A software tool for human-robot shared-workspace collaboration with task precedence constraints

Terrin Babu Pulikottil*, Stefania Pellegrinelli, Nicola Pedrocchi

Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council of Italy, Milano, 20133, Italy

ARTICLE INFO

Keywords:
Human-robot collaboration
Goal precedence constraints
Partially observable markov decision process

ABSTRACT

A key challenge in human-robot shared workspace is defining the decision criterion to select the next task for a fluent, efficient and safe collaboration. While working with robots in an industrial environment, tasks may comply with precedence constraints to be executed. A typical example of precedence constraint in industry occurs at an assembly station when the human cannot perform a task before the robot ends on its own. This paper presents a methodology based on the Maximum Entropy Inverse Optimal Control for the identification of a probability distribution over the human goals, packed into a software tool for human-robot shared-workspace collaboration. The software analyzes the human goal and the goal precedence constraints, and it is able to identify the best robot goal along with the relative motion plan. The approach used is, an algorithm for the management of goal precedence constraints and the Partially Observable Markov Decision Process (POMDP) for the selection of the next robot action. A comparison study with 15 participants was carried out in a real world assembly station. The experiment focused on evaluating the task fluency, the task efficiency and the human satisfaction. The presented model displayed reduction in robot idle time and increased human satisfaction.

1. Introduction

Human Robot Collaboration (HRC), although far from being fully exploited, is considered the enabler of a safe and effective task execution, with reduced tedium and strain for the human operator in the industrial environment [1–3] (Fig. 1). Cooperation efficiency is still limited to all those applications in which the operations are sequential and simple [4,5], and few softwares for industrial applications rely on task precedence constraints, which the robot has to take into account before executing its tasks [6]. Among others, the efficiency of HRC solutions characterized by set of tasks with many constraints is related to predict the human actions [7], to plan and continuously replan safe robot trajectories [8], and to plan sequences of the tasks according to explicit temporal constraints with large variability [9]. In other words, there is the need for the robot to reason explicitly on human environments and on its own capacities to achieve tasks in a collaborative way with a human partner [10–12].

It is worth noting that many papers in the literature address jointly the ability to predict human actions and to plan the robot motion [13–15], while only few ones do consider the influence deriving from possible constraints in the sequence of operations, *i.e.* few softwares integrate a model for complex HRC processes. Ziebart et al. [16] proposed an approach for predicting future pedestrian trajectories using Maximum Entropy Inverse Optimal Control (MaxEntOIC). Lasota and Shah [17] modeled the human action and decision making process as a stochastic transition function with an Markov Decision Process (MDP). Similarly, Mainprice et al. [18] proposed a framework based on the prediction of human workspace occupancy by computing the swept volume of learned human motion trajectories and the planning of the robot trajectories was then computed by minimizing the penetration cost. Karami et al. [19] presented a new framework for controlling a robot while cooperating with a human in the accomplishment of a common mission; the approach utilized a Partially Observable Markov Decision Process (PODMP) to infer user goals and to select the next robot task in shared mission domains. Nguyen et al. [20] and Macindoe et al. [21] applied this POMDP model for creating agents in cooperative games. Bandyopadhyay et al. [22] investigated a new class of motion planning problems by assuming a finite set of unknown intentions. Sotiris Makris et al. [23] introduced an Artificial Intelligence based framework for effectively distributing the work while deriving optimized motion plans in flexible robotic assembly lines. Chengyu Hu et al. [24] addressed robot navigation based on particle swarm optimization in static or dynamic surroundings by generating each optimal step from initial to goal position. Yibin Li et al. [25] used a new hybrid algorithm for dynamic obstacle avoidance with a mutation-based evolutionary polynomial Artificial Neural Networks for short-term prediction and its

Fig. 1. Human robot collaboration in assembly.

output integrated into the reward function of a POMDP.

All the approaches described here have been developed and tested in unstructured environments, such as homes, supermarkets, etc., where there are no constraints among the human and the robot tasks, *i.e.* the robot can execute its actions without having knowledge of the set of tasks already accomplished by the human. However, this information is necessary and mandatory in a lot of industrial contexts in order to achieve the desired product quality and functionality.

Considering the planning of the task sequence, the modeling of the constraints relies on artificial intelligence techniques [26] or on the scheduling of highly structured industrial processes [27,28]. On the one hand, plan-based controllers, *e.g.* T-REX [29], rely on temporal planning mechanisms capable of dealing with coordinated task actions and temporal flexibility. Unfortunately, these systems do not have an explicit representation of uncontrollable features such as human actions and therefore their applicability is still limited. On the other hand, because of the interaction with humans, the linear programming techniques used in the planning and scheduling of industrial tasks cannot model easily the variability in the robot task execution [30].

Within such a large research field, the goal of this paper is to make a step forward in the direction of HRC involving task precedence constraints while improving task efficiency and human satisfaction. Specifically this paper proposes a real-time tool for human-robot cooperation where a distribution over the human's intention is continuously inferred and considered to let the robot act, while taking into account existing precedence constraints. The tool will identify the best robot goal along with the motion plan considering the workspace objects, the human motion, all possible robot goals and the precedence constraints. This work will serve as the first step towards introducing the concept of real-time task planning and real-time motion planning in HRC for industrial applications.

This paper is organized as follows: in Section 2. the approach used for human goal prediction, task precedence constraint and robot action is explained in detail; Section 3 describes the software architecture of the tool using the proposed algorithm; Section 4 describes the implementation of the software tool used in the experiments; Section 5 presents a real world experiment carried out on an assembly station to investigate the effectiveness of the tool; finally, Section 6 details the results and discussion, and Section 7 presents conclusions and future work.

2. Approach

The current work is an extension of the author's previous work published in [31], where task precedence constraints were included for both robot and human and the work was presented as a software tool to be used in industry. This work is an attempt to bridge the gap between robot motion planning and task planning in human-robot collaboration. In this research work, it is assumed that both human and robot have a set of goals to be performed on objects eventually and that a set of constraints exists for the completion of these goals. The constraints include:

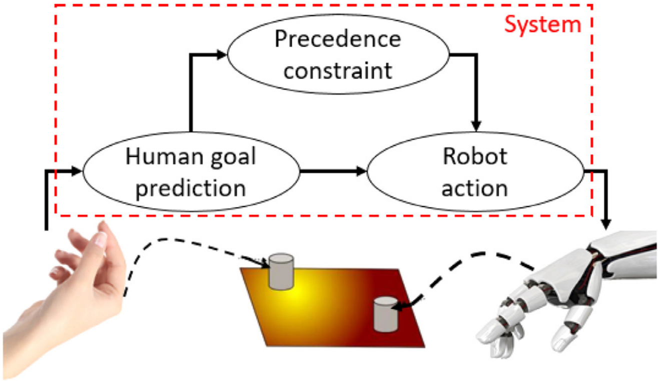

Fig. 2. Framework of the proposed algorithm.

- human and robot cannot perform their goal on an object at the same time;
- some objects have task precedence constraint,*i.e.* the robot cannot perform its goal on some objects before the human has performed his/her goal and the human cannot perform his/her goal on some objects before the robot has performed its goal.

In such a context and for an effective human-robot collaboration, it is extremely relevant to (i) predict the human goal and his/her probability distribution over the goals, (ii) evaluate the possible set of goals for the robot based on human and robot accomplished tasks, (iii) identify the best robot goal and motion plan to achieve the goal. These three aspects that represent the core of the developed methodology (Fig. 2) are described hereafter.

2.1. Human goal prediction

The aim is to infer a probability distribution $P(g^h)$ over human goals $g^h \in G^h$. With such an aim, the trajectory ξ executed by the human (*e.g.* a sequence of hand poses) from the initial position p_0 to the current position p is analyzed and used together with the Bayes's rule:

$$P\left(g^h \mid \xi_{p_o \to p}\right) \propto P\left(\xi_{p_o \to p} \mid g^h\right) P(g^h) \qquad (1)$$

The methodology, based on the Maximum Entropy Inverse Optimal Control, is explained in detail in the work published by Dragan and Srinivasa [32].

2.2. Possible robot goals

Each object may be free of any precedence constraint (*free constraint*) or may be assigned a task precedence constraint depending on the task to be performed on the object. Constraints on each object may be one of the following:

- *robot constraint*: the robot cannot perform its goal on the object before the human;
- *human constraint*: the human cannot perform its goal on the object before the robot.

The algorithm for evaluating the possible set of goals that the robot may perform at a given time considering task precedence constraint is presented in algorithm I. Each object O_i has a set of robot goals G_i^r (*e.g.* various grasp poses for an object) and a human goal G_i^h to be performed

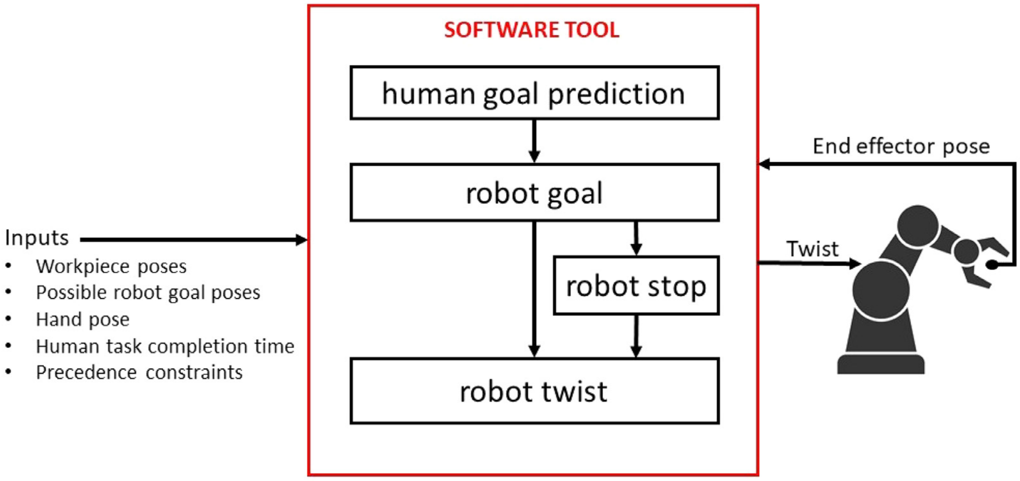

Fig. 3. Software architecture.

considering its task precedence constraint TPC_{O_i}. If TPC_{O_i} is not a robot constraint and G_i^r has not yet been performed, then G_i^r is added to the list of possible robot goals $\overline{G^r}$. This process is performed for all objects O and then $\overline{G^r}$ is published. In parallel, the task precedence constraint TPC is updated constantly considering the human goals G^h. When the human goal probability over an object is higher than the human goal probability threshold value $P^T(G^h)$ (set at 0.9), then it is assumed that the human is performing its task on that object. If an object O_i has a robot constraint and the human goal probability over that object $P(G_i^h)$ is higher than the threshold value, then a timer is started. The timer stops once $P(G_i^h)$ is lower than the threshold value. If the timer time recorded is higher than the minimum time required to complete the human goal on that object $(T_{\min})_i^h$, then its task precedence constraint TPC_{O_i} is updated from robot constraint to free constraint. Both the task precedence constraints and the list of possible robot goals are updated constantly till all the robot goals are performed.

2.3. Best robot goal

The robot must select the best goal from the possible set of goals $\overline{G^r}$ at that given time and define the best trajectory to reach the goal from the current position. With such an aim and considering the human probability distribution over the goals, as well as the task precedence constraints, the algorithm used in Javdani et al. [33] was adopted.

Specifically, the problem was encoded as a Partially Observable Markov Decision Process (POMDP) that minimizes the expected robot cost for the (unknown) human goal. Formally, we can define the continuous state $s_t \in S$ as $S = X \times G^h$, where $x \in X$ is the robot continuous state, *i.e.* robot position and $g^h \in G^h$ is the human goal. Since the human goal g^h is not known in advance, a probability distribution over the human goal g^h of the system state (the belief b) is used. Finally, we can introduce $a_t \in A$ as the continuous robot action, *i.e.* robot twist. Based on the previous definitions, the total expected robot cost can be expressed as $\mathbb{E}[\Sigma_t C^r(s_t, a_t)]$, where C^r is the cost function. Since the resolution of such POMDP was intractable, *i.e.* it would require to identify the optimal action considering all the possible belief state b with continuous state and action spaces, hindsight optimization [34,35] (also referred as QMDP approximation [36]) was used.

3. Software architecture

The software tool was developed using the Robot Operating System (ROS - kinetic version), which requires a Linux distribution base operating system; Ubuntu 16.04 was used for the development of the tool. The inputs and outputs are exchanged as ROS topics.

Inputs are listed below:

- Pose of the human hand [geometry_msgs/Pose - representation of pose in free space, composed of point position and quaternion orientation]
- Pose of the workpieces [geometry_msgs/PoseArray - an array of poses]
- Possible robot end-effector goal poses on each workpieces [array of geometry_msgs/PoseArray - each PoseArray element represents the possible robot end effector goal poses on a workpiece]
- Current end effector pose of the robot [geometry_msgs/Pose]
- Precedence constraint on each workpiece [integer array - each integer element (value ranging from 1 to 3) represents precedence constraint on a workpiece (robot constraint-1, human constraint-2, free constraint-3)]
- Human task completion time (minimum) on each workpiece [float array]

The Output of the Software tool is the robot end-effector twist (float array of 6 elements). The first three elements of the array give the linear twist and the next three elements give the angular twist of the robot in x, y and z directions.

The Software architecture of the tool can be divided into 4 ROS nodes (Fig. 3),

Human goal prediction - The human goal prediction node subscribes to the pose of the workpieces and human hand and gives a probability distribution of the human's goal over the workpiece. This information is important for the effective selection of the robot actions. The probability distribution is evaluated according to Eq. (1)

Robot goal - This node generates the list of possible goals the robot can perform at the given time. It receives feedback from the robot (*i.e.* robot action module, discussed in Section 4.4) on the completion of each goal and updates the possible goal list. This node also measures the time the human has spent on a workpiece to infer whether the human has performed its goal or not. Simplifying, the node verifies whether the human has spent enough time (a predefined time) on a task to complete it. The task precedence constraints are managed in this node using the algorithm presented in Section 2.2.

Robot stop - The robot stop node analyzes the human goals probability distribution and the possible robot goals and provides a stop command to the robot if required. The node decides when the robot has to stay idle without performing any task. This situation might arise when the robot has no possible goal to perform at that time or only one possible goal, which is in constraint because the human is currently working on it.

Robot twist - This node is responsible for the motion of the robot. Assuming the human has decided which goal to perform, the robot task

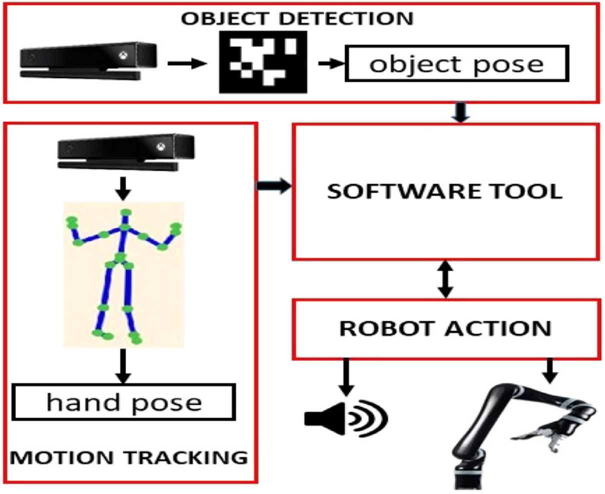

Fig. 4. Structure of the system.

begins accomplishing a goal from the list of possible goals. The POMDP described in Section 2.3 is used to evaluate the best robot twist, *i.e.* end effector velocity.

4. Implementation

A 6 dof Kinova JACO^2 robotic arm was selected due to its very limited weight and its compliance to safety standards for human-robot cooperation. Two Microsoft Kinects[37], consisting of several sensors including a RGB sensor, a 3D depth sensor, multi-array microphones and an accelerometer, were used for object detection and human motion tracking. They were placed at a distance of 1.5 m from the workspace. An internal speaker on the laptop was used to give feedback to the human as soon as the robot completed its goal. ROS was used to control both the robotic arm and the kinect sensors.

The implementation, illustrated in Fig. 4, consists of 4 modules: (1) the object detection, (2) the human motion tracking, (3) the software tool, and (4) the robot action. These four modules are described in details hereafter.

4.1. Object detection module

The “iai_kinect2” package [38] was used to receive data from the kinect sensor. The package provides a collection of tools and libraries to calibrate the kinect (also for human motion tracking) and to interface it with ROS. “AprilTags” (a visual fiducial system) [39] was used for the detection of the objects in the workspace. Targets (tags) can be created from an ordinary printer, and the AprilTag detection software computes the precise tag 3D position and orientation relative to the kinect and identifies the object. The “VISP hand2eye calibration” package [40] was used for the extrinsic calibration (kinect with respect to the robot). Possible robot end-effector goal poses for each workpiece are generated using the “Moveit! simple grasps” package [41]. In this module also the precedence constraints for each workpiece are defined by the user according to the application. Hence, the Object Detection Module provides the pose of the workpieces, possible robot end-effector goal poses on each workpieces and precedence constraint on each workpiece as input to the software tool.

4.2. Human motion tracking module

In this module, the kinect sensor works under a windows operating system using the “kinect windows SDK2.0”; the “k2_bridge” software package [42] was used to integrate the kinect in ROS. This method was chosen to exploit the kinect manufacturer support softwares, which works only in Windows OS. “k2_bridge” software package is divided into two parts. One part runs on a windows machine and dumps the data over the network, while the second one runs on a linux machine which reads the stream and publishes appropriate ROS topics. Using kinect, 6 different users can be tracked in the workspace with different colors and numbers. Each skeleton contains 15 bony landmarks including: head, neck, torso, shoulder, elbow, hand, etc. A node was created to access the bony landmarks coordinates and to publish the pose of the hand closest to the robot. Human task completion time (minimum) for each workpiece was computed and published as a ROS topic. This can be provided by the user through past experience or computed during a trial run. This information is required to check whether the human has spent the minimum amount of time near the workpiece needed to perform the task. Hence, the Human Motion Tracking Module provides the pose of the human hand and human task completion time (minimum) on each workpiece as input to the software tool.

4.3. Software tool module

The ROS topics published by the Object Detection and Human Motion Tracking modules along with the current end-effector pose of the robot provided by the Robot Action module are analyzed by the Software tool to define the end-effector twist. The details of this module are presented in Section 3.

4.4. Robot action module

This module, subscribing to the Software Tool module, sends to the robot the command to move according to the defined end-effector twist. In addition, it defines the next task to be performed after achieving its goal (e.g. stamping or grabbing the workpiece). This module publishes continuously the robot current pose needed by the Software Tool module and further gives sound feedback to the human once the robot has completed its task on a workpiece.

5. Experiments

The system was tested on a real-world human-robot collaborative environment with a typical industrial application. The aim of the experiment was to investigate the effectiveness of the algorithm used in the software tool. A secondary aim was to show that the tool can operate in real time successfully even with naive users. The following two algorithms were measured:

Algorithm 1 The developed POMDP model with hindsight optimization and task precedence constraint used in the software tool

Algorithm 2 A control algorithm without human motion tracking executing a fixed sequence of tasks that considers task precedence constraint (hence, takes into account the human goal completion)

The sequence of tasks (or workpiece) the robot has to perform is predefined (*i.e.* the order of execution) in Algorithm 2. The Algorithm 2 tries to replicate an algorithm which doesn’t consider the human movement and hence, doesn’t change its path due to human intervention (The robot takes the shortest path to complete all its tasks). In order to satisfy the robot constraint, Algorithm 2 like Algorithm 1 takes into account the human goal completion. To compare the two algorithms, the user performance was evaluated using objective and subjective metrics, namely the task efficiency, the human satisfaction and the task fluency. It was hypothesized that the outcome measures would improve with Algorithm 1, *i.e.* the algorithm used in the software tool.

5.1. Task

An industrial assembly station with a vision system was replicated for the experiment (Fig. 5). The workpieces (WP) at the assembly

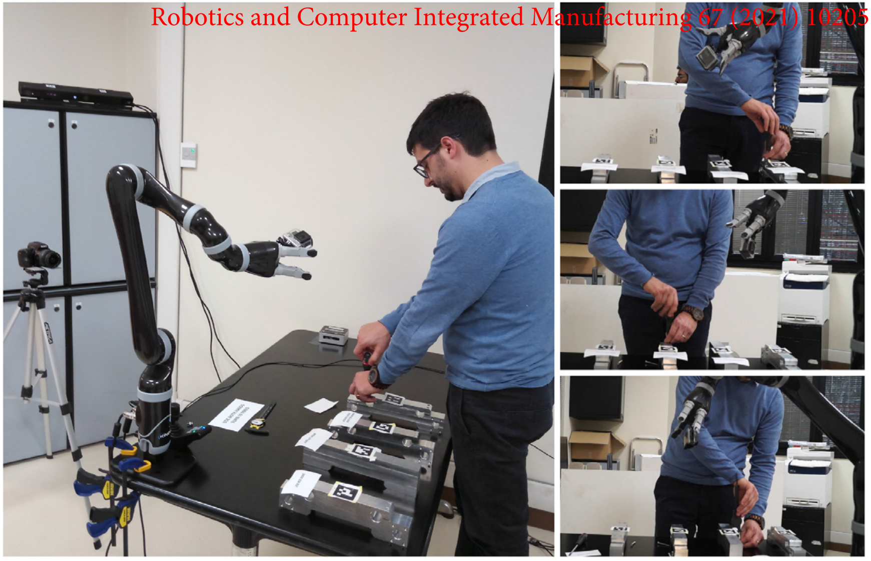

Fig. 5. Participant performing a collaborative assembly task on the selected setup.

station were provided with tags attached to their top for object detection. The robot and the human were not allowed to perform their tasks simultaneously on the same workpiece. In addition, each workpiece may be characterized by a precedence constraint that could impact human and robot on how the action is performed. All the precedence constraints are listed in Table 1.

The task for the robot was to perform a quality check on the workpieces. The robot has to take a photo of the workpieces from the top using the camera attached to its end-effector. If a workpiece has a robot constraint (WP.No. 1 and WP.No. 4), then the robot cannot take the photo of that workpiece before the human has performed his/her task. The task for the human was to carry out a set of operations on the workpieces (the operations differed for each workpiece; see Table 1). If the workpiece has a human constraint (WP. No. 3), then the human cannot perform his/her task on that workpiece before the robot has performed its task. After the robot had completed each task a sound feedback was given to allow the human to keep track of the completed robot tasks. In the experiment, workpieces were placed in a row for this experiment but the positions of the workpieces could vary (within the workspace of the robot) before the start of the experiment. The set of tasks to be performed by the human and the robot as well as the precedence constraints were not changed during the experiment.

5.2. Procedure

Fifteen subjects participated in the study. After obtaining the informed consent, the participants were introduced to the task by the researcher. They performed the task at the assembly station using both algorithms (Fig. 5); *i.e.* same set of tasks were performed twice (one trial for each algorithm). For each participant, which algorithm to use first was selected randomly. Immediately after each trial, before starting the next one, the participants completed a eight question Likert-scale survey to evaluate their collaboration with the robot. They were further asked to provide verbal feedback about the trial just performed. At the end of the experiment, after completion of the last survey, they were asked to provide verbal feedback comparing the two trials. All trials and verbal feedback were video recorded for a later evaluation.

5.3. Measurement

The task fluency and efficiency, and the subjective human satisfaction, defined here after, were chosen as primary outcome measures.

Task efficiency - Task efficiency is defined as the time needed by the human and the robot to carry out all the predefined tasks. An objective key performance indicator would be the measure of the human and robot idle time during the trial. Idle time is the waiting time during which the robot (or the human) is not performing a task. This may occur when the robot (or the human) has completed all the tasks before the human (or the robot) or when no task may be performed due to constraints. The idle time was measured from the recorded videos. Idle times for human and robot were analyzed using one way ANOVA analysis as a percentage of the total task time.

Task fluency - Task fluency involves seamless coordination of the action [31]. A measure of task fluency is the distance between the human hand and the end-effector during each trail. This distance can give us an idea about how close the human and robot interacted during the completion of the task. A low average or minimum distance between human and robot can indicate a higher proportion of task time spent in collision (Assuming that a collision occurs when the distance between the robot's end effector and the human's hand goes below a certain threshold).

Human satisfaction - A seven-point Likert scale survey was used to evaluate human satisfaction, which inherently is a subjective measure. The survey assesses the perceived safety and a sense of collaboration. Verbal feedback after each trial was also evaluated to further validate the survey results. The list of survey questions includes four questions for safety and four questions for the sense of collaboration (Table 3). To prevent response biases, questions were made both in the positive and the negative(*e.g.* æKINOVA got in my wayg). After the survey, the results of the negative questions (four questions) were revise-coded so that their scales matched the positive questions.

6. Results and discussion

Two participant results were excluded from task fluency and task efficiency analysis due to technical issues occurred during the experiments.

Robot idle time is measured as the percentage of the time robot was idle to the total time taken to complete all the tasks. The results showed a reduction in robot idle time by approximately 40% using Algorithm 1 compared to Algorithm 2 (see Table 2). This result was found to be statistically significant ($F(1, 24) = 79.52$, $p < 0.001$). This indicates a better task efficiency of the robot in the completion of the task using the software tool.

```
Inputs $t$: time, $t_0$: start time
$\delta t$: time interval between two evaluating steps
$O$: set of objects
$O_j$: $j^th$ object in $O$
$G^r$: set of robot goals
$G_i^r$ : $i$th robot goal in $G^r$
$\overline{G^r}$: set of possible robot goals
$G^h$: set of human goals
$G_i^h$ : $i$th human goal in $G^h$
$P(G^h)$: human goal probabilities over $O$
$P(G_j^h)$: human goal probability on $O_j$
$P^T(G^h)$: human goal probability threshold value
$TPC_{O_j}$: task precedence constraints for $O_j$
$(T_{\min})_j^h$: minimum time required to complete the human goal on $j$th
object
------------------------------------------------------------
$t \leftarrow t_0$
$P^T(G^h) \leftarrow 0.9$
do
            \\Update TPC
  for each $G_i^h$ in $G^h$ do
    $O_j \leftarrow$ calculate the object index given for $G_i^h$
    if $TPC_{O_j}$ is robot constraint
    if $P(G_j^h) > P^T(G^h)$
      Timer start
      if $P(G_j^h) < P^T(G^h)$
        Timer stop
      if Timer time > $(T_{\min})_j^h$
        $TPC_{O_j} \leftarrow$ free constraint
    end if
  end for
            \\Calculate $\overline{G^r}$
$\overline{G^r} \leftarrow$ empty
  for each $G_i^r$ in $G^r$ do
    $O_j \leftarrow$ calculate the object index given for $G_i^r$
    if $G_i^r$ is to be performed and $TPC_{O_j}$ is not robot constraint
      append $G_i^r$ to $\overline{G^r}$
    end if
  end for
  Publish $\overline{G^r}$
until all robot goal is performed
$t \leftarrow t_0 + \delta t$
```

Algorithm 1. Possible robot goals.

Table 1
Tasks and constraints.

WP No	Human Task	Robot Task	Constraint
1	Fasten 4 screws	Take WP photo	Robot constraint
2	Measure WP width and thickness	Take WP photo	Free constraint
3	Unfasten 4 screws	Take WP photo	Human constraint
4	Fasten 4 screws	Take WP photo	Robot constraint

Table 2
Task-efficiency and task-fluency analysis.

	Alg. 1	Alg. 2
Robot idle time (%)	42.3(± 4.1)	81.8(± 3.7)
Human idle time (%)	17.1(± 6.8)	5.2(± 3.5)
Average distance between hand and end-effector (cm)	63.9(± 3.6)	61.3(± 4.6)
Minimum distance between hand and end-effector (cm)	29.4(± 5.8)	26.6(± 5.2)

Table 3
Seven point Likert scale survey - mean and standard deviation.

	Alg. 1	Alg. 2
KINOVA was a good partner	4.6(± 1.3)	4.6(± 1.3)
I felt that KINOVA kept a safe distance from me	4.4(± 1.3)	4.33(± 1.4)
KINOVA got in my way	3.6(± 1.66)	4.33(± 1.6)
KINOVA moved too fast	1.2(± 0.4)	1.53(± 0.88)
I think KINOVA and I worked well as a team	4.4(± 1.4)	4.2(± 1.3)
I felt uncomfortable working so close to KINOVA	1.93(± 1.33)	2.73(± 1.5)
I am dissatisfied with how KINOVA and I worked together	2.67(± 1.56)	2.93(± 1.4)
I trust KINOVA	5.4(± 1.4)	5.47(± 1.2)

Note: Questions are rated from 1 (not at all) to 7 (very much)

Human idle time is measured as the percentage of the time the human was idle to the total time taken to complete all the tasks. Human idle time was found to be higher for Algorithm 1 compared to Algorithm 2 (approx. 12%, see Table 2). This result was also found to be statistically significant ($F(1, 24) = 8.18$, $p = 0.009$). This is a further confirmation of the fact that Algorithm 1 improves task fluency. Indeed, in Algorithm 1, idle time occurred only after the human accomplished all his/her task, thus demonstrating an improvement in human working conditions.

The analysis of the distance between the end-effector and the human hand indicates high data scatter for both average and the minimum distance measurement; therefore no statistically significant effect (average distance: $F(1, 24) = 1.98$, $p = 0.173$ and minimum distance: $F(1, 24) = 1.2$, $p = 0.284$) was noticed between the Algorithm 1 and Algorithm 2(see Table 2). These results may be affected by the position of the robot, Indeed, the robot was placed at the corner of the workstation and hence, favored robot motion towards object 1 and 2 in both algorithms. Due to insignificant results, improvements in task fluency with the help of the software tool cannot be proved.

The evaluation of the survey results was found to be encouraging since Algorithm 1 was either rated equal or better than Algorithm 2. The average result of the survey is given in Table 3. The results indicate that the participants felt Algorithm 1 to be either equal or much safer and more comfortable compared to Algorithm 2 thanks to a reduction of the general robot speed and to the interference of the robot with the human. Moreover, the verbal feedback was in support of the survey results as almost all the participants preferred Algorithm 1 over Algorithm 2.

7. Conclusions

In this paper, a new software tool for human-robot shared-workspace collaboration considering task precedence constraints was proposed. The tool used an extension of Bayers rule for human goal prediction, a developed algorithm for evaluating the set of available tasks for the robot based on task precedence constraints and a POMDP for identifying the robot goal and the motion plan to reach the goal. The algorithm of the tool was then tested on a real-world environment comparing it with a control algorithm that does not consider the human motion tracking and executes the task in a fixed sequence considering task precedence constraints. The results indicated a 40% reduction in robot idle time and better human satisfaction. Statistical analysis of these results was carried out to assess the significance of the results.

This work can be extended to incorporate a model of how the human infers the robots goal into the existing framework. Such a model has led to more fluent human-robot collaboration [43]. Another area of extension of this work is in the field of task planning, with a focus on the study of trajectory time variability. None of the existing papers, to the best of the author's knowledge, has focused on utilizing the POMDP model on such a study.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome. We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us. We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property. We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept email from terrinbabu.pulikottil@stiima.cnr.it.

Acknowledgment

This research was partially funded by Future Homes for Future Communities (Accordo Quadro diCollaborazione tra Regione Lombardia e Consiglio Nazionale delle Ricerche Convenzione Operativa no 19365/RCC).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.rcim.2020.102051.

References

[1] S. Pellegrinelli, F.L. Moro, N. Pedrocchi, L. Molinari Tosatti, T. Tolio, A probabilistic approach to workspace sharing for human-robot cooperation in assembly tasks, CIRP Ann. 65 (1) (2016) 57–60, https://doi.org/10.1016/j.cirp.2016.04.035.

[2] C. Byner, B. Matthias, H. Ding, Dynamic speed and separation monitoring for collaborative robot applications–concepts and performance, Rob. Comput. Integr. Manuf. 58 (2019) 239–252.

[3] L. Gualtieri, E. Rauch, R. Vidoni, Emerging research fields in safety and ergonomics in industrial collaborative robotics: a systematic literature review, Rob. Comput.

Integr. Manuf. 67 (2020) 101998.

[4] R. Alami, A. Clodic, V. Montreuil, E.A. Sisbot, R. Chatila, Task planning for human-robot interaction, Proceedings of the 2005 Conference on Smart Objects and Ambient Intelligence, ACM, 2005, pp. 81–85.

[5] P. Tsarouchi, S. Makris, G. Chryssolouris, Human–robot interaction review and challenges on task planning and programming, Int. J. Computer Integr. Manuf. 29 (8) (2016) 916–931.

[6] A. Cesta, L. Molinari Tosatti, A. Orlandini, N. Pedrocchi, S. Pellegrinelli, T. Tolio, A. Umbrico, Planning and execution with robot trajectory generation in industrial human-robot collaboration, 4th Italian Workshop on Artificial Intelligence and Robotics, ACM, 2017.

[7] D. Kulić, Y. Nakamura, Incremental learning of human behaviors using hierarchical hidden Markov models, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, 2010, pp. 4649–4655.

[8] B. Schmidt, L. Wang, Contact-less and programming-less human-robot collaboration, Procedia CIRP 7 (2013) 545–550.

[9] M. Cialdea Mayer, A. Orlandini, A. Umbrico, Planning and execution with flexible timelines: a formal account, Acta Inf. 53 (6) (2016) 649–680, https://doi.org/10.1007/s00236-015-0252-z.

[10] R. Alami, A. Clodic, V. Montreuil, E.A. Sisbot, R. Chatila, Toward human-aware robot task planning. AAAI Spring Symposium: To Boldly Go Where no Human-Robot Team Has Gone Before, (2006), pp. 39–46.

[11] S. Takata, T. Hirano, Human and robot allocation method for hybrid assembly systems, CIRP Ann. 60 (1) (2011) 9–12.

[12] C. Schlenoff, Z. Kootbally, A. Pietromartire, M. Franaszek, S. Foufou, Intention recognition in manufacturing applications, Rob. Comput. Integr. Manuf. 33 (2015) 29–41.

[13] R. Meziane, M.J.-D. Otis, H. Ezzaidi, Human-robot collaboration while sharing production activities in dynamic environment: spader system, Rob. Comput. Integr. Manuf. 48 (2017) 243–253.

[14] E. Kim, R. Kirschner, Y. Yamada, S. Okamoto, Estimating probability of human hand intrusion for speed and separation monitoring using interference theory, Rob. Comput. Integr. Manuf. 61 (2020) 101819.

[15] X. Zhang, H. Lin, Performance guaranteed human-robot collaboration with POMDP supervisory control, Rob. Comput. Integr. Manuf. 57 (2019) 59–72.

[16] B.D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J.A. Bagnell, M. Hebert, A.K. Dey, S. Srinivasa, Planning-based prediction for pedestrians, Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, IEEE, 2009, pp. 3931–3936.

[17] P.A. Lasota, J.A. Shah, Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration, Hum. Factors 57 (1) (2015) 21–33.

[18] J. Mainprice, D. Berenson, Human-robot collaborative manipulation planning using early prediction of human motion, Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, IEEE, 2013, pp. 299–306.

[19] A.-B. Karami, L. Jeanpierre, A.-I. Mouaddib, Partially observable Markov decision process for managing robot collaboration with human, Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on, IEEE, 2009, pp. 518–521.

[20] T.-H.D. Nguyen, D. Hsu, W.S. Lee, T.-Y. Leong, L.P. Kaelbling, T. Lozano-Perez, A.H. Grant, CAPIR: Collaborative action planning with intention recognition, AIIDE, (2011).

[21] O. Macindoe, L.P. Kaelbling, T. Lozano-Pérez, POMCOP: Belief space planning for sidekicks in cooperative games, AIIDE, (2012).

[22] T. Bandyopadhyay, K.S. Won, E. Frazzoli, D. Hsu, W.S. Lee, D. Rus, Intention-aware motion planning, Algorithmic Foundations of Robotics X, Springer, 2013, pp. 475–491.

[23] N. Kousi, D. Dimosthenopoulos, A.-S. Matthaiakis, G. Michalos, S. Makris, Ai based combined scheduling and motion planning in flexible robotic assembly lines, Procedia CIRP 86 (2019) 74–79.

[24] C. Hu, X. Wu, Q. Liang, Y. Wang, Autonomous robot path planning based on swarm intelligence and stream functions, Lecture Notes in Computer Science, Springer, Berlin, 4684 (2007).

[25] Y. Li, C. Li, R. Song, A new hybrid algorithm of dynamic obstacle avoidance based on dynamic rolling planning and RBFNN, 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO), (2007), pp. 2064–2068.

[26] L. Johannsmeier, S. Haddadin, A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes, IEEE Robotics and Automation Letters 3766 (c) (2016) 1, https://doi.org/10.1109/LRA.2016.2535907. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7421993

[27] S. Pellegrinelli, T. Tolio, Pallet operation sequencing based on network part program logic, Robotics and Computer-Integrated Manufacturing 29 (5) (2013) 322–345. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736584513000252f#

[28] W. Xu, Q. Tang, J. Liu, Z. Liu, Z. Zhou, D.T. Pham, Disassembly sequence planning using discrete bees algorithm for human-robot collaboration in remanufacturing, Rob. Comput. Integr. Manuf. 62 (2020) 101860.

[29] C. PyFrédéric, R. Kanna, McGann, A systematic agent framework for situated autonomous systems, Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, Toronto, Canada, (2010), pp. 583–590.

[30] S. Pellegrinelli, C. Cenati, L. Cevasco, F. Giannini, K. Lupinetti, M. Monti, D. Parazzoli, L. Molinari, Configuration and inspection of multi-fixturing pallets in flexible manufacturing systems - Evolution of the network part program approach, Rob. Comput. Integr. Manuf. 000 (May) (2017) 1–11, https://doi.org/10.1016/j.rcim.2017.05.014.

[31] S. Pellegrinelli, H. Admoni, S. Javdani, S. Srinivasa, Human-robot shared workspace collaboration via hindsight optimization, Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, IEEE, 2016, pp. 831–838.

[32] A.D. Dragan, S.S. Srinivasa, A policy-blending formalism for shared control, Int. J. Rob. Res. 32 (7) (2013) 790–805.

[33] S. Javdani, S.S. Srinivasa, J.A. Bagnell, Shared autonomy via hindsight optimization, arXiv:1503.07619(2015).

[34] E.K. Chong, R.L. Givan, H.S. Chang, A framework for simulation-based network control via hindsight optimization, Decision and Control, 2000. Proceedings of the 39th IEEE Conference on, vol. 2, IEEE, 2000, pp. 1433–1438.

[35] S.W. Yoon, A. Fern, R. Givan, S. Kambhampati, Probabilistic planning via determinization in hindsight. AAAI, (2008), pp. 1010–1016.

[36] M.L. Littman, A.R. Cassandra, L.P. Kaelbling, Learning policies for partially observable environments: scaling up, Machine Learning Proceedings 1995, Elsevier, 1995, pp. 362–370.

[37] G. Du, P. Zhang, Markerless human–robot interface for dual robot manipulators using Kinect sensor, Rob. Comput. Integr. Manuf. 30 (2) (2014) 150–159.

[38] iai kinect package, 2018, [Online]. Available: https://github.com/code-iai/iaikinect2.

[39] apriltags website, 2018, [Online]. Available: https://april.eecs.umich.edu/software/apriltag.html.

[40] VISP calibration package, 2018, [Online]. Available: http://wiki.ros.org/visp_hand2eye_calibration.

[41] Moveit! simple grasps package, 2018, [Online]. Available: https://github.com/davetcoleman/moveit_simple_grasps.

[42] k2 bridge package, 2018, [Online]. Available: https://github.com/personalrobotics/k2_bridge.

[43] A.D. Dragan, S. Bauman, J. Forlizzi, S.S. Srinivasa, Effects of robot motion on human-robot collaboration, Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, ACM, 2015, pp. 51–58.