Preprint version of the IEEE 2023 European Conference on Mobile Robots (ECMR), Coimbra, Portugal 4 – 7 September 2023

Dynamic Human-Aware Task Planner for Human-Robot Collaboration in Industrial Scenario

Alberto Gottardi[1,2,†], Matteo Terreran[1], Christoph Frommel[3],
Manfred Schoenheits[3], Nicola Castaman[2], Stefano Ghidoni[1], Emanuele Menegatti[1]

***Abstract*— The collaboration between humans and robots in industrial scenarios is one of the key challenges for Industry 4.0. In particular, industrial robots offer accuracy and efficiency, while humans have experience and the capability to manage complex situations. Combining these features can enhance the industrial process by avoiding the user manipulates heavy weights and allowing him to dedicate his efforts to tasks where flexibility, quality and experience make the difference in the final product. However, the collaboration between humans and robots raises several new problems to be addressed like safety, tasks scheduling and operator ergonomics. For example, human presence in the robot workspace introduces various elements of complexity into robot planning due to its dynamism and unpredictability. Planning must take into account how to coordinate the tasks between the robot and the human and be quick in re-planning to respond reactively to the operator's trigger. For this purpose, this work proposes a hierarchical Human-Aware Task Planner framework capable of generate a suitable plan to complete the process and manage user interrupts in order to have a constantly updated plan. The method is evaluated in a real industrial scenario and in a specific complex assembly task like the draping of carbon fiber plies.**

***Index Terms*— Human-Aware Task Planner, Human Action Recognition, Human-Robot Collaboration, Dynamic industrial scenario**

I. INTRODUCTION

Human-Robot Collaboration (HRC) in the industrial scenario is one of the most important technological challenges of recent years. The synergy between the robot's abilities, like precision, accuracy, efficiency and repeatability, along with human intelligence, flexibility and experience provides several advantages because it reduces the operator's effort and improves ergonomics during the operations, ensures the production quality and accuracy [1]. To be able to fully benefit from these advantages, while at the same time ensuring user safety when working with the robot, intelligent task coordination between humans and robots is required. A task planner takes over this intelligent coordination of activities. Moreover, this implies that it is necessary to use a planner that takes into account the user.

† Corresponding Author
[1] Intelligent Autonomous System Lab, Department of Information Engineering, University of Padova, 35131 Padua, Italy. gottardial, matteo.terreran, ghidoni, emg@dei.unipd.it
[2] IT+Robotics srl, 36100 Vicenza, Italy. nicola.castaman@it-robotics.it
[3] Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Augsburg, Germany. christoph.frommel, manfred.schoenheits@dlr.de
979-8-3503-0704-7/23/$31.00

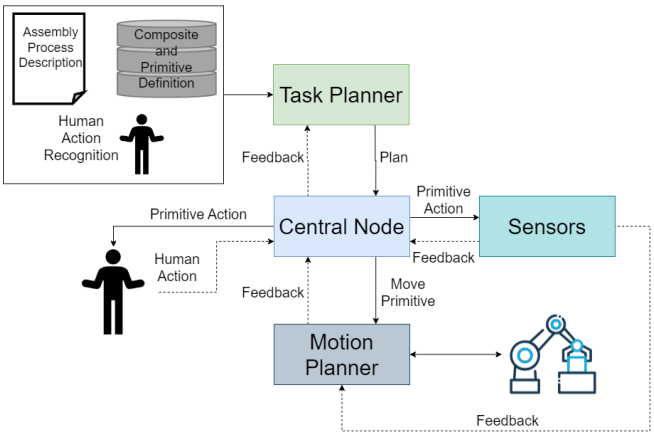


Fig. 1: Dynamic Human-Aware Task Planning Framework for Human-Robot Collaboration.

Since a shared industrial environment between humans and robots is a highly dynamic scenario, the classical task planner approaches are infeasible: they assume that the workspace is deterministic, the state is fully-observable, the robot is the only agent that can change the workspace, and actions are instantaneous. Therefore, to be usable in dynamic environments, task planners must deal with unpredictable and partially uncontrollable situations, especially due to human behaviour [2]. Several approaches have been investigated over the last few years to handle the dynamic scenario: from Artificial Intelligent Planning like PDDL [3] or Markov Decision Process [4] through Finite State Machine [5] to timeline-based approaches [6]. Multi-level programming [7] and Task Allocation [8] solutions are also very common approaches. Finally, game-theory and Reinforcement Learning (RL) models and methodologies are widely applied to multi-agent task scheduling problems [9], [10].

This paper proposes the Dynamic Human-Aware Task Planner framework for HRC in an industrial scenario summarized in Fig. 1. In particular, it focuses on the dynamic scheduling of shared human-robot activities within a manufacturing environment where humans and robots have to collaborate to complete complex tasks like object sorting, production line assembly [11] or draping [12].

Draping is one of the most complex operations in carbon fibre manufacturing. It is carried out by transporting the carbon ply onto the mould and adapting its shape to the mould. Nowadays, this process is completely manual and performed by expert human operators. In addition, the human operator is in charge of transporting the plies from a table

to the mould and then draping it. The EU project DrapeBot[1] aims at developing an HRC system capable of assisting an operator working on the draping of carbon fibre parts. In order to manage that process, a collaborative task planner must be used.

To address the requirements outlined above, we propose a hierarchical task planner that exploits the symbolic description of the process, the definition of Primitives and Composites actions and human commands to generate a suitable and continuously updated plan for the assembly process. In detail, our contribution follows:

- Dynamic Human-Aware Task Planner framework, which is able to compute human and robot activities and handle the user commands to update the plan.
- Using primitive actions to create more complex, so-called composite actions, which contribute to the creation of the final plan.
- Intelligent Action Recognition to trigger activities or robot behaviour not foreseen in the plan but which the expert user wants to perform.

II. RELATED WORKS

In recent years, task planning problems for HRC have been investigated. Existing works like [13], [14] tried to explore the knowledge encoded in the CAD model to extract the product's assembly sequence. Other works focused on sub-problems such as scheduling human and robot actions through Petri Nets [5], [15], or cooperative planning at a symbolic level [16], [17]. These approaches work better only in a classical static environment. Indeed, they cannot handle dynamism, uncertainty and the possibility of the user triggering unforeseen actions as a Human-Aware dynamic scenario requires and as proposed in our approach.

Nikolakis et al. [18] proposed a hierarchical method based on multi-criteria decision-making for an offline task allocation and a dynamic replanning due to unexpected events. Related works which used multi-criteria decision-making framework are [19], [20]. In these works, the authors considered robots and humans as resources. They developed task allocation approaches capable of handling unexpected events but not capable of handling specific commands/actions desired by the user. An unexpected event can be considered as a generic trigger where different events could correspond to a generic reaction. A user's command, instead, is a specific trigger, i.e., each command corresponds to a specific reaction. However, this part is crucial because the operator is an important subject inside the process, and his ability is fundamental to improve the process. Our method proposes solving this gap using the action recognition module connected to the task planner.

Graph-based approaches are described in [21], [22]. The modelling of the process takes place via AND/OR graph that can handle the parallelism of two actions assigned to two different resources. However, they cannot handle the order of precedence constraints typical of assembly tasks and how our approach aims to address and resolve. In other work, instead of using graph-based approaches, the authors exploit the advantages of the Behaviour Tree (BT) [23], [24]. In particular, Lamon et al. [24] have combined the BT approach with a Mixed-Integer Linear Programming (MILP) based role allocation method that allows individual and collaborative roles within the same formulation. However, human uncertainty is not modelled and considered. But, an intelligent system has to consider human intentions in its decision-making rather than force the operator to follow a strict, predefined assembly plan. In our proposed method, the operator can directly interact with the system and force the robot to execute some tasks by the action recognition module. Human intentions are typically modelled through the Partially Observed Markov Decision Process (POMDP) [25]. Approaches which shared similarities with Cramer et al. [25] modelled the collaborative task like hierarchical task network (HTN) [26], [27], [28]. The latter directly employs first-order logic to enable the robot to estimate its partner's goals and anticipate correctly in the presence of human variability and non-deterministic sensing. Another work related to the HTN is [29], where the task planner is able to divide the plan into multiple streams for multiple agents, humans included.

Most of the approaches described above have an implicit representation of the time. Actions are supposed to be instantaneous so that the action effects become true when the action itself is applied and changes the environment or the situation. Therefore, states and goals are not supposed to have a temporal extension such that they hold only for a limited temporal interval, or that they must be achieved within known temporal bounds. The planners that follow this approach are temporal planning, and the main feature is that they synthesize plans by combining causal reasoning with time and resource reasoning [30], [6].

In [6], Umbrico et al. proposed a timeline-based planner called PLATINUm with the ability to deal with temporal uncertainty at the planning and plan execution levels. The same authors then improved that tool by proposing an evolution of it, called TENANT [31], capable of setting objectives, defining tasks and establishing operational constraints, despite the inherent complexity required in planning and robotics. Although these works are evaluated in industrial applications, these approaches are not able to be adapted or rescheduled based on real-time observations by the operator. Indeed, adapting plans on the fly can be difficult, especially when the original plan heavily relies on strict time constraints like in these approaches.

Finally, significant advances in Deep Reinforcement Learning (DRL) have been witnessed in many outstanding large-scale sequential decision-making problems [10], [32]. Hu et al. in [33] exploited the combination of timed-place Petri nets with the deep Q-network with GCN to manage the dynamic scheduling problem of an industrial manufacturing scenario. In the same application, Kim et al. in [34] proposed the RL approach where intelligent agents evaluate the priorities of jobs and distribute them through negotiation.

[1] https://www.drapebot.eu/

Fig. 2: Example of a plan with composite and primitive actions.

III. SYSTEM ARCHITECTURE

This section presents the Dynamic Human-Aware Task Planner framework that handles the entire process and human-robot planning. Fig. 1 depicts an overview of the system proposed.

The Task Planner module coordinates the human and robot activities. It is responsible for creating a continuously updated plan that serves as a guideline for the workflow and will be composed of the sequence of actions to achieve the assigned task. Finally, the Task Planner must manage the human intentions to adapt the computed task plan to meet the collaboration needs dynamically or to handle unexpected situations and use recovery actions to return to a safe state. A simple example of the plan for a draping process is shown in Fig. 2 where it is composed of composite action like *Transport* that represents the activity to transfer the carbon fibre ply from a picking table to the mould, and primitives like *Draping* and *Inspection* that represent the actions performed by the human operator that drapes the ply into the mould and checks that no defects have formed during the previous activity.

The second important module is Action Recognition, which recognises the gestures associated with triggering specific actions. The associated command is sent to the Central Node, which is the module that monitors the operation of the system and sends the commands to the other modules that are in charge of performing the action in the plan. The Central Node manages the information the sensors provide in the workcell. Finally, a state-of-art Motion Planner is involved in order to generate a collision-free trajectory for the robot.

A. Task Planner

The task planner structure is outlined in Fig. 3 and is developed following a hierarchical approach consisting of 3 different layers:

- A low layer consisting of Primitive Actions.
- A middle layer consisting of Composite Actions.
- A high layer consisting of the Plan of the entire process.

Each layer has its own distinctive characteristics and a different level of abstraction with respect to the final task.
In the lower layer, we have the *Primitive actions* which represents the activities to be performed (e.g. *Move*, *Draping*, *Inspection*, etc.). The robot and the human alone could execute this activity, or both agents are required. A series of preconditions and effects characterise a primitive action. The preconditions are verified directly by the primitive itself, while the effects describe the state changes. When one of the preconditions is not satisfied, the current state is invalid and the primitive cannot be sent to execution, i.e. the related activity cannot be performed. Therefore, the primitive itself notifies the Central Node that the state is invalid. The central

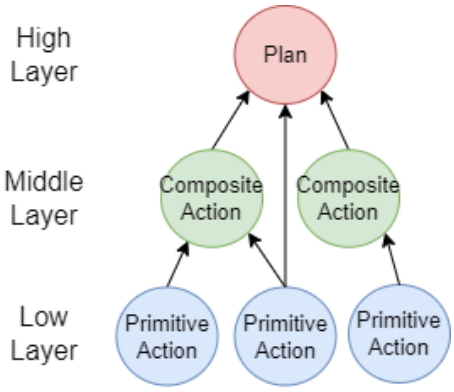


Fig. 3: Hierarchical structure of the Task Planner: the primitive actions are in the lower layer, the composite actions are in the middle layer and the final plan is in the highest layer.

node triggers the related recovery behaviour in order to return to a valid state. A recovery behaviour is a specific and simple action, strictly related to the primitive itself, that is responsible for returning the system to a valid state, thus allowing the process to continue while limiting external intervention to a minimum. For example, if a robot's gripper fails to grasp an object, recovery could be deactivating the gripper and retrying the grasping operation, i.e., reverting the state, performing the object detection again and re-evaluating the precondition primitives for the grasp action. Another more complex example could be the *Piece-Detection* primitive, which involves localizing the object on the pick-up table and providing a suitable grasping point. If the algorithm does not find the desired object, two recovery behaviours can be activated: the first starts a second scan of the table by moving the camera in a slightly different position; the second, if the process allows (i.e. without violating precedence constraints) searches for the next object to pick-up. The action associated to recovery behaviour is defined as a primitive, with its precondition (if needed) and effects. After the intervention of a recovery behaviour, it is verified whether the current plan is still valid and whether the preconditions of that action are now valid. If this happens, the primitive is re-executed, otherwise a re-planning is required. A set of specific configuration files defines the primitives and their structure.

In the middle layer, we have the *Composite actions* defined as a logical sequence of primitive actions. The composite has both preconditions and effects, corresponding to the first and last primitives, respectively. Similar to the primitives, the composite has a set of recovery behaviours triggered when a transition between one primitive and the next fails. The sequence of the primitives is defined in an external configuration by an expert operator who has to collaborate with the robot in the workcell. The definition of primitives and composite actions are provided in input to the Task Planner as shown in Fig. 1. The symbolic language used to model the primitives is the PDDL [35] because its action is precisely defined by a set of parameters, preconditions, and effects required by the Task Planner. Fig. 4 depicts an example of the *Transport* composite action which includes the primitives

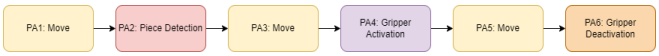


Fig. 4: Example of the Transport Composite Action (CA) as the sequence of Primitive Actions (PA).

Move, *Piece-Detection*, *Activation* and *Deactivation* of the gripper.

Finally, in the highest layer, we have the *Plan* for the entire process. By construction, this is the most abstract layer and where the definitions of composites and primitives are used together with information from the assembly process, environment and human operator to create the process plan. Assembly applications contain precedence constraints to manage and define the order in which the main components should be mounted. For these reasons, it was decided to use a Direct Acyclic Graph (DAG) approach with weighted arcs to represent all the possible alternative plans. In particular, each node of the graph represents an action (primitive or composite), while each arc represents the dependencies that must be fulfilled. For example, some objects must be placed before others in assembly tasks. Therefore, during the building of the graph, the task planner has to take into account that aspect. For example, as shown in Fig. 5, the *Action 5* must be performed only after *Action 3* and *Action 4*. In addition, each arc has an associated cost representing the effort of the robot and the user respectively in performing the action associated with the transition between the two nodes. The goal is to create a plan that minimises the user's effort and maximises the robot's effort by exploiting the possibility of having the robot perform some actions while the user performs others in the same collaborative workcell. Therefore, using the DAG (Alg1 - line 1) it is possible to find a topological ordering which describes the sequence of actions to complete the process (Alg1 - line 2). However, a DAG may contain more than one valid topological ordering. For this reason, the Depth-first search (DFS) algorithm was used for the topological search and optimised the cost function. The cost function used to calculate effort is the same for user and robot and it is the sum of the weights on the arcs in the DAG in Fig. 5. The mathematical formulation of the cost function follows:

$$C(w) = \sum_{u} w_{i,u} + \sum_{ru} w_{i,ru} - \sum_{u} w_{i,r} \qquad (1)$$

where $w_{i,j}$ represents the weight of i^{th} arc and j represents to which agent the weight is associated, whether to the user (u), the robot (r) or both (ru) when that action is to be performed by the two agents together. In order to optimize the cost function and obtain the plan that minimizes the user's effort, the $\arg\min_w C(w)$ is taken (Alg1 - line 3). In this way, a plan can be found to complete the process and described by the sequence of actions to be performed by the robot and user.

B. Human Action Recognition

The Human Action Recognition module monitors human activities during the collaboration, such as phases of the process (e.g., draping) or particular gestures to provide commands to the robot (e.g., request a new ply). Such information allows the task planner to be constantly updated on the current activities of the human operator: the task planner can periodically check whether the human is still engaged in particular tasks (e.g., draping) or whether through the use of gestures it is requesting specific actions from the robot that require the generation of a new task plan.

The human action recognition module is based on a previous work [36], where a graph convolutional neural network was proposed to recognize common human actions and gestures which arise in a collaborative manufacturing scenario. Such network takes as input a sequence of human 3D poses (i.e., skeletons) and tries to classify human movements according to a set of actions of interest by analyzing both spatial and temporal information.

In this work, human poses are provided by the state-of-

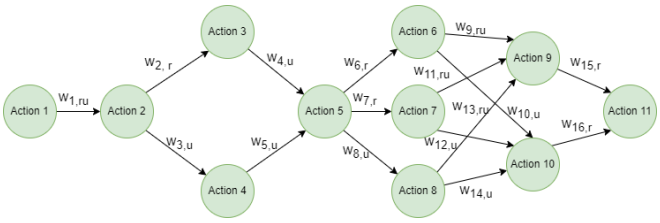


Fig. 5: Plan described by the Direct Acyclic Graph (DAG) where the action is represented by the node and the effort associated by the arc's weight.

Algorithm 1 Task Planner

Input: $\boldsymbol{PA}$ Primitive Action set, $\boldsymbol{CA}$ Composite Action set, $\boldsymbol{P}_d$ Process description, $state$ current state, a action failed, h human command
Output: $\mathcal{P}$ Plan
1: $\mathcal{G} \leftarrow buildDAG(\boldsymbol{PA}, \boldsymbol{CA}, \boldsymbol{P}_d, state, a, h)$
2: $< \mathcal{TP}, C(w) > \leftarrow findAllTopologicalOrdering(\mathcal{G})$
3: $\mathcal{P} \leftarrow \arg\min_w C(w) \in < \mathcal{TP}, C(w) >$
4: return $\mathcal{P}$

Algorithm 2 Central Node

Input: $\boldsymbol{PA}$ Primitive Action set, $\boldsymbol{CA}$ Composite Action set, $\boldsymbol{P}_d$ Process description, $\boldsymbol{H}$ human command set
1: $\mathcal{P} \leftarrow TaskPlanner(\boldsymbol{PA}, \boldsymbol{CA}, \boldsymbol{P}_d)$
2: **for all** $a \in \mathcal{P}$ **do**
3: $valid \leftarrow evaluatePrecondition(a)$
4: **if** $valid$ **then**
5: $state \leftarrow Execute(a)$
6: **else**
7: $(state, valid) \leftarrow RecoveryBehaviour(a)$
8: **if** $valid$ **then**
9: ***go to*** 3
10: **else**
11: $\mathcal{P} \leftarrow TaskPlanner(\boldsymbol{PA}, \boldsymbol{CA}, \boldsymbol{P}_d, state, a, null)$
12: **end if**
13: **end if**
14: $h \leftarrow HumanActionRecognition(), \quad h \in \boldsymbol{H}$
15: **if** h **then**
16: $\mathcal{P} \leftarrow TaskPlanner(\boldsymbol{PA}, \boldsymbol{CA}, \boldsymbol{P}_d, state, null, h)$
17: **end if**
18: **end for**

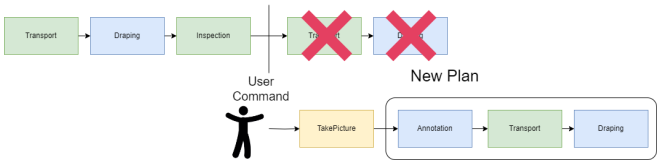


Fig. 6: Example of replanning due to user command.

the-art monocular 3D pose estimator MeTRAbs [37]. Such estimator outputs human poses composed of 19 keypoints describing the main body joints (e.g., torso, arms, legs).

Differently from [36], where actions were recognized using an ensemble of various models specialized for different body parts (e.g., body and hands), in this work we focused only on the body, as during a real manufacturing application the hands are thickly occluded and difficult to estimate accurately.

C. Central Node

The Central Node executes the plan by activating the correct primitives and supervising their execution. In addition, this module is always aware of the state of the workcell through the sensors present in the scene, e.g. a camera network positioned around the robot workcell, laser scanners, etc. A second purpose of the central node is to handle invalid state situations one may find oneself in during the execution of the process plan. When a primitive precondition is not satisfied (Alg2 - line 3), it is notified that the state is invalid and the central node will trigger the corresponding recovery behaviour in order to return to a valid state (Alg2 - line 7). When this happens, the primitive is asked to re-verify whether the preconditions are satisfied (Alg2 - line 9) in order to verify whether the current plan is still valid or if a re-planning is necessary (Alg2 - line 11). In the last case, the central node notified the task planner module that the current plan was unfeasible and a new plan was required. A similar situation occurs when the user wants to make the robot perform a task not foreseen in the plan (Alg2 - line 14). In this case, through the Human Action Recognition system, the user executes a specific command associated with a specific action to be done, be it a primitive or a composite. For example, as depicted in Fig. 6, when the *Inspection* action was finished, the user noticed some defects and decided to take a picture of the area where the defects were present. In order to perform this action, which was not included in the original plan, he performed the specific gesture associated with the composite *TakePicture*. The central node receives this information and sends it to the Task Planner module which is in charge of creating a new plan where the requested action is the first action to be performed. The requested action is treated as a precedence constraint to add to the DAG.

IV. EXPERIMENTS

The Dynamic Human-Aware Task Planner developed in this work was tested in a specific assembly scenario which is the draping of fibre carbon plies. First, we described the draping process and the actions involved, then we evaluated the performance of the Task Planner analyzing the computational time spent to create a suitable plan, considering also the replanning phase, and the performance of the Human Action Recognition system. Finally, a qualitative analysis in a real scenario is provided.

A. Case Study

Draping is a complex industrial operation that requires an advanced skilled user who is not only responsible for draping onto the mould but also for a series of activities such as inspecting the part, noting by text and/or photos of certain areas if they have slight defects, and checking that the orientation of the fibres is correct. The transport of a ply could be executed by the operator or robot alone, or it could be a collaborative transport where human and robot are involved. Thus, the coordination of the activities, like the robot's motion, activation/deactivation of the gripper to perform the pick and place and the detection of the piece in the picking table is crucial.

In addition, the operator could use a gesture to trigger an action which it was not in the original plan (Fig. 6) or to notify the central node that the current action has been completed and to move on to the next one. An example of this situation is when the operator has finished draping the ply and wants to notify the central node so that it can perform the next action. This is a simple and intuitive way for the user to interact with the robots and provide his/her experience into the system. Therefore, analyzing the process and the activities to be done, a set of gestures of interest has been defined based on the possible human interactions which can arise during the process. In particular, a set of 6 human actions has been considered: *Detection*, *Inspection*, *Transport*, *Draping*, *Drape Next* and *Take Picture*.

As described in the previous paragraph, a configuration file is used to represent the precedence constraints and define the order in which the plies are draped. Also, the sets of primitives, composite and gesture are defined in order to perform the entire draping process. Table I provides an overview of the actions used to evaluate the Task Planner, while a detailed list of the human actions of interest and their description is provided in Table II.

Action name	Description
Move	Primitive that represents the motion of the robot and/or the operator
Gripper Activation	Primitive that represents the activation of the gripper to pick up the ply
Gripper Deactivation	Primitive that represents the deactivation of the gripper to place the ply
Piece Detection	Primitive that represents the detection of the plies on the table
Draping	Primitive that represents the draping of the ply onto the mould by the operator
Transport	Composite that represents the transport of the ply from the picking table to the mould
TakePicture	Composite that represents the saving of an image of a certain mould area
Annotation	Composite that represents the saving of information by the user operator
Inspection	Composite that represents the inspection of the draping plies onto the mould

TABLE I: Overview of actions used to evaluate the Task Planner.

B. Task Planner Evaluation

The Task Planner was evaluated through the entire draping process using the actions shown in Table I and 20 independent process trials were performed. For each trial, the plan considered the draping of 5 plies. As mentioned above,

Action name	Meaning	Description
Detection	Trigger the Object Detection	Clap with stretched arms above head
Inspection	Stop current robot operation	Raise one hand with a stretched arm
Transport	Collaborative transport	Move while holding one side of the ply
Draping	Manual draping	Drape the ply on the mould
Drape Next	Require next ply	Move the right arm bend 90°
Take Picture	Take a photo of the ply status	Point at the desired location to be framed

TABLE II: Set of actions of interest considered for evaluating the human action recognition module in the proposed case study.

Trial	First Plan Time [ms]	Replanning Time [ms]
1	3.36	3.16
2	3.58	4.08
3	3.79	5.9
4	3.81	4.1
5	2.95	6.79
6	2.14	5.72
7	2.5	5.95
8	2.4	3.43
9	3.52	3.13
10	3.04	3.5
11	2.42	3.49
12	3.86	5.09
13	3.15	3.81
14	3.62	4.18
15	3.35	5.86
16	2.32	3.67
17	2.42	4.37
18	2.61	6.12
19	3.76	6.52
20	2.70	5.56
Average	3.065	4.7215
Validity	19/20	19/20

TABLE III: Time for the first plan (left) and average time for replanning (right) in ms.

the computational time is used to evaluate the performance and the interaction with the user was done by the Human Action Recognition system. Therefore, when the Central Node received the gesture it would either perform the next action in the plan or send a message to the Task Planner that a re-planning was necessary. Task Planner and Central Node were running in a Lenovo ThinkPad with 11^{th} Intel Core i7 processor and 16GB of RAM. The results obtained are summarized in Table III where the time is expressed in milliseconds (ms).

As shown in Table III, the planner demonstrated high efficiency in generating the first plan, with an average computational time of 3.065 ms. However, when the planner had to replan in response to a user's command, it was slightly slower, with an average computational time of 4.72 ms. The minimum and maximum computational times observed were 2.14 ms and 3.86 ms for the first plan, and 3.13 ms and 6.79 ms for the replanning phase, respectively. All the values in the replanning column are the average of all replanning that happened during the trial. In fact, in this way, it is possible to consider when the rescheduling has taken place. If you replan at the beginning of the process, there are a lot of tasks

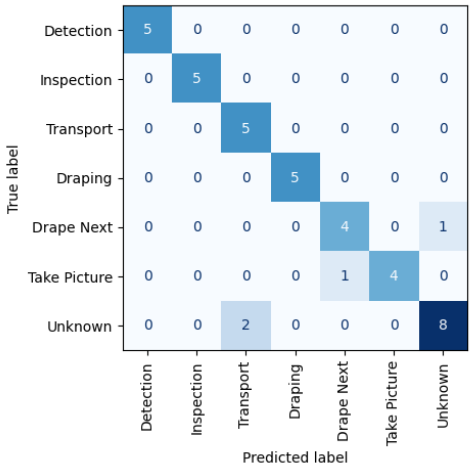


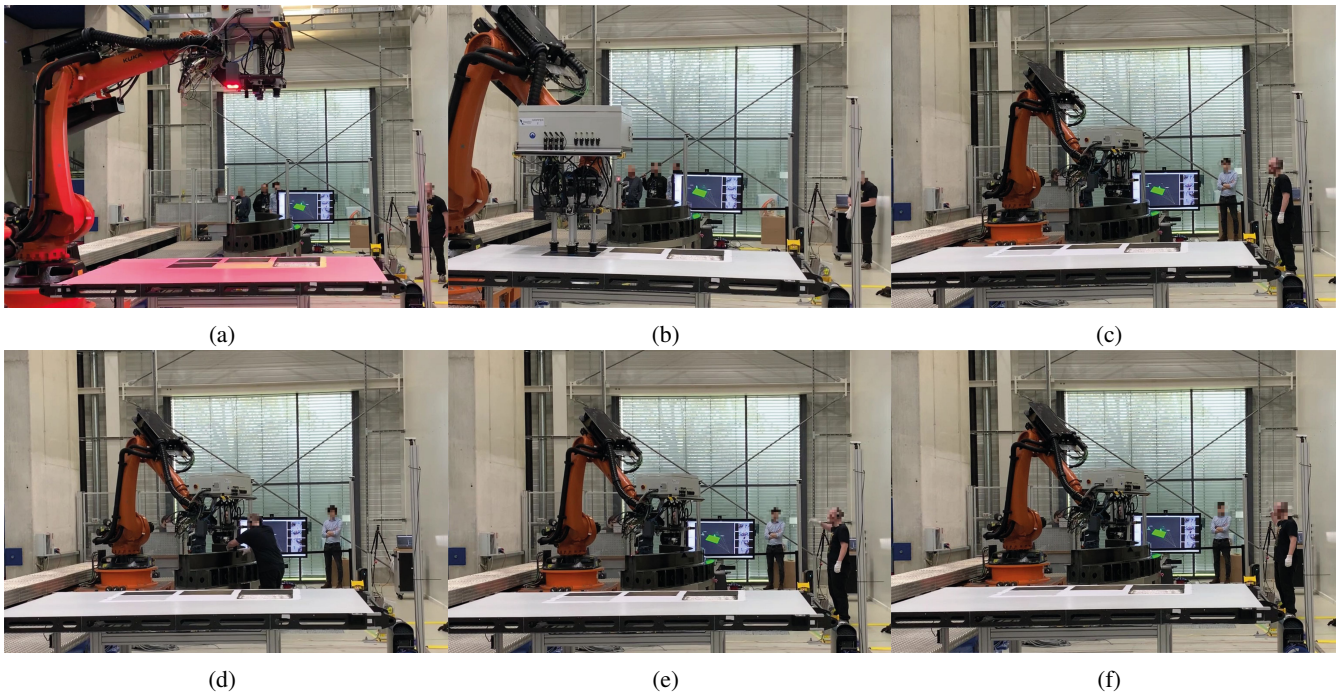
Fig. 7: Evaluation of the action and gesture recognition performance on the test set as confusion matrix.

afterwards, so it takes a lot of time; vice versa if you replan close to the end of the process, it will be faster because there are fewer tasks left to complete the process. Out of all the trials conducted, there was only one instance (Trial 13) where the Task Planner provided an invalid plan both as the first plan and during the replanning phase. The computational time was recorded in this case, and the trial was aborted. These findings highlight the efficiency of the planner in generating initial plans, with most plans being valid and feasible. The slight increase in computational time during the replanning phase suggests that the process of revising and generating a new plan takes slightly more time than the initial planning stage. Additionally, the occurrence of an invalid plan during replanning emphasizes the importance of thorough testing and verification to ensure the reliability and safety of the system.

C. Human Action Recognition Evaluation

For evaluating the Human Action Recognition module, a dataset has been collected for 5 different subjects, each one performing five times all the human actions considered in Table II. In order to improve the reliability of the action classifier, also a general "unknown" class has been considered in the dataset acquisition to learn better to distinguish the movements associated with the gesture from movements related to general movements of the worker within the workcell not related to the overall process as walking and standing. We considered 10 sequences for each subject for evaluating performance on the "unknown" gesture since it includes a larger variability of possible movements.

The action recognition module is trained on the collected dataset, using the sequence relative to 4 subjects. The sequences related to the fifth subject are reserved as a test set, on which the action recognition classifier is evaluated in terms of accuracy. This allows to evaluate the action classifier on a set of data not used to train the model, assessing the classifier's ability to generalize on novel data. The total number of test sequences is 40: five sequences for each of



(a) (b) (c)

(d) (e) (f)

Fig. 8: Evaluation during a draping task: the robot detects the required ply (a) and moves to the "pick" position (b); the operator waits until the robot reaches the final position on the mould (c) and then starts the manual draping phase (d); when draping is finished, the worker requests a new ply using a *Drape Next* gesture (e); while the robot starts moving towards a new ply, the human raise the right arm to trigger a manual "inspection" causing a replanning (f).

the six human actions defined in Table II, and ten sequences involving common movements annotated as "unknown".

The action recognition module has been evaluated in terms of Top1 and Top3 accuracy. The former represents the percentage of correctly predicted gestures in the test set. At the same time, the Top3 accuracy is the percentage of actions whose correct prediction falls in the three highest softmax scores estimated by the network. The performance achieved are Top1=90.00% and Top3=97.50%. As shown in the confusion matrix, Fig. 7, the classifier performs well in terms of accuracy on the considered test set since most of the actions of the fifth subject are correctly recognized.

D. Experimental Validation

The proposed framework has been validated in a real industrial scenario, targeting a collaborative draping task shown in Figure 8. In particular, the human operator and the robot work together to drape a series of plies on a mould: the robot provides the material transport and accurate placement on the mould (Figure 8c), while the human operator performs the actions that require high manual dexterity, such as manually draping the material over the mould (Figure 8d). At any time, the user can request particular actions from the robot by means of gestures, such as a request for a new ply (Figure 8e) or a request to stop the current operation to allow the operator to manually inspect the draping quality (Figure 8f). The proposed framework is able to monitor the worker's activity and handle sudden requests from the operator. For example, in the experimental validation shown in Figure 8, at the end of the manual draping, the operator makes a *Drape Next* gesture, thus triggering the task planner to generate a plan to move the robot towards a new ply; immediately afterwards, when the robot starts to move, the user makes a new *Inspection* gesture forcing the task planner to delete the previous plan and generate a new one.

V. CONCLUSIONS

In this paper, we proposed a Human-Aware Task Planner for Human-Robot Collaborative industrial applications. The advantages of this approach are the ability to create a plan starting from the description of the process and the actions in order to share that activities both from humans and robots. In addition, it is able to handle user interaction through the dynamic rescheduling of the plan following user interruptions or to handle unexpected events. The user commands are handled by an intelligent Human Action Recognition module based on Deep Learning technique. Another main contribution is the ability of the planner to create actions starting from primitives. The framework has been validated in an industrial collaborative scenario derived from the DrapeBot European research project. The results obtained demonstrate the applicability and effectiveness of the proposed approach. In future works, we plan to integrate the Task Planner with an ergonomic Motion Planner in order to evaluate the complete Task and Motion Planner (TAMP) application in a dynamic industrial scenario where one of the collaborative activities between robot and human is the collaborative transport of plies.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101006732, "DrapeBot – COLLABORATIVE DRAPING OF CARBON FIBER PARTS".

REFERENCES

[1] G. Chryssolouris, N. Papakostas, and D. Mavrikios, "A perspective on manufacturing strategy: Produce more with less," *CIRP Journal of Manufacturing Science and Technology*, vol. 1, no. 1, pp. 45–52, 2008.

[2] S. Pellegrinelli, A. Orlandini, N. Pedrocchi, A. Umbrico, and T. Tolio, "Motion planning and scheduling for human and industrial-robot collaboration," *CIRP Annals*, vol. 66, no. 1, pp. 1–4, 2017.

[3] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtos, and M. Carreras, "Rosplan: Planning in the robot operating system," in *Proceedings of the international conference on automated planning and scheduling*, vol. 25, 2015, pp. 333–341.

[4] B. Lacerda, D. Parker, and N. Hawes, "Optimal and dynamic planning for markov decision processes with co-safe ltl specifications," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1511–1516.

[5] V. A. Ziparo, L. Iocchi, P. U. Lima, D. Nardi, and P. F. Palamara, "Petri net plans: A framework for collaboration and coordination in multi-robot systems," *Autonomous Agents and Multi-Agent Systems*, vol. 23, pp. 344–383, 2011.

[6] A. Umbrico, A. Cesta, M. Cialdea Mayer, and A. Orlandini, "Platinu m: A new framework for planning and acting," in *AI* IA 2017 Advances in Artificial Intelligence: XVIth International Conference of the Italian Association for Artificial Intelligence, Bari, Italy, November 14-17, 2017, Proceedings 16*. Springer, 2017, pp. 498–512.

[7] M. S. Malvankar-Mehta and S. S. Mehta, "Optimal task allocation in multi-human multi-robot interaction," *Optimization Letters*, vol. 9, pp. 1787–1803, 2015.

[8] F. Chen, K. Sekiyama, F. Cannella, and T. Fukuda, "Optimal subtask allocation for human and robot collaboration within hybrid assembly system," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1065–1075, 2013.

[9] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[10] T. Yu, J. Huang, and Q. Chang, "Optimizing task scheduling in human-robot collaboration with deep multi-agent reinforcement learning," *Journal of Manufacturing Systems*, vol. 60, pp. 487–499, 2021.

[11] L. Tagliapietra and E. Tosello, "Curami: human-robot collaboration for intelligent assembly tasks."

[12] M. Terreran, S. Ghidoni, E. Menegatti, V. Enrico, P. Nicola, N. Castaman, A. Gottardi, E. Christian, V. Luca, S. Giuseppe, *et al.*, "A smart workcell for cooperative assembly of carbon fiber parts guided by human actions," in *Atti della 4ª Conferenza Italiana di Robotica e Macchine Intelligenti*, 2022.

[13] M. H. Arbo, Y. Pane, E. Aertbeliën, and W. Decré, "A system architecture for constraint-based robotic assembly with cad information," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 690–696.

[14] G. Michalos, J. Spiliotopoulos, S. Makris, and G. Chryssolouris, "A method for planning human robot shared tasks," *CIRP journal of manufacturing science and technology*, vol. 22, pp. 76–90, 2018.

[15] A. Casalino, A. M. Zanchettin, L. Piroddi, and P. Rocco, "Optimal scheduling of human–robot collaborative assembly operations with time petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 1, pp. 70–84, 2019.

[16] Y. Cheng, L. Sun, and M. Tomizuka, "Human-aware robot task planning based on a hierarchical task model," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1136–1143, 2021.

[17] A. Ham and M.-J. Park, "Human–robot task allocation and scheduling: Boeing 777 case study," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1256–1263, 2021.

[18] N. Nikolakis, N. Kousi, G. Michalos, and S. Makris, "Dynamic scheduling of shared human-robot manufacturing operations," *Procedia CIRP*, vol. 72, pp. 9–14, 2018.

[19] P. Tsarouchi, G. Michalos, S. Makris, T. Athanasatos, K. Dimoulas, and G. Chryssolouris, "On a human–robot workplace design and task allocation system," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 12, pp. 1272–1279, 2017.

[20] P. Tsarouchi, S. Makris, and G. Chryssolouris, "On a human and dual-arm robot task planning method," *Procedia CIRP*, vol. 57, pp. 551–555, 2016.

[21] L. Johannsmeier and S. Haddadin, "A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 41–48, 2016.

[22] K. Darvish, E. Simetti, F. Mastrogiovanni, and G. Casalino, "A hierarchical architecture for human–robot cooperation processes," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 567–586, 2020.

[23] F. Rovida, B. Grossmann, and V. Krüger, "Extended behavior trees for quick definition of flexible robotic tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6793–6800.

[24] E. Lamon, F. Fusaro, E. De Momi, and A. Ajoudani, "A comprehensive architecture for dynamic role allocation and collaborative task planning in mixed human-robot teams," *arXiv preprint arXiv:2301.08038*, 2023.

[25] M. Cramer, K. Kellens, and E. Demeester, "Probabilistic decision model for adaptive task planning in human-robot collaborative assembly based on designer and operator intents," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7325–7332, 2021.

[26] R. Caccavale and A. Finzi, "Flexible task execution and attentional regulations in human-robot interaction," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 1, pp. 68–79, 2016.

[27] J. Cacace, R. Caccavale, A. Finzi, and R. Grieco, "Combining human guidance and structured task execution during physical human–robot collaboration," *Journal of Intelligent Manufacturing*, pp. 1–15, 2022.

[28] M. Rizwan, V. Patoglu, and E. Erdem, "Human robot collaborative assembly planning: An answer set programming approach," *Theory and Practice of Logic Programming*, vol. 20, no. 6, pp. 1006–1020, 2020.

[29] R. Lallement, L. de Silva, and R. Alami, "Hatp: hierarchical agent-based task planner," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018)*, 2018.

[30] S. Stock, M. Mansouri, F. Pecora, and J. Hertzberg, "Online task merging with a hierarchical hybrid task planner for mobile service robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 6459–6464.

[31] E. Foderaro, A. Cesta, A. Umbrico, and A. Orlandini, "Simplifying the ai planning modeling for human-robot collaboration," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. IEEE, 2021, pp. 1011–1016.

[32] H. Oliff, Y. Liu, M. Kumar, M. Williams, and M. Ryan, "Reinforcement learning for facilitating human-robot-interaction in manufacturing," *Journal of Manufacturing Systems*, vol. 56, pp. 326–340, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612520301084

[33] L. Hu, Z. Liu, W. Hu, Y. Wang, J. Tan, and F. Wu, "Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network," *Journal of Manufacturing Systems*, vol. 55, pp. 1–14, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612520300145

[34] Y. G. Kim, S. Lee, J. Son, H. Bae, and B. D. Chung, "Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system," *Journal of Manufacturing Systems*, vol. 57, pp. 440–450, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612520301916

[35] C. Aeronautiques, A. Howe, C. Knoblock, I. D. McDermott, A. Ram, M. Veloso, D. Weld, D. W. SRI, A. Barrett, D. Christianson, *et al.*, "Pddl— the planning domain definition language," *Technical Report, Tech. Rep.*, 1998.

[36] M. Terreran, M. Lazzaretto, and S. Ghidoni, "Skeleton-based action and gesture recognition for human-robot collaboration," in *Intelligent Autonomous Systems 17: Proceedings of the 17th International Conference IAS-17*. Springer, 2023, pp. 29–45.

[37] I. Sárándi, T. Linder, K. O. Arras, and B. Leibe, "Metrabs: metric-scale truncation-robust heatmaps for absolute 3d human pose estimation," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 1, pp. 16–30, 2020.