Probabilistic Decision Model for Adaptive Task Planning in Human-Robot Collaborative Assembly Based on Designer and Operator Intents

Martijn Cramer, Karel Kellens, and Eric Demeester

***Abstract*—In the manufacturing industry, the era of mass customization has arrived. Combining the complementary strengths of humans and robots will allow to cope with growing product diversity and fluctuating demands. When collaborating, humans and robots operate in a shared workspace to assemble a common product and it has to be decided how tasks are distributed between them. In previous work, human operators have often been modeled as controllable agents whose actions can be planned. To achieve a natural collaboration, we believe that the robot has to estimate the operator's assembly intentions and should behave accordingly. This work presents a framework for planning collaborative robot tasks in assembly, considering both the designer and operator intents. From the designer's CAD data, a set of potential assembly plans is automatically derived and translated into a state graph from which the operator intentions follow. To enable the robot to act optimally under a given belief of operator intentions, a partially observable Markov decision process (POMDP) is formulated, whose state space is represented by the intention graph. Experimental results obtained with our approach demonstrate the potential to estimate the true operator intent based on observations about the parts and tools manipulated during collaborative assembly.**

***Index Terms*—Human-robot collaboration, assembly, task planning, intention recognition, probabilistic inference.**

I. Introduction

In recent years, the manufacturing industry is shifting towards mass customization. This customer-driven market demands for adaptive assembly lines for which traditional manual labor or full automation often fall short. Combining the complementary strengths of both worlds—human's intelligence and adaptability along with robot's precision, repeatability, and strength—will allow to deal with growing product diversity and fluctuating demands [1].

When collaborating in the manufacturing industry, humans and robots may operate in a shared workspace to assemble a common product. To achieve a more natural and adaptive robot behavior: (1) the robot should be aware of possible assembly plans, its own and its human partner's capabilities, and the work environment [2] that could result from the product and assembly information specified by the designer (referred to as the "designer intent"); and (2) the robot should actively anticipate in performing assembly operations by estimating the operator's preferred sequence of assembly [3] (referred to as the "operator intent").

Previous research [4], [5] often regards human operators as highly dexterous manipulators whose tasks can be planned and scheduled in advance therewith assuming that they always will strictly follow the imposed schedule, similar to how robots behave. In [6], [7], collaborative assembly plans are represented as AND/OR graphs on which the A* heuristic graph-search algorithm is unleashed to find the most optimal assignment of agents to assembly operations in human-robot teams. The allocation of tasks is steered by cost metrics that account for human workload and risk, execution time and complexity [8], and resource costs. Deviating online from these fixed task sequences is prohibited. What assembly tasks to perform by whom and in what order are predefined.

Some documented approaches offer human co-workers partial freedom in terms of how, where, or when to complete the assigned assembly operations. In [2], [9], the robot is able to adapt its motion according to its human partner's preferences by recognizing the human's activities and tracking the manipulated objects. [10], on the other hand, estimates when the operator is most likely to start the preassigned tasks to schedule robot activities according to the operator's timing.

In this work, operators are free to assemble the product in the order they desire, provided that this sequence is within their physical capabilities, and practically feasible i.e. the required tools are at hand, and the subassemblies constructed during operation may not conflict with topological, geometrical, and technical constraints (see Section II-B). These constraints not only impose the feasibility of a (sub)assembly but also significantly reduce the number of assembly sequences that would otherwise amount to the factorial of the assembly's size.

The robot has to consider human intentions in its decision-making rather than to force the operator to follow a strict, predefined assembly plan. This assembly sequence may vary between or even during executions. Approaches that share some similarities with this research modeled the collaborative task as a hierarchical task network (HTN), including AND/OR

Manuscript received February 24, 2021; accepted June 20, 2021. Date of publication July 8, 2021; date of current version August 2, 2021. This letter was recommended for publication by Associate Editor N. Yamanobe and Editor G. Yamanobe upon evaluation of the reviewers' comments. This work was supported by the Research Foundation - Flanders (FWO) under grant agreement 1SA6919N. *(Corresponding author: Martijn Cramer.)*

The authors are with the KU Leuven, Diepenbeek Campus, Dept. of Mechanical Engineering, Research unit ACRO, B-3000 Leuven, Belgium (e-mail: martijn.cramer@kuleuven.be; karel.kellens@kuleuven.be; eric.demeester@kuleuven.be).

Digital Object Identifier 10.1109/LRA.2021.3095513

graphs [11], or self-designed variants [12], [13]. In [11], [12], this task representation is converted to a probabilistic graphical model, whereas [13], [14] directly employ 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.

From the aforementioned literature, only some discussed the generation of assembly plans from prior knowledge (i.e. the designer intent). In [12], task hierarchies are constructed from instructor-provided action sequences learned by demonstration, which are often difficult and time-consuming to obtain. [5] directly extracts product characteristics (i.e. weight, size, and flexibility) from CAD data for allocating humans and robots to assembly tasks. Finally in [7], an assembly plan is deduced using manually annotated CAD data with connectivity information.

The presented research builds towards a more natural and efficient human-robot collaboration by taking into account both designer and operator intents. Section II introduces the concept of designer intent with an overview of useful product and assembly information for constructing collaborative assembly plans. Next, the representation of assembly plans as AND/OR graphs and their automatic generation from the assembly product's CAD data is described. Throughout this work, Bourjault's ballpoint pen [15], which has evolved into a benchmark among researchers, is used as an example use case. Even though this assembly consists of only six components, its assembly complexity is representative to numerous industrial products with many more components. Indeed, the assembly of an entire product is often divided over several workstations. The size and complexity of such subassembly are constrained by the workstation's physical size, the number of operators and their abilities [16]. Therefore, the authors believe that this six-part pen represents a typical product that could be assembled by a single workstation, which forms the scope of this work. In Section III, the assembly's AND/OR graph is translated into the state graph from which a novel representation of operator intentions follows. The set of intentions is translated into an intention graph which is used to formulate a probabilistic decision model (POMDP) for planning robot actions in presence of operator intention ambiguity and perception uncertainty. Section IV experimentally validates the proposed framework in a collaborative assembly scenario. Section V discusses the obtained results and points out future research directions to overcome current limitations and simplifications when scaling up to more complex assemblies. Finally, Section VI concludes with a summary of this work.

II. Designer Intent

The designer intent includes the designer's knowledge about both the individual components as well as the entire assembly process, and is useful for several purposes:

- Geometric dimensions and CAD models of the assembly components can be used for object recognition and localization, motion and grasp planning, and assembly planning.
- Geometric tolerances between the different assembly components ensure practical manufacturability that is useful for allocating tasks between human and robot. For instance, if the robot is unable to establish a peg-in-hole fitting due to sensorimotoric limitations, this task should be outsourced to a human operator.
- Material properties, such as an assembly component's density and volume, allow determining whether the human or robot is capable of performing the assembly operation ergonomically and safely. Based on Young's modulus, it can be opted to leave the handling of soft materials to humans.
- Spatial relationships between assembly components can be used for motion and grasp planning, instruction generation, and assembly planning and verification.
- Connector objects (e.g. circlip, bolt, screw, nut) often require the use of dedicated tools that one of the agents may or may not have at hand.
- The assembly tree possesses the structure of subassemblies created by the designer that can be used for assembly planning.

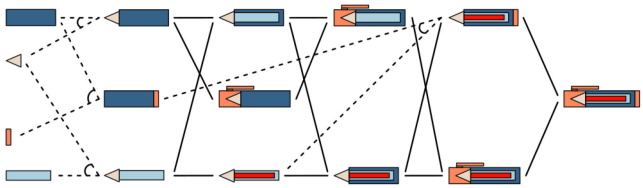

Fig. 1. AND/OR graph of Bourjault's six-part ballpoint pen [15]. Dashed lines connected by an arc denote the AND relationship between parent subassemblies and their children. Solid lines concisely represent AND relationships in which edges to child vertices of individual components are omitted. Multiple hyperarcs emerging from the same parent subassembly indicate the OR relationships.

In order to construct the assembly plan, we extract this component and assembly information using an in-house created workbench in FreeCAD: the open-source, cross-platform 3D CAD/CAE parametric modeler built on top of Open Cascade's geometric kernel. The reader is referred to the code repository[1] and accompanying wiki for details regarding the setup, usage, and source code of this workbench.

For now, this work only uses geometric information and spatial relationships to generate the assembly plan. However, the modular design of this workbench allows to scale to the full designer intent as well as additional information, such as the estimated time per assembly operation, work cell layout, and robot specifications.

A. Representing Assembly Plans

In the early 1990s, Homem de Mello and Sanderson [17] represented for the first time decomposable production systems, such as (dis)assembly plans, as AND/OR graphs. When viewed from left to right, Fig. 1 depicts the assembly plan of Bourjault's ballpoint pen; when viewed in the opposite direction, the disassembly plan can be deduced.

The AND/OR graph provides a dense representation of a product's possible (dis)assembly sequences, and can be formally described as a directed hypergraph $H = (V, A)$. The vertices

[1] https://github.com/martcram/FreeCAD-APLAN

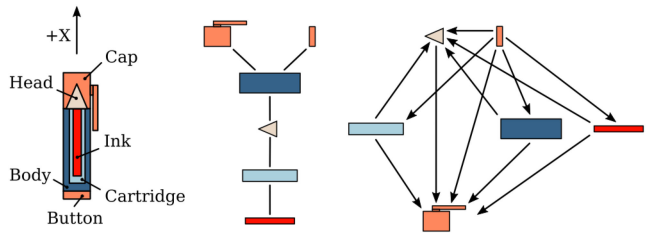

Fig. 2. Connection diagram (middle) and obstruction diagram along the positive x-axis (right) of Bourjault's six-part ballpoint pen (left).

$V_H \subseteq \mathcal{P}(C) \setminus \{\emptyset\}$ of this hypergraph, with C being the set of assembly components, symbolize feasible subassemblies. These consist of the individual components (i.e. H's sink vertices), combinations of individual components, and the complete assembly (i.e. H's source vertex). As the name suggests, an AND/OR graph's vertices relate in two ways: (1) AND relationship: every hyperarc $a_H \in A_H \subseteq \{(X,Y)|X \in V_H \wedge Y \subset V_H \wedge X \notin Y\}$ represents the disassembly of one parent subassembly X into multiple child subassemblies Y; (2) OR relationship: a subassembly can often be shaped in various ways as represented by two or more hyperarcs originating from the same parent vertex and pointing to different sets of child vertices.

B. Generating Assembly Plans

Lambert and Gupta [18] thoroughly discussed different techniques for constructing (dis)assembly plans H using precedence relationships. In order to keep the generation of possible assembly sequences combinatorially tractable, the reverse cut-set algorithm [18, p. 188] has been applied to automatically construct AND/OR graphs, instead of its non-reversed counterpart. In addition, only feasible subassemblies were considered by checking against three types of constraints.

1) Topological constraints are graphically represented by the connection diagram [15]: an undirected graph $G_C = (V,E)$ whose vertices $V_{G_C} = C$ depict the individual assembly components C, and whose edges $E_{G_C} \subseteq \{\{x,y\}|x,y \in C \wedge x \neq y\}$ represent the (mechanical) connections between these components (Fig. 2, middle). A subassembly can be viewed as a subgraph $G_S = (V,E)$ from the connection diagram G_C where $V_{G_S} \subseteq V_{G_C}$ and $E_{G_S} \subseteq E_{G_C}$, and is topologically feasible if $\forall v \in V_{G_S}$: $\delta(v) > 0$, with $\delta(v)$ being the degree of vertex v. For instance, the subgraph composed of the ball pen's cap and button is disconnected, consequently their subassembly is topologically unfeasible.
2) A (dis)assembly operation along a certain direction is geometrically constrained when a component's movement is obstructed by other components. Identifying these geometric constraints is important to avoid deadlocks: a situation wherein further assembly is halted in case undoing previous assembly operations is prohibited. These geometric constraints are derived from the product's obstruction diagrams [18]: directed graphs $G_O = (V,E)$ whose vertices correspond to the assembly's individual components $V_{G_O} = C$, and whose directed edges $E_{G_O} \subseteq \{(x,y)|(x,y) \in C^2 \wedge x \neq y\}$ point to the components y that block the parent component x when disassembled along the related direction (Fig. 2, right). For instance, the ball pen's head is obstructed by the cap when disassembled along the positive x-axis. Obstruction diagrams are constructed along the three positive Cartesian directions; those along the opposite directions are these graphs' transposes, e.g. $(G_O^{X-}) = (G_O^{X+})^T$.
3) There may exist some undesired operations from a technical perspective due to the product's structure, physical properties, or the absence of certain tools [18]. These cases are handled by incorporating suitable technical constraints when generating the assembly plan, which are represented by the pairs $T \subseteq \{(R_1,R_2)|R_1 \subseteq C \wedge R_2 \in C \wedge R_2 \notin R_1\}$ such that subassembly $S \subseteq C$ is technically unfeasible if $\exists (R_1,R_2) \in T : R_1 \subseteq S \wedge R_2 \notin S$. The ball pen is technically constrained by preventing the insertion of the ink into its cartridge before the head is mounted. This leads to the fact that in the AND/OR graph (Fig. 1) no subassembly exists where both the ink and cartridge R_1 are present without the pen's head R_2. Finally, these technical constraints are manually added, while the connection and obstruction diagrams are automatically computed using our FreeCAD workbench.

III. Operator Intent

A. Representing Assembly States

The assembly state graph $G_{ST} = (V,E)$ is built from the previously constructed assembly plan H as described by Alg. 1. The vertices $V_{G_{ST}} \subset \mathcal{P}(V_H) \setminus \{\emptyset\}$ of this directed acyclic graph symbolize the different states (i.e. the collection of subassemblies at that moment) in which the assembly can reside. The edges $E_{G_{ST}} \subseteq \{(X,Y)|(X,Y) \in V_{G_{ST}}^2 \wedge X \neq Y\}$ connecting these vertices, correspond to the different assembly operations (i.e. the AND relationships A_H in the AND/OR graph) that result in transitions between assembly states. This mapping is defined by m_{ST} (Alg. 1).

Fig. 3 (top) shows the assembly state graph of Bourjault's ballpoint pen. The state graph diverges from the initial assembly state that consists of the set of individual components; the graph terminates in the final assembly state that represents the completed product.

B. Representing Assembly Intentions

In order for the robot to anticipate the actions of its human partner properly, it is important to consider his/her intentions in its decision-making. An operator intention is modeled as the sequence of consecutive assembly states that the operator has in mind from the current moment on.

This graphically translates into various paths through the assembly state graph, starting from the current state and ending into the final goal state. Note that this intention representation shares similarities with the proposed driver intentions in [19] for assisting wheelchair drivers in navigating. In Fig. 3 (top), the yellow and green paths show two examples of operator intentions

Algorithm 1 Assembly state graph generation

Input: AND/OR graph $H = (V, A)$
Output: Assembly state graph $G_{ST} = (V, E)$,
Map $m_{ST} : E_{G_{ST}} \mapsto A_H$

$G_{ST} \leftarrow$ empty directed graph (V, E)
H's source vertices:
$\mathbb{V}_H^{src} \leftarrow \{V^{src} | V^{src} \in V_H \wedge \delta_H^-(V^{src}) = 0\}$
Open assembly states: $\mathbb{S}_O \leftarrow \{\{V^{src}\} | V^{src} \in \mathbb{V}_H^{src}\}$
while $\mathbb{S}_O \neq \emptyset$ **do**
  $S_n \leftarrow$ last element of $\mathbb{S}_O$
  for all subassembly $X \in S_n$ **do**
    for all outgoing hyperarc $a_H^+ \in A_H^+(X)$ **do**
      $Y \leftarrow$ terminal vertices of a_H^+
      $S'_n \leftarrow (S_n \setminus X) \cup Y$
      $e_{ST} \leftarrow (S'_n, S_n)$
      $V_{G_{ST}} \leftarrow V_{G_{ST}} \cup \{S_n, S'_n\}$
      $E_{G_{ST}} \leftarrow E_{G_{ST}} \cup \{e_{ST}\}$
      $m_{ST} : e_{ST} \mapsto a_H^+$
      $\mathbb{S}_O \leftarrow \mathbb{S}_O \cup \{S'_n\}$
  $\mathbb{S}_O \leftarrow \mathbb{S}_O \setminus S_n$

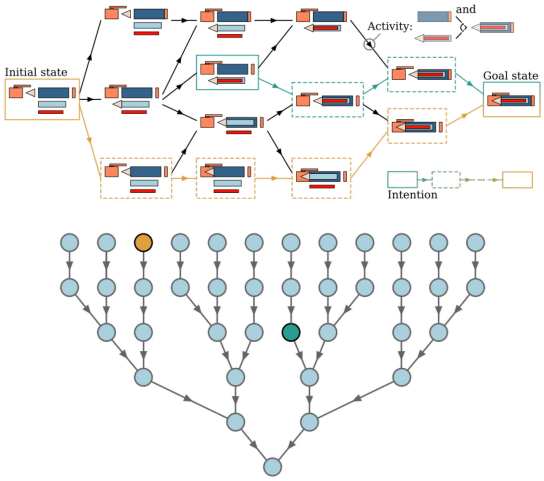

Fig. 3. Assembly state graph of Bourjault's ball pen, starting with the individual components and ending with the complete assembly, depicting two possible operator intentions at different points in time (top). The derived intention graph highlighting the vertices that correspond to these two operator intentions (bottom).

for the ballpoint pen's assembly. From the assembly state graph G_{ST}, the directed intention graph $G_I = (V, E)$, shown in Fig. 3 (bottom), can be constructed as described by Alg. 2. The vertices $V_{G_I} \subset \mathcal{P}(V_{G_{ST}}) \setminus \{\emptyset\}$ and edges $E(G_I) \subseteq \{(X, Y) | (X, Y) \in V_{G_I}^2 \wedge X \neq Y\}$ represent the set of operator intentions, and the possible transitions between these intentions, respectively. m_I (Alg. 2) defines the mapping between the intention graph's edges E_{G_I} and the assembly operations A_H (i.e. AND relationships).

C. Formulating the Decision Model

Since the operator intent is inherently part of the mental state, it is not directly observable through sensing or perception and therefore modeled as a hidden variable in a partially observable

Algorithm 2 Intention graph generation

Input: Assembly state graph $G_{ST} = (V, E)$,
Map $m_{ST} : E_{G_{ST}} \mapsto A_H$
Output: Intention graph $G_I = (V, E)$,
Map $m_I : E_{G_I} \mapsto A_H$

$G_I \leftarrow$ empty directed graph (V, E)
G_{ST}'s sink vertices:
$\mathbb{V}_{G_{ST}}^{snk} \leftarrow \{V^{snk} | V^{snk} \in V_{G_{ST}} \wedge \delta_{G_{ST}}^+(V^{snk}) = 0\}$
Open intentions: $\mathbb{I}_O \leftarrow \{\{V^{snk}\} | V^{snk} \in \mathbb{V}_{G_{ST}}^{snk}\}$
while $\mathbb{I}_O \neq \emptyset$ **do**
  $I_n \leftarrow$ last element of $\mathbb{I}_O$
  $Y \leftarrow$ last element of I_n
  for all incoming edge $e_{G_{ST}}^- \in E_{G_{ST}}^-(Y)$ **do**
    $X \leftarrow$ origin vertex of $e_{G_{ST}}^-$
    $I'_n \leftarrow I_n \cup \{X\}$
    $e_I \leftarrow (I'_n, I_n)$
    $V_{G_I} \leftarrow V_{G_I} \cup \{I_n, I'_n\}$
    $E_{G_I} \leftarrow E_{G_I} \cup \{e_I\}$
    $m_I : e_I \mapsto m_{ST}(e_{G_{ST}}^-)$
    $\mathbb{I}_O \leftarrow \mathbb{I}_O \cup \{I'_n\}$
  $\mathbb{I}_O \leftarrow \mathbb{I}_O \setminus I_n$

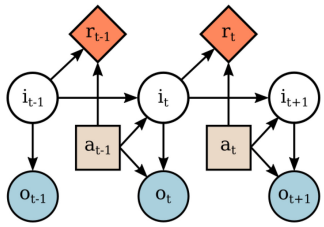

Fig. 4. Dynamic Bayesian network representation of the employed POMDP in this work. The hidden layer consists of the different operator intentions whereby the observable layer represents the parts and tools handled when performing assembly operations.

Markov decision process (POMDP). A POMDP allows to probabilistically plan robot actions in presence of uncertainty (due to operator intentions, or imperfect perception and actuation) by observing operator actions and maximizing the expected sum of future rewards over a finite time horizon. The resulting robot policy defines the best action to perform under a given belief.

The POMDP designed in this work (Fig. 4), is formally described by the seven-tuple, $\{I, A, T, R, \Omega, O, \gamma\}$:

- I is the set of operator intentions, modeled as hidden variables in the POMDP. An operator intent $i \in I$ represents the mental path through the assembly state graph from the current assembly state towards the goal state. As a result, the collection of operator intentions corresponds to the intention graph's vertices: $I = V_{G_I}$.
- A is the robot's action repertoire: the set of feasible assembly operations A_H the robot is capable of executing, extended with a wait action a_w: $A = A_H \cup \{a_w\}$.
- $T : I \times A \times I \to \mathbb{R}$, the state transition function that gives the probability of the operator changing intentions in the next timestep $i_{t+1} \in I$ given the current intention $i_t \in I$ and the assembly operation $a_t \in A$ executed by the robot

(see Alg. 3, line 6). Possible transitions between different intentions are deduced using the intention graph (Fig. 3, bottom). In this work, a uniform probability function over potential successive intentions I_{t+1} is employed.

- $R : I \times A \to \mathbb{R}$, the reward function that specifies the immediate reward received by the robot when performing assembly operation $a \in A$ under operator intention $i \in I$ (see Alg. 3, line 15). Desired robot actions A_I for a given intent i follow from the directed edges $E^+_{G_I}(i)$ pointing outwards from the corresponding vertex in the intention graph. Moreover, the designer is requested beforehand to list assembly operations $A_U \subseteq A$ that are unfavorable (e.g. non-ergonomic, dangerous, or repetitive) to humans. The robot will receive the highest positive reward (+10.0) when performing such action $a \in A_U$, provided that it is aligned with the operator's intent $a \in A_I$. Contrary behavior $a \notin A_U$ will be punished with negative rewards: -2.0 and -50.0, depending whether or not the robot action a matches the operator's intent: $a \in A_I$ or $a \notin A_I$. By adjusting the latter cost, the user can alter the robot's eagerness to act. At higher values, the system is more likely to wait until it is almost sure of the operator's intent; at lower values, the system will start to act after a few observations. These actions may mismatch the operator's intent, which may require mutual adaptation including from the human operator. At any time, waiting $a = a_w$ is tolerated with no reward (0.0). These reward values are determined empirically.
- Ω is the set of observations perceived by the robot. Observations consist of the different assembly components and tools picked from the storage by the human operator when performing an assembly operation. The mapping between assembly operations and observations is given by $m_\Omega : A \to \Omega$. These observations can be categorized as: (1) extending an existing subassembly that is done using only a single (disjoint) component and the necessary tool, (2) creating a new subassembly that is done using two (disjoint) components and the necessary tool, and (3) combining two existing subassemblies that is done using only the necessary tool.
- $O : I \times A \times \Omega \to \mathbb{R}$, the observation function that gives the probability of observing operator action ω_t under intention i_t when the robot performed action a_{t-1} in the previous timestep (see Alg. 3, line 19). In this work, a uniform probability function p_{uni} over potential observations Ω_I is employed, whereby a low probability ($p_{uni}/4$) is assigned in case the assembly operation was performed correctly, but did go unnoticed by the vision system. At that moment, the vision system will register the same (blank) observation as perceived after performing the wait action (i.e. $\omega_t = m_\Omega(a_w)$).
- $\gamma \in [0, 1]$ represents the discount factor used to penalize future rewards for which a value of 0.9 is selected.

Algorithm 3 POMDP formulation

1: Intention graph: $G_I = (V, E)$
2: Operator intentions: $I \leftarrow V_{G_I}$
3: Robot actions: $A \leftarrow A_H \cup \{\text{wait action } a_w\}$
4: Unfavorable operator actions: $A_U \subseteq A$
5: Maps $m_\Omega : A \mapsto \Omega$, and $m_I : E_{G_I} \mapsto A_H$
6: **function** $\mathbf{T}(i_t \in I, a_t \in A, i_{t+1} \in I)$
7: Intermediate intentions after a_t: $I_{t+0.5} \leftarrow \emptyset$
8: **if** $a_t = a_w$ **then**
9: $I_{t+0.5} \leftarrow \{i_t\}$
10: **else**
11: $I_{t+0.5} \leftarrow \{i | i \in N^+_I(i_t) \wedge m_I((i_t, i)) = a_t\}$
12: Potential intentions after a_t under i_t:
$I_{t+1} \leftarrow \bigcup(\{I_{t+0.5}\} \cup \{N^+_I(i) | i \in I_{t+0.5}\})$
13: $p \leftarrow \begin{cases} 1/|I_{t+1}| & \text{if } i_{t+1} \in I_{t+1}, \\ 0.0 & \text{else.} \end{cases}$
14: **return** p
15: **function** $\mathbf{R}(i \in I, a \in A)$
16: Desired robot actions under i:
$A_I \leftarrow \{a_w\} \cup \{m_I(e) | e \in E^+_{G_I}(i)\}$
17: $r \leftarrow \begin{cases} 10.0, & \text{if } a \in A_I \wedge a \in A_U, \\ -2.0, & \text{if } a \in A_I \wedge a \notin A_U, \\ -50.0, & \text{if } a \notin A_I, \\ 0.0, & \text{if } a = a_w. \end{cases}$
18: **return** r
19: **function** $\mathrm{O}(i_t \in I, a_{t-1} \in A, \omega_t \in \Omega)$
20: Wait (blank) observation: $\omega_w \leftarrow m_\Omega(a_w)$
21: Potential operator actions after a_{t-1} under i_t:
$A_t \leftarrow \{m_I((i_{t-0.5}, i_t)) | i_{t-0.5} \in N^-_{G_I}(i_t) \wedge$
$i_{t-1} \in N^-_{G_I}(i_{t-0.5}) \wedge a_{t-1} = m_I((i_{t-1}, i_{t-0.5}))\}$
22: Potential observations after a_{t-1} under i_t:
$\Omega_I \leftarrow \{\omega_w\} \cup \{m_\Omega(a_t) | a_t \in A_t\}$
23: Identity: $p_{uni} \cdot |\Omega_I \setminus \omega_w| + p_{uni}/4 \cdot |\{\omega_w\}| \equiv 1.0$
24: $p \leftarrow \begin{cases} p_{uni}/4, & \text{if } \omega_t \in \Omega_I \wedge \omega_t = \omega_w, \\ p_{uni}, & \text{if } \omega_t \in \Omega_I \wedge \omega_t \neq \omega_w, \\ 0.0, & \text{if } \omega_t \notin \Omega_I. \end{cases}$
25: **return** p

IV. Experiments

The performance of the presented intention recognition framework for collaborative task planning was evaluated in a real assembly scenario in which a human operator and robot take turns assembling the six-part ballpoint pen in three different orders.

From the ball pen's CAD models, the corresponding assembly plan and POMDP model were automatically constructed without user intervention. The interested reader is referred to the code repository[2] for more details about our programmatic implementation. Technical constraints were excluded during plan construction. The CAD models of all products discussed in this work, are available in FreeCAD's part library on GitHub.[3] The generated POMDP for this six-part assembly is characterized by 69 operator intentions (states), 23 assembly operations (actions), and 12 different observations. The POMDP's state space graphically translates into the intention graph presented by

[2]https://github.com/martcram/hrc_pomdp
[3]https://github.com/FreeCAD/FreeCAD-library

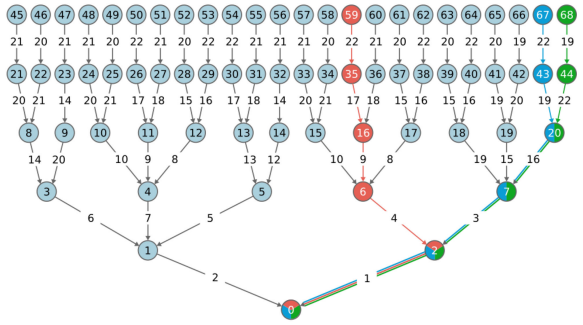

Fig. 5. Intention graph of Bourjault's six-part ballpoint pen together with the evolution of true operator intents for three different assembly sequences.

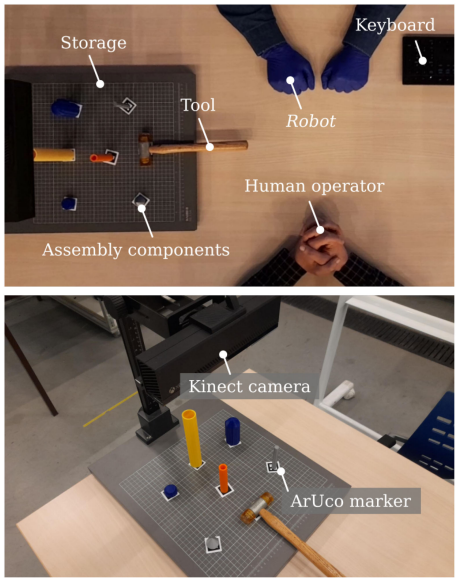

Fig. 6. Top-down view of our collaborative assembly setup (top). Vision system for detecting the absence of individual components or tool (bottom).

Fig. 5 of which the colored paths relate to the three experiments carried out.

Fig. 6 (top) presents the top-down view of our experimental setup. In order to demonstrate the effectiveness of our planner, without involving grasp and motion planning, the robot is impersonated by a person wearing blue gloves (referred to as "the robot"). The Robot Operating System's (ROS) action protocol is used for bi-directional communication between the POMDP planner, the vision system, and the robot. The planner (action client) starts by requesting the vision system's action server to observe the workplace and to identify the assembly operation previously performed by the human operator. Given this observation, the planner determines the next optimal action using its policy and sends a request to the robot's action server to perform an assembly operation or to wait.

At this moment, the robot's action server replies with textual instructions about the task at hand, which are displayed to the person taking the robot's role. The robot confirms the successful execution of the task using the keyboard. In the near future, these action specifications will be extended with the necessary

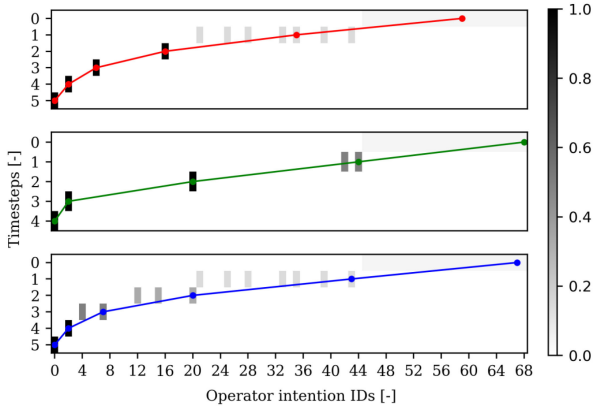

Fig. 7. Evolution of the posterior belief $bel(i_t)$ (in grayscale): the probability distribution over the set of operator intentions at time t given the most recent robot action a_t and observation ω_t, for the three assembly experiments. True operator intentions are denoted by colored lines.

instructions for a real robotic manipulator to correctly perform the assigned assembly operations.

The vision system observes the operator's actions by registering the parts or tools picked from the storage that contains the individual components of the product. Fig. 6 (bottom) offers a detailed view of our vision system in which each object covers a unique ArUco marker. These 4x4-sized markers become visible when components or tools are used by one of the agents and are identified using the *aruco_detect* ROS package on the RGB images (1080p, 30 fps) from the Kinect v2 camera that is mounted at the top of the setup. The *iai_kinect2* driver[4] serves for communication between ROS and this camera. An assembly operation ends when the available tool (i.e. hammer) is grasped and its ArUco marker is detected. This signals the system to record the components and tool taken and to send this observation to the planner.

The point-based SARSOP algorithm from the approximate POMDP planning (APPL) toolkit was chosen for computing the robot policy, which defines the best action to perform under a given belief. After converting the problem representation to the XML-based POMDPX file format, an approximate value function after 1000 trials was found offline in (131.1 ± 3.3) seconds (mean $\pm$ SD, n $= 5$) that is comprised of (382 ± 121) α-vectors (mean $\pm$ SD, n $= 5$). The time needed to create the AND-OR, assembly state, and intention graphs, as well as the POMDP model, constitutes only a fraction of the time required for solving this probabilistic model, namely $(1047.1 \pm 1.9) \times 10^{-3}$ seconds (mean $\pm$ SD, n $= 100$). The inner product of the α-vectors and the current belief results in the robot policy. The experiments were performed on a PC with an i7-7820HQ Intel processor (2.90 GHz) and 32 GB memory.

Fig. 7 pictures the probability distributions over the set of operator intentions at the different timesteps for the three assembly processes. The darker the cell's color the more probable the operator intention. Each subplot is related to a certain assembly

[4]https://github.com/code-iai/iaikinect2

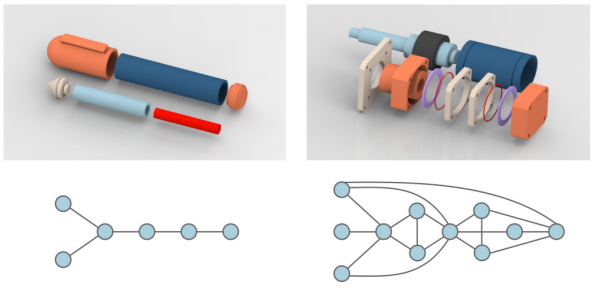

Fig. 8. Connection diagrams of Bourjault's ballpoint pen (left) and the eleven-part pneumatic cylinder (right).

TABLE I
SPECIFICATIONS OF VARIOUS ASSEMBLIES: (1) BOURJAULT'S BALLPOINT PEN, (2) PNEUMATIC CYLINDER, (3) WEAKLY CONNECTED PRODUCT, (4) STRONGLY CONNECTED PRODUCT

Product no.	#1	#2	#3	#4
Parts	6	11	11	11
Subassemblies	18	60	66	1,023
Assembly states	18	225	1,024	678,570
Actions	23	106	220	86,526
Intentions	69	112,497	-	-
Complexity α	1.67	3.27	1.82	10.0

sequence followed by the operator. The evolution of true operator intents relate to the paths in Fig. 5 and are drawn in matching colors for clarity. During the first two experiments, the robot correctly infers the human's intention after performing only two assembly operations by the operator. This is shown by the unimodal probability distribution over operator intentions from timestep 2 onward. In the third experiment, the same observation is made from timestep 4. After these points in time, the robot is sure about what assembly sequence the operator desires and starts taking collaborative actions towards the common goal, which follows from our reward function. A supportive video of the conducted experiments can be found via following link.[5]

V. DISCUSSION

The previous sections offered an insight into the proposed framework for estimating operator intentions and planning collaborative robot tasks for assembly. In contrast to related work [6], the operator is regarded as an uncontrollable entity whose actions cannot be planned and scheduled beforehand. It is the robot's task to recognize the operator's intent about what assembly sequence to follow and to act accordingly. We demonstrated that from the CAD model of a six-part product, a probabilistic model can be constructed automatically that is able to assess the true operator intent, for three real assembly scenarios, after only a few observations. From that moment on, the robot starts participating and offers collaborative actions to the user.

In addition, the scalability of our approach to a larger industrial assembly is examined. The comparison between the ballpoint pen and an eleven-part pneumatic cylinder (Fig. 8, top) shows that for an assembly consisting of five additional parts, the number of operator intentions significantly increases (Table I: #1 vs. #2). This can be explained by the index of complexity $\alpha = 2K/N$ that defines the average number of connections (K) per component (N) [18].

In practice, this quantity possesses a value between two and four [20], which is closer to the theoretical lower bound value defined by an eleven-part weakly connected product than to the upper bound value defined by a strongly connected product of the same size (Table I: #3 vs. #4). The pneumatic cylinder's greater α denotes that on average almost twice as many connections between the cylinder's components exist than for the ball pen (Fig. 8, bottom). More connections result in more feasible subassemblies, which in turn leads to more assembly states and operator intentions. By introducing technical constraints, the number of subassemblies, assembly states, and actions can be reduced below the lower bound of an eleven-part weakly connected product (Table I: #2 vs. #3). Although the SARSOP algorithm has shown to scale up well to large state spaces for non-trivial robotic applications [21], future work will consist of reducing the number of operator intentions and therewith the POMDP's state space.

Point-based POMDP solvers have shown to benefit from domain structure [22]. One option could consist of adopting a factored POMDP in which the state variable is split into two components: (1) the current assembly state in which the world resides, and (2) an altered representation of the operator intention, e.g. as the sequence of consecutive assembly states that the operator has in mind from the start of the assembly process. The latter part graphically translates into the path going from the initial assembly state (consisting of the set of basic parts) to the final state (consisting of the final assembled product). This allows a different transition function for each state variable. For the ball pen's assembly, the factored POMDP's state space is composed of 17 assembly states together with 24 operator intentions instead of 69 operator intentions using a flat state representation. The pneumatic cylinder's factored POMDP is 225 assembly states and 40,320 operator intentions in size compared to the initial 112,497 different states.

Factored POMDPs [23] reduce the complexity of the planning task by decomposing the state variable into independent features. Hierarchical POMDPs [24] are another interesting approach in this regard that could leverage the inherent structure of the assembly domain through action or state hierarchies.

A further reduction can be achieved by performing some kind of initialization. The most connected component defines the assembly's base part, e.g. the pneumatic cylinder's housing (Fig. 8, bottom right). When only assembly sequences starting with this base part are allowed, the number of intentions for the cylinder's factored POMDP decreases to 24,948.

The dynamics of the model: the state transition, observation, and reward functions, were initialized with uniform probability distributions and rules designed by the authors. The usage of prior domain knowledge allowed to successfully plan robot tasks

[5]https://www.youtube.com/channel/UCe00cHGgGXY0qJhgmg_hNRw

for different operator intentions in the absence of extensive training data. However, future work will examine methods for tuning the model to operator preferences using observations recorded online. Furthermore, techniques to enable parallel operator and robot actions, and to recover from incorrect assembly operations will be studied. This will contribute to the system's robustness and its acceptance by the industry.

Finally, the effectiveness of our approach is shown in a collaborative assembly scenario wherein the robot's role was taken by a person. This allowed to first focus on the details and performance of the framework rather than to deal with motion and grasp planning. Future work will involve validating our approach with a real robotic manipulator. Finally, users' experiences will be subjected to a qualitative analysis involving human participants of different ages and levels of expertise as also being perceived in the manufacturing industry.

VI. Conclusion

To enable a more natural and efficient human-robot collaboration, the human operator is regarded as an uncontrollable agent whose actions cannot be planned and scheduled beforehand. The robot has to estimate the operator's intent about the sequence of assembly he/she desires and should act accordingly. This work proposes a framework for planning collaborative robot tasks in assembly considering both the designer and operator intents.

Based on the product and assembly information, mainly obtained from CAD data, the assembly's AND/OR graph is automatically constructed by the reversed cut-set algorithm considering topological, geometrical, and technical constraints. This assembly plan is subsequently translated into the assembly state graph from which a novel representation of operator intentions—as mental paths of consecutive assembly states—follows. To enable the robot to select the most optimal action under a given belief of operator intentions, a probabilistic decision model (POMDP) is formulated, whose state space is graphically represented by an intention graph. This paper demonstrated that the presented approach is able to assess the true operator intent in a real collaborative assembly after only a few observations. From that moment on, the robot starts participating and offers collaborative actions to the user.

Future work consists of improving scalability by leveraging the inherent structure of the assembly domain in the probabilistic model and validating the proposed framework in a collaborative assembly scenario with a robotic manipulator.

References

[1] M. Cramer, J. Cramer, K. Kellens, and E. Demeester, "Towards robust intention estimation based on object affordance enabling natural human-robot collaboration in assembly tasks," Procedia CIRP, vol. 78, pp. 255–260, 2018.

[2] S. C. Akkaladevi, M. Plasch, A. Pichler, and B. Rinner, "Human robot collaboration to reach a common goal in an assembly process," in *Proc. 8th Eur. Starting AI Researcher Symp., ser. Front. Artif. Intell. Appl.*, vol. 284, IOS Press, 2016, pp. 3–14.

[3] J. Shah, J. Wiken, B. Williams, and C. Breazeal, "Improved human-robot team performance using chaski, a human-inspired plan execution system," in *Proc. 6th ACM/IEEE Int. Conf. Hum.-Robot Interaction*, 2011, pp. 29–36.

[4] P. Tsarouchi, A.-S. Matthaiakis, S. Makris, and G. Chryssolouris, "On a human-robot collaboration in an assembly cell," *Int. J. Comput. Integr. Manuf.*, vol. 30, no. 6, pp. 580–589, Jun. 2017.

[5] G. Michalos, J. Spiliotopoulos, S. Makris, and G. Chryssolouris, "A method for planning human robot shared tasks," *CIRP J. Manuf. Sci. Technol.*, vol. 22, pp. 76–90, 2018.

[6] L. Johannsmeier and S. Haddadin, "A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes," *IEEE Robot. Automat. Lett.*, vol. 2, no. 1, pp. 41–48, Jan. 2017.

[7] R. A. Knepper, D. Ahuja, G. Lalonde, and R. Daniela, "Distributed assembly with AND/OR graphs," in *Proc. Workshop AI Robot. Int. Conf. Intell. Robots Syst.*, 2014.

[8] A. A. Malik and A. Bilberg, "Complexity-based task allocation in human-robot collaborative assembly," *Ind. Robot*, vol. 46, no. 4, pp. 471–480, Jun. 2019.

[9] S. Pellegrinelli, H. Admoni, S. Javdani, and S. Srinivasa, "Human-robot shared workspace collaboration via hindsight optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 831–838.

[10] A. Casalino, F. Cividini, A. M. Zanchettin, L. Piroddi, and P. Rocco, "Human-robot collaborative assembly: A use-case application," IFAC-PapersOnLine, vol. 51, no. 11, pp. 194–199, 2018.

[11] K. P. Hawkins, S. Bansal, N. N. Vo, and A. F. Bobick, "Anticipating human actions for collaboration in the presence of task and sensor uncertainty," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 2215–2222.

[12] B. Hayes and B. Scassellati, "Autonomously constructing hierarchical task networks for planning and human-robot collaboration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 5469–5476.

[13] K. Darvish, E. Simetti, F. Mastrogiovanni, and G. Casalino, "A hierarchical architecture for human-robot cooperation processes," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 567–586, Apr. 2021.

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

[15] A. Bourjault, "Contribution à une approche méthodologique de l'assemblage automatisé: élaboration automatique des séquences opératoires," Ph.D. dissertation, Faculty of Science and Technology, Université de Franche-Comté, Besançon, France, Nov. 1984.

[16] J. Nicholas, *Lean Production for Competitive Advantage: A. Comprehensive Guide to Lean Methodologies and Management Practices*, 2nd Edition. New York, NY, USA: Taylor & Francis, 2018.

[17] L. S. Homem de Mello and A. C. Sanderson, "AND/OR graph representation of assembly plans," *IEEE Trans. Robot. Automat.*, vol. 6, no. 2, pp. 188–199, Apr. 1990.

[18] A. J. D. Lambert and S. M. Gupta, *Disassembly modeling for assembly, maintenance, reuse, and recycling," Ser. The St. Lucie Press Series Resour. Manage*. Boca Raton, FL, USA: CRC Press, 2005.

[19] E. Demeester, "Mobile robot path planning with explicit consideration of uncertainty due to state space and action space discretisation," in *Proc. 50th Int. Symp. Robot.*, 2018, pp. 314–321.

[20] D. E. Whitney, "Assembly in the 21st century," 2001, oral presentation at the International Symposium on Assembly and Task Planning.

[21] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Proc. Robotics: Sci. Syst.*, 2008, pp. 65–72.

[22] T. S. Veiga, M. T. J. Spaan, and P. U. Lima, "Point-based POMDP solving with factored value function approximation," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2512–2518.

[23] G. Shani, "Task-based decomposition of factored POMDPs," *IEEE Trans. Cybern.*, vol. 44, no. 2, pp. 208–216, Feb. 2014.

[24] J. Pineau, N. Roy, and S. Thrun, "A hierarchical approach to POMDP planning and execution," in *Proc. Workshop Hierarchy Memory Reinforcement Learn.*, 2001.