Journal of Intelligent Manufacturing (2023) 34:3053–3067
https://doi.org/10.1007/s10845-022-01989-y

Combining human guidance and structured task execution during physical human–robot collaboration

Jonathan Cacace[1] · Riccardo Caccavale[1] · Alberto Finzi[1] · Riccardo Grieco[1]

Received: 30 November 2020 / Accepted: 2 July 2022 / Published online: 5 August 2022

Abstract

In this work, we consider a scenario in which a human operator physically interacts with a collaborative robot (CoBot) to perform shared and structured tasks. We assume that collaborative operations are formulated as hierarchical task networks to be interactively executed exploiting the human physical guidance. In this scenario, the human interventions are continuously interpreted by the robotic system in order to infer whether the human guidance is aligned or not with respect to the planned activities. The interpreted human interventions are also exploited by the robotic system to on-line adapt its cooperative behavior during the execution of the shared plan. Depending on the estimated operator intentions, the robotic system can adjust tasks or motions, while regulating the robot compliance with respect to the co-worker physical guidance. We describe the overall framework illustrating the architecture and its components. The proposed approach is demonstrated in a testing scenario consisting of a human operator that interacts with the Kuka LBR iiwa manipulator in order to perform a collaborative task. The collected results show the effectiveness of the proposed approach.

Keywords Human–robot collaboration · Physical human–robot interaction · Collaborative task execution · Human intention estimation

Introduction

Collaborative robotic systems (CoBots) enable humans and robots to safely work in close proximity during the execution of shared tasks (Corrales et al., 2012) merging their complementary abilities (Romero et al., 2016). Depending on the application domain, human-robot collaboration (HRC) may require both cognitive and physical interaction, from coordinated execution of independent or sequential activities to physical and responsive collaboration in co-manipulation operations. While collaborative robotic platforms ensuring safe and compliant physical human-robot interaction are spreading in service robotics applications (De Santis et al., 2007), the collaborative execution of structured collaborative tasks still poses relevant research challenges (Johannsmeier & Haddadin, 2017). In these settings, activities of humans and robots should be monitored, coordinated, and suitably orchestrated with respect to the task and the human guidance. An efficient and fluent collaboration demands operators and CoBots to continuously estimate their reciprocal intentions to decide whether to commit to shared tasks and subtasks, when to switch towards different targets, or how to regulate compliant interactions during co-manipulation. These issues are particularly relevant in industrial scenarios, where tasks are usually well defined and explicitly formalized (Vernon & Vincze, 2016), while their execution should be continuously and fluently adjusted to the human activities and interventions in a shared workspace.

In this work, we address these issues considering a scenario in which a human operator interacts with a lightweight robotic manipulator through physical interaction in order to accomplish hierarchically structured collaborative activities. During task execution, human interventions can be associated with different purposes, e.g., lead the robot, slightly adjust its motion, change the target of the actions, speed

✉ Alberto Finzi
alberto.finzi@unina.it

Jonathan Cacace
jonathan.cacace@unina.it

Riccardo Caccavale
riccardo.caccavale@unina.it

Riccardo Grieco
riccardo.grieco@unina.it

[1] Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione (DIETI), Università degli Studi di Napoli "Federico II", via Claudio 21, 80125 Naples, Italy

up the execution, or use the manipulator as a passive tool. The intentions conveyed with these physical inputs should be continuously interpreted with respect to the planned activities and motions, while the robot behavior should be adapted accordingly. When the human physical guidance is assessed as aligned with respect to the planned activities, these are maintained along with trajectories and targets. Otherwise, depending on the assessment of the operator's aims, the robotic system may switch tasks, change targets, adjust trajectories, while suitably regulating the robot's compliance with respect to human guidance.

This fluent and direct interaction between human workers and CoBots should be associated with a flexible execution of collaborative tasks, which needs to be continuously adapted at different levels of abstraction. Different methods have been proposed in the robotics literature to enable collaborative plan execution, the dominant paradigm relies on activity replanning when the human behavior diverges from the one planned (Karpas et al., 2015; Carbone et al., 2008; Shah et al., 2011; Lallement et al., 2014). On the other hand, continuous replanning is usually computationally expensive and can affect the naturalness and effectiveness of the interaction, which is crucial when humans and robots work in close proximity with frequent physical contacts. In order to harmonize flexible task execution and fluent interaction with humans, we leverage the framework proposed in Caccavale and Finzi (2017); Caccavale et al. (2019); Caccavale and Finzi (2019, 2022), which exploits supervisory attention and contention scheduling (Norman & Shallice, 1986; Cooper & Shallice, 2006) to monitor human behaviors and suitably orchestrate multiple hierarchically structured tasks with respect to the interpreted human interventions. In this setting, supervisory attention permits the smooth integration of autonomous guidance and human guidance through top-down and bottom-up regulations, in so enabling flexible, adaptive, and interactive execution of collaborative plans.

Collaborative task adaptation occurs while simultaneously interpreting the human interventions with respect to the activities proposed by the flexible plan. Depending on the operational state and the environmental stimuli, the supervisory system enables possible subtasks, targets and trajectories, which are continuously evaluated by intention recognition processes. Specifically, in the proposed framework each possible trajectory is assessed by a Long Short Memory Network (LSTM) that, upon receiving as input the robot motion and the operator contact forces, infers the intention of the operator to follow/contrast the manipulator motion towards a target point, deviate from the latter, or use the robot manipulator in direct hand-guided control to prepare other activities. In this scenario, when the human interventions and the current plan targets are aligned, the robotic system can keep executing the current plan, while suitably adjusting its motion trajectory following the corrections provided by the human. Otherwise, different targets, trajectories, and subtasks should be selected to adjust the estimated human indications with respect to the activities enabled by the collaborative plan. When the human guidance remains unclear in the context of the task, the robotic system should remain passive and fully compliant with the human guidance.

In order to demonstrate the proposed framework, we designed an experimental setup inspired by an industrial scenario, where a human operator cooperates with a Kuka *LBR iiwa* robot for the coordinated execution of multiple tapping operations in a shared workspace. In this scenario, we proposed a pilot study to evaluate the users and system performance in different settings (passive, guided, proactive) considering both quantitative (effort and execution time) and qualitative (questionnaire) assessments. The collected results show the advantage of the proposed assisted modalities with respect to the passive one. Interestingly, the guided setting emerges as the preferred mode despite the advantage of the proactive mode in terms of physical effort and execution time.

In summary, in this manuscript we propose a novel human-robot collaboration framework which seamlessly combines human intention interpretation and activity orchestration during physical interaction for adaptive execution of structured collaborative tasks. The proposed system integrates and develops different approaches to collaborative task execution and human intention recognition (Cacace et al., 2019, 2018). The LSTM-based intention recognition method proposed in this work extends the contact force classification technique introduced in Cacace et al. (2019). While the previous method is reactive and provides instantaneous intention classification from current features, in the novel approach the sequence of past interactions are exploited to assess the human intent during task execution. A preliminary approach to collaborative execution of structured co-manipulation tasks is proposed in Cacace et al. (2018), where the human physical interventions are interpreted in the context of hierarchically structured tasks exploiting simple disambiguation and task switching policies, which do not involve attention-based influences. In contrast, in this work attention regulation mechanisms and intention recognition processes are fully integrated to assess human intentions and to smoothly regulate the compliant execution of collaborative activities at different levels of abstraction. In the extended framework, we can define different interaction modes which are then tested and compared in an experimental setup. Such assessment of the system along with the enabled interaction modes is another original contribution of this work.

The remainder of this paper is organized as follows. In Related works section, a brief overview of related works is presented, in Collaborative human–robot manipulation section the overall system is described presenting the architecture and the associated components. Operator intention estimation section details the human intention estimation process. An experimental case study is proposed and discussed in Experiments and results section. Finally, the conclusion provides a summary of the proposed results along with some lines of future research.

Related works

In the human-robot interaction literature, different frameworks have been proposed to support human-aware planning and collaborative plan execution (Karpas et al., 2015; Carbone et al., 2008; Shah et al., 2011; Lallement et al., 2014; Clodic et al., 2008; Caccavale & Finzi, 2017; Caccavale et al., 2016; Sisbot et al., 2007), in our work, we focus on physical human-robot interaction for the collaborative execution of hierarchically structured tasks. In this setting, the human physical guidance and the plan guidance should be strictly integrated in order to enable an effective and fluent human-robot collaboration. Related to the scenario considered in this work, in Johannsmeier and Haddadin (2017) the authors propose a framework for human-robot collaborative execution of industrial assembly processes. However, in this case, the main focus is on the allocation and coordination of human-robot activities, while we are concerned with natural and compliant human-robot collaboration during the execution of a shared work plan. In this respect, we propose an approach to combine plan and human guidance by means of a continuous interpretation of the human physical interventions during the interactive execution of a task. Intention estimation is considered as a crucial issue for natural human-robot collaboration in a shared workspace (Hoffman & Breazeal, 2004, 2007). Different approaches have been proposed in the robotics literature to integrate physical interaction interpretation with adaptive and compliant control. For instance, in Peternel et al. (2016), the authors present a method in which the robot behavior is regulated based on the estimated human fatigue, but operators' trajectories and targets are not inferred. Instead, in Nicolis et al. (2018) a proactive system assisting the human operator during path navigation is enabled by predicting and classifying human-robot cooperative motions, given a set of predefined goals and data of human movements. Co-manipulation with multiple virtual guides is addressed in Raiola et al. (2015); here the authors estimate the most likely workspace trajectories and locations using learned Gaussian mixture to guide the user during the execution of the shared pick-and-place tasks. Differently from these approaches, in our framework, the estimation of the operator intentions is blended with structured task orchestration mechanisms and exploited at different levels of abstraction to support trajectory, target, and task/subtask selection. Another related approach can be found in Park et al. (2019), where intention and motion prediction methods support adaptive human aware motion planning. In this case, analogously to our approach, an integrated architecture that combines task and motion planning with intention assessment is proposed, however, physical human guidance and co-manipulation tasks are not considered. Other related, but complementary, works concern compliant control methods supporting physical human-robot collaboration. Among them, starting from the prior works by Colgate and Hogan (1989, 1988), admittance controllers are traditionally adopted for this scope. In this context, the regulation of the damping of the admittance controller can be online adapted to increase the effectiveness of the co-manipulation system (Grafakos et al., 2016; Cacace et al., 2019; Cacace et al., 2019). In this work, we adopt a classical admittance control schema to enable compliant physical interaction during the collaborative execution of structured tasks.

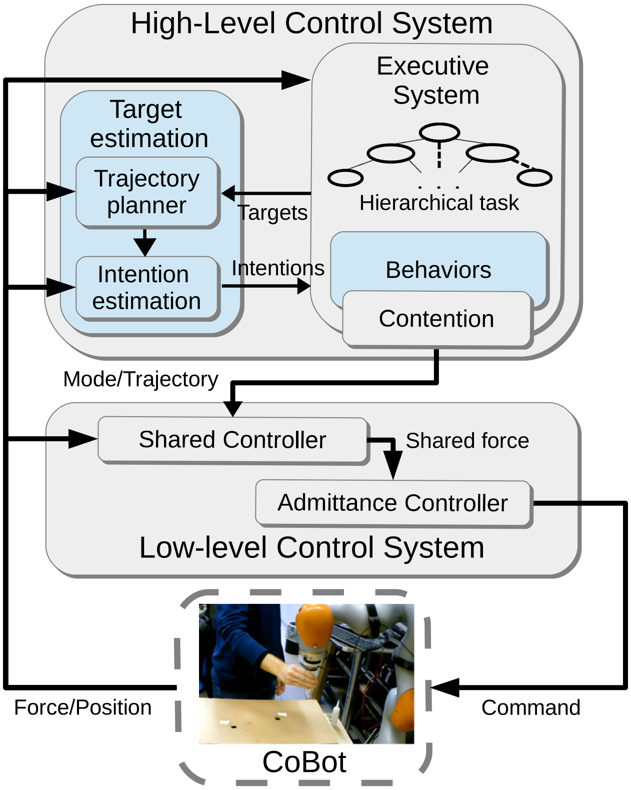

Fig. 1 The system architecture is structured in two main layers. The high-level control system manages and supervises task generation and collaborative task execution; the low-level control system enables compliant execution of primitive operations

Collaborative human–robot manipulation

In this section, we describe the human-robot collaborative framework presenting the overall architecture along with its main components.

The collaborative system is structured in two main control layers (see Fig. 1) working at different levels of abstraction. The *High-Level Control System* (HLC) is a deliberative layer responsible for task generation, decomposition, orchestration, and interaction. The *Low-Level Control System* (LLC) is concerned with the actual execution of the primitive operations selected by the HLC while maintaining the robotic system compliant with respect to the human interventions.

The *Executive System* is an HLC module that manages the orchestration of multiple collaborative tasks taking into account both the environmental changes and human activities. During task execution, the human operator can physically interact with the CoBot and these interventions (force/position feedback) are simultaneously interpreted at the different layers of the architecture. Depending on the task, the environmental context, and the human interventions, the Executive System (top-down) proposes a set of primitive operations/processes (*Behaviors*) that compete for the execution (*Contention*). Each proposed behavior is associated with a target position (*Target*) and an activation value, the latter representing an attentional weight, which summarizes the relevance of that activities given the current execution state. The *Target estimation* module generates a trajectory (*Trajectory Planner*) for each proposed target and assesses it (*Intention Estimation*) considering the current human guidance in order to estimate the most aligned with respect to the human interventions. The classification results (*Intentions*), along with the associated attention weights, are then exploited to influence behavior selection (*Contention*) with the associated target position for the CoBot. Finally, the LLC implements a *Shared Controller* aimed at mixing the inputs generated by the human operator with the ones needed to perform robot motion (*Shared force*). An *Admittance Controller* integrates the human and the robot guidance. In the following sections, we further detail the Executive System, the Target Estimation process, and the Low-Level Control System.

Executive system

The Executive System is responsible for task retrieving, decomposition, monitoring, orchestration, and regulation. Specifically, we rely on the supervisory attention framework proposed in Caccavale and Finzi (2015); Cacace et al. (2018); Caccavale et al. (2019) for human-robot collaboration. In this setting, the executive system is decomposed into an *Attentional Executive System* and an *Attentional Behavior-based System*. The first one manages the execution of hierarchically structured tasks along with the associated activations (top-down attentional regulations); the latter collects the active robot processes (behaviors), each associated with an activation value (obtained as a combination of top-down and bottom-up attentional regulations).

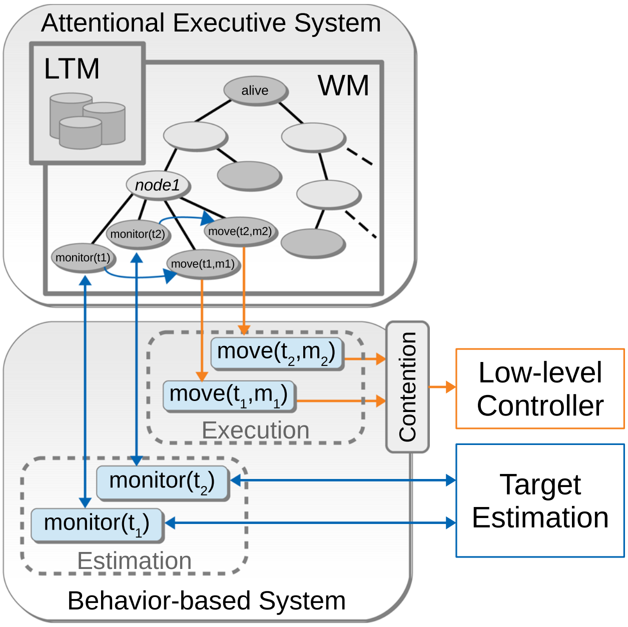

Fig. 2 The executive system manages the execution of multiple hierarchically structured tasks

A representation of the Executive System is proposed in Fig. 2, where we can highlight three main components: a Long Term Memory (LTM), a Working Memory (WM), and a Behavior-based System (BS). The LTM collects the system procedural knowledge, i.e., the specification of the tasks available to the robot. A task can be either abstract (to be further decomposed) or concrete (a real sensorimotor process). Each task is defined in the LTM by a predicate **schema**(m, l, p), where m is the name of the task, l is a list of m_i subtasks along with associated enabling conditions r_i (releasers), i.e. $l = \langle (m_1, r_1), \ldots, (m_n, r_n) \rangle$, while p is a postcondition used to check task accomplishment.

The WM is a data structure that collects hierarchically decomposed tasks instantiated and allocated for execution. The task set in WM along with the associated state variables characterizes the current execution state of the system. The WM is represented by an annotated rooted directed graph (r, B, E), whose nodes in B represent allocated tasks/subtasks, E are parental relations among subtasks, while $r \in B$ is the root process that manages the WM structure. Each node $b \in B$ is represented as a 5-tuple $(m_b, r_b, p_b, x_b, \mu_b)$, where m_b is the name of the allocated task, r_b and p_b represent the task precondition and postcondition respectively, x_b is the set of sub-behaviors generated by m_b, while μ_b is an activation value assigned to the task. Leaves in the WM structure correspond to *attentional behaviors* devoted to the execution of sensorimotor processes.

The BS collects all the allocated, active, and concrete behaviors which compete for the execution. Behaviors can be marked as *enabled*, if the associated precondition in WM along with the ancestors' preconditions are satisfied, *disabled* otherwise. An enabled behavior is *accomplished* if its postcondition is satisfied. Enabled behaviors which are not accomplished can be dispatched and executed by the executive system once the associated resources are allocated (actuators, input/output devices, control variables, etc.).

Since multiple behaviors can be active at the same time, they may conflict in accessing non-shareable resources. We rely on *contention scheduling* mechanism (Norman & Shallice, 1986; Cooper & Shallice, 2000) to regulate this competition. For this purpose, we exploit the behaviors' activation values. When a conflict arises, following a winner-takes-all approach, the behavior associated with a higher activation value is selected for exclusive access to a contended resource.

In WM, the activation value of a node is given by the weighted sum of all contributions for that node:

$$\mu_b = \sum_i w_{i,b} c_{i,b}, \tag{1}$$

where contributions $c_{i,b}$ can be either inherited from the connected nodes (with $i \neq b$) in the WM structure (top-down) or generated by the node itself ($i = b$) from external or internal stimuli (bottom-up), while $w_{i,b}$ are the contribution-specific weights. This way, given a shared resource or variable v and the set of competing behaviors $B(v)$ for that variable, the behavior acquiring v is

$$b_{win} = \arg\max_{b \in B(v)}(\mu_b). \tag{2}$$

Overall, the executive system works as follows: when a new task is allocated in the WM for the execution, the associated schemata are recursively retrieved from the LTM and allocated into the WM until primitive sensorimotor processes. Preconditions and postconditions associated with allocated tasks are continuously monitored by the executive system in order to establish the set of subtasks that are active and enabled in the current operative context. The behaviors belonging to the enabled subtasks are then associated with specific activation values, which are used to regulate their competition in case of conflicts. This induces soft scheduling where the most emphasized behaviors (i.e., the one's better fitting the executive context) are prioritized. For instance, let assume a cooperative task where a robotic arm is tasked to pick two objects from a table and to put them into a basket following the human physical guidance. Assuming the two tasks enabled (i.e., both preconditions are satisfied and the tasks are not yet accomplished), in the absence of human interaction, the robot may be attracted by the nearest object on the table due to its better accessibility (i.e., bottom-up stimulated by object proximity). However, during the movement towards the proximal target, the operator can physically interact with the robotic arm, pushing it toward the second (less accessible) object. In this case, the human intervention would elicit an additional activation influence inducing the robot to switch towards the intended target. This mechanism is further detailed below.

Classification and regulations

We exploit object accessibility, task-based constraints, and human intention recognition to suitably single out, among the allocated tasks, the ones consistent with respect to the executive context and the user interventions. For this purpose, we distinguish the following types of influences to activations of the nodes in WM:

– *Task-based influence* (t_i), which is top-down provided to node i by the allocated tasks/subtasks in WM to be accomplished.
– *Human-based influence* (h_i), which is provided to node i by the physical interaction between the human and the robot; it emphasizes enabled nodes, which are also coherent with respect to the human guidance.
– *Accessibility-based influence* (a_i), which is provided by the environment, it emphasizes enabled nodes whose targets (e.g., objects, locations or trajectories) are more accessible (e.g., closer).

The *task-based influence* is the weighted sum of the contributions inherited from the other nodes in WM, i.e., $t_i = \sum_{j \neq i} w_{i,j} c_j$. Instead, the human and the accessibility influences are combined together into a unique contribution $c_{i,i}$ due to external stimuli. This is obtained by the following convex combination:

$$c_{i,i} = m_h \cdot h_i + m_a \cdot a_i \tag{3}$$

with $m_a, m_h \in [1, 0]$ and $m_a = 1 - m_h$. This weighted sum is exploited to mediate between accessibility and human guidance. The *accessibility-based* influence drives the robot towards the closest location where an operation can be performed (target), as specified by Eq. 4:

$$a_i = \frac{d_{MAX} - d(i)}{d_{MAX}} \tag{4}$$

where $d(i)$ is the length of the trajectory calculated to reach the target of the node i, and d_{MAX} is the maximum reachable distance in the robot workspace.

The *human-based influence* should induce the CoBot to move towards the target pointed by the operator guidance. In our framework, each possible target location is associated with a score $s_h(i) \in [0, 1]$, obtained from the assessment of the human physical guidance given the target associated to node i. Such score is an output provided by the LSTM classifier and estimates how likely the user is driving the robot to that location (as detailed in Operator intention estimation section). The human-based influence is then defined as follows:

$$h_i = \frac{max(0, s_h(i) - \lambda)}{1 - \lambda}, \tag{5}$$

where $\lambda \leq 1$ is a suitable threshold used to discriminate the reliability of the score.

Depending on the balances of weights in Eq. 3 we can introduce the following execution setups:

- *Human-guided:* enabled when $m_h >> m_a$; in this case, the CoBot is more prone to follow the human guidance rather than possible alternatives enabled by the plan and suggested by the environment (i.e., targets accessibility).
- *Target-guided:* associated with $m_a >> m_h$; in this mode, the CoBot tends to act according to the plan guidance and the environmental stimuli, rather than following the operator inputs.
- *Balanced:* when $m_a \approx m_h$ the system is not biased towards accessible targets or human guidance, but the robotic behavior is equally sensitive to both of them.

The combined effect of the human and the accessibility influences can be exemplified considering the scenario depicted in Fig. 3, which represents target points in a workspace to be reached by the robot end-effector with human assistance. The collaborative task is decomposed into 5 behaviors, each associated with a target location (the colored points depicted in the figure). We assume all behaviors are always enabled (satisfied preconditions) with the same task-based influence since the goal is to reach all the target points without a specific ordering. During the execution, the operator can physically interact with the robot to drive its end effector toward the desired location (e.g., from $WP1$ to $WP4$).

The development of the activation values in this scenario is illustrated in Fig. 4. The first two charts plot the temporal evolution (in timesteps sampled at 120 Hz) of the distance and the human intention stimuli for each active concrete behavior associated with a target point ($WP1$, ..., $WP5$). The other three charts plot the evolution of the combined contributions assuming 85–15%, 60–40% and 25–75% of balance between the accessibility and the human intention stimuli; these three cases are examples of the *target-guided*, *balanced*, and *human-guided modes*, respectively.

Coming back to Fig. 3, it shows the robot end effector starting from a position close to $WP1$ and $WP2$ to progressively reach $WP4$ while passing near $WP3$. The associated activations are plotted in Fig. 4. As illustrated in the second chart of Fig. 4, initially the human guidance is neglected ($s_h < \lambda$), and the robot behavior is mainly affected by the proximal waypoints, at about half a second (around 40 steps) the human guidance is also considered ($s_h > \lambda$) and $WP3$, $WP4$ and $WP5$ are recognized as possible targets. Notice that $WP1$ and $WP2$ are opposed to human guidance therefore they do not receive stimuli, while $WP4$ is the target that better fits the guidance, hence it receives the higher stimulus. In the *target-guided* setting (third plot), alternative targets compete and the robot has to reach a certain distance from $WP3$ before the desired target $WP4$ becomes the one winning. Instead, in the *balanced* an *human-guided* settings (last two plots), since the score of the human intention estimation is higher, the behavior associated with $WP4$ immediately wins the competition among the other targets.

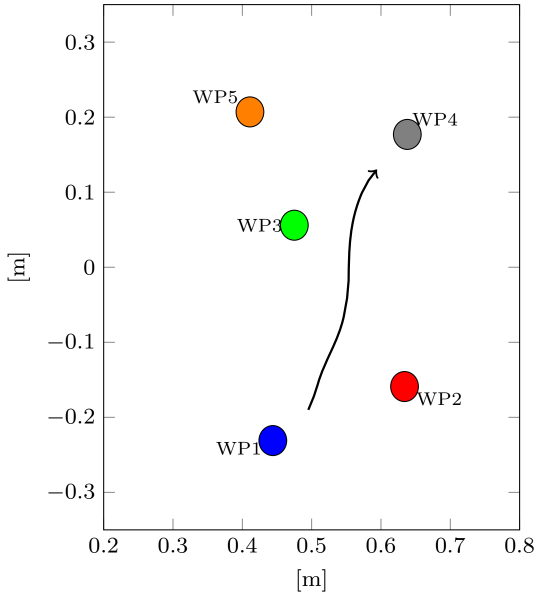

Fig. 3 Example scenario: combined human and accessibility influences while the robot moves from $WP1$ to $WP4$

Low level control

In this work, we assume that the robotic system is controlled in position and the control input of the system is represented by the desired position of its end-effector. Here, a compliant behavior is deployed by means of the classical admittance control schema (Siciliano et al., 2008) to allow the human operator to physically interact with the manipulator. In this context, the system dynamics is described by Eq. 6 that maps the overall forces acting on the end-effector with its position in the Cartesian space as follows:

$$f_{tot} = Mp + D\dot{p} + K\ddot{p} \tag{6}$$

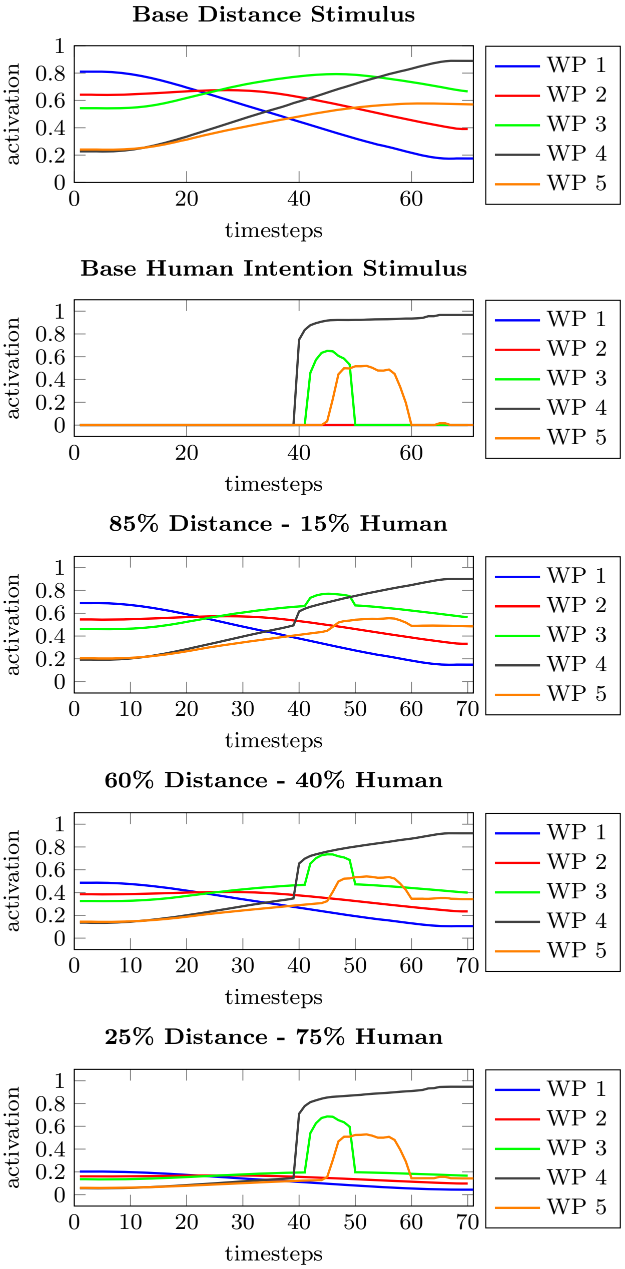

Fig. 4 Temporal evolution of behaviors' activations with respect to target accessibility (distance stimuli) and human influence

where p, $\dot{p}$ and $\ddot{p}$ respectively represent the position, velocity, and acceleration of the robot, while M, D, and K are three control gains describing the mass, damping, and spring of the second order virtual mechanical system. Starting from Eq. 6, the desired acceleration of the robot is trivially calculated in order to obtain the position to command. Pursuing our main goal, the f_{tot} forces are generated by the *Shared Controller* module of the system architecture as a combination of two components: the external forces f_{ext} exerted by the human operator during the co-manipulation task; the forces f_a generated by the autonomous system during the autonomous motion. The latter are properly calculated to reach the target provided by the High-Level Control layer (the one winning the contention, as described above). In particular, once a new desired waypoint is selected, the *Shared Controller* module generates a 3D geometric trajectory connecting the current position of the manipulator and the waypoint. Hereby, the goal of f_a is to constrain the robot along the generated path. Such forces are calculated as a function of the euclidean distance between the end-effector current position x_c and the desired position along the path in a given time, as reported in Eq. 7:

$$f_a = K_p(x_d - x_c) + K_d(\dot{x}_d - \dot{x}_c) \tag{7}$$

where K_p and K_d are the proportional and derivative gains, respectively. This formula defines the dynamic relationship between the applied forces and the motion of the robot thought a virtual inertia (K_p) and damping (K_d) values. Differently, x_c and $\dot{x}_c$ represent the current position and velocity of the manipulator, while x_d and $\dot{x}_d$ are the desired ones. Moreover, when no target has been selected by the higher layer of the architecture, the autonomous forces are nullified, allowing the robotic system to respond to the human forces only, in so enabling the passive control mode.

Operator intention estimation

In our framework, human-robot collaboration is supported by the interpretation of the operator's intention from his/her physical guidance during the execution of the shared task. To assess human intentions, we follow and extend the approach proposed in our previous work (Cacace et al., 2018), where human physical interventions on the robot are evaluated with respect to targets and related trajectories exploiting a neural network. Specifically, the human interventions, are classified by the network in the following categories depending on the concordance of the operator inputs with respect to targets and trajectories:

- *Concorde* (C): Human guidance follows the trajectory.
- *Deviation Concorde* (D_C): The operator wants to modify the trajectory without changing the active target.
- *Opposite* (O): The operator wants to go against robot motion.
- *Deviation Opposite* (D_O): The operator wants to switch target.

In Cacace et al. (2018), the classification is performed by a three-layered Fully-Connected Feed-Forward (FF) Neural Network composed of an input layer, a middle layer, and an output layer. The input layer takes an *interaction* *snapshot* made up by the human force magnitude $\|F_t\|$, the angle between human force direction and planned motion $d_p = \angle(\mathbf{d}_d, \mathbf{d}_p)$, and the distance between the position of the end-effector, and the closest point of the trajectory $d_h = \|X_p - X_c\|$. The middle layer consists of 25 nodes considering the sigmoidal activation function. Finally, 4 nodes corresponding to the possible classes make up the output layer. The proposed network model tries to generalize human intention classification taking into account only one single step of the interaction, i.e., one vector $h = (d_p, d_h, \|F_t\|)$. The approach is reactive and provides satisfactory results (Cacace et al., 2018), on the other hand, the classification is instantaneous and does not exploit the history of past interactions to disambiguate the human intent. In this work, we extend this approach in order to enhance the intention recognition process by exploiting the flow of data collected during the human-robot interaction. Indeed, data about previous interactions may not only support the interpretation of the current intervention but also reduce possible observational errors, caused either by the sensors or by the way the human touches the robot during collaborative task execution. In this direction, we propose to deploy Recurrent Neural Network (RNN) based on LSTM nodes, which are particularly suited for time series classification. LSTM networks have been introduced to address the vanishing gradient problem in RNNs exploiting gates that selectively retain relevant information while forgetting irrelevant information. Specifically, each LSTM node is composed of a memory *cell* and 3 different networks called *gates* (i.e., *input gate*, *forget gate*, *output gate*) acting as regulators for the manipulation and the utilization of the memory.

Our intention classification network consists of an input layer, a hidden layer made up of LSTM cells, and an output layer associated with a *softmax* function. Notice that a new classification network is allocated for each trajectory/target to be assessed, therefore, in order to limit computational effort and memory usage, the desiderata is to deploy simple and small network structures. For this purpose, we designed a method for sequence classification that enables online deployment of such networks.

Given an input sequence $\mathbf{h}$=$(h_1, \ldots, h_n)$, where each h_i represents the i-th human interaction snapshot, and given its corresponding classification sequence $\mathbf{s}$=$((y_{1,1}, y_{1,2}, y_{1,3}, y_{1,4}), \ldots, (y_{n,1}, y_{n,2}, y_{n,3}, y_{n,4}))$, where each 4-tupla represents the outputs related to the 4 classes introduced above, the class c assigned to $\mathbf{h}$ is the first one for which there exists a subsequence $(y_{t_0,c}, \ldots, y_{t_0+\Delta,c})$ such that for all $t \in [t_0, t_0 + \Delta]$, we have that $y_{t,c} > \lambda$ holds. That is, the sequence $\mathbf{h}$ is assigned to the class c, such that the classification result c remains coherent for a fixed time windows Δ, with confidence always greater than a fixed threshold λ in that window.

Algorithm 1 RNN training algorithm.

```
1:  procedure TRAIN($train\_steps$,$epochs$,$h\_size$)
2:      $net = random\_weights(h\_size)$
3:      $best\_acc = 0$
4:      $batches = split\_into\_batches(train\_x, train\_steps)$
5:      for $i \leftarrow 1$ to $epochs$ do
6:          for $j \leftarrow 1$ to $length(batches)$ do
7:              $net = train(net, batches[j], train\_y)$
8:          $acc = evaluate\_network(net, test\_x, test\_y)$
9:          if $acc > best\_acc$ then
10:             $best\_net = net$
11:             $best\_acc = acc$
12:         $net = reset\_states(net)$
13:     return $net, best\_acc$
```

In our experimental setting, Δ was empirically set to 40 steps (about half a second and about one third of the length of all the sequences in the dataset), while λ was set to 0.5. In the training phase different hyperparameters, such as hidden layer size and a number of training epochs, have been tested in order to select the network with satisfactory accuracy in the proposed application. The dataset has been generated by physically interacting with the robot during its motion from one point to another following a simple trajectory while recording data at 100 Hz. Since sequences with different lengths can be collected, these were divided into subsequences of a fixed length to be used in batch for learning. Specifically, we used 120 timesteps (about 1 second), shortest sequences were discarded, while sequences longer than 120 timesteps were divided into subsequences of the fixed length. Notice that in the experimental setup, the fixed length was chosen considering, on the one hand, the statistics of the dataset (average length and percentiles), on the other hand, the latency given to the system to assess the human intention (about 1 second). The collected dataset was then randomly split into a training set and test set, covering the 80% and 20% of the data and an amount of 443 and 111 sequences, respectively.

The procedure adopted to train the classifier is reported in Algorithm 1. Here, a trained network for a given configuration of the hyperparameters is obtained by running batch-training over different learning epochs. Specifically, at the beginning of the process, a network is generated with a h_size number of LSTM nodes for the hidden layer and random weights in the interval $[-1, 1]$ (line 2–3). Then, the training set is split into batches by grouping together subsequences of $train\_steps$ size (line 4). For each epoch (line 5), batch-training is performed through forward and back-propagation (lines 6–7), then the performance of the trained network is evaluated (line 8). At the end of the epoch, the best regulation is updated (lines 9–11), and the states of the LSTM network are reset (line 11). Finally, the best network and the best accuracy found during the training process are returned (line 12).

Table 1 Accuracies with different hyperparameters

LSTM nodes	Steps	Epochs	Accuracy (%)	Average F1
8	1	150	81.08	0.83
8	5	300	84.68	0.85
8	10	850	83.78	0.84
8	20	850	83.78	0.84
8	30	400	81.98	0.83
8	40	850	83.78	0.85
8	60	850	78.38	0.79
8	120	800	73.87	0.75
16	1	400	83.78	0.85
16	5	700	86.49	0.87
16	10	650	84.68	0.85
16	20	200	83.78	0.84
16	30	300	79.28	0.82
16	40	950	81.98	0.84
16	60	1300	81.98	0.83
16	120	850	81.98	0.82
24	1	650	83.78	0.85
24	5	550	84.68	0.86
24	10	200	82.88	0.83
24	20	200	84.68	0.85
24	30	250	82.88	0.84
24	40	350	81.92	0.83
24	60	800	85.59	0.86
24	120	600	82.88	0.82

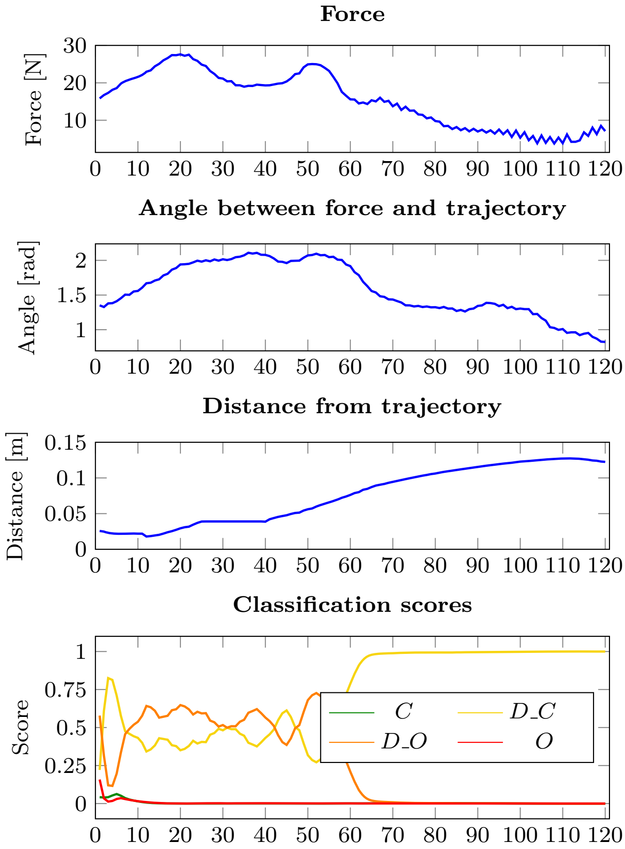

Fig. 5 Example of LSTM classification

Table 1 reports the different accuracies reached with different train steps, different training epochs, and different sizes of the hidden layer. For ease of comprehension, not all of the combinations are reported. The obtained results show that, on average, satisfactory accuracies are reached faster when shorter training steps are exploited. On the other hand, with longer training steps more epochs are needed, while we empirically observed that accuracies may also get worse. In particular, learning on shorter subsequences seems a better choice for on-line classification since the network tends to classify with less information and to better manage its memory during the interactive execution. In this respect, its worth recalling that the intention classifier is designed to be a component of the overall interaction system and to balance accuracy, computational effort, memory usage, and reactivity. The intention classification results are indeed continuously integrated by the executive system with other influences (i.e., accessibility and task-based guidance) to affect internal regulations and behavior selections (see Classification and regulations section).

An example of how the trained network classifies a sequence is illustrated in Fig. 5. Here, the network seems to start from a state of indecision between the *Deviation Concorde* and the *Deviation Opposite* classes. However, the decrease of the angle and the force breaks the ambiguity, determining *Deviation Concorde* as the winning class. Notice that this behavior is satisfactory for the interaction, indeed the two classes are very close, hence their classification can remain not defined until the operator solves the ambiguity with a physical intervention aligned or against the proposed trajectory.

Finally, to further highlight the importance of the interaction history in the intention estimation process, we can compare the proposed *LSTM* network with respect to the *FF* network presented in Cacace et al. (2018). As expected, the LSTM-based classifier reaches a higher accuracy (86.49%. vs 80.84%) once assessed over the same test set, but it also provides additional advantages. The difference between the two approaches can be exemplified in Fig. 6, where the same input sequence is classified deploying the two networks. The first three plots of the figure illustrate the input data (force, angle, distance), while the remaining two plots show the highest score and the associated class obtained with the two methods. In the reported case, we can observe how the *FF* network easily tends to be confident about the class of a single snapshot, but noise in the input data (e.g., spikes in the angle values) may provide sudden changes in classifica-

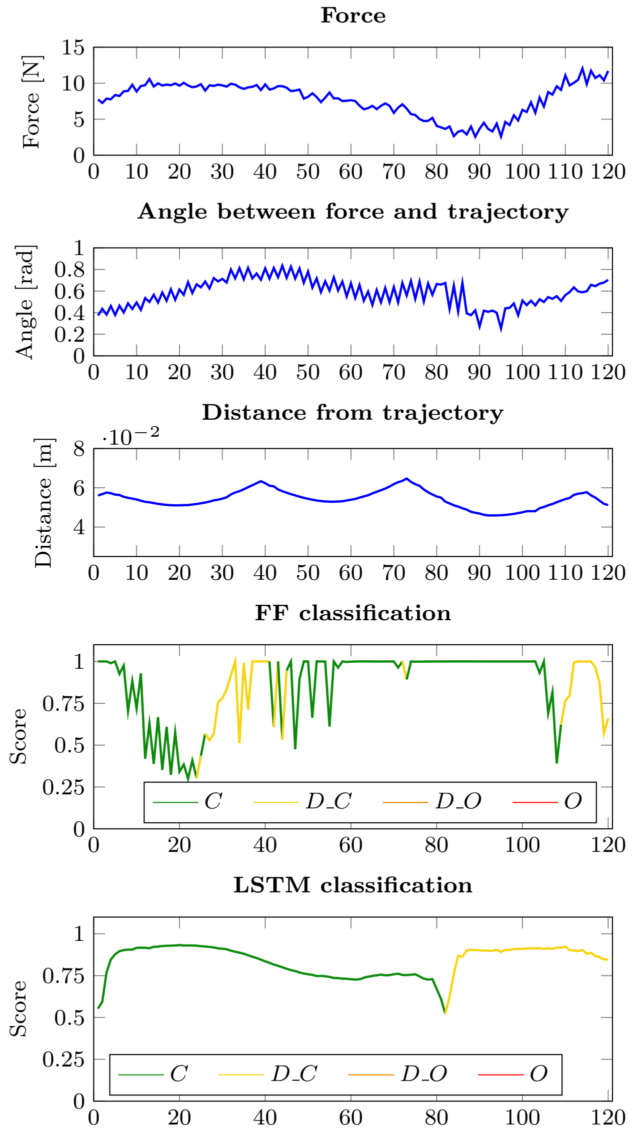

Fig. 6 An input sequence classified by the LSTM network and the FF network presented in Cacace et al. (2018). The first three charts represent the inputs of the classifiers, while the last two are the scores of the winning class respectively by the FF and the LSTM network

tion. In contrast, the *LSTM* network shows lower scores for the same input sequence, but it seems more robust to noise, which is a desirable behavior since the classification result coherence influences the smoothness and consistency of the overall interaction.

Experiments and results

In this section, the evaluation of the proposed approach is discussed. We designed an experimental setup inspired by an industrial scenario in which a human operator interacts with a CoBot for the collaborative execution of different operations.

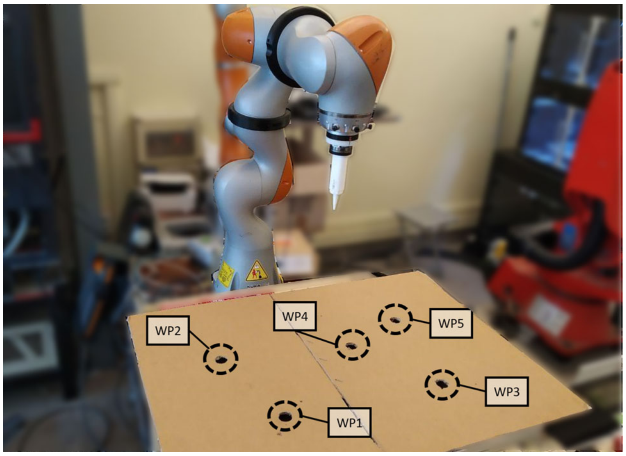

Fig. 7 Experimental setup with labeled points

The shared workspace is depicted in Fig. 7, it illustrates a mockup, which is mainly made up of the cardboard cutout, representing a surface with 5 holes, where operations must be executed. Since the main interest in this work is human-robot collaboration, such operations were simulated, and a mockup 3D-printed tool has been attached to the robot's flange. In all of the experiments, a *balanced* setup for robot's behavior was used, with 50% of importance given to environmental and human influences, in order to analyze the users' interaction in a uniform setting where both the human and the robot guidance are active without a preset bias.

Tests were performed using the Kuka LBR iiwa manipulator, controlled via ROS middleware (Joseph & Cacace, 2018) running on a standard version of Ubuntu 18.04 GNU/Linux OS. The ATI Mini 45 Force sensor has been used to detect the human input to command the robot. The LSTM network has been implemented using TensorFlow library through Keras high-level interface programmed in Python language.

Task description

The experimental task consists in the execution of mockup tapping operations on all of the holes on the cardboard, following a specific sequence provided to the operator only. Since such sequence is unknown to the robotic system, the human co-worker is to intervene throughout the experiment to assure the desired order of execution. In this context, his/her job is to physically suggest, when needed, the next target point to the robot, through corrective hand-guided interventions on the end-effector. In this scenario, the tapping operation is simulated by a movement of the robot end-effector entering and exiting inside and outside the hole. Figure 8 illustrates a snapshot of the WM structure during the execution of the collaborative task. In this context, dashed-border inner nodes represent abstract tasks and subtasks, while leaves are executable actions (behaviors allocated in

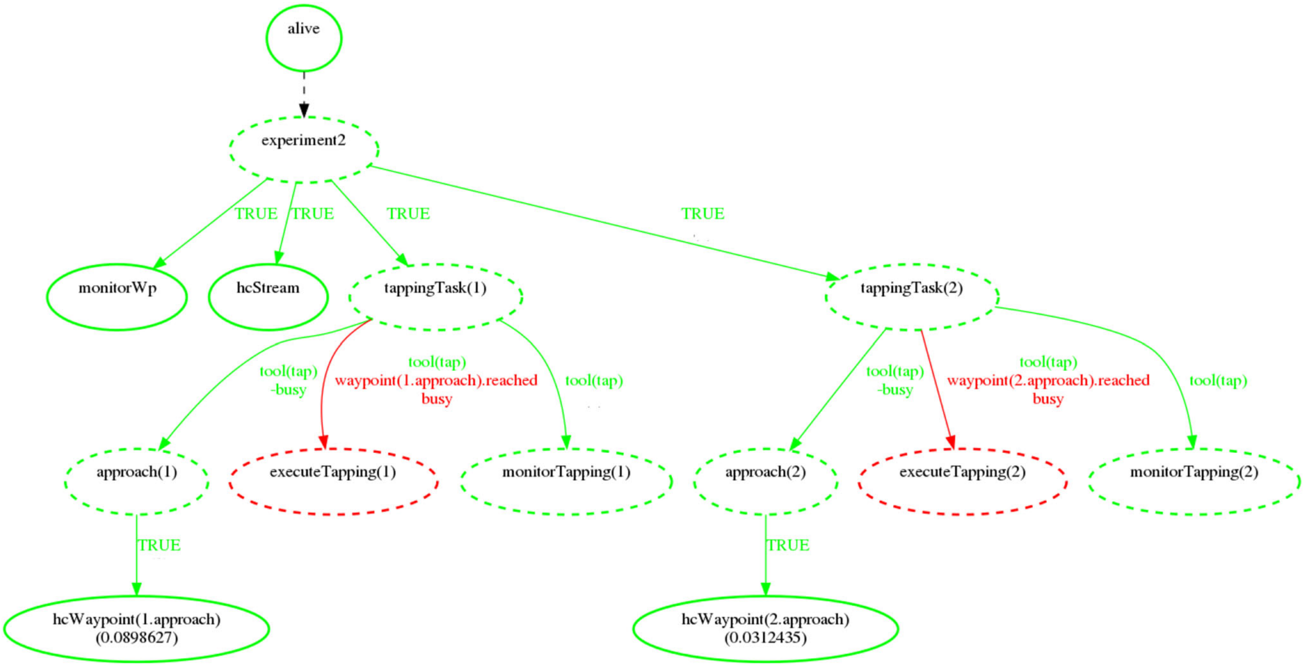

Fig. 8 Working memory during the experiment execution. The green ovals are enabled tasks/subtasks, while the read ones are disabled; dotted and solid ovals are for abstract and concrete activities respectively. In this stage, two tapping subtasks are concurrently enabled and two associated operations (approach waypoint 1 or waypoint 2) are in contention

WM). Here, a node is red when not all of its preconditions are satisfied, and all its subtasks are hence disabled; green nodes are enabled tasks to be executed; edges between nodes capture the dependencies between a task and its subtasks; these are labeled with preconditions (green if satisfied, red otherwise). The overall experiment is represented by a task made up of five subtasks of the type *tappingTask*, each corresponding to the execution of the tapping of a single hole, which is labeled with an ID. The *tappingTask* itself includes the following subtasks: approaching the point where the operation has to take place (e.g., *approach*(1)), execute the required operation (e.g., *executeTapping*), while monitoring of the operation itself (e.g., *monitoringTapping*). The *approach* subtask is dedicated to reaching the point of interest. It is enabled when the tool equipped on the flange is the correct one, i.e., when the *tool(tap)* variable is true, and the robot is not *busy*. The CoBot is considered *busy* when it is in the area of interest of a hole. The *executeTapping* subtask represents the actual execution of the operation. It is enabled when the correct tool is mounted on the robot end-effector and the robot is close to the hole area (*waypoint(ID.approach).reached*). The correct execution of the tapping operation is supervised by the *monitorTapping* subtask exploiting a set of condition-action rules. Finally, in Fig. 8, *monitorWp* monitors when a target area is entered/exited by the robot manipulator, while *hcStream* detects contentions among concrete behaviors in WM.

Experiments

For each experimental session, testers are asked to execute the task in three modalities: *Passive*, *Guided*, and *Proactive*, each associated with a specific robot attitude in selecting and approaching target points depending on the human guidance. In the *Passive* mode, the user is to physically bring the robot end-effector close to the hole, then the robot can autonomously adjust its position and execute the tapping operation. In contrast, in *Guided* and *Proactive* modes, the collaborative system can assist the user during the approach of each target. In this phase, the target intended by the human is recognized exploiting the LSTM-based assessment of the operator's physical guidance. Once the user perceives the robot moving towards a target, he/she can either continue to hand guide the CoBot (to adjust the trajectory or the target) or let the robot approach such point autonomously. In the *Guided* setting, once a subtask (tapping operation) is accomplished, the robot always waits for the user hand guidance to continue the execution towards the next target. Instead, in the *Proactive* setting, the CoBot does not wait for any user intervention and directly proceeds towards the next target enabled by the plan and suggested by the operational context. For each setting, the operator is to exploit his/her hand guidance to induce the robot to execute the tapping operations in the requested order. The sequence follows the waypoints numbering reported in Fig. 7, where consecutive targets at

Table 2 Questionnaire provided to testers

Questionnaire	
Personal information	
Age?	
Gender?	
Level of education?	
How familiarized are you with robotic applications?	
How familiarized are you with cobots?	
Personal feelings	
Safety	How safe did you find the system during the execution of the test?
Comfort	How manageable was using the system?
Intuitiveness	How intuitive was learning how to use the system?
Frustration	How frustrating was interacting with the robot while executing the task?
Utility	How much do you think the system helped in executing the task?
Performance evaluation	
Reliability	How coherent was the robot, compared to your actions?
Readability	How understandable was the behavior of the robot during the execution of the test?
Efficiency	How much do you think the system helped in completing the task faster?
Effectiveness	How much do you think the system helped in completing the task correctly?
Physical demand	How much physical effort was needed to execute the task?
Mental demand	How much mental demand was required to execute the task?
Satisfaction	How much success did you have in executing the task?
Pressure	How much under pressure did you feel while executing the task?

different ranges have been considered to provide a realistic and general layout.

The experiment was carried out by 40 testers, graduate and post-graduate students (with participants' age ranged from 18 to 35), not always acquainted with robotics applications. After each test, we asked testers to fill the questionnaire showed in Table 2, which is inspired by the NASA Task Load Index (NASA-TLX) (Hart & Staveland, 1988) and suitably adapted/extended with typical questions proposed in HRI literature (Steinfeld et al., 2006; Young et al., 2011; Broquère et al., 2014; Chen & Kemp, 2010; Maurtua et al., 2017) to assess the interaction and the performance. A 5-point Likert multi-item scale was employed for the survey.

For each test, we also measured the time to complete the task (*execution time*) and the overall physical effort exerted by users (*overall effort*) during the experiments. The latter has been calculated as the cumulative impulse (force over the time interval) applied to the robot by the operator throughout the experiment.

Results and discussion

In this section, we discuss the collected empirical results including users' performance and users evaluations.

As for performance data, in Table 3, for each modality we report the mean and standard deviation values of the *execution time* and the *overall effort* applied by testers during the experiments. In this respect, data from Table 3 shows that the system's assistance can reduce, on average, the users' *overall effort* by about 50% in *Guided* mode and by almost 70% in *Proactive* mode. Such outcome is strengthened by users' answers illustrated in Table 4, that reports, for each modality, mean and standard deviation of the scores provided by testers. Specifically, even though the execution of the task was considered a little physically demanding in all modalities (physical demand in Table 4), significance results[1] ($p < 0.001$) show that users perceived the effort reduction due to the introduction of the system.

As far as task performance is concerned, in Table 3 we observe that the system assistance seems to not significantly affect the time performances (the slightly positive effect is not significant enough). Nevertheless, users perceived the system's assistance to bring significant improvements in performances: in Table 4 the *Passive* mode is considered one point below the others two modes for both efficiency and effectiveness ($p < 0.001$). Similarly, the *Proactive* and *Guided* setups are generally assessed as more useful than the passive one (utility in Table 4). These results suggest that even in the absence of a significant improvement in time performances, users still have an overall positive feeling about the support provided by the *Proactive* and *Guided* systems during task execution. On the other hand, from the comparison between the *Proactive* and *Guided* modalities does not emerge a significant preference. We expected the *Proactive* mode to be assessed as more useful (utility in Table 4) than *Guided* one, but this is not supported by the significance test ($p > 0.1$). Similar results apply to efficiency and effectiveness.

[1] The Wilcoxon signed-rank test (Wilcoxon, 1945) with Pratt modification (Pratt, 1959) (for zero differences in Liker values) was deployed since the collected questionnaire results are paired and not normally distributed.

Table 3 Data collected during the experiments

Modality	Overall effort (Ns)	Execution time (s)
Passive	97.35 ± 26.48	93.33 ± 9.60
Guided	52.44 ± 22.59	89.87 ± 9.94
Proactive	29.20 ± 12.20	88.44 ± 7.86

Table 4 Questionnaire results: mean and standard deviation values of the users' answers

	Modality		
	Passive	Guided	Proactive
Safety	4.53 ± 0.82	4.58 ± 0.64	4.2 ± 0.88
Comfort	3.88 ± 0.94	4.33 ± 0.66	4.30 ± 0.69
Intuitiveness	4.63 ± 0.74	4.63 ± 0.59	4.60 ± 0.67
Frustration	1.85 ± 1.17	1.53 ± 0.91	1.60 ± 1.00
Utility	3.35 ± 1.41	4.33 ± 0.73	4.50 ± 0.94
Reliability	4.60 ± 0.78	4.70 ± 0.46	4.28 ± 0.82
Readability	4.67 ± 0.58	4.74 ± 0.44	4.31 ± 0.86
Efficiency	3.23 ± 1.19	4.15 ± 0.62	4.33 ± 0.83
Effectiveness	3.73 ± 1.36	4.58 ± 0.68	4.50 ± 0.85
Physical demand	2.15 ± 1.37	1.53 ± 0.96	1.33 ± 0.73
Mental demand	1.75 ± 1.08	1.43 ± 0.81	1.45 ± 0.88
Satisfaction	4.35 ± 0.98	4.73 ± 0.51	4.65 ± 0.62
Pressure	1.48 ± 0.85	1.45 ± 0.75	1.5 ± 0.75

Concerning users' experience while interacting with the robot, the collaborative system is mostly considered as safe (safety in Table 4), with a mean value of about 4.5/5, while comfort values show that the robot remains easy to handle even during autonomous motion. The interaction was generally felt a little frustrating, with mean scores of such parameters below 2. These results show that testers are usually not afraid or intimidated by the physical interaction with the CoBot, instead, this is assessed as safe, comfortable, and fluent. Moreover, the system is on average considered as always intuitive to use for each of the three setups. Positive reviews on reliability and readability parameters were given, with mean scores between 4 and 5. The small differences in mean scores for reliability show that, even when the robot moves autonomously, the users feel the robot understands their intentions. The same considerations apply to readability, whose values suggest that users can understand what the robot does, and how it reacts according to human intention. As for the comfort of the interaction, the assisted modes are moderately preferred over the passive one; in this case, the preference of the *Guided* mode with respect to the *Passive* mode seems more pronounced ($p < 0.001$) than the *Proactive* mode preference ($0.001 < p < 0.005$).

In order to determine a preference order between the modalities, total scores were calculated for each of them. Scores given to *negative* parameters, such as frustration and physical demand, have been inverted, while the total user score for a modality has hence been calculated as the sum of his/her associated votes (the maximum total score is 65). Table 5 shows, for each modality, the mean and standard deviation values of the total scores from each user. Such results show that robot assistance is generally preferred and the *Guided* mode is on average considered the best one.

Table 5 Total scores for each execution modality

Mean sum of scores		
P	G	A
53.68±7.69	58.78±4.84	57.75±5.67

To summarize, *Guided* and *Proactive* modes provide a collaborative system that allows the robot to assist the user throughout the execution of tasks, instead of being simply passive while guided. We also observed that such mechanisms reduce the overall human effort, and such improvement has been perceived by the users too. Users have shown not to be afraid of interacting with the robot in the assisted modes and judged the interaction itself as intuitive and not frustrating; in these settings, the robot's behavior has been assessed as understandable and reliable. Finally, overall scores derived from users' questionnaires, suggest, as expected, that robot-assisted modalities are preferred by users, likely because they perceive the effort reduction and appreciate the robot to work as autonomously as possible for the completion of the task. Between the two robot-aided modalities, the *Guided* one has generally obtained better scores. Looking at Table 4, it is observable that the *Proactive* mode on average wins at the task level, since it is considered more efficient and useful, and less physically demanding. On the other hand, the *Guided* mode has been judged generally more safe, reliable, readable, and satisfactory, and users have felt less frustration, pressure, and mental demand. The reasons for such outcomes can be found in the fact that users may dislike not having full control of the system. When the CoBot moves on its own, more attention is needed in supervising the system, and this may stress users. In contrast, during the *Guided* modality, the robot always waits for a user physical input before acting, hence the human working times are better followed, in so enabling a more comfortable interaction.

Conclusion

In this paper, we proposed a human-robot interaction system for CoBots that interprets the human physical guidance during the execution of hierarchically structured collaborative tasks. In this setting, human interventions are continuously assessed with respect to shared tasks to be accomplished at different levels of abstraction, while the robot behavior and compliance are regulated accordingly. In the proposed framework, both human and robot activities are supervised and orchestrated by an executive attentional system, which enables multiple task execution and smooth task switching depending on the environmental stimuli and the operator interventions. Specifically, the executive system is endowed with attention regulation mechanisms affected by task guidance, targets' accessibility, and human guidance. These influences are continuously assessed, combined, and suitably weighted in order to balance the tendency of the system to follow human intentions or autonomously executing the scheduled actions. In this setting, the human guidance is monitored by LSTM networks that classify operator physical interventions with respect to targets and trajectories admitted by the allocated tasks. These classification results are simultaneously exploited by attention-based regulation/selection processes to align adaptive task orchestration with respect to the estimated human intentions.

We illustrated the proposed human-robot collaboration framework detailing the overall architecture, its main components along with the associated interpretation and regulation mechanisms. In order to assess the performance of the proposed system, we designed an experimental setup inspired by an industrial scenario, where a human operator physically interacts with a lightweight manipulator during the execution of multiple tapping operations. In this setting, we carried out a pilot study to evaluate system performance and human-robot interaction in three different setups, namely, *Proactive*, *Guided*, and *Passive* mode, each associated with a different attitude of the robotic system with respect to the task and the human guidance. From performance evaluation, we observed that both *Proactive* and *Guided* assistance can significantly reduce the overall effort of the human operator during collaborative task execution with respect to the *Passive* mode, which is used as baseline. The benefit of the two assisted modes with respect to the *Passive* one is confirmed by the user experience evaluation, which also shows a preference for the *Guided* setting despite the *Proactive* advantage in effort reduction and task execution support. In this respect, the *Guided* mode seems to provide a better balance between natural interaction (more intuitive, readable, reliable, less mentally demanding) and effective task execution (safer, more effective, satisfactory).

Notice that in this work we focused on physical interaction and physical guidance only, as a future work, we are interested in investigating whether in a multimodal interaction setting the proposed assisted modalities can be evaluated differently by users. For instance, visual and audio feedback may provide additional information about the robot state to improve readability, safety, and reliability of the assisted modes. Moreover, gesture-based and speech-based interaction modalities may complement physical interaction to enable a more natural human-robot communication, while enhancing the robustness of intention estimation.

Concerning the limitations of the proposed evaluation, it should be noted that we deployed a laboratory prototype on a mockup scenario to get an initial assessment of the proposed human-robot collaboration modalities involving generic users (non-specialized testers) for the execution of a generic structured task. Since we aim at an intuitive and natural interaction experience, generic users provide valuable feedback. As a future work, we plan to move this technology from laboratory research to industrial scenarios. In this direction, more focused case studies will be designed involving expert workers in the evaluation process. In these settings, the user experience of specialized workers may diverge from the one of generic testers, while technology acceptance in a real workspace is another relevant issue to be considered and investigated.

Acknowledgements The research leading to these results has been partially supported by the projects HARMONY (H2020-ICT-46-2020, grant agreement 101017008), ICOSAF (PON R&I 2014-2020) and HYFLIERS (H2020-ICT-779411).

References

Broquère, X., Finzi, A., Mainprice, J., Rossi, S., Sidobre, D., & Staffa, M. (2014). An attentional approach to human-robot interactive manipulation. *International Journal of Social Robotics, 6*(4), 533–553.

Cacace, J., Caccavale, R., Finzi, A., & Lippiello, V. (2018). Interactive plan execution during human–robot cooperative manipulation. *IFAC-PapersOnLine, 51*(22), 500–505.

Cacace, J., Caccavale, R., Finzi, A., & Lippiello, V. (2019). Variable admittance control based on virtual fixtures for human–robot co-manipulation. In Proceedings of the IEEE international conference on systems, man and cybernetics (SMC 2019), pp. 1569–1574

Cacace, J., Finzi, A., & Lippiello, V. (2019). Enhancing shared control via contact force classification in human–robot cooperative task execution. In F. Ficuciello, F. Ruggiero, & A. Finzi (Eds.), *Human friendly robotics* (pp. 167–179). Springer International Publishing.

Caccavale, R., & Finzi, A. (2015). Plan execution and attentional regulations for flexible human–robot interaction. In Proceedings of the IEEE international conference on systems, man, and cybernetics (SMC 2015), IEEE, pp. 2453–2458

Caccavale, R., & Finzi, A. (2017). Flexible task execution and attentional regulations in human–robot interaction. *IEEE Transactions on Cognitive and Developmental Systems, 9*(1), 68–79.

Caccavale, R., & Finzi, A. (2019). Learning attentional regulations for structured tasks execution in robotic cognitive control. *Autonomous Robots, 43*(8), 2229–2243.

Caccavale, R., & Finzi, A. (2022). A robotic cognitive control framework for collaborative task execution and learning. *Topics in Cognitive Science, 14*(2), 327–343.

Caccavale, R., Cacace, J., Fiore, M., Alami, R., & Finzi, A. (2016). Attentional supervision of human-robot collaborative plans. In Proceedings of the IEEE international symposium on robot and human interactive communication (RO-MAN 2016), pp. 867–873

Caccavale, R., Saveriano, M., Finzi, A., & Lee, D. (2019). Kinesthetic teaching and attentional supervision of structured tasks in human–robot interaction. *Autonomous Robots, 43*(6), 1291–1307.

Carbone, A., Finzi, A., Orlandini, A., & Pirri, F. (2008). Model-based control architecture for attentive robots in rescue scenarios. *Autonomous Robots, 24*(1), 87–120.

Chen, T. L., & Kemp, C. C. (2010). Lead me by the hand: Evaluation of a direct physical interface for nursing assistant robots. In Proceedings of the 5th ACM/IEEE international conference on human–robot interaction (HRI 2010), pp. 367–374

Clodic, A., Cao, H., Alili, S., Montreuil, V., Alami, R., & Chatila, R. (2008). SHARY: A supervision system adapted to human-robot interaction. *ISER, Springer, Springer Tracts in Advanced Robotics, 54*, 229–238.

Colgate, E., & Hogan, N. (1989). An analysis of contact instability in terms of passive physical equivalents. In Proceedings of the international conference on robotics and automation (ICRA 1989), Vol. 1, pp. 404–409

Colgate, J. E., & Hogan, N. (1988). Robust control of dynamically interacting systems. *International Journal of Control, 48*(1), 65–88.

Cooper, R., & Shallice, T. (2000). Contention scheduling and the control of routine activies. *Cognitive Neuropsychology, 17*(4), 297–338.

Cooper, R. P., & Shallice, T. (2006). Hierarchical schemas and goals in the control of sequential behavior. *Psychological Review, 113*(4), 887–916.

Corrales, J. A., Garcia Gomez, G. J., Torres, F., & Perdereau, V. (2012). Cooperative tasks between humans and robots in industrial environments. *International Journal of Advanced Robotic Systems, 9*, 1–10.

De Santis, A., Siciliano, B., Luca, A., & Bicchi, A. (2007). An atlas of physical human–robot interaction. *Mechanism and Machine Theory, 43*(3), 253–270.

Grafakos, S., Dimeas, F., & Aspragathos, N. (2016). Variable admittance control in phri using emg-based arm muscles co-activation. In Proceedings of the IEEE international conference on systems, man, and cybernetics (SMC 2016), pp. 1900–1905

Hart, S., & Staveland, L. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Human Mental Workload Advances in Psychology, 52*, 139–183.

Hoffman, G., & Breazeal, C. (2004). Collaboration in human-robot teams. In Proceedings of the AIAA 1st intelligent systems technical conference, pp. 1–18

Hoffman, G., & Breazeal, C. (2007). Effects of anticipatory action on human-robot teamwork: Efficiency, fluency, and perception of team. In Proceedings of the 2nd ACM/IEEE international conference on human–robot interaction (HRI 2007), pp. 1–8

Johannsmeier, L., & Haddadin, S. (2017). A hierarchical human–robot interaction-planning framework for task allocation in collaborative industrial assembly processes. *IEEE Robotics and Automation Letters, 2*(1), 41–48.

Joseph, L., & Cacace, J. (2018). Mastering ROS for robotics programming: Design, build, and simulate complex robots using the robot operating system, 2nd edn. Packt Publishing

Karpas, E., Levine, S. J., Yu, P., & Williams, BC. (2015). Robust execution of plans for human-robot teams. In: Proceedings of the twenty-fifth international conference on international conference on automated planning and scheduling (ICAPS 2015), AAAI Press, pp. 342–346

Lallement, R., de Silva, L., & Alami, R. (2014). HATP: An HTN planner for robotics. In: Proceedings of the 2nd ICAPS workshop on planning and robotics (PlanRob 2014), pp. 20–27

Maurtua, I., Ibarguren, A., Kildal, J., Susperregi, L., & Sierra, B. (2017). Human-robot collaboration in industrial applications: Safety, interaction and trust. *International Journal of Advanced Robotic Systems, 14*, 1–10.

Nicolis, D., Zanchettin, AM., & Rocco, P. (2018). Human intention estimation based on neural networks for enhanced collaboration with robots. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS), pp. 1326–1333

Norman, D. A., & Shallice, T. (1986). Attention to action. In: Consciousness and self-regulation, pp 1–18. Springer

Park, J. S., Park, C., & Manocha, D. (2019). I-planner: Intention-aware motion planning using learning-based human motion prediction. *The International Journal of Robotics Research, 38*(1), 23–39.

Peternel, L., Tsagarakis, N., Caldwell, D., & Ajoudani, A. (2016). Adaptation of robot physical behaviour to human fatigue in human–robot co-manipulation. In: Proceedings of the IEEE-RAS 16th international conference on humanoid robots (humanoids 2016), pp. 489–494

Pratt, J. (1959). Remarks on zeros and ties in the Wilcoxon signed rank procedures. *Journal of the American Statistical Association, 54*(287), 655–667.

Raiola, G., Lamy, X., & Stulp, F. (2015). Co-manipulation with multiple probabilistic virtual guides. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS 2015), pp. 7–13

Romero, D., Bernus, P., Noran, O., Stahre, J., & Fast-Berglund, Å. (2016). The operator 4.0: human cyber-physical systems & adaptive automation towards human-automation symbiosis work systems. In: Proceedings of the IFIP international conference on advances in production management systems, Springer, pp. 677–686

Shah, J., Wiken, J., Williams, B., & Breazeal, C. (2011). Improved human-robot team performance using chaski, a human-inspired plan execution system. In: Proceedings of the 6th ACM/IEEE international conference on human–robot interaction (HRI 2011), pp. 29–36

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2008). *Robotics: Modelling, planning and control* (1st ed.). Springer Publishing Company, Incorporated.

Sisbot, E. A., Marin-Urias, L. F., Alami, R., & Simeon, T. (2007). A human aware mobile robot motion planner. *IEEE Transactions on Robotics, 23*(5), 874–883.

Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., & Goodrich, M. (2006). Common metrics for human-robot interaction. In: Proceedings of the 1st ACM SIGCHI/SIGART conference on human–robot interaction (HRI 2006). ACM, pp 33–40

Vernon, D., & Vincze, M. (2016). Industrial priorities for cognitive robotics. In EUCognition, CEUR-WS.org, CEUR workshop proceedings, Vol. 1855, pp. 6–9

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1*(6), 80–83.

Young, J. E., Sung, J., Voida, A., Sharlin, E., Igarashi, T., Christensen, H. I., & Grinter, R. E. (2011). Evaluating human–robot interaction. *International Journal of Social Robotics, 3*(1), 53–67.

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