides the specifics of the underlying hardware. Consequently, this allows the robot software developers to easily transfer/reuse their code on different robots. All the R-HAL implementations are built as a shared library loaded at runtime according to what specified in a configuration file. In particular, the R-HAL implementation for the KUKA IIWA has been provided relying on the FRI (Fast Robot Interface) to transfer data between the robot controller and the external computer. It is important to mention that the control was done in non-RT mode because the FRI API communication provided by KUKA is intrinsically not RT safe. However, a porting of the KUKA API to the RT domain could be possible with no particular issues ensuring always no changes in the XBotCore plugin developed. The control behavior is implemented using two different XBot plugin. One of them is used to realize the Cartesian impedance controller. The second plugin implements the FSM behavior exchanging data (pose to reach, external force detected) with the other plugin.

[2]https://github.com/ADVRHumanoids/XBotCore

IV. EXPERIMENTS

We conducted experiments using a KUKA LBR IIWA 14 robotic arm equipped with an underactuated hand, i.e., the Pisa/IIT SoftHand [23] in a debris removal setup. An ASUS Xtion Pro RGB-D sensor was attached in the front part of the robot arm base and calibrated with respect to the end-effector. In the default mode, the diagonal components of the stiffness matrix ($\boldsymbol{K}_c$) were set to $100N/m$ on the translational components, and to $20Nm/rad$ on the rotational components. The choice of update rate parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in (6) and (7) was done experimentally with the aim of minimizing the Cartesian error within the trajectory period of the current state. Different debris lengths and weights were used in our experiments.

Fig. 4 illustrates the different phases of the task execution with reference to the FSM states. The leftmost figure at the top (4.1), shows the robot in the "Homing" state, i.e, its initial configuration. The robot end-effector then reaches the

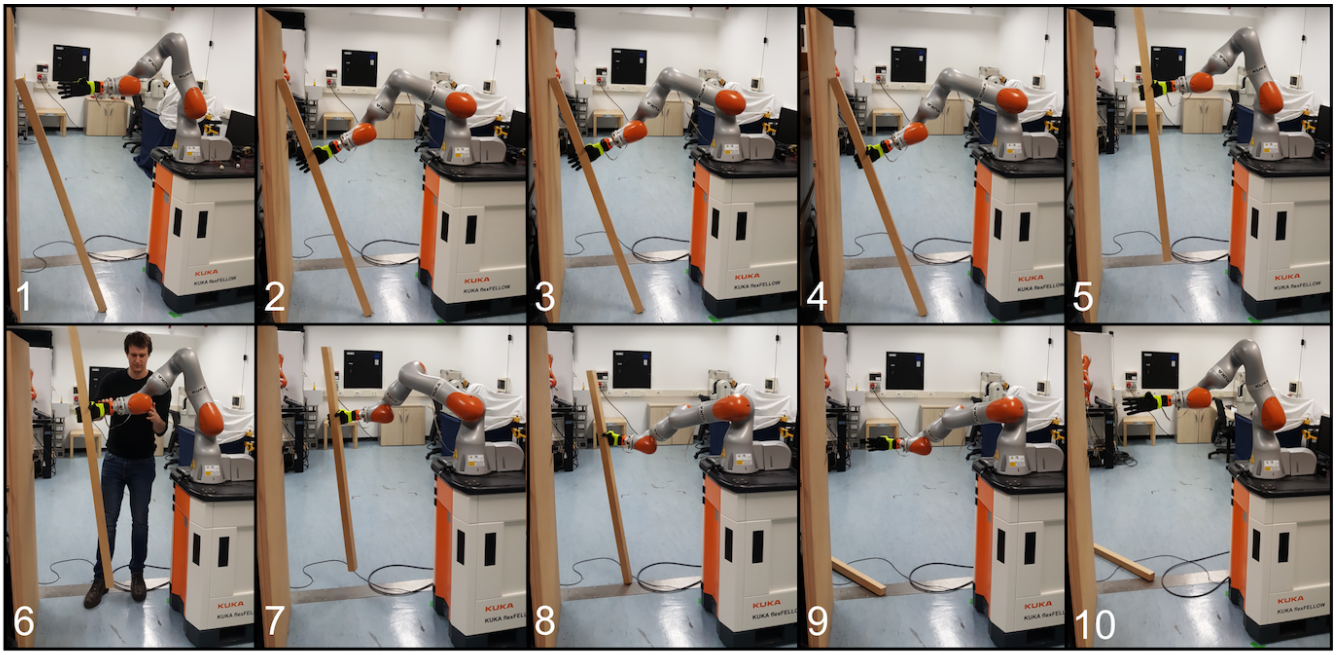

Fig. 4: Debris removal sequences performed with KUKA LBR IIWA 14 robotic arm: Homing (1), Reach (2), Adjust (3), Grasp (4), Pick (5), Unexpected external disturbances (6), Move Away (7), Place Down (8), Ungrasp (9), Homing (10). Please refer to the text for a detailed explanation of the listed motion primitives.

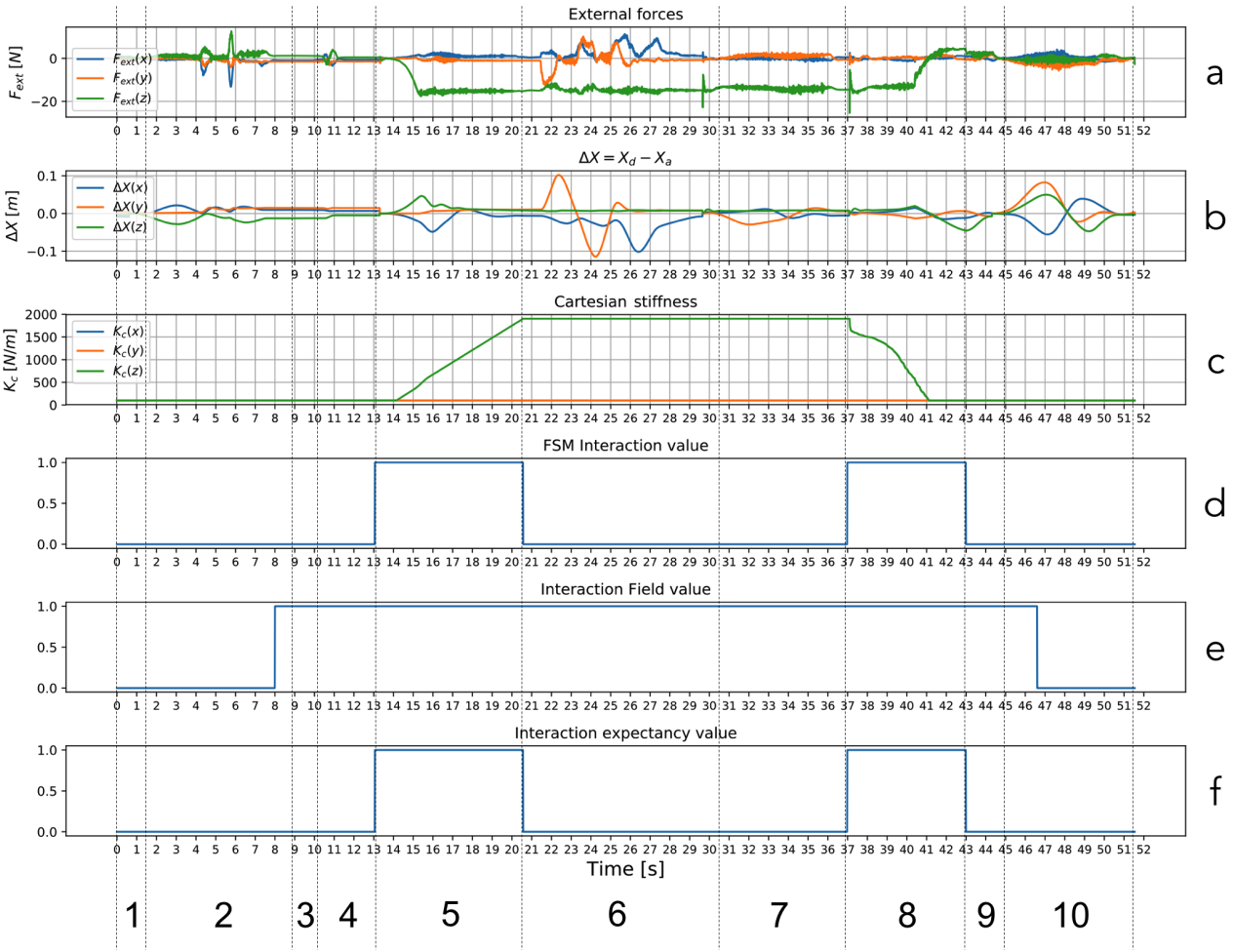

Fig. 5: Experiment 1 - External forces in the Cartesian space (a), the Cartesian displacement between desired and actual position (b), the curve trend of the Cartesian stiffness (c), FSM interaction value (d), interaction field value (e) and interaction expectancy value (f). The dashed vertical lines highlight when the different FSM states are performed referring to Fig. 4.

debris grasp pose received by the object localization module (Fig. 4.2, "Reach" state), entering the interaction field that surrounds the debris area and therefore activating the interaction field value as shown in Fig. 5.e. The interaction expectancy value (Fig. 5.f) is not yet activated since the other component of the logical AND, i.e., the FSM interaction value (Fig. 5.d), is still False and will be active in future states ("Pick" and "Place Down"). The sensor fusion of force and visual localization unit then checks if the grasp is ready to be executed (Fig. 4.3, "Adjust" state). Once the debris is in contact with the end-effector, the grasping action is performed (Fig. 4.4, "Grasp" state) and the object is picked (Fig. 4.5, "Pick" state). Since the latter state enables a positive FSM interaction value, the Cartesian stiffness (Fig. 5.c) is now subject to changes: $\boldsymbol{\Delta F}_{ext,t} < 0$ on z-axis, so $\boldsymbol{K_c}(z)$ is increased following (6) until the end of the current state when the FSM interaction value is deactivated.

At the end of the "Pick" state the FSM interaction value becomes False and so does the interaction expectancy value. To show that the Cartesian stiffness would not change even if unexpected external interactions occur, the framework leaves the robotic manipulator in the current configuration for 10 seconds allowing a person to interact with it. In Fig. 4.6 a subject perturbs the robot in all the Cartesian axes as the external forces plot (Fig. 5.a) shows from time $20.5s$ to time $30.5s$ and the Cartesian stiffness is not affected by such changes, regardless of the force and position displacements.

After these manual perturbations, the FSM restarts its natural control flow to move away the debris (Fig. 4.7, "Move Away" state) and place it down (Fig. 4.8, "Place Down" state). In this state, the FSM interaction value becomes again True and, since the object is still in the interaction field, the interaction expectancy value is activated. The Cartesian stiffness (Fig. 5.c) is again subject to changes, this time $\boldsymbol{\Delta F}_{ext,t} > 0$ on z-axis, so $\boldsymbol{K_c}(z)$ is decreased following (7) until a lower bound set to $100N/m$ (as the initial value) is reached. This behavior results in the object being gently placed on the ground. At the end of the "Place Down" state the FSM interaction value is set back to False and the "Ungrasp" state opens the end-effector and releases the object (4.9). As soon as the debris exits the interaction field, the relative interaction field value is set to False. Fig. 4.10 shows the final setup with the object laying on the ground and the robot back to the "Homing" state ready to start the

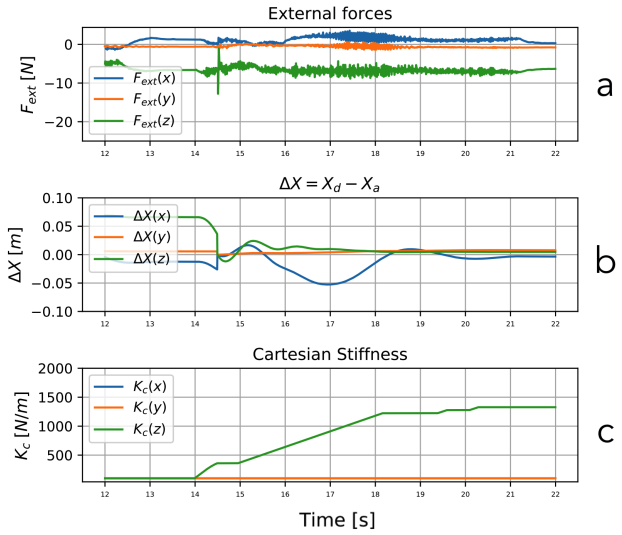

Fig. 6: Experiment 2 - External forces in the Cartesian space (a), Cartesian displacement between desired and actual position (b), and curve trend of the Cartesian stiffness (c) during state "Pick".

task over.

A similar experiment was conducted with another object of different weight ($0.7Kg$) and length, to demonstrate the system flexibility to varying task conditions. The results of the first phase, where the interaction field was active, are shown in Fig. 6 ("Pick" state). $\boldsymbol{K}_c(z)$ reaches a maximum value of $\approx 1330 N/m$ in comparison with the $\approx 1900 N/m$ of the first experiment where the weight of the object was $2Kg$. Due to the similarity of the interaction expectancy profiles, the plots are not repeated here.

V. DISCUSSION AND CONCLUSION

In this work, a novel self-tuning impedance controller was proposed to improve robot autonomous adaptability to varying task conditions. The selective adaptation of the Cartesian impedance parameters was carried out based on local force and vision sensory data. To improve the situation-awareness of the robot in responding to expected interactions and complying with the external disturbances, the interaction expectancy module combined two interaction values from vision and FSM to make robot controller respond appropriately and accordingly. The choice of update rate parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ were carried out experimentally in this work, that must be improved to enhance robot autonomy. In this direction, the parameters will be tuned based on the task and on the object material, retrieved by the perception module. The effectiveness of the proposed control framework in adapting to varying task conditions, in the presence of external disturbances, was experimentally carried out in a debris removal task.

REFERENCES

[1] E. Ackerman and E. Guizzo, "Darpa robotics challenge finals: Rules and course," *IEEE,[Online]. Available: http://spectrum. ieee. org/automaton/robotics/humanoids/drcfinals-course*, 2015.

[2] C. C. Kemp and A. Edsinger, "Robot manipulation of human tools: Autonomous detection and control of task relevant features," in *Proc. of the Fifth Intl. Conference on Development and Learning*, 2006.

[3] D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, and A. Stentz, "Perceiving, learning, and exploiting object affordances for autonomous pile manipulation," *Aut. Robots*, vol. 37, no. 4, pp. 369–382, 2014.

[4] D. Katz, Y. Pyuro, and O. Brock, "Learning to manipulate articulated objects in unstructured environments using a grounded relational representation," *Robotics: Science and Systems IV*, p. 254, 2009.

[5] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human-robot interaction," *IEEE transactions on haptics*, vol. 7, no. 3, pp. 367–380, 2014.

[6] A. Dietrich, K. Bussmann, F. Petit, P. Kotyczka, C. Ott, B. Lohmann, and A. Albu-Schäffer, "Whole-body impedance control of wheeled mobile manipulators," *Aut. Robots*, vol. 40, no. 3, pp. 505–517, 2016.

[7] J. Lee, A. Ajoudani, E. M. Hoffman, A. Rocchi, A. Settimi, M. Ferrati, A. Bicchi, N. G. Tsagarakis, and D. G. Caldwell, "Upper-body impedance control with variable stiffness for a door opening task," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 713–719.

[8] A. Ajoudani, J. Lee, A. Rocchi, M. Ferrati, E. M. Hoffman, A. Settimi, D. G. Caldwell, A. Bicchi, and N. G. Tsagarakis, "A manipulation framework for compliant humanoid coman: Application to a valve turning task," in *14th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2014, pp. 664–670.

[9] L. Righetti, M. Kalakrishnan, P. Pastor, J. Binney, J. Kelly, R. C. Voorhies, G. S. Sukhatme, and S. Schaal, "An autonomous manipulation system based on force control and optimization," *Aut. Robots*, vol. 36, no. 1-2, pp. 11–30, 2014.

[10] R. J. Anderson and M. W. Spong, "Hybrid impedance control of robotic manipulators," *IEEE Journal on Robotics and Automation*, vol. 4, no. 5, pp. 549–556, 1988.

[11] C. Schindlbeck and S. Haddadin, "Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 440–447.

[12] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 918–930, 2011.

[13] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013, pp. 4948–4953.

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

[15] W. He and Y. Dong, "Adaptive fuzzy neural network control for a constrained robot using impedance learning," *IEEE Trans. on neural networks and learning systems*, vol. 29, no. 4, pp. 1174–1186, 2018.

[16] A. Dietrich, C. Ott, and A. Albu-Schäffer, "An overview of null space projections for redundant, torque-controlled robots," *The International Journal of Robotics Research*, vol. 34, no. 11, pp. 1385–1400, 2015.

[17] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2003, pp. 3704–3709.

[18] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[19] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, TUM, Germany, October 2009.

[20] D. Kanoulas, J. Lee, D. G. Caldwell, and N. G. Tsagarakis, "Center-of-Mass-Based Grasp Pose Adaptation Using 3D Range and Force/Torque Sensing," *Intl. Journal of Humanoid Robotics*, p. 1850013, 2018.

[21] L. Muratore, A. Laurenzi, E. M. Hoffman, A. Rocchi, D. G. Caldwell, and N. G. Tsagarakis, "XBotCore: A Real-Time Cross-Robot Software Platform," in *IEEE Int. Conf. on Robotic Computing*, 2017.

[22] G. Rigano, L. Muratore, A. Laurenzi, E. Hoffman, and N. Tsagarakis, "Towards a robot hardware abstraction layer (r-hal) leveraging the xbot software framework," *IEEE Int. Conf. on Robotic Computing*, 2018.

[23] A. Ajoudani, S. B. Godfrey, M. Bianchi, M. G. Catalano, G. Grioli, N. Tsagarakis, and A. Bicchi, "Exploring teleimpedance and tactile feedback for intuitive control of the pisa/iit softhand," *IEEE transactions on haptics*, vol. 7, no. 2, pp. 203–215, 2014.