Flexible Scheduling and Tactile Communication for Human-Robot Collaboration*

Riccardo Maderna[a,**], Maria Pozzi[b], Andrea Maria Zanchettin[a], Paolo Rocco[a], Domenico Prattichizzo[b]

[a] *Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Milano, Italy (name.surname@polimi.it).*
[b] *University of Siena, Department of Information Engineering and Mathematics, Siena, Italy, and Istituto Italiano di Tecnologia (IIT) (name.surname@unisi.it).*

Abstract

Human-robot collaboration (HRC) is expected to add flexibility and agility to production lines in manufacturing plants. In this context, versatile scheduling algorithms are needed to organize the increasingly complex work-flow and to exploit the gained flexibility, ensuring the optimal use of resources and the smart management of failures. Moreover, intuitive user interfaces are needed to communicate with the human worker, informing him/her of the next operation to perform. Usually, grounded or wearable screens are used to this aim. Whenever human sight is impaired or needs to be free, other sensory channels could be used as well. In this work, we present a new dynamic scheduler that adapts to the system variability, and a novel way of communicating instructions to the human operators based on haptic guidance. The proposed strategies were applied to a complex assembly task involving three agents and compared to baseline methods with an experimental campaign involving 16 subjects. Results show the clear advantage of using dynamic scheduling over the static one and suggest that a combination of visual and tactile stimuli is a viable and effective solution for displaying instructions in complex HRC scenarios.

Keywords: Human-robot collaboration, Task scheduling, Haptic interfaces,

*This work was supported by Progetto Prin 2017 "TIGHT: Tactile InteGration for Humans and arTificial systems", prot. 2017SB48FP.

**Corresponding author

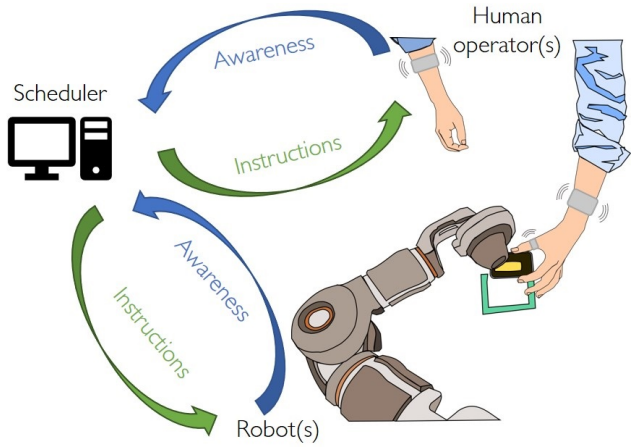


Figure 1: Communication paradigm: the scheduler communicates instructions to the human(s), through haptic interfaces and/or a screen, and to the robot(s). Agents make the scheduling algorithm aware of their status, communicating operation completion and other information, like robot faults.

Assembly

1. Introduction

Human-Robot Collaboration (HRC) has gained increasing attention in the robotics community, as it is expected to add flexibility to manufacturing processes. Today's collaborative robots are machines that are safe to work with, but are still far from fluently interacting with human co-workers [1][2]. On the one hand, versatile scheduling algorithms are needed to organize the complex flow of human and robot operations and obtain a seamless collaboration, going beyond mere workspace sharing [3]. On the other hand, multimodal communication interfaces become essential to guarantee safety while not interrupting the task flow [4] and to allow for a rich exchange of information between the agents [5].

In this paper, a novel dynamic scheduler is coupled with a visuo-haptic interface to perform complex HRC tasks involving multiple human operators and robots. Fig. 1 shows the general principles behind the proposed paradigm.

The most important feature of a scheduling algorithm is its capability of outputting plans that are robust against the uncertainties, while attaining good performance. One way to achieve this is to apply dispatching rules [6], which are based on predetermined heuristics, with the advantage of being easy to apply at runtime. However, when dealing with HRC, high flexibility is required to rapidly adapt to the variability of human behaviour and the occurrence of robot faults. Hence, the need for more complex dynamic schedulers that must work in real-time.

The scheduling algorithm introduced in this work leverages a Digital Twin (DT) of the cell, based on Petri Nets (PN), to predict the future evolution of the system and determine the optimal control action with a receding horizon approach. In particular, the proposed method:

1. Accounts for the variability in the duration of human tasks and the occurrence of robot faults;
2. Allows the concurrent assembly of multiple products to increase productivity;
3. Considers many feasible sequences to complete the assembly. The optimal one for each product is selected in real-time and can change during operation;
4. Dynamically solves on-line both task allocation and sequencing to adapt the plan to the process variability.

To the best of our knowledge, no other work in HRC embeds all these features to allow the highest degree of flexibility, which results in smart use of resources and in the minimization of the overall cell idle time.

In the HRC scenario proposed in this paper, the human agent needs to be informed, through a suitable interface, of the next operation he/she must execute. While previous works on collaborative assembly usually rely only on visual interfaces, i.e., screens on which instructions are displayed through images and text [7], here we propose to combine visual and tactile information, introduc-ing a new method to convey instructions through wearable haptic interfaces.

A general rule is conceived to associate each operation to a vibration pattern displayed either on the left or on the right arm of the user. The wearability of interfaces allows human body parts to move freely and perform the assigned task without difficulty.

The paper is organized as follows. Sec. 2 discusses related works, Sec. 3 describes the proposed dynamic scheduler, Sec. 4 presents two different modalities of giving instructions to the human: through visual cues only or through a combination of visual and tactile stimuli, and Sec. 5 and 6 present and discuss experimental results.

2. Related Works

The emergence of HRC environments results in multiple agents, that are both humans and robots, working together in dynamic and complex workspaces. In [8], an exhaustive classification for HRC scenarios in product assembly with respect to the agents' multiplicity and initiative is discussed. In addition, the authors envision a collaborative environment with intuitive multimodal interfaces and a DT that can be leveraged for monitoring and planning. More in general, a DT supports the entire life cycle of HRC systems by providing a virtual counterpart that can be exploited for fast and reliable validation and control [9].

Effective task allocation and scheduling become crucial to orchestrate the operations of such manufacturing processes. These problems can be taken into account starting from the workplace design phase. For instance, in [10], a multi-criteria decision-making framework is employed for the automatic optimization of the workplace layout through the allocation of tasks to humans and robots. In [11], instead, a method for static task allocation based on task complexity, ergonomics, payload, and repeatability is proposed. In the static scheduling approach, the sequence of operations, and possibly the agent responsible for them, are fixed in advance according to the product to assemble and to suitable decision criteria. Thanks to their simplicity, static schedulers do not require

constant monitoring of agents' operations and are a common choice in manufacturing plants. However, when a high degree of flexibility in the production cell is required, and especially when the HRC paradigm is adopted, static schedulers are not robust to the uncertainties occurring in the system. In these cases, the use of dynamic schedulers is preferable.

To reduce computational load, decentralized market-based approaches are presented in [12, 13], where auctions are exploited for dynamic task allocation. In [14], the authors propose a distributed architecture where each agent ex-ploits a neural network to schedule dynamic operations with real-time sensor data. The use of AI algorithms improves decision-making and learning abilities. In [15], a multi-agent algorithm that combines offline proactive scheduling with online reactive repair according to predetermined heuristics is proposed. In [16], a similar centralised approach for the dynamic scheduling of shared human-robot activities is introduced. Multiple alternatives are evaluated offline to find the best nominal plan, which is then corrected online in case of unexpected events.

A satisficing algorithm for task scheduling in multi-robot teams is proposed in [17], whereas a hierarchical framework for the optimal task assignment and sequencing in human-robot assembly is described in [18]. The latter work does not consider the variability in the duration of operations, nor the possibility of faults, which are instead taken into account in this paper. To account for the uncertainties related to human operations, [3] proposes a scheduling algorithm that leverages prediction of human intentions from visual tracking data, whereas [19] introduces a method to adapt the robot schedule to the operator's preferences by switching between pre-computed alternatives. Similar to the approach followed in this paper, [3, 20] exploit PN to model the assembly process and support the scheduling problem. Also, [21] applies transition-timed PN and A* to search for a near optimal schedule for remanufacturing on parallel processing machines.

The role of human-machine interfaces in achieving effective HRC is often neglected by works that focus on scheduling performance. However, the choice and design of interfaces are crucial when scheduling algorithms are applied in

practice to real HRC scenarios. Suitable interfaces can be used to create a bidirectional exchange of information between the scheduling system and the human operator, ensuring human-aware scheduling of the operations. In [22], a method for the allocation of sequential tasks without concurrency is presented, and human safety is achieved through gesture commands, which are used for starting, stopping, and guiding the robot action. Gestures, however, distract the operator from his/her work and require the continuous tracking of the operator. In this work, instead, we decided to focus on a multimodal interface including visual and tactile displays which do not interrupt or obstruct the operator's motions. While the former are widely used in HRC contexts [7, 23], the latter are less exploited.

The advantages of tactile feedback, however, are several. Previous work showed that it outperforms visual and auditory signals for sensory substitution [24], while a combination of visual-tactile feedback leads to a reduction in reaction times and a better attention allocation with respect to visual feedback alone [25, 26]. In addition, haptic feedback can be employed when other senses are busy or impaired and it has been shown that wearable haptic interfaces can be successfully exploited for spatial guidance in human-robot teams [27], operator awareness in collaborative assembly [28, 29], and command acknowledgement for the control of wearable robots [30].

3. Scheduling Algorithm

In the proposed method, a collaborative robotic cell consists of a set of resources R (humans, robots, tools, shared workspace) used to repeatedly complete a job. A job is a finite set of tasks (or operations) $\Omega = \{o_1, \ldots, o_{n_o}\}$ required to assemble a product $P = \{p_1, \ldots, p_{n_p}\}$ composed of n_p parts. A task is a fixed sequence of actions that transforms a stable intermediate assembly into another stable intermediate assembly. A typical example is the pick and place of a part onto a partial product. To increase productivity, resources work on multiple jobs concurrently.

Let o_i^k be the i-th task of the k-th instance of the job and t_i^k its start time. Then, the solution to the scheduling problem is the optimal set $S = \{(o_i^k, t_i^k), (o_j^k, t_j^k), \dots\}$ of future tasks that minimizes a cost function J over a planning horizon. The scheduler sends commands to the agents according to the computed schedule, and the single agents are responsible for the actual execution of the operations. In case of tasks assigned to robots, commands from the scheduler start robot programs (with a one-to-one correspondence) that are implemented locally in the robot control unit. In this sense, the scheduler constitutes a higher-level layer of the overall control system and remains agnostic to the specific implementation of each task. This approach makes the control system modular and allows to fully exploit the capabilities of each resource. Also, it makes it easier to modify and upgrade the implementation of specific tasks without affecting the overall control system.

3.1. Job definition

When computing the optimal schedule, constraints among tasks arise from the assembly structure of the product, the workspace layout, and the current state of the production. Thus, a model able to describe the cell state and its evolution is needed, which is applicable to a wide variety of use-cases.

In general, a given product can be completed following different sequences of tasks. To give a compact representation of all viable assembly plans, one can use AND/OR trees [31], which in this work have been expanded to include information on resource requirements and workspace layout. An AND/OR tree is a hypergraph $H = (N, E)$, where N is the set of nodes and E the set of (hyper)edges. Each node in the graph represents a step in the assembly process. It is uniquely associated with an intermediate assembly, i.e., Work-in-Progress (WIP) part, and the physical buffer where it is stored between operations. A node is thus described by $n = (W, c_B)$, where $W \subseteq P$ defines the WIP part and c_B the buffer capacity. Starting with all parts $p_i \in P$ disconnected from each other, we join them to form the final assembled product P. Thus, leaves of the tree are n_p nodes with $W_i = p_i$ for $i = 1, ..., n_p$, while the root node has

$W_{|N|} = P$.

The graph has an edge for each task. A task is described by $o = (d, R_O, R_F)$, where d is its expected duration and R_X stands for a set (possibly empty) of resources. Namely, R_O collects resources that must be available at the beginning of the task and are needed for its execution, whereas R_F collects resources that are released at the end of it. The two sets can be different, so to account for cases where a resource is kept occupied along multiple tasks and is not released at the end of each one. An assembly operation connects a single parent node $f \in N$ to a set of children nodes $C \subset N$. Each edge is thus defined by $e = (o, C, f)$. For the AND/OR graph to be well-defined, each edge must be such that $W_i \cap W_j = \emptyset, \forall i \neq j \in C$, and $W_f = \bigcup_{c \in C} W_c$, i.e., the set describing the WIP part associated to the father node is the union of the sets of all WIP parts coming from the children nodes.

A feasible assembly sequence for the job Ω is the set $\Omega_S \subseteq \Omega$ of tasks associated to the edges of a minimal sub-graph that includes the root node and all the leaves of the complete graph, i.e., a graph describing a set of operations to obtain the complete product from its base parts. The job is considered to be completed after the execution of all and only the tasks belonging to one of its feasible assembly sequences. Usually, the optimal Ω_S is determined offline based on nominal behaviour. In this work, the scheduler dynamically makes this choice for each product to improve flexibility and adaptation to the current status of the production.

Fig. 2 shows an example of augmented AND-OR graph for a product composed of three parts p_1, p_2, p_3. These are specified in the leaves with the relative buffer capacities. The root represents the complete product $W_6 = \{p_1, p_2, p_3\}$, and the intermediate nodes contain the WIP parts $W_4 = \{p_1, p_2\}$ and $W_5 = \{p_2, p_3\}$. Robot r_1 performs operation o_2, the human operator r_2 performs o_1 and o_4, and a third agent r_3 performs o_3. To assemble the product part p_3 a tool r_4 is needed. Thus, the tasks that operate on p_3, i.e., o_2 and o_3, cannot be performed simultaneously. Overall, in the presented illustrative example, two assembly sequences are possible, namely (o_1, o_3) and (o_2, o_4).

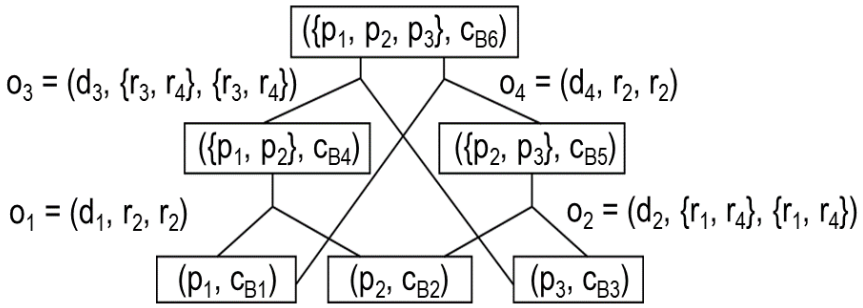


Figure 2: Example of augmented AND-OR tree.

3.2. Cell model

The AND/OR graph allows us to define the job structure in a simple, yet rigorous, way that is suitable also for non-expert users. However, it cannot be used to track the state of the robotic cell and the production, as it does not include state or timing information. As a consequence, AND/OR graphs are unable to model the concurrency of several products in the assembly cell. Given the augmented AND/OR graph description of the job, it is possible to automatically build a partially controllable PN model that can be seen as the DT of the complete assembly cell. The PN describes both the physical structure of the workspace and the evolution of the assembly process, considering *i)* the state of all concurrent WIP products, *ii)* the situations originating from human actions, and *iii)* the occurrence of robot failures. During operation, the DT tracks the state of the workspace in real-time based on information coming from the robot controllers and the human interfaces (see Sec. 4).

A PN is a tuple $\mathcal{M} = (\Pi, \Theta, I, \boldsymbol{m}_0)$, where Π is the set of places, Θ the set of transitions, I the $|\Pi| \times |\Theta|$ incidence matrix and $\boldsymbol{m}_0$ the $|\Pi|$-dimensional initial marking vector. In the PN, the presence of many tokens allows for modelling a composition of individual states related to products, resources and operations. A resource is modelled by a place that is marked when the resource is available. A buffer comprises two places, to track free and occupied slots, with the sum of the markings equal to the buffer capacity. The basic structure of operations

is composed of a start and a stop transition plus a place that is marked as long as the operation is in progress. Moreover, additional n_F^{Π} places and n_F^{Θ} transitions are included to model possible robot failures and associated recovery actions. Several operation places can be marked at the same time to model the simultaneous execution of independent operations by different resources.

The cardinality of the set of places and that of transitions is:

$$|\Pi| = 2|N| + |E| + |R| + n_F^{\Pi} \qquad |\Theta| = 2|E| + n_F^{\Theta}$$

The set Θ is partitioned as $\Theta = \Theta_C \cup \Theta_U$, $\Theta_C \cap \Theta_U = \emptyset$, with Θ_C the set of controllable transitions, whose firing is decided by the scheduler, and Θ_U that of uncontrollable transitions related to exogenous events. In this work, controllable transitions determine the start of new tasks, whereas uncontrollable ones relate to tasks completion and faults occurrence.

The evolution of the state of the cell is tracked in real-time through subsequent transitions firings. Given the $|\Theta|$-dimensional transition vector θ_k having in the j-th position the number of times transition j is to fire, the updated marking vector is computed as $\boldsymbol{m}_{k+1} = \boldsymbol{m}_k + I\theta_k$.

Fig. 3 shows a PN that models a simple cell with three tasks (o_1, o_2, and o_3) performed by two resources (r_1 and r_2, highlighted in red). Both resources can take a WIP part of type W_2 from the input buffer and complete a task to obtain a W_4 part, while the first resource can also assemble a W_3 part by combining W_1 and W_2 parts. For each buffer W_i, tokens in the upper place indicate available parts, tokens in the lower place indicate the number of free slots, and their sum is equal to the buffer capacity. Based on the initial marking (Fig. 3a), all the controllable transitions (θ_1, θ_2, and θ_3) are enabled: at least a WIP part is in the input buffer, and the resources are free (marked place). If, for instance, θ_1 fires (Fig. 3b), a marked place indicates that the operation o_1 is ongoing. Also, one place of the input buffers is freed and the resource r_1 becomes busy. While o_1 is in progress, task o_3 can start in parallel, since another W_2 part is present in the buffer. Conversely, θ_2 is no longer enabled, as r_1 is busy. The uncontrollable transition $\bar{\theta_1}$ fires upon task completion (Fig. 3c): the operation place empties,

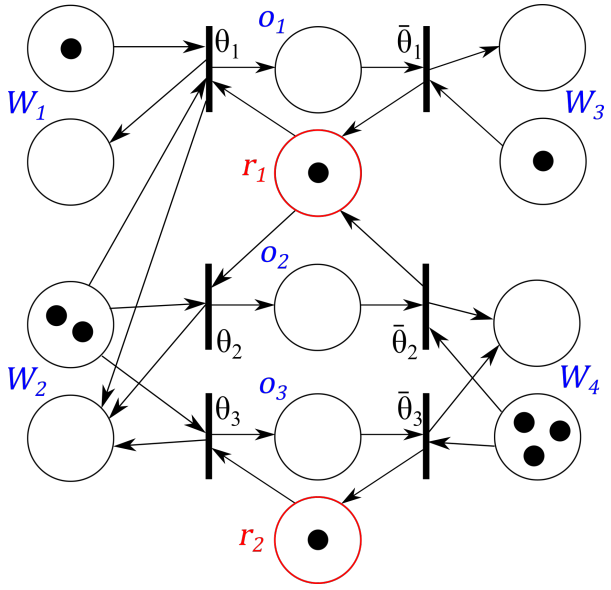
the resource returns available and the output buffer contains the W_3 part.

3.3. Scheduling problem solution

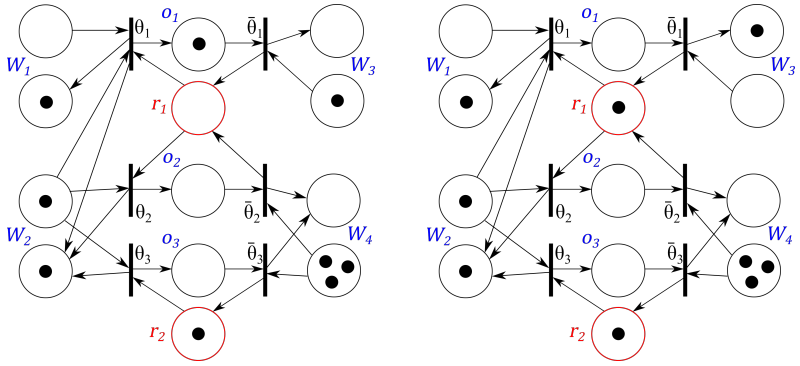
The solution to the scheduling problem is based on an MPC-like approach. The underlying idea is to predict the evolution of the system over the planning horizon from the current state to determine the control input sequence that minimizes a cost function. The exact form of the cost function depends on the specific application. Usually, minimization of the production cycle time is the main objective in assembly processes, but other considerations can be included, such as penalization of energy consumption or a measure of the level of human effort. The value of the planning horizon, which is expressed in terms of the number of operations in the sequence, is guided by a trade-off between computational load and performance. At each step, only the first command is applied and a new prediction starts from the new state of the system. The receding horizon paradigm allows the schedule to adapt to the natural variability of the process (e.g., task durations) and unforeseen events, such as faults.

To simulate the future evolution of the robotic cell, and the firing of uncontrollable transitions in particular, a temporal characterization must be included in the model. Therefore, a Timed Petri Net (TPN) description is adopted for the prediction phase [32]. In a TPN, each transition $\theta_i \in \Theta$ can only fire with a delay of at least d_i seconds after it has been enabled. Specifically, controllable transitions have zero delay, uncontrollable transitions linked to task completion have delay equal to the expected task duration, while fault-related transitions have infinite delay, i.e., are prevented to fire, as they describe an abnormal behaviour of the system.

The initial state of the TPN is defined by the current marking $\boldsymbol{m}_k$ and the remaining time to completion of ongoing operations, i.e., the initial delay of the associated uncontrollable transitions. This information reflects the current state of the real system and is retrieved by monitoring the workspace with the help of a database of past task durations, which is updated each time a new one is completed.



(a)



(b) (c)

Figure 3: Example of PN DT for a cell with two resources (in red) and three operations. Initial state (a), state after θ_1 fires (b) and state after $\bar{\theta}_1$ fires (c).

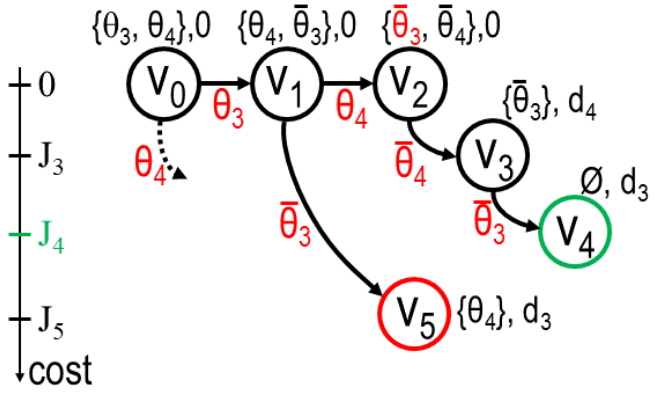


Figure 4: Partial RT for the product in Fig. 2 (Hyp. $d_3 > d_4$). Nodes are numbered in order of exploration, labels indicate enabled transitions and arrival time. Costs are computed from eq. (1). $\bar{\theta}_3$ is enabled in v_2 but would lead to an unfeasible state, thus the related branch is not explored. Expansion stops at v_5 as its cost is higher than the current best J_4.

As already stated, the decisions of the scheduler are limited to the firing of controllable transitions. Instead, uncontrollable transitions always fire with the minimum delay. In addition, since an unnecessary delay of controllable transitions increases the idle time of resources, the only decision at each step is either to fire one enabled controllable transition immediately, or wait for the firing of an uncontrollable one. The latter choice means waiting for the end of an ongoing operation, that can enable new tasks. If several uncontrollable transitions are enabled at the same time, the scheduler can only choose to wait for the one with minimum delay. Otherwise, the uncontrollable transitions with smaller delay are no longer able to fire at the correct time. This is equivalent to say that the scheduler has to choose one transition among all the enabled ones, that is that the scheduling problem reduces to determining the optimal sequence of transition firings.

From the initial state, feasible system evolutions are found by exploring the Reachability Tree (RT) of the TPN [3, 21]. The compact matrix representation and simple update rule of the PN model allow for an efficient simulation of the system. The RT is a pair (V, A), where V is a set of nodes connected by arcs

in A. Each node v_i represents a reachable state for the TPN and is described by $(\boldsymbol{m}_{k,i}, t_i, J_i)$, that is a marking, the arrival time to the state, and its cost. Arc a_{ij} represents the transition whose firing brings the system from v_i to v_j. A path between two nodes v_i and v_j, referred to as $\langle v_i, v_j \rangle$, can be equivalently described by the set of connected nodes (i.e., states) or the sequence connecting arcs (i.e., transitions).

For a node v_b to be the children of node v_a, the transition associated with a_{ab} must be enabled in v_a. The farthest enabling ancestor of the node v_b, is the node $v_e \in \langle v_0, v_b \rangle$, such that $\langle v_e, v_b \rangle$ is the largest subpath that contains only nodes for which the transition leading to v_b is enabled. Then, the arrival time to the state v_b is equal to $t_b = t_e + \delta_{ab}$, where t_e is the arrival time to the farthest enabling ancestor of v_b and $\delta_{ab} = \max\{t_a, d_{ab}\}$ is the delay with which the transition described by arc a_{ab} fires. A node is a leaf, i.e., has no children, either in case no transitions are enabled in that node, or the expansion depth limit has been reached.

The optimal path in the RT is $\langle v_0, v_* \rangle$ from the root node to the leaf v_* with minimum cost J_*. The cost function minimized in the generation of the optimal schedule is:

$$J(v_i) = J(\langle v_0, v_i \rangle) = k_S t_S + k_I \sum_{j=1}^{|R|} c_j t_j^I + k_W \sum_{j=1}^{|N|} t_j^W \tag{1}$$

where k_S, k_I, k_W are weight coefficients and t_S, t_j^I, t_j^W are functions of the arrival times to nodes in $\langle v_0, v_i \rangle$, which in turn depend on the sequence of fired transitions, i.e., the control inputs. The first term reduces the time t_S of completion of the last operation in the plan. The second term penalizes the idle time of resources, being t_j^I the idle time of resource j in the interval $(0, t_S)$ and c_j the unit idle cost for resource j. Finally, the last term facilitates product flow, being t_j^W the total time a part waits in the j-th buffer.

Pruning strategies are implemented to speed up the exploration of the RT without losing optimality. The tree is explored depth-first to rapidly find a feasible evolution of the system, then new nodes are generated only if their

cost is smaller than the current optimal one. Also, the exploration of duplicate branches is avoided by noting that changing the order of two subsequent controllable transitions generates equivalent evolutions. Fig. 4 shows an example of RT exploration.

The final schedule S is obtained from the optimal path $\langle v_0, v_* \rangle$, as the sequence of controllable transitions along the path and their firing time. Commands are sent to the free resources, then the cell state is updated and the plan recomputed.

4. Communication paradigm

As already mentioned, this paper aims at creating a bilateral communication among humans and robots, mediated by the scheduler described in Sec. 3. The scheduler outputs instructions for humans and robots, while agents communicate their status to the scheduler (see Fig. 1). With this information, the scheduler can continue planning the next operations.

From the robot side, communication is enabled by TCP/IP sockets between the CPU running the scheduling algorithm and the robot controller that stores the robot programs for the single operations. When a robot must perform a new operation, the scheduler sends the associated command to the agent. In turn, the robot sends periodic updates on its status to the scheduler. Specifically, feasible robot states are *i) idle*; *ii) busy* (with the indication of the operation being performed); *iii) failed* (with the indication of the specific error occurred). A transition from *busy* to *free* marks the completion of the ongoing operation, a transition to *failed* marks the occurrence of an error or fault.

From the human side, communication is enabled by the wearable devices shown in Fig. 7a. The ring contains three push-buttons and a 3 mm vibration motor (Precision Microdrives) that are controlled through an Arduino Pro Mini. The same Arduino controls the two 25 mm vibration motors (Precision Microdrives) of the bracelet. The communication between the central CPU and the wearable devices is wireless, thanks to two XBee® RF modules (Digi

International Inc.).

4.1. Human input to the scheduler

While [3, 33] estimate the duration of human tasks through human intention prediction based on visual sensing, here we choose a simpler, yet effective, solution, that does not require continuous monitoring of the operator. The human explicitly communicates the completion of his/her current task by pressing a button on the ring. This allows keeping track of the current process state, as described in Sec. 3. The signal fires the uncontrollable transition that marks the end of the ongoing task, updating the availability of WIP parts and re-sources for the next operation. Besides, the change in the cell status triggers the re-computation of the schedule, which can adapt to the actual duration of the just-finished task. As the human presses the button, a vibration burst (duration = 150 ms, frequency = 200 Hz, amplitude = 0.6g) from the motor of the ring acknowledges that the pressure was correctly recognized. If, by any chance, the human presses the wrong button, different vibration patterns are sent to the user to notify the error. The importance of the acknowledgement has been recognized by the users during the experiments (see Sec. 6) and the wearability of the interface allowed for a prompter user input with respect to using a push-button located on the workbench.

4.2. Instructions from the scheduler to human operators

Two different modalities for displaying information to the human are employed: visual (through images on a screen) and tactile (through vibration patterns transmitted by haptic bracelets). Images with overlaid information (Fig. 5) and vibration patterns (Fig. 6) are automatically displayed to the human as soon as the scheduler has dispatched his/her next operation.

Visual instructions are static images that include the name of the operation and a photo of the workspace highlighting the main stages of the task (e.g., which parts are required and where to place the product at the end). Fig. 5 shows an

Figure 5: Example of visual instruction provided to the operator. Circles highlight the parts required to perform the operation, the arrow indicates the feeder where to put the obtained intermediate assembly.

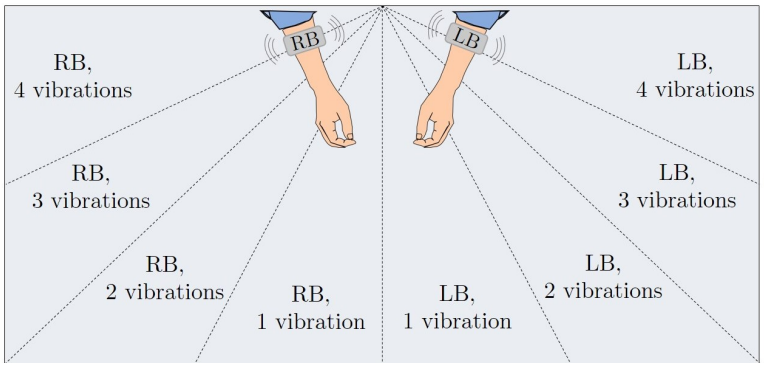


Figure 6: Haptic communication principle: the operator is guided towards the sector of the workspace associated with the next operation that he/she must perform (LB = left bracelet, RB = right bracelet).

example of visual instruction provided to the operator. Different background colours emphasize waiting and fault-related commands to ease identification.

Two vibrotactile bracelets, worn on the two arms, are used to provide tactile cues to the operator indicating his/her next task or the occurrence of unexpected events. The approach we use to send haptic signals depends on the type of information to be transmitted and mixes typical features of spatial guidance paradigms [27] with techniques to send complex messages through *tactons* [34].

For failures, emergencies, or other operations that require particular attention or immediate intervention, a fast sequence of several vibrations is sent. In the experiments, for example, we used 10 vibrations lasting 50 ms with an interval of 50 ms between them. This type of signal was found to be suitable to communicate the importance of an event [35]. Depending on which bracelet is activated and for how long the train of vibrations lasts, different messages can be conveyed.

For assembly operations, i.e., those that are part of the productive cycle and are not responses to unexpected events, the approach sketched in Fig. 6 is followed. When the left (right) bracelet vibrates, the human has to work in the left (right) part of the workspace. The number of vibration bursts indicates the sector associated with the starting point of the operation to be performed, where the user should move the hands. In our experiments, vibration bursts lasted 100 ms, had a frequency of 220 Hz, an amplitude of 0.7g, and were equally spaced of 100 ms.

To the best of our knowledge, this is the first attempt to combine visual and tactile cues to communicate instructions during a HRC task supervised by a dynamic scheduler. The proposed paradigm is easy to learn and general, as it unequivocally associates each operation to a specific tactile cue using a simple rule, and it can be applied to different set-ups, provided that each portion of the workspace corresponds to one operation maximum. Though this may seem a strict constraint, it is well suited to most structured activities, such as industrial assembly tasks, as the workspace layout can be designed accordingly.

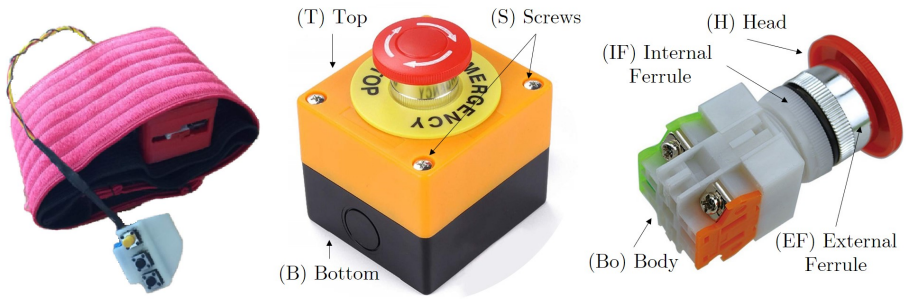


Figure 7: (a) Haptic devices: ring with a vibrating motor and three push-buttons and bracelet containing the controller box and two vibrating motors. (b) Emergency button and its parts.

5. Experimental setup and protocol

The proposed method was tested in a collaborative task where a human operator assembles an emergency button (Fig. 7b) with the help of two robots (ABB dual-arm YUMI and IRB140). The human wears haptic bracelets on both arms and a vibrotactile ring with push-buttons on the non-dominant hand. Tactile interfaces and robots are connected to a CPU, where the scheduling algorithm is implemented. Fig. 8 shows the experimental setup.

The assembly of the emergency button consists of 5 steps:

1. Screw the *Internal Ferrule* onto the *Body*;
2. Position the *Top* and fix the *External Ferrule*;
3. Fasten the *Bottom* with the *Screws*, then fix the *Head*;
4. Pack the completed product inside a *Container Box*;
5. When the *Box* is full, change it for an empty one.

Fig. 9 shows the AND/OR graph that defines all viable assembly plans for the product and the tasks of which they are composed. Edge colours specify the agent that performs the task: YuMi (red), IRB (blue) or human (green). Some steps can be performed independently by more than one agent to increase flexibility. For instance, step 2 is performed either by YuMi or IRB alone, or with

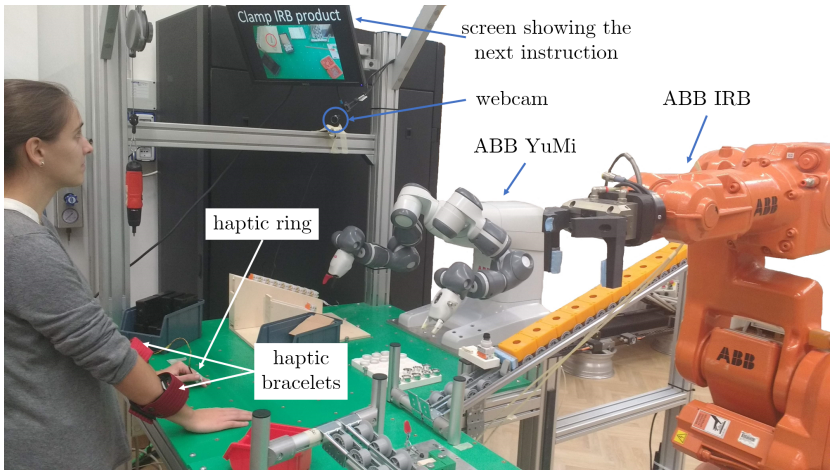


Figure 8: Experimental setup.

a combination of a human and an IRB task. A constraint of mutual exclusion holds among step 2 tasks to avoid the simultaneous access to the buffer that stores the *Top* part, which is shared between the two robots.

Note that when more than one operation per agent is present in the same assembly step (e.g., o_8 and o_{12}), their definition is not identical, as they differ at least with respect to the R_O and R_F sets. As a consequence, each task is a distinct edge in the AND/OR graph, as it is shown in Fig. 9. In turn, each task is described by different places and transitions in the PN. Therefore, when the scheduler explores the future evolution of the assembly process, it automatically determines which operation to include in the plan based on the optimal path in the PN reachability tree.

In total, 12 assembly operations are possible in the considered use-case, for 6 assembly sequences. Also, 5 additional human tasks are added to recover from as many robot faults that may occur in the execution of steps 2 and 4. Table 1 lists the robot tasks, while Table 2 details those available to the human and the associated haptic signals according to Sec. 4.2.

During the experiments, faults have been injected to consistently test the behaviour of the schedulers. Specifically, at the sixth step 2 task performed by

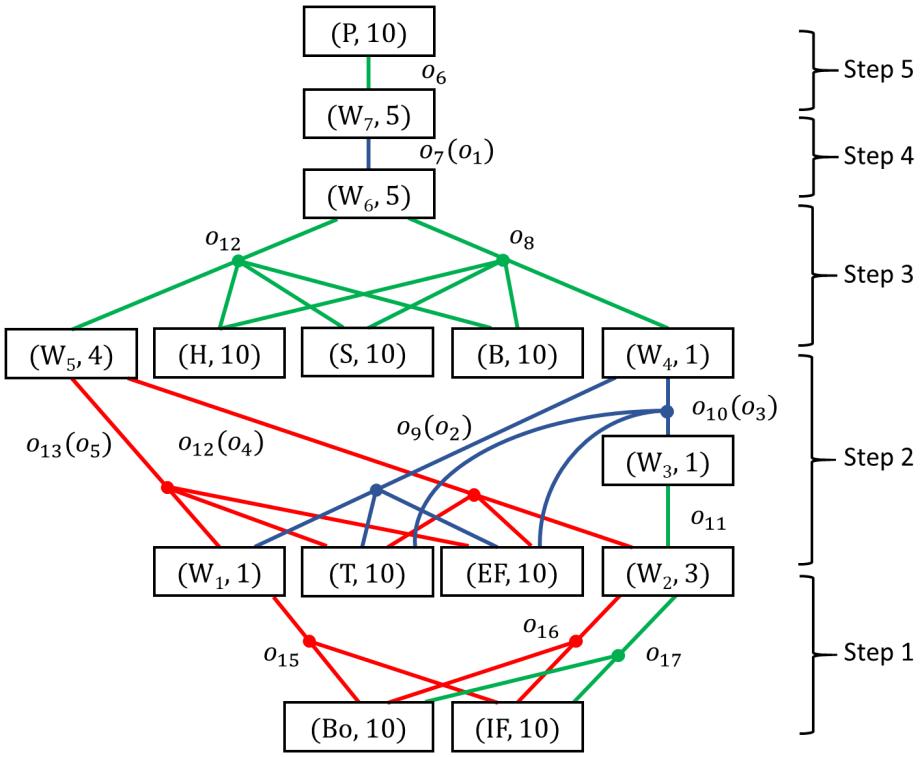


Figure 9: AND/OR graph of the job considered in the experiments. Product part names refer to Fig. 7b. Edge labels identify operations and the associated fault recovery actions (in brackets, if present). Colours specify the agent that performs the task: YuMi (red), IRB (blue) or human (green).

YuMi, the gripper got stuck and was not able to grasp the product, the robot had to be reset by the operator, but the product was still available for processing. To recover from such a failure, a human operator had to use the robot teach pendant to perform a sequence of steps able to restore the gripper functionality and restart the robot program. Conversely, IRB executed the fourth step 4 operation in a wrong way: the robot was still working, but the product needed human intervention to be completed. In this case, the operator had to move to pick up the invalid product, adjust it, place it in the final container, and finally return to his/her working position.

We tested three different conditions: static scheduling with visual feedback

Table 1: List of robot operations.

ID#	Robot	Operation description	Assembly step
15-16	YuMi	Start new product	1
12-13	YuMi	Assemble Top	2
9	IRB	Clamp product & assemble Top	2
10	IRB	Assemble Top	2
7	IRB	Store finished product in box	5

Table 2: List of human operations and haptic signals.

ID#	Operation description	Assembly step	Haptic signal
17	Start new product	1	3 left
11	Clamp product for IRB task	2	2 left
12	Complete product from YuMi	3	1 left
8	Complete product from IRB	3	1 right
6	Change full box	5	3 right
4-5	Recover failed YuMi	-	fast left
1-2-3	Recover failed IRB	-	fast right

(S), dynamic scheduling with visual feedback (V), and dynamic scheduling with visual and haptic feedback (H). In the S condition, the optimal production plan was computed off-line as the sequence of operations that minimizes the cost function throughout the entire trial. To do so, the nominal duration of human operations was used. When a fault occurred during the experiments, the associated recovery action was inserted in the plan as soon as possible, without interrupting the current human task. Other common solutions for fault-tolerant static schedulers, such as reserving slack time or switch to precomputed contingency plans, were deemed inadequate for the specific application.

Instead, the dynamic scheduler introduced in Sec. 3 was employed in V and H conditions. The complete TPN that models the assembly process results in 50 places and 34 transitions. The scheduling problem was solved in less than 0.02s using an off-the-shelf laptop (Intel Core i7-8550U CPU 1.80GHz, 8 GB RAM) with the planning horizon set to 14 tasks. In all three conditions, the next operation for the human was displayed on a screen above the workbench (see Fig. 8). In the H condition, participants also received tactile feedback according to Table 2. The association between haptic signals and operations was learnt by the user before testing the H condition. Participants were instructed to press the button on the wearable ring upon completion of each operation.

Twelve volunteers (3 females and 9 males, age range 23-33) participated in our study. Informed consent was obtained from all of them and the experimental evaluation protocol followed the Declaration of Helsinki. Participants did not perceive any payment and were able to leave the experiment at any moment. None of them had prior experience with the scheduling algorithms, nor with the selected assembly task, nor with the proposed use of haptic bracelets. The authors of this paper did not take part in the study. During a preliminary *training phase*, volunteers 1) were taught how to complete the different human operations and tried each of them twice, 2) performed a complete trial (10 products) to get comfortable with all tasks and associated visual instructions. Then, in the *test phase*, we divided the participants into 3 groups of 4. Each group tested a pair of conditions in different orders. For example, four participants tested

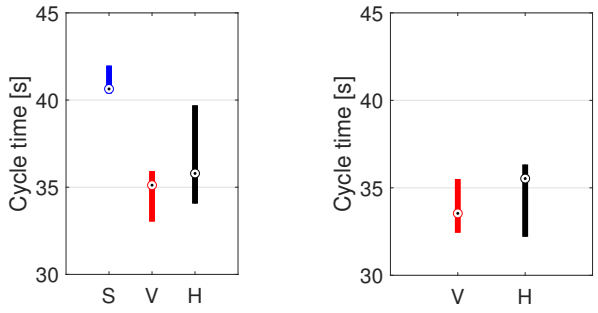


Figure 10: Cycle time obtained for the different conditions (S, V and H) with 12 (a), and 16 (b) participants. Boxplots show median and quartiles.

conditions S and V, two in the order SV and two in the order VS. The same approach was applied to the groups that tested S and H, and V and H. Thus, each participant performed two test experimental trials. A single experimental trial consisted of the assembly of 10 emergency buttons, and participants were asked to finish as quickly as possible. The average duration of a trial was about 7 min.

6. Experimental results and discussion

6.1. Comparison between dynamic and static scheduling

Firstly, we analysed the experimental results based on the cycle time, i.e., the average time between the completion of two subsequent products. This is measured from the end of the first product assembly to the end of the 9^{th} product assembly. The initial and final parts of the experiments have been neglected to remove transient behaviour.

Fig. 10a shows the measured values for the cycle time for the three conditions. The advantages of the dynamic scheduling over the static one are evident: the use of a dynamic scheduler (V, H) attains better performance than the static one (S). For instance, the average cycle time in the V condition, which shares the same communication interface with the S condition, decreases by 15.6%

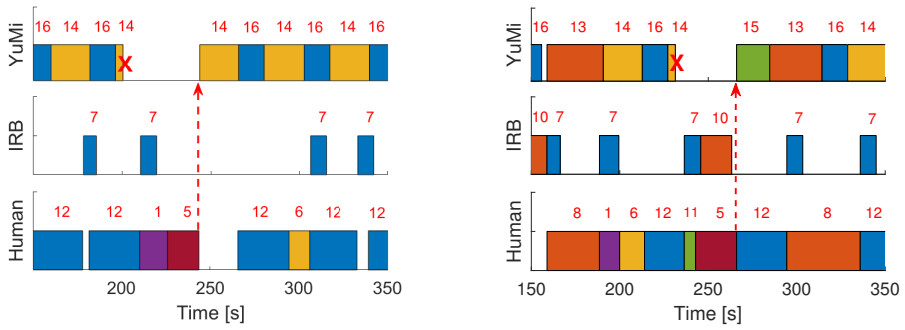


Figure 11: Portions of plans from experiments showing how the static (a) and dynamic (b) schedulers handle a failure of YuMi (red crosses). Arrows mark when the robot is reset. The dynamic scheduler minimizes the overall idle time by delaying the recovery action.

($p < 0.0002$, Wilcoxon test), from 40.6 s to 34.3 s. The improvement is related to two main properties of the dynamic scheduler: it adapts to the actual duration of human tasks and it optimizes the management of the fault recovery actions, which can be delayed to avoid a complete stop of the production, giving priority, e.g., to the preparation of a WIP part for another agent before restoring the failed one.

An example of the latter situation is shown in Figs. 11a and 11b, which compare two typical cases for the S and V conditions, respectively. With the static scheduler, the human recovery action (o_5) is planned as soon as possible after the failure of task o_{14} (at $t \approx 200$ s). However, this leads to a complete stop in the production and long idle times of all three agents, since IRB has no feasible operation to perform. Conversely, the dynamic scheduler inserts task 11 in the human plan before the recovery action to allow IRB to continue the assembly of a product, which is then ready for the human task o_{12}. In this way, the sum of the idle times of all resources is minimized.

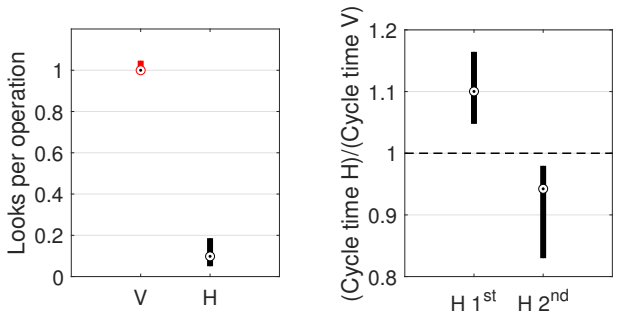
On the other hand, the comparison between conditions V and H is more challenging. The reason is twofold. First, cycle time data show high variability in the H case. This can be due either to the use of haptic interfaces or to the

small amount of available data. Second, since the difference between conditions V and H lies in a change of interface, it is critical to understand how the two approaches influence not only the performance, but also the attitude of the operator. To gather more data for the comparison between the interfaces, the experimental campaign was extended to other 4 novice volunteers (different from those that performed the first 12 tests), for a total of 16. Two of them tried the pair VH, and the other two the pair HV. We obtained the informed consent also for these additional participants and the experimental protocol was the same applied for the other 12. Fig. 10b shows the cycle times achieved with conditions V and H considering all 16 participants. One can notice that the large dispersion displayed in Fig. 10a is no longer present in the results obtained for the H condition and the performance in the two conditions is comparable.

Two other aspects were additionally taken into account, which are discussed in the following: the frequency with which the experimenter directed his/her gaze towards the screen, and the users' subjective impressions on the performed HRC task. The first was measured from videos acquired by a webcam mounted below the screen (see Fig. 8). The second was assessed through questionnaires filled in by volunteers just after their trials.

6.2. The role of tactile communication

Fig. 12a reports the average number of looks the operators gave to the screen per operation. When only the visual interface is present (V case), the average number of looks remains constant, slightly above 1. This is reasonable since the operator had to look at the screen at least once per operation, to know which task to perform. The extra gazes mostly account for occasional double checks. Conversely, when the tactile instructions are introduced, experimenters consistently gave fewer looks to the screen, with a median of 0.1 looks per operation that constitutes a significant reduction over the V case ($p = 0.0010$). Residual looks concentrate during idle time periods, between the end of a task and the reception of the next haptic command, or occasionally to check the received haptic command.



(a) (b)

Figure 12: (a) Average number of looks per operation for V and H conditions and (b) cycle time variation from V to H.

Fig. 12b reports the ratio between the cycle time achieved in condition H and the one obtained in condition V by the same subject. Data show a significant reduction in the H-to-V cycle time ratios between the subjects that tested the H condition first and those that tested it as the last trial ($p = 0.0286$). Instead, the median cycle time in the V condition does not change significantly between the two groups ($p = 0.4857$). Thus, the level of expertise of the operator influences more the performance attained using haptic interfaces and allows more expert users to improve results, with a median reduction of the cycle time of 5.8%.

To obtain information on the participants' subjective opinions, we relied on two questionnaires: one focusing on the acceptance and usage of the haptic devices, the other comparing users' impressions on the two feedback modalities. The four subjects that never tested the H condition were not considered for the survey as they had no experience of the haptic interface, whereas the S and V conditions were considered as equal as they both rely on the visual interface only.

The first questionnaire was composed of 10 questions (Table 3, B1-B5, R1-R5) formulated as five-level Likert items, with answers varying from "strongly disagree" to "strongly agree", similarly to [28]. Results are shown in Fig. 13. Overall, the interfaces were not perceived as cumbersome or obstructing, as only

Table 3: Questionnaires answered by the 12 subjects.

Questions on the haptic interfaces (5 levels Likert items)	
B1/R1:	I found the vibrotactile bracelets/ring very cumbersome to use.
B2/R2:	Vibrations produced by the bracelets/ring are easy to distinguish.
B3/R3:	Wearing the bracelets/ring impedes some movements.
B4/R4:	The vibration is annoying.
B5:	With the haptic feedback I can avoid looking at the screen.
R5:	The tactile acknowledgement produced by the ring is useful.
Questions on the two feedback modalities (7 levels linear scale)	
F1:	It is easy to use.
F2:	Using it is effortless.
F3:	I don't notice any inconsistencies as I use it.
F4:	I can recover from mistakes quickly and easily.
F5:	I can use it successfully every time.
F6:	I learned to use it quickly.
F7:	I easily remember how to use it.
F8:	I am satisfied with it.
F9:	The collaboration proceeded smoothly.
F10:	I complete the operation more quickly when using it.

one person agreed with statements B1 (subject #6) and R1 (subject #11), and nobody agreed with B3 and R3. Vibrations produced by the ring were easily distinguishable and not annoying (nobody disagreed with R2 and nobody agreed with R4), while vibrations produced by the bracelets were perceived as difficult to distinguish by two subjects (#6 and #11), and annoying by the same two people plus another one (#8). Nobody disagreed with statement R5.

The questionnaire to compare the two feedback modalities was formulated as a linear scale going from 1 (only visual feedback) to 7 (visual plus tactile feedback), with statements taken in part from the USE Questionnaire [36] (Table 3,

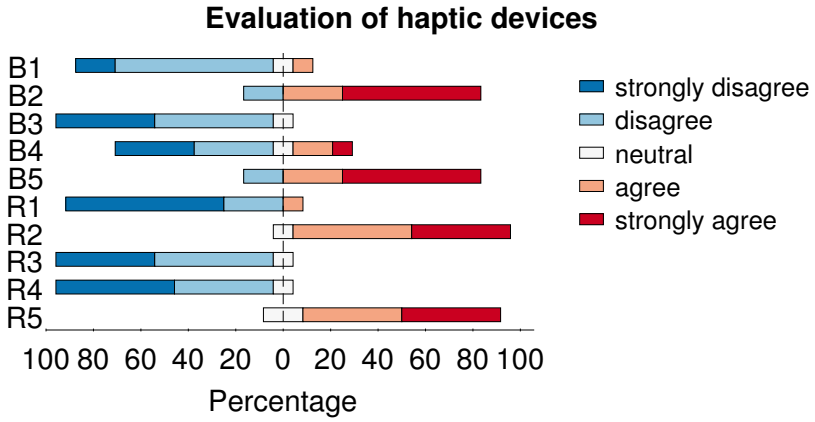
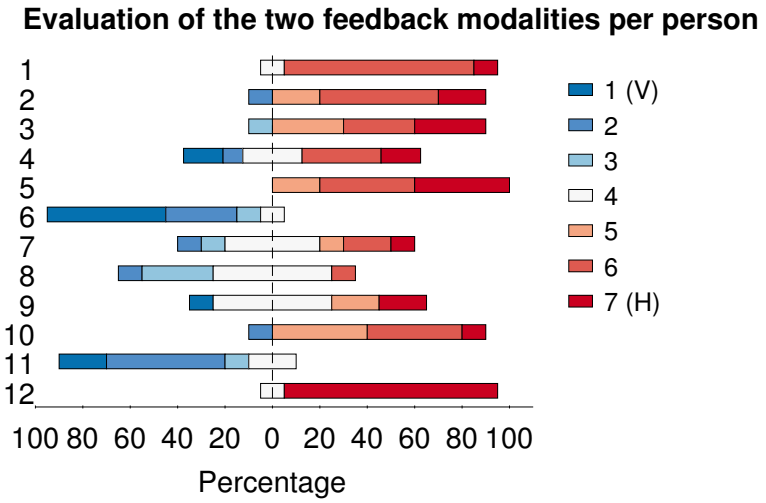


Figure 13: Answers of the 12 subjects that tested the H condition to the questions on the vibrotactile devices (Table 3, B1-B5 for the bracelets, R1-R5 for the ring). Ratings express the level of agreement with the statement according to a five-level Likert scale.

F1-F10). The distribution of the replies of each subject is shown in Fig. 14. By adding the ratings given to all questions, ranging from a minimum score of 10 to a maximum of 70, we distinguished between two attitudes: preference for the V condition (score 10–40, subjects $\#6, 8, 11$) and preference for the H condition (score 41–70, the other 9 subjects). Although some people preferred the V condition, there was an overall preference for the H condition, as 75% of the subjects appreciated the use of haptic feedback. The bottom part of Fig. 14 includes a table summarizing the trials performed by the 12 experimenters that tested the haptic feedback and the percentage of looks given to the screen during the H trial with respect to the other one. Nine out of twelve subjects have a percentage of looks that is $\ll 50\%$, meaning that while testing the H condition they followed the haptic instructions, without the need of looking at the screen.

The relation between the order of execution and task performance has already been discussed, and we noticed that it might have also influenced the acceptance of condition H, as the two most negative evaluations came from subjects that tested the haptic interface in the first trial ($\#6, 11$). Both of them negatively evaluated the haptic bracelets (they disagreed with statement B5) and relied more on the screen than on the haptic interfaces during their H trial,



#	Order	Looks (%)	#	Order	Looks (%)
1	HS	56.8	7	VH	8.7
2	SH	14.5	8	VH	4.8
3	SH	26.4	9	VH	10.5
4	HV	0.0	10	HV	5.0
5	HV	22.2	11	HV	100
6	HS	73.7	12	VH	12.0

Figure 14: Top: answers of the 12 subjects that tested the H condition to the questions on feedback modalities (Table 3, F1-F10) according to a linear scale going from 1 (only visual) to 7 (visual+tactile). Bottom: trials performed by each subject, and percentage of the number of looks in H condition over the condition with visual feedback only.

as indicated by their percentage of looks. As expected, in the subjective evaluation of the interfaces, the acceptance of the feedback devices plays a fundamental role.

Overall, results show that the haptic interface has been appreciated and effectively used by most of the participants, despite the complexity of the task and the variety of commands conveyed through the vibrotactile paradigm. In particular, subjects that tested the H case as the second trial were able to exploit the haptic interfaces to increase performance, showing the crucial role of operators' confidence with the task.

7. Conclusions

This work builds upon the concept of human-robot collaboration, proposing a framework that goes beyond the pure human-robot coexistence, achieving a flexible and human-aware integration of different agents in a work-cell. The contribution is twofold. First, a dynamic scheduler able to plan the operations of multiple agents with on-line adaptation to human completion times and flexible management of failures is introduced. Second, a new tactile communication paradigm based on the association between operations and the portion of the workspace in which they are performed is presented. The scheduler and tactile communication were tested in a realistic HRC scenario through an experimental campaign that involved 16 participants, for a total of about 6 hours of assembly time.

The improvement brought in by the proposed dynamic scheduler with respect to a static one used as baseline is evident: the cycle time decreases by 15.6% when using the dynamic scheduler. This is a consequence of the higher adaptability and optimal fault management of the latter. The differences between the adopted feedback methods (visual *vs* visual+tactile) are more subtle. It was demonstrated that combining vibrotactile and visual inputs to give instructions during HRC tasks is the solution that is preferred by participants, and is viable even in complex scenarios with many agents and operations. In

addition, more expert users tend to rely more on the haptic interfaces, thus avoiding looking at the screen. This allows them to improve their performance.

This paper is a first step towards the understanding of the implications of using vibrotactile feedback in combination with dynamic scheduling for complex collaborative assembly tasks. To advance the described results and ensure the usability and acceptance of the proposed collaboration paradigm in real-world contexts, further analyses on the visuo-haptic interface should be conducted. These include, for example, *i)* studying whether and how the operators' level of expertise influences their preference for a certain type of interface, *ii)* investigating whether and how the use of tactile information during the training of workers can affect their learning curve, and *iii)* determining what is the maximum number of different vibration patterns (corresponding to different operations) that users can discriminate and remember without compromising their performance. Another possible future research direction consists in examining whether the use of haptic feedback can improve the interaction and cooperation between visually-impaired users and collaborative robots.

References

[1] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, J. Pérez-Oria, Working together: a review on safe human-robot collaboration in industrial environments, IEEE Access 5 (2017) 26754–26773. doi:10.1109/ACCESS.2017.2773127.

[2] E. Magrini, F. Ferraguti, A. J. Ronga, F. Pini, A. D. Luca, F. Leali, Human-robot coexistence and interaction in open industrial cells, Robotics and Computer-Integrated Manufacturing 61 (2020) 101846. doi:https://doi.org/10.1016/j.rcim.2019.101846.

[3] A. Casalino, A. M. Zanchettin, L. Piroddi, P. Rocco, Optimal scheduling of human-robot collaborative assembly operations with time petri nets, IEEE Trans. Aut. Science and Eng. (2019) 1–15 doi:10.1109/TASE.2019.2932150.

[4] E. Matsas, G.-C. Vosniakos, D. Batras, Prototyping proactive and adaptive techniques for human-robot collaboration in manufacturing using virtual reality, Robotics and Computer-Integrated Manufacturing 50 (2018) 168–180. doi:https://doi.org/10.1016/j.rcim.2017.09.005.

[5] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Ko-suge, O. Khatib, Progress and prospects of the human–robot collab-oration, Autonomous Robots 42 (5) (2018) 957–975. doi:10.1007/s10514-017-9677-2.

[6] C. Ferreira, G. Figueira, P. Amorim, Optimizing dispatching rules for stochastic job shop scheduling, in: International Conference on Hybrid Intelligent Systems, 2018, pp. 321–330. doi:10.1007/978-3-030-14347-3_31.

[7] B. Sadrfaridpour, Y. Wang, Collaborative assembly in hybrid manufacturing cells: An integrated framework for human-robot interaction, IEEE Transactions on Automation Science and Engineering.

[8] X. V. Wang, Z. Kemény, J. Váncza, L. Wang, Human–robot collabora-tive assembly in cyber-physical production: Classification framework and implementation, CIRP annals 66 (1) (2017) 5–8.

[9] A. A. Malik, A. Brem, Digital twins for collaborative robots: A case study in human-robot interaction, Robotics and Computer-Integrated Manufacturing 68 (2021) 102092. doi:https://doi.org/10.1016/j.rcim.2020.102092.

[10] P. Tsarouchi, G. Michalos, S. Makris, T. Athanasatos, K. Dimoulas, G. Chryssolouris, On a human–robot workplace design and task allocation system, International Journal of Computer Integrated Manufacturing 30 (12) (2017) 1272–1279.

[11] M. Dianatfar, J. Latokartano, M. Lanz, Task balancing between human and robot in mid-heavy assembly tasks, Procedia CIRP 81 (2019) 157–161.

[12] Y. Chen, X. Mao, F. Hou, Q. Wang, S. Yang, Combining re-allocating and re-scheduling for dynamic multi-robot task allocation, in: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2016, pp. 395–400. doi:10.1109/SMC.2016.7844273.

[13] E. Nunes, M. McIntire, M. Gini, Decentralized allocation of tasks with temporal and precedence constraints to a team of robots, in: 2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), 2016, pp. 197–202. doi:10.1109/SIMPAR.2016.7862396.

[14] T. Zhou, D. Tang, H. Zhu, Z. Zhang, Multi-agent reinforcement learning for online scheduling in smart factories, Robotics and Computer-Integrated Manufacturing 72 (2021) 102202. doi:https://doi.org/10.1016/j.rcim.2021.102202.

[15] P. Lou, Q. Liu, Z. Zhou, H. Wang, S. X. Sun, Multi-agent-based proactive–reactive scheduling for a job shop, Int. Journal of Advanced Manufacturing Technology 59 (1) (2012) 311–324. doi:10.1007/s00170-011-3482-4.

[16] N. Nikolakis, N. Kousi, G. Michalos, S. Makris, Dynamic scheduling of shared human-robot manufacturing operations, Procedia CIRP 72 (2018) 9 – 14, 51st CIRP Conference on Manufacturing Systems. doi:https://doi.org/10.1016/j.procir.2018.04.007.

[17] M. C. Gombolay, R. J. Wilcox, J. A. Shah, Fast scheduling of robot teams performing tasks with temporospatial constraints, IEEE Transactions on Robotics 34 (1) (2018) 220–239. doi:10.1109/TRO.2018.2795034.

[18] L. Johannsmeier, S. Haddadin, A hierarchical human-robot inter-action-planning framework for task allocation in collaborative industrial assembly processes, IEEE Robtics and Automation Letters 2 (1) (2017) 41–48. doi:10.1109/LRA.2016.2535907.

[19] R. Wilcox, S. Nikolaidis, J. Shah, Optimization of temporal dynamics for adaptive human-robot interaction in assembly manufacturing, Robotics 8 (2013) 441. doi:10.15607/RSS.2012.VIII.056.

[20] H. Kim, J. Lee, T. Lee, A petri net-based modeling and scheduling with a branch and bound algorithm, in: 2012 IEEE International Conference on Systems, Man, and Cybernetics, 2012, pp. 1779–1784. doi:10.1109/ICSMC.2012.6377995.

[21] S. Peng, T. Li, J. Zhao, Y. Guo, S. Lv, G. Z. Tan, H. Zhang, Petri net-based scheduling strategy and energy modeling for the cylinder block remanufacturing under uncertainty, Robotics and Computer-Integrated Manufacturing 58 (2019) 208–219. doi:https://doi.org/10.1016/j.rcim.2019.03.004.

[22] P. Tsarouchi, A.-S. Matthaiakis, S. Makris, G. Chryssolouris, On a human-robot collaboration in an assembly cell, International Journal of Computer Integrated Manufacturing 30 (6) (2017) 580–589.

[23] S. A. Matthaiakis, K. Dimoulas, A. Athanasatos, K. Mparis, G. Dimitrakopoulos, C. Gkournelos, A. Papavasileiou, N. Fousekis, S. Papanastasiou, G. Michalos, et al., Flexible programming tool enabling synergy between human and robot, Procedia Manufacturing 11 (2017) 431–440.

[24] L. Meli, C. Pacchierotti, D. Prattichizzo, Sensory subtraction in robot-assisted surgery: fingertip skin deformation feedback to ensure safety and improve transparency in bimanual haptic interaction, IEEE Trans. on Biomedical Eng. 61 (4) (2014) 1318–1327. doi:10.1109/TBME.2014.2303052.

[25] A. Sklar, N. Sarter, Good vibrations: Tactile feedback in support of attention allocation and human-automation coordination in event-driven domains, Human Factors 41 (4) (1999) 543–552. doi:10.1518/001872099779656716.

[26] J. L. Burke, M. S. Prewett, A. A. Gray, L. Yang, F. R. B. Stilson, M. D. Coovert, L. R. Elliot, E. Redden, Comparing the effects of visual-auditory and visual-tactile feedback on user performance: A meta-analysis, in: Proceedings of the 8th International Conference on Multimodal Interfaces, 2006, p. 108–117. doi:10.1145/1180995.1181017.

[27] S. Scheggi, M. Aggravi, D. Prattichizzo, Cooperative navigation for mixed human-robot teams using haptic feedback, IEEE Trans. on Human-Machine Systems 47 (4) (2017) 462–473. doi:10.1109/THMS.2016.2608936.

[28] A. Casalino, C. Messeri, M. Pozzi, A. M. Zanchettin, P. Rocco, D. Prattichizzo, Operator awareness in human–robot collaboration through wearable vibrotactile feedback, IEEE Robotics and Automation Letters 3 (4) (2018) 4289–4296. doi:10.1109/LRA.2018.2865034.

[29] K. Katayama, M. Pozzi, Y. Tanaka, K. Minamizawa, D. Prattichizzo, Shared haptic perception for human-robot collaboration, in: I. Nisky, J. Hartcher-O'Brien, M. Wiertlewski, J. Smeets (Eds.), Haptics: Science, Technology, Applications, Springer International Publishing, Cham, 2020, pp. 536–544.

[30] L. Franco, G. Salvietti, D. Prattichizzo, Command acknowledge through tactile feedback improves the usability of an emg-based interface for the frontalis muscle, in: IEEE 2019 World Haptics Conference, 2019. doi:10.1109/WHC.2019.8816133.

[31] L. S. H. de Mello, A. C. Sanderson, And/or graph representation of assembly plans, IEEE Transactions on Robotics and Automation 6 (2) (1990) 188–199. doi:10.1109/70.54734.

[32] D. Lefebvre, F. Basile, Design of control sequences for timed petri nets based on tree encoding, IFAC-PapersOnLine 51 (7) (2018) 218 – 223, 14th IFAC Workshop on Discrete Event Systems. doi:https://doi.org/10.1016/j.ifacol.2018.06.304.

[33] R. Maderna, P. Lanfredini, A. M. Zanchettin, P. Rocco, Real-time monitoring of human task advancement, in: Intelligent Robots and Systems (IROS), 2019 IEEE/RSJ International Conference on, 2019. doi:10.1109/IROS40897.2019.8967933.

[34] L. A. Jones, N. Sarter, Tactile displays: Guidance for their design and application, Human Factors 50 (1) (2008) 90–111. doi:10.1518/001872008X250638.

[35] S. Hameed, T. Ferris, S. Jayaraman, N. Sarter, Supporting interruption management through informative tactile and peripheral visual cues, Proceedings of the Human Factors and Ergonomics Society Annual Meeting 50 (3) (2006) 376–380. doi:10.1177/154193120605000335.

[36] A. M. Lund, Measuring usability with the use questionnaire, Usability interface 8 (2) (2001) 3–6.