2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

A Proxy-Tactile Reactive Control for Robots Moving in Clutter

Giammarco Caroleo[1#*], Francesco Giovinazzo[2#], Alessandro Albini[1], Francesco Grella[2], Giorgio Cannata[2] and Perla Maiolino [1,3]

***Abstract*— Robots performing tasks in challenging environments must be supported by control or planning algorithms that exploit sensor feedback to effectively plan the robot's actions. In this paper, we propose a reactive control law that simultaneously utilizes proximity and tactile feedback to perform a pick-and-place task in an unknown and cluttered environment. Specifically, the presented solution leverages proximity sensing obtained from distributed Time of Flight (ToF) sensors to avoid collision when this does not interfere with the pick-and-place task. Safety is guaranteed by a higher-priority task using tactile feedback that reduces contact forces when a collision occurs. Additionally, we compare the effectiveness of this control scheme with a collision detection and reaction scheme based solely on tactile sensing. Our results demonstrate that the proposed approach reduces the collisions with the environment and the task execution time of the pick-and-place operation.**

I. Introduction

Autonomous tasks performed in cluttered environments require robots equipped with sensing capabilities allowing for detecting the presence of obstacles and guaranteeing safe operations [1]–[7]. In this respect, two types of feedback are generally considered in the literature. The majority of the approaches exploit cameras or proximity sensors capable of measuring the distance between the robot and obstacles in the environment [1]–[3], [6], [8], [9]. Generally, these works assume that no collisions between the robot and the environment occur since they lack the ability to sense contact forces and consequently regulate them. However, this approach is too conservative - real-world applications related to pipe inspection, reaching and grasping in clutter or human-robot interaction, where physical contact is likely occurring, cannot be based solely on contact-free assumptions. This aspect is addressed by approaches leveraging force or tactile feedback, where the robot's task is executed by taking into account the control of contact forces arising from unexpected collisions with the environment [4], [7], [10]–[14]. As a drawback, the lack of information about the surrounding environment leads to robot motions that may provoke unnecessary collisions with the environment.

Combining proximity and tactile feedback would allow for compensating for the drawbacks of both, thus improving the robot's behavior when moving in challenging environments [15]–[17]. In this paper, we aim to investigate how proximity and tactile feedback can be combined to safely perform a pick-and-place task. The environment depicted in Figure 1 is cluttered, and the position of the obstacles is unknown to the robot. The robot must react to the presence of obstacles using distributed Time of Flight (ToF) and tactile sensors, partially covering its body. The general idea is that contact with the environment should be avoided whenever possible - i.e., when obstacles are detected by ToFs and when not affecting the correct execution of the trajectory. In the event of a collision, tactile feedback can then be leveraged to safely manage contact forces without interrupting the task.

Existing literature has combined proximity and tactile feedback mainly by implementing switching mechanisms based on an if-else logic that can trigger pre-programmed behaviours for the robot [8], [15], [16], [18]. For example, in [8], [15], [18], the robot slows down when getting closer to obstacles or humans. A repulsive control action is then

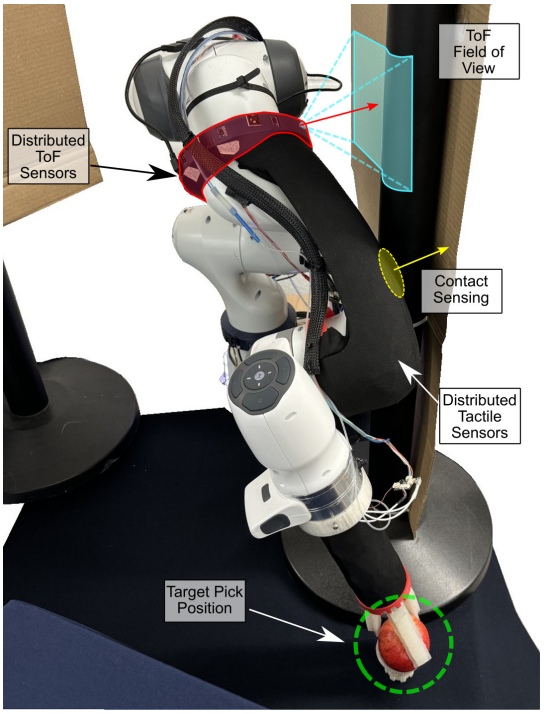

Fig. 1. The robot performs a pick-and-place task in an unstructured and unknown environment, utilizing distributed tactile and ToF sensors. The proposed controller leverages both distance and contact sensing to ensure safe task execution. The yellow arrow represents a force vector detected by the tactile sensing system during collisions with the environment. The light blue square denotes the FoV of a single ToF sensor, while the red arrow indicates the distance vector between the arm and the obstacle.

[1] are with the Oxford Robotics Institute, University of Oxford, UK.
[2] are with the DIBRIS, University of Genoa, IT.
[3] is with the DIMEC, University of Genoa, IT.
[#]Equally contributed to the paper.
[*]Corresponding author e-mail: giammarco@oxfordrobotics.institute
This work was supported by the SESTOSENSO (HORIZON EUROPE Research and Innovation Actions under GA number 101070310).

commanded when a collision is detected. However, this type of approach is not suitable in our scenario since the task execution is stopped as soon as a contact is detected. The work in [16] implements a similar strategy - there is a clear distinction between the pre-touch and touch phases, and a different control action is triggered depending on the context. Another possibility is to combine proxy-tactile information in the form of virtual force fields [17]. This allows for defining a joint representation of the two sensing modalities and can be used to avoid obstacles and to move away from contacts. This solution does not fit our scenario in which tactile and distance sensing affects different control tasks.

The approach proposed in this paper leverages the task priority framework [19] and consists of defining three tasks with different priorities. The lowest priority is given to a controller performing obstacle avoidance using proximity information. The pick-and-place is assigned with a mid-priority and consists of a path following performed with the end-effector. The highest priority is instead given to the task using tactile feedback to minimize contact forces applied to the environment. In this way, the expected behaviour is as follows: the robot performs the pick-and-place, while at the same time, it tries to avoid obstacles using the kinematic redundancy. When a collision occurs, the contact force minimization task takes over. Compared to the use of tactile sensing alone [4], [7], [13], we expected to reduce the collisions with the environment and, consequently, the time to execute the task. In this respect, we compared the proposed approach with the tactile-based collision detection and reaction scheme implemented in our previous work [7].

The remainder of the paper is structured as follows. Section II provides an overview of the type of feedback provided by the tactile and ToF sensors. The controller is described in Section III. Section IV introduces the experimental setup and describes the experiments we performed to validate the approach. Results and discussions are presented in Section V. Conclusions follow.

II. Overview on the Sensing System

The robot is assumed to be partially covered by (i) distributed tactile sensors, measuring the normal component of forces applied to the robot arm, and (ii) distributed ToF sensors, providing a small resolution depth map of the robot surroundings.

A. Distributed Tactile Sensing System

The tactile sensing system is composed of independent elements known as *taxels*, each providing a measurement related to the intensity of a lumped normal force acting upon it. Additionally, taxels are assumed to be spatially calibrated, allowing to associate a force measurement with the corresponding position on the robot's body [20], [21]. As depicted in Figure 2, when a set of adjacent taxels is involved in contact, we compute the normal force applied on the area as the resultant of the measurements provided by the taxels. In the Figure, $\mathbf{x}_{\mathbf{c}j} \in \Re^3$, $\mathbf{f}_j \in \Re^3$ and $\mathbf{n}_{\mathbf{c}j} \in \Re^3$ represent respectively the contact centroid, the resultant force

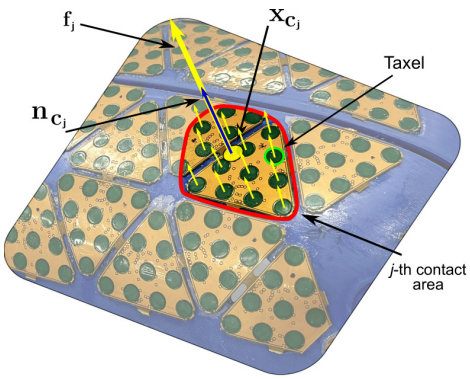

Fig. 2. It is shown how the output of the tactile sensing system is processed. Each green element is a taxel (details on the specific transduction technology are provided in Section IV). The red contour highlights the contact area. Each taxel provides a measurement related to the normal force applied, represented by small yellow arrows. The yellow circle and arrow represent the contact centroid and the resultant force, respectively, while the blue arrow indicates its direction.

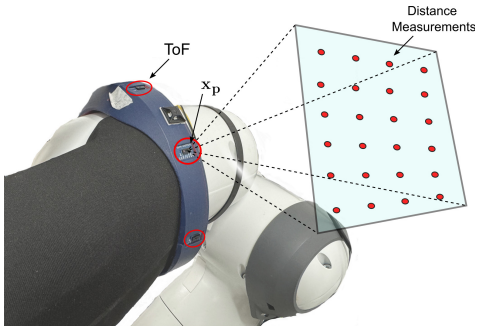

Fig. 3. The robot arm embeds ToF sensors, each of which provides a low-resolution depth map containing distance measurements captured within a square FoV, represented by the red dots.

applied on the area and the normal at the surface for the j-th contact involving a sensorized part of the robot[1]. It is worth noting that the computation of $\mathbf{f}_j$ is not specific for the adopted sensing technology (described in Section IV) - the same computation can be performed with any type of tactile sensing system based on taxels, such as [22]–[25].

B. Distributed ToF System

The sensing system providing proximity feedback comprises a set of independent ToF sensors. As shown in Figure 3, each ToF has a squared FoV and provides a low-resolution depth-map, where each element represents the distance between the location of the sensor $\mathbf{x}_\mathbf{p} \in \Re^3$ and obstacles or parts of the environment falling into the FoV. Assuming the intrinsic parameters of the ToF to be known, it is possible to convert the distance measurements into points in space referred to the robot base frame. This generates a point cloud-like representation of the environment surrounding the robot.

The measurement of the ToF sensors are then filtered by removing the points located farther than threshold distance $\bar{d}$

[1]All vectors and positions defined in the paper are assumed to be expressed with respect to the base of the robot. Furthermore, we do not explicitly specify the time dependency to maintain lightweight notation.

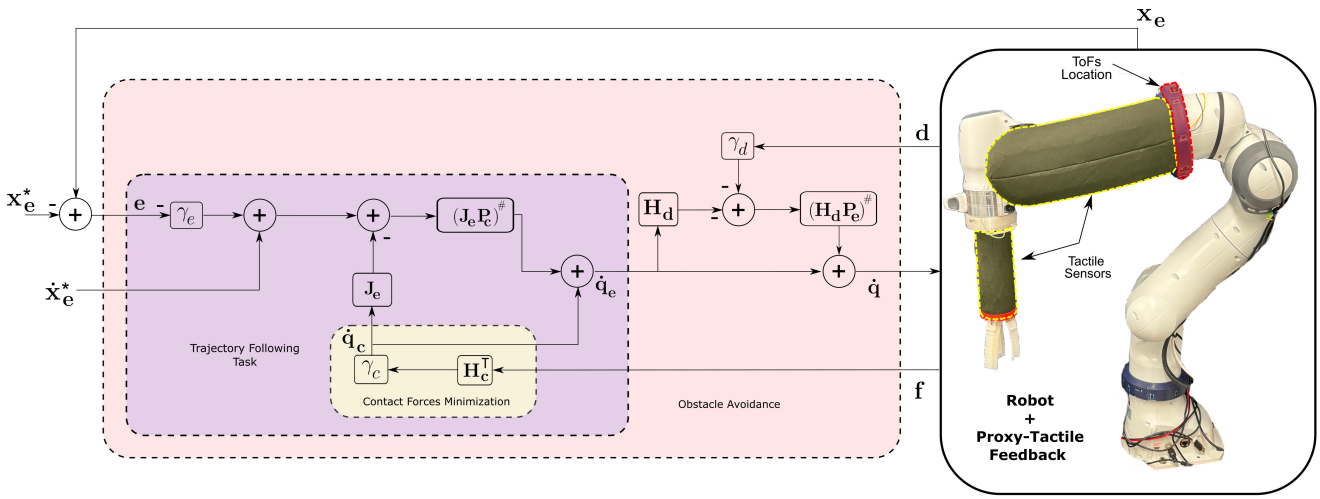

Fig. 4. Overview of the control scheme derived as explained in Section III. The control action $\dot{\mathbf{q}}$ is designed to exploit both distance and tactile feedback to perform the trajectory task. It is designed based on the control objectives defined in Section III and to deal with the different priorities. The yellow, blue and red blocks highlight the part of the scheme contributing to the different tasks. Furthermore, the picture of the robot shows the area sensorized with tactile and ToF sensors highlighted in yellow and red respectively.

from the origin of the sensor $\mathbf{x_p}$. This generates a sparse set of points approximating the local geometry of the objects in the proximity of the robot. The points are successively clustered [26] to find the centroids $\mathbf{x_{o}}_i$ with $i = \{1,2,\ldots,N\}$ and N the number of clusters. This allows for computing a distance vector $\mathbf{d}_i \in \Re^3$ as

$$\mathbf{d}_i = \mathbf{x_{o}}_i - \bar{\mathbf{x}}_{\mathbf{p_i}}, \tag{1}$$

where $\bar{\mathbf{x}}_{\mathbf{p}i}$ is the position of the ToF closest to $\mathbf{x_{o}}_i$. The distance vector is represented in Figure 1 as a red arrow.

III. Proxy-Tactile Controller

As previously explained, the controller has to satisfy three control objectives having different priorities. The highest-priority controller deals with the management of the contact forces. Similarly to [7], the objective is to minimize the squared sum of contact forces acting on the robot arm when in contact with the environment. In this respect, the control objective can be written as the following Lyapunov function:

$$V_c = \sum_{j=1}^{M} f_j^2, \tag{2}$$

with $f_j = \mathbf{n_c}_j^\mathsf{T}\mathbf{f}_j \in \Re$ and M the number of contacts occurring on the robot.

The mid-priority task is related to the control of the position of the end-effector $\mathbf{x}_e \in \Re^3$ to follow the reference trajectory $\mathbf{x}_e^* \in \Re^3$. This can be written as a control objective that minimizes the error $\mathbf{e} = \mathbf{x}_e - \mathbf{x}_e^*$ [27]:

$$V_e = \frac{1}{2}\mathbf{e}^\mathsf{T}\mathbf{e}. \tag{3}$$

Finally, the lowest priority is given to the obstacle avoidance task based on the ToF feedback. The goal is to increase the distance from the object as much as possible. Therefore, the objective is to maximize the following function:

$$V_d = \sum_{i=1}^{N} d_i^2, \tag{4}$$

where $d_i = \mathbf{n_d}_i^\mathsf{T}\mathbf{d_i} \in \Re$ and $\mathbf{n_d}$ an unitary vector having the same direction of $\mathbf{d_i}$.

The full control scheme computing the joint level control action $\dot{\mathbf{q}}$ is reported in Figure 4. The three terms $\dot{\mathbf{q}}_\mathbf{c}$, $\dot{\mathbf{q}}_\mathbf{e}$ and $\dot{\mathbf{q}}$ in the Figure, are related to the solution of the three Lyapunov functions defined above. The remainder of the Section describes how the control action that respects the different priorities is computed starting from the control objectives defined in Equations (2) to (4).

A. Contact Task

To achieve the minimization of Equation (2) the control action must impose the following condition at each time instant:

$$\dot{V}_c = \sum_{j=1}^{M} f_j \dot{f}_j < 0. \tag{5}$$

If this condition is satisfied, V_c exhibits a decreasing trend that eventually leads to its minimization over time. The solution to Equation (5) is the same reported in our previous work [7], and highlighted with the yellow block in Figure 4, consisting of the following joint-level control action:

$$\dot{\mathbf{q}}_\mathbf{c} = \gamma_c \mathbf{H_c}^\mathsf{T}\mathbf{f}, \tag{6}$$

where $\gamma_c > 0$ is a gain factor and $\mathbf{H_c}$ and $\mathbf{f}$ are defined as follow:

$$\mathbf{f} = \begin{bmatrix} f_1 & \ldots & f_M \end{bmatrix}^\mathsf{T} \in \Re^M,$$

$$\mathbf{H_c} = \begin{bmatrix} \mathbf{n}_1^\mathsf{T}\mathbf{J_{c}}_1 \\ \vdots \\ \mathbf{n}_M^\mathsf{T}\mathbf{J_{c}}_M \end{bmatrix} \in \Re^{M\times 6}, \tag{7}$$

with $\mathbf{J_{c}}_j$ being the Jacobian at the j-th contact point.

B. Trajectory Following Task

The minimization of Equation (3) can be achieved by applying the following velocity at the end-effector [27]:

$$\dot{\mathbf{x}}_e = \mathbf{J_e}\dot{\mathbf{q}}_e = -\gamma_e \mathbf{e} + \dot{\mathbf{x}}_e^*, \tag{8}$$

where $\mathbf{J_e}$ is the Jacobian of the robot end-effector. The joint velocities $\dot{\mathbf{q}}_e$ can be computed as described in [19] to take into account the higher priority assigned to the contact task:

$$\dot{\mathbf{q}}_\mathbf{e} = \dot{\mathbf{q}}_\mathbf{c} + \left[\mathbf{I} - \mathbf{H_c}^{\#}\mathbf{H_c}\right]\dot{\mathbf{z}}_\mathbf{e} = \dot{\mathbf{q}}_\mathbf{c} + \mathbf{P_H}\dot{\mathbf{z}}_\mathbf{e}, \tag{9}$$

where $(\cdot)^{\#}$ is the pseudo-inverse operation and $\mathbf{I}$ is a 7×7 identity matrix. Since $\mathbf{P_H}$ is an orthogonal projector, the minimization of Equation (2) is not affected by $\dot{\mathbf{z}}_\mathbf{e}$ which can be arbitrarily chosen. By combining Equations (8) and (9) we get:

$$\dot{\mathbf{z}}_\mathbf{e} = (\mathbf{J_e P_H})^{\#}(-\gamma_e \mathbf{e} + \dot{\mathbf{x}}_\mathbf{e}^* - \mathbf{J_e}\dot{\mathbf{q}}_\mathbf{c}), \tag{10}$$

then, since $\mathbf{P_H}(\mathbf{J_e P_H})^{\#} = (\mathbf{J_e P_H})^{\#}$ [19], we get the following expression for $\dot{\mathbf{q}}_\mathbf{e}$:

$$\dot{\mathbf{q}}_\mathbf{e} = \dot{\mathbf{q}}_\mathbf{c} + (\mathbf{J_e P_H})^{\#}(-\gamma_e \mathbf{e} + \dot{\mathbf{x}}_\mathbf{e}^* - \mathbf{J_e}\dot{\mathbf{q}}_\mathbf{c}). \tag{11}$$

C. Obstacle Avoidance Task

Similarly to Section III-A, we can derive the control action related to the obstacle-avoidance task by considering the time derivative of V_d:

$$\dot{V}_d = \sum_{i=1}^{N} d_i \dot{d}_i > 0, \tag{12}$$

where in this case the control action needs to impose $\dot{V}_d$ strictly positive to maximise the distance. The time derivative of d_i corresponds to:

$$\dot{d}_i = \mathbf{n_d}_i^\intercal \dot{\mathbf{d}}_i + \underbrace{\dot{\mathbf{n}}_{\mathbf{d}_i}^\intercal \mathbf{d}_i}_{=\mathbf{0}} = \mathbf{n_d}_i^\intercal \dot{\mathbf{d}}_i = -\mathbf{n_d}_i^\intercal \dot{\mathbf{x}}_{\mathbf{p}i}. \tag{13}$$

In the equation above, $\dot{\mathbf{n}}_{\mathbf{d}_i}^\intercal \mathbf{d}_i = \mathbf{0}$ since the two vectors are orthogonal. Furthermore, obstacles in the environment are fixed, therefore, $\dot{\mathbf{d}}_\mathbf{i} = -\dot{\mathbf{x}}_{\mathbf{p}i}$ since $\dot{\mathbf{x}}_{\mathbf{o}i} = \mathbf{0}$. This allows to rewrite Equation (12) as:

$$\dot{V}_d = -\sum_{i=1}^{N} d_i \mathbf{n_d}_i^\intercal \dot{\mathbf{x}}_{\mathbf{p}i} = -\sum_{i=1}^{N} d_i \mathbf{n_d}_i^\intercal \mathbf{J_p}_i \dot{\mathbf{q}} > 0, \tag{14}$$

where $\mathbf{J_p}_i$ is the Jacobian related to the position of the ToF $\mathbf{x_p}_i$. The previous Equation can be written in a more compact form by defining the following quantities:

$$\mathbf{d} = \begin{bmatrix} d_1 & \dots & d_N \end{bmatrix}^\intercal \in \Re^N,$$

$$\mathbf{H_d} = \begin{bmatrix} \mathbf{n_d}_1^\intercal \mathbf{J_p}_1 \\ \vdots \\ \mathbf{n_d}_N^\intercal \mathbf{J_p}_N \end{bmatrix} \in \Re^{N \times 6}, \tag{15}$$

thus leading to Equation (12) rewritten as:

$$\dot{V}_d = -\mathbf{d}\mathbf{H_d}\dot{\mathbf{q}} > 0. \tag{16}$$

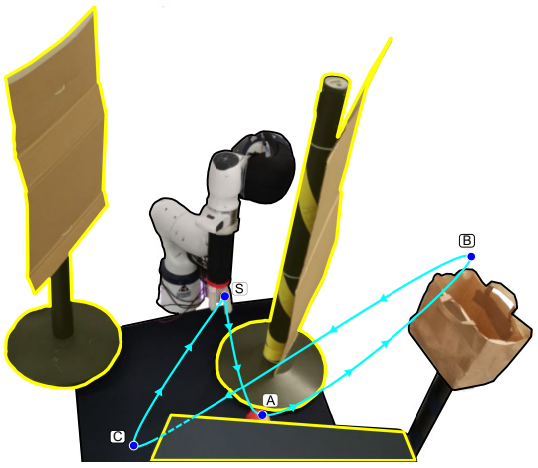

Fig. 5. The environment used to validate the proposed approach. The obstacles are highlighted in yellow. It is also shown one of the three starting configurations of the robot, the waypoints and the trajectory. The pick and place positions are labelled as **A** and **B** respectively.

The inequality in Equation (16) can be satisfied by applying the following control action at each time instant:

$$\dot{\mathbf{q}} = -\gamma_d \mathbf{H_d}^{\#}\mathbf{d}, \tag{17}$$

where $\gamma_d > 0$ is a gain factor. However, since this task must be executed with lower priority than the trajectory following task, the joint velocities are computed as:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{e} + \left[\mathbf{I} - (\mathbf{J_e P_H})^{\#}\mathbf{J_e P_H}\right]\dot{\mathbf{z}}_\mathbf{d} = \dot{\mathbf{q}}_\mathbf{e} + \mathbf{P_e}\dot{\mathbf{z}}_\mathbf{d}, \tag{18}$$

where $\mathbf{P_e}$ is an orthogonal projector and $\dot{\mathbf{z}}_\mathbf{d}$ an arbitrary vector. The final expression for $\dot{\mathbf{q}}$ can be obtained by combining Equations (17) and (18) and following the same steps described for the trajectory following task in Section III-B, thus leading to:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{e} + (\mathbf{H_d P_e})^{\#}(-\gamma_d \mathbf{d} - \mathbf{H_d}\dot{\mathbf{q}}_\mathbf{e}). \tag{19}$$

The part of the control scheme dedicated to the computation of Equation (19) is highlighted with a red block in the control scheme (see Figure 4).

IV. Experimental Setup and Experiments Description

The approach was validated on a Franka Robotics Panda robotic arm equipped with distributed ToFs on the elbow and tactile sensors on the forearm and end-effector. The experimental platform is shown in Figure 4 where the part covered with tactile and ToF sensors are highlighted in yellow and red respectively.

Our integration consists of 8 ToF sensors VL53L5CX from STMicroElectronics, featuring a square FoV of 65°, a maximum range of 4 m and an output consisting of an 8×8 depth map. ToF sensors communicate with the host PC at 15 Hz and have been integrated into the blue ring mounted on the robot elbow and are equally spaced as seen in Figure 3. ToF measurements have been processed as described in Section II to extract the position of the objects

$\mathbf{x}_{\mathbf{o}i}$. Moreover, if the robot's body falls into the sensors' FoV, the relative measurements are filtered out in a similar way to what is described in [28].

The distributed tactile sensing system visible in Figure 2 is presented in [29] and composed of capacitive taxels with a 3.5 mm diameter and a pitch of 7.5 mm. In total, our integration features 590 taxels on the forearm and 600 at the end-effector, with data acquisition occurring at a rate of 20 Hz. The sensor is not force-calibrated, therefore, the magnitude of the force applied on the robot is computed from the raw response provided by the sensors which is a dimensionless value in the range of [0-25000] units[2].

The robot is commanded to follow a trajectory and perform a pick-and-place task in the cluttered environment shown in Figure 5, using a soft pneumatic gripper as the end-effector. The peak velocity of the trajectory is set to $0.02\,\mathrm{m/s}$. Obstacles have been placed as highlighted in Figure 5 in yellow. Since the robot is not fully sensorised, we placed the obstacles such that contacts will occur on sensitive parts. Without this assumption, the validation of the approach is hindered since we cannot regulate contact forces or detect obstacles. The labels in Figure 5 represent the waypoints of the trajectory and pick or place positions. The robot moves from the starting point **S** to point **A** to pick up the object, which is then released at point **B**. Subsequently, it moves to point **C** before returning to the starting position.

The controller's gains were set to $\gamma_c = 5 \times 10^{-6}$, $\gamma_e = 2.0$ and $\gamma_d = 0.16$. Lower gains lead to slower robot movements with the consequence of being slow to react to sensors' feedback. On the contrary, too high values may lead to jerky movements. The gains were manually tuned to find a trade-off. Additionally, the contact force minimization and obstacle avoidance tasks should only intervene when sensor measurements exceed a certain threshold. Therefore, we utilize activation functions to enable/disable the tasks [31].

The proposed control scheme was compared against the purely tactile-based solution presented in [7] to show the advantage of the combined use of proximity and tactile feedback compared to a solution relying solely on tactile feedback. Both controllers are evaluated in the same experimental conditions. The task is executed starting from 3 different arm configurations: in one the starting point **S** is close to the obstacle on the right as in Figure 5, in another one it is closer to the obstacle on the left and in the third **S** is equidistant from the two obstacles. For each starting position, the experiment was repeated 5 times, resulting in a total of 15 experiments for each controller.

V. Results and Discussion

Figure 6 shows the behaviour of the controller and the different phases of the task execution when starting from the initial configuration shown in Figure 5. Furthermore, we also report plots showing the following trends over time:

- Top: the norm of the error $\|\mathbf{e}\|$. Furthermore, we highlighted with black circles the instants where the robot reaches the waypoints shown in Figure 5.
- Middle: The average of the contact forces applied by the robot on the environment, computed as explained in Section II. The spiky trend is related to the fact that the robot is rigid and the controller suddenly reacts when a force is detected. It is also worth noting that the value reported is dimensionless since it is computed from the raw measurements of the tactile sensor (see Section IV).
- Bottom: the distance of the object closest to the robot's elbow. The line in this subplot is not continuous since objects can fall outside the FoV of the sensors. Furthermore, we reported the distance of objects within 0.5 m. The plot also reports a red dashed line, representing the threshold distance $\bar{d}$ causing the activation of the task if the object gets closer than 0.3 m.

In the initial position **S** an obstacle is detected approximately at 0.2 m from the manipulator's elbow. Accordingly, the obstacle avoidance task is active and increases the distance between the robot arm and the obstacle until the elbow is farther than the distance threshold. This is highlighted with the red area in the plot. At the same time, after a short transient, the error on the trajectory starts decreasing. At approximately 8 s, the robot is approaching the object, but a part of its forearm collides with the environment. This activates the contact task, which, given its higher priority, leads to a deviation in the trajectory following, corresponding to an increase in $\|\mathbf{e}\|$. This segment of the task is highlighted with the yellow area on the left side.

Waypoints are considered to be reached when $\|\mathbf{e}\| \leq 0.02\,\mathrm{m}$ as this is enough to succeed in pick and place operations due to the compliance of the soft gripper. The spike in the error occurs when the robot stops at the waypoints thus resulting in a short transient before reaching again the steady state. This pattern is also visible for the other waypoints.

The oscillatory behavior observed in the distance subplot between 25 and 35 s is caused by the robot being in a configuration where small movements of the elbow, induced by the obstacle avoidance task, result in large variations in the computation of $\mathbf{x}_{\mathbf{o}i}$ (see Section II). Nevertheless, this causes just small oscillations in $\|\mathbf{e}\|$ that do not affect the task significantly. This could be fixed by applying a filtering algorithm on top of the computation of $\mathbf{x}_{\mathbf{o}i}$.

At 38 s the robot almost reached the place position. It is visible from the plot that the distance is way below the safety threshold. However, since the robot arm is almost fully extended, (as shown in Figure 6) there are not enough degrees of freedom to properly execute the obstacle avoidance task. The magenta box in the plot highlights the period where the obstacle avoidance task is violated.

Before reaching the waypoint **C**, the end-effector collides with the box. Similarly to the previous case, the force is detected by the tactile sensor and the controller works to reduce the applied force. Eventually, after reaching the last

[2]The measurements are acquired with 16-bit resolution. The effective range is between 1000 and 35000 units. When we processed the senors' measurements we removed the baseline of 10000. To retrieve the absolute force value is out of the scope of this paper since additional characterization and calibration procedures are required and usually hardware dependent [30].

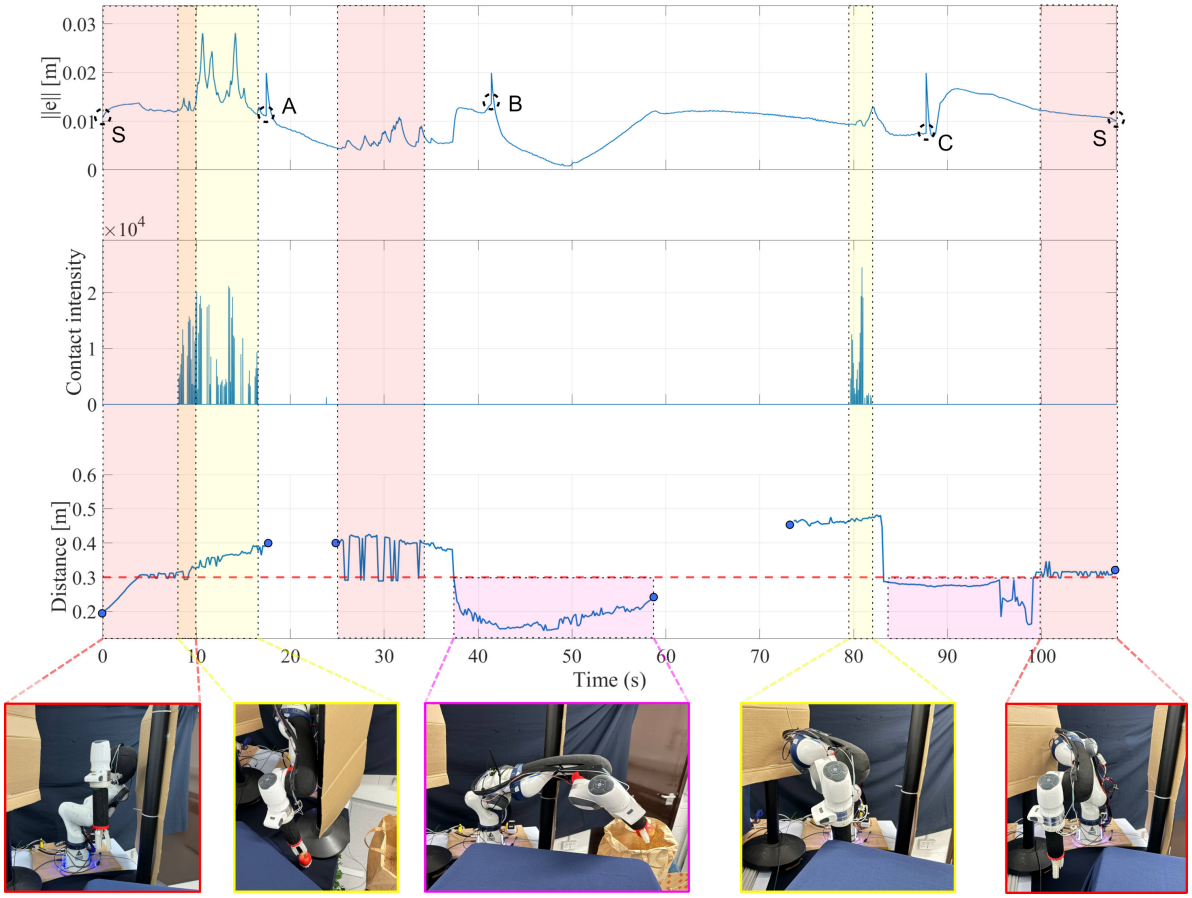

Fig. 6. The plot shows the trend of (Top) $\|\mathbf{e}\|$, (Middle) the average contact intensity of all contacts captured by the tactile sensors and (Bottom) the distance of the ToF sensors to the closest obstacle. The red areas emphasize the time in which the obstacle avoidance task is active. The yellow areas identify the activation of the contact force minimization task. Black circles in the subplot on top also highlight the instants in which the robot reaches the waypoints. The blue dots in the bottom subplot denote the point at which the obstacle gets in or out of the sensors' FoV while the dashed red line represents the threshold value. Figures on the bottom represent snapshots of the task execution.

TABLE I

COMPARISON BETWEEN THE CONTROLLERS

Poses	*Controller*	*t [sec] (avg ± std)*	*t_c [sec] (avg ± std)*	*$\|\bar{\mathbf{e}}\|$ [m] (avg ± std)*
1	Proxy-Tact	107.98 ± 0.27	8.21 ± 0.36	0.010 ± 0.000
	Tact [7]	118.25 ± 0.49	25.66 ± 0.55	0.018± 0.000
2	Proxy-Tact	111.09 ± 0.23	5.78 ± 1.09	0.010 ± 0.001
	Tact [7]	150.28 ± 6.90	47.46 ± 9.05	0.014 ± 0.003
3	Proxy-Tact	83.38 ± 0.43	4.77 ± 1.59	0.008 ± 0.003
	Tact [7]	83.86 ± 0.06	7.82 ± 0.17	0.007± 0.000

waypoint, the robot returns to the starting position. As visible from the plot, after a transient, the error decreases and is not affected by the execution of the obstacle avoidance task, which can intervene at 100 s and bring the elbow away from the obstacle. A video showing the motion of the robot is provided as supplementary material.

The results of the 30 trials related to the comparative experiments are presented in Table I. In particular, *Proxy-Tact* refers to the controller proposed in this paper, while *Tact* is the controller described in [7].

For each trial, we monitored the following metrics:

- t, the time required to complete the task;
- t_c, the time the robot spends in contact with the obstacles in the environment;
- $\|\bar{\mathbf{e}}\|$, the norm of the error averaged over the whole experiment.

For each metric, we reported in Table I their mean value and the standard deviation across the 5 experiments repeated starting from the same pose. The controller achieving the best performance is highlighted with a light blue color in the Table. Results show that the proposed approach outperforms the control presented in [7]. In particular, on average, it provides a lower execution time and significantly less time spent in contact. In particular, when starting from pose 2, the tactile-based controller can spend up to more than 45 s in contact with the environment, compared with less than 6 s achieved by the proposed approach. This shows that the integration of both proximity and tactile feedback allows for generating safer motions for the robot reducing the collision with the environment. It is also worth noting that, [7] provides slightly better accuracy in trajectory tracking (1 mm on average) when starting from the third pose. However, in the other two cases, $\mathbf{x}_{\mathbf{e}}^{*}$ is tracked with higher accuracy by

the *Proxy-Tact* controller.

VI. Conclusion

In this paper, we proposed a controller that exploits both proximity and tactile feedback gathered from distributed ToF and tactile sensors to perform a trajectory-following task in a cluttered and unknown environment. The solution allows for the use of proximity sensing to keep the robot away from obstacles when it does not interfere with the path-following execution and utilizes tactile sensing only when strictly necessary to manage collisions with the environment. We compare this approach against a controller that exploits tactile feedback only, demonstrating that the proposed solution provides better performance by executing the task faster while simultaneously reducing collisions with the environment.

While this controller is purely reactive, future research aims to explore the use of planning techniques to apply on top of the proposed solution to further reduce the possible collisions.

Also, in future work, we intend to cover a larger area of the body of the robot with the lightweight ToF sensors to improve obstacle detection, minimize occlusions and avoid external infrastructures.

References

[1] J. Kim and Y. Do, "Moving obstacle avoidance of a mobile robot using a single camera," *Procedia Engineering*, vol. 41, pp. 911–916, 2012.

[2] R. Lagisetty, N. Philip, R. Padhi, and M. Bhat, "Object detection and obstacle avoidance for mobile robot using stereo camera," in *2013 IEEE International Conference on Control Applications (CCA)*. IEEE, 2013, pp. 605–610.

[3] V. Lumelsky and E. Cheung, "Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 1, pp. 194–203, 1993.

[4] M. D. Killpack, A. Kapusta, and C. C. Kemp, "Model predictive control for fast reaching in clutter," *Autonomous Robots*, vol. 40, pp. 537–560, 2016.

[5] E. Magrini and A. De Luca, "Human-robot coexistence and contact handling with redundant robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4611–4617.

[6] T. Schmidt, R. Newcombe, and D. Fox, "Dart: Dense articulated real-time tracking," 07 2014.

[7] A. Albini, F. Grella, P. Maiolino, and G. Cannata, "Exploiting distributed tactile sensors to drive a robot arm through obstacles," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4361–4368, 2021.

[8] S. Tsuji and T. Kohama, "Proximity and contact sensor for human cooperative robot by combining time-of-flight and self-capacitance sensors," *IEEE Sensors Journal*, vol. 20, no. 10, pp. 5519–5526, 2020.

[9] S. E. Navarro, S. Mühlbacher-Karrer, H. Alagi, H. Zangl, K. Koyama, B. Hein, C. Duriez, and J. R. Smith, "Proximity perception in human-centered robotics: A survey on sensing systems and applications," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1599–1620, 2022.

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

[11] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 999–1004.

[12] F. Romano, G. Nava, M. Azad, J. Čamernik, S. Dafarra, O. Dermy, C. Latella, M. Lazzaroni, R. Lober, M. Lorenzini *et al.*, "The codyco project achievements and beyond: Toward human aware whole-body controllers for physical human robot interaction," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 516–523, 2017.

[13] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, "Reaching in clutter with whole-arm tactile sensing," *The International Journal of Robotics Research*, vol. 32, no. 4, pp. 458–482, 2013.

[14] Q. Leboutet, E. Dean-Leon, F. Bergner, and G. Cheng, "Tactile-based whole-body compliance with force propagation for mobile manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 330–342, 2019.

[15] S. Tsuji and T. Kohama, "Proximity skin sensor using time-of-flight sensor for human collaborative robot," *IEEE Sensors Journal*, vol. 19, no. 14, pp. 5859–5864, 2019.

[16] J. Liang, J. Wu, H. Huang, W. Xu, B. Li, and F. Xi, "Soft sensitive skin for safety control of a nursing robot using proximity and tactile sensors," *IEEE Sensors Journal*, vol. 20, no. 7, pp. 3822–3830, 2019.

[17] Y. Zhou, J. Zhao, P. Lu, Z. Wang, and B. He, "Tacsuit: A wearable large-area, bioinspired multi-modal tactile skin for collaborative robots," *IEEE Transactions on Industrial Electronics*, 2023.

[18] S. Tsuji and T. Kohama, "Proximity and tactile sensor combining multiple tof sensors and a self-capacitance proximity and tactile sensor," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 18, no. 5, pp. 797–805, 2023.

[19] S. B. Slotine and B. Siciliano, "A general framework for managing multiple tasks in highly redundant robotic systems," in *proceeding of 5th International Conference on Advanced Robotics*, vol. 2, 1991, pp. 1211–1216.

[20] A. Albini, S. Denei, and G. Cannata, "Towards autonomous robotic skin spatial calibration: A framework based on vision and self-touch," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 153–159.

[21] L. Rustler, B. Potocna, M. Polic, K. Stepanova, and M. Hoffmann, "Spatial calibration of whole-body artificial skin on a humanoid robot: comparing self-contact, 3d reconstruction, and cad-based calibration," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, 2021, pp. 445–452.

[22] G. Cannata, M. Maggiali, G. Metta, and G. Sandini, "An embedded artificial skin for humanoid robots," in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Aug 2008, pp. 434–438.

[23] P. Mittendorfer and G. Cheng, "Humanoid multimodal tactile-sensing modules," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 401–410, June 2011.

[24] Y. Ohmura, Y. Kuniyoshi, and A. Nagakubo, "Conformable and scalable tactile sensor skin for curved surfaces," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 1348–1353.

[25] T. Mukai, M. Onishi, T. Odashima, S. Hirano, and Z. Luo, "Development of the tactile sensor system of a human-interactive robot "riman"," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 505–512, April 2008.

[26] M. Hahsler, M. Piekenbrock, and D. Doran, "dbscan: Fast density-based clustering with R," *Journal of Statistical Software*, vol. 91, no. 1, pp. 1–30, 2019.

[27] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[28] S. Kumar, S. Arora, and F. Sahin, "Speed and separation monitoring using on-robot time-of-flight laser-ranging sensor arrays," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 1684–1691.

[29] "Cyskin," https://www.cyskin.com/.

[30] J. Kangro, S. Traversaro, D. Pucci, and F. Nori, "Skin normal force calibration using vacuum bags," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 401–406.

[31] E. Simetti, G. Casalino, S. Torelli, A. Sperinde, and A. Turetta, "Floating underwater manipulation: Developed control methodology and experimental validation within the trident project," *Journal of Field Robotics*, vol. 31, no. 3, pp. 364–385, 2014.