MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

PACE: Proactive Assistance in Human-Robot Collaboration through Action-Completion Estimation

De Lazzari, Davide; Terreran, Matteo; Giacomuzzo, Giulio; Jain, Siddarth; Falco, Pietro; Carli, Ruggero; Romeres, Diego

TR2025-064 May 20, 2025

Abstract

This paper introduces the Proactive Assistance through action-Completion Estimation (PACE) framework, de- signed to enhance human-robot collaboration through real-time monitoring of human progress. PACE incorporates a novel method that combines Dynamic Time Warping (DTW) with correlation analysis to track human task progression from hand movements. PACE trains a reinforcement learning policy from limited demonstrations to generate a proactive assistance policy that synchronizes robotic actions with human activities, minimizing idle time and enhancing collaboration efficiency. We validate the framework through user studies involving 12 participants, showing significant improvements in interaction fluency, reduced waiting times, and positive user feedback compared to traditional methods.

IEEE International Conference on Robotics and Automation (ICRA) 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

PACE: Proactive Assistance in Human-Robot Collaboration through Action-Completion Estimation

Davide De Lazzari[1], Matteo Terreran[1], Giulio Giacomuzzo[1], Siddarth Jain[2], Pietro Falco[1], Ruggero Carli[1], Diego Romeres[2]

***Abstract*— This paper introduces the Proactive Assistance through action-Completion Estimation (PACE) framework, designed to enhance human-robot collaboration through real-time monitoring of human progress. PACE incorporates a novel method that combines Dynamic Time Warping (DTW) with correlation analysis to track human task progression from hand movements. PACE trains a reinforcement learning policy from limited demonstrations to generate a proactive assistance policy that synchronizes robotic actions with human activities, minimizing idle time and enhancing collaboration efficiency. We validate the framework through user studies involving 12 participants, showing significant improvements in interaction fluency, reduced waiting times, and positive user feedback compared to traditional methods.**

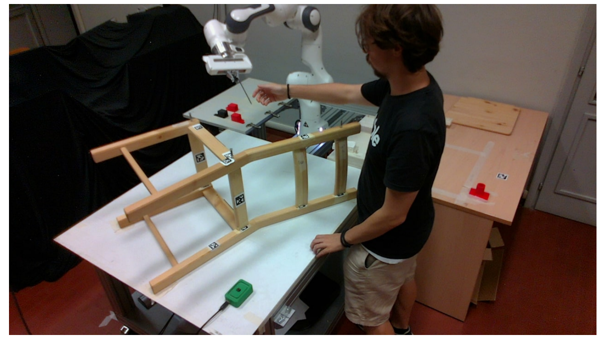

Fig. 1. Collaborative assembly process of an IKEA wooden chair. The robot provides timely assistance to the human operator, helping to move large/heavy parts or passing tools when needed.

I. Introduction

Recent advancements in robotics and artificial intelligence have significantly increased the use of robots alongside humans in industrial settings [1]. In particular, assembly tasks provide an ideal scenario for human-robot collaboration (HRC) applications, exploiting robot precision and speed to complement dexterity and adaptability of human workers. This multi-agent collaboration can lead to enhanced productivity and improved work quality for humans by allowing robots to take over the most repetitive and physically demanding tasks [2].

However, effective human-robot collaboration requires careful task planning and assignment, which is challenging due to the inherent variability in human performance. For instance, different operators may require varying amounts of time to complete the same task, even when following a predefined sequence of actions. Mutual understanding between humans and robots is essential for successful collaboration, and timing plays a critical role in shaping these interactions. In particular, humans are highly sensitive to the timing and smoothness of interactions, making time synchronization crucial for sharing resources and physical space [3].

Our work focuses on enabling the robot to proactively support the human operator, aiming to minimize idle times and reduce human effort. To achieve this, two primary challenges must be addressed: (i) accurately perceiving and predicting human actions and their temporal progression, and (ii) planning the robot's actions in such a way that assistance is synchronized with the human's needs.

This work was supported by the European Union – NextGenerationEU
[1]Department of Information Engineering, Università di Padova, Italy {delazzarid,terreran,giacomuzzo, falcopietro, carlirug,}@dei.unipd.it
[2]Mitsubishi Electric Research Laboratories, Cambridge, MA, USA {jain,romeres}@merl.com

To tackle these challenges, we first estimate human action completion percentage by employing a real-time adaptation of the Dynamic Time Warping (DTW) algorithm, specifically designed to track and accommodate variations in human hand movements. Then, we utilize data collected from demonstrations to train an optimal policy through reinforcement learning (RL), aiming to minimize idle times for both human and robot.

Our work offers the following key contributions:

- We present OS-DTW$_{\mathrm{WP}}$, a novel open-ended DTW algorithm for real-time human action completion estimation, incorporating a Windowed-Pearson (WP) distance.
- We introduce the Proactive Assistance through action-Completion Estimation (PACE) framework, which trains an RL agent that proactively assists a human operator by monitoring their progress with OS-DTW$_{\mathrm{WP}}$, minimizing waiting times and ensuring seamless human-robot synchronization.
- We validate the PACE framework with real-world experiments involving a chair assembly task with human participants. Our results demonstrate significant improvements in collaborative efficiency, supported by both quantitative metrics and subjective evaluations.
- We show that OS-DTW$_{\mathrm{WP}}$ outperforms existing DTW-based methods on real experimental data.

II. Related Works

Human-robot collaboration in assembly processes has traditionally relied on predefined plans to ensure predictability and safety. While this structured approach provides a clear framework for task execution, it often lacks the flexibility needed to adapt to the variable nature of human actions.

A significant body of research bypasses this limitation by focusing on task assignment, where specific tasks are allocated to either humans or robots to optimize collaboration efficiency. For instance, genetic algorithm-based strategies [4] and trust-based dynamic sub-task allocation methods [5] have been proposed to minimize assembly time and costs.

More recent studies have sought to enhance flexibility by introducing methods for re-planning robot actions based on human behavior, thereby improving adaptability and efficiency [6]–[10]. Many of these approaches operate under a leader-follower paradigm, where the human leads and the robot follows. However, research has shown that human workers prefer to maintain control over the task flow while expecting the robot to anticipate their needs and assist proactively [11]. This underscores the importance of enabling robots to act proactively, reducing reliance on explicit commands and enhancing collaboration fluidity. Yet, none of these existing methods actively monitor the human during action execution, limiting their ability to recognize and adapt to the current progress.

To our knowledge, only two studies have explored real-time monitoring of human movements at the action level. The first [12] employs Open-End Dynamic Time Warping (OE-DTW) [13] to estimate the progression of human tasks and dynamically re-plan robot actions. The second [14] proposes a method based on a Sigma log-normal model of human movements, claiming that it outperforms DTW for estimating action completion time. The same authors further utilize this approach for task planning in a collaborative assembly application [15], where tasks are assigned to the human by the robot. In this work, we demonstrate that OE-DTW, by itself, is not robust to the inherent variability of human movements. To address this limitation, we propose a new DTW-based algorithm capable of overcoming these challenges. Moreover, we integrate this method into a reinforcement learning framework, aiming to optimize robot performance directly without explicitly estimating action completion times as an intermediate step.

III. Problem Definition

We consider a scenario in which a human and a robotic manipulator perform separate tasks concurrently. The robot executes a sequence of M *robot-task* actions $\{\mathrm{a}_i^R\}_{i=1}^M$, which is repeated indefinitely. Simultaneously, the human performs a sequence of N *human* actions, denoted by $H = \{\mathrm{a}_j^H\}_{j=1}^N$. The human requires the robot's assistance to complete a subset of these actions, referred to as *joint* actions and denoted by $J = \{\mathrm{a}_l^J\}_{l=1}^L$, where $J \subseteq H$. We define the operator $\alpha(\cdot)$ to map the index of a joint action to the corresponding human action index, such that $\mathrm{a}_{\alpha(l)}^H = \mathrm{a}_l^J$. Without loss of generality, we assume that the last human action is a joint action (i.e., $\mathrm{a}_N^H = \mathrm{a}_L^J$), and no two consecutive human actions are joint actions (i.e., if $\mathrm{a}_j^H \in J$, then $\mathrm{a}_{j+1}^H \notin J$).

To assist the human, the robot must first complete the current robot-task action a_i^R before pausing its ongoing task. Once paused, the robot performs a *preparatory* action (e.g., repositioning or collecting a tool) to prepare for the joint action. After completing the joint action, the robot executes a *homing* action before either resuming with the robot-task or preparing for the next joint action. The sets of preparatory and homing actions are denoted as $\{\mathrm{a}_l^P\}_{l=1}^L$ and $\{\mathrm{a}_l^E\}_{l=1}^L$, respectively. A depiction of the collaborative task in the form of a hierarchical state machine is provided in Fig. 2.

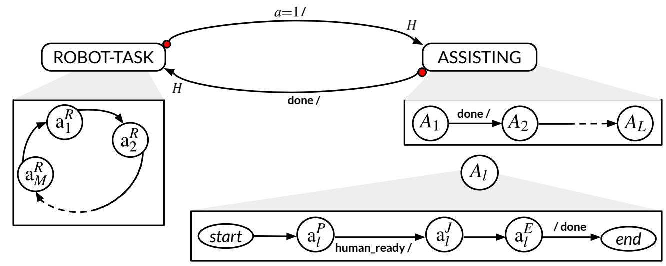

Fig. 2. Hierarchical state machine depicting the collaborative task from the robot perspective. The robot transitions from *ROBOT-TASK* to *ASSISTING* in between states a_i^R if $a = 1$. Once an (*assist*) action A_i is completed, the robot goes back to its task. The state machine follows the conventions as in [16]. Each transition is labeled with *guard / effect*. A transition is taken on a reaction only if the *guard* holds true.

Additionally, we assume to have access to a set of Q demonstrations, consisting of human trajectories $Y_j = \{\mathbf{y}_k^j\}_{k=1}^Q$ for each *non-joint* human action $H \setminus J$. Each trajectory consists of the Cartesian position of the human hand along the x-, y-, and z-axes.

The objective is to minimize idle times for both robot and human operator, i.e., the time each agent waits for the other before starting the joint actions. We define the cost function:

$$C(\Delta^R_{\text{total idle}}, \Delta^H_{\text{total idle}}) := \Delta^R_{\text{total idle}} + \lambda \Delta^H_{\text{total idle}}, \tag{1}$$

where $\Delta^R_{\text{total idle}}$ and $\Delta^H_{\text{total idle}}$ are the total robot and human idle times, respectively, and $\lambda > 0$ is an arbitrary weighting coefficient that balances their relative importance.

IV. Proactive Assistance through Action-Completion Estimation Framework

We introduce PACE, a framework designed to address the outlined problem by learning a proactive RL policy to provide timely robotic assistance. A key innovation of our approach is the integration of a real-time estimate of human action completion percentage into the policy's input. The framework comprises two main components: (i) a novel Dynamic Time Warping (DTW) algorithm for real-time human completion percentage estimation, and (ii) a formulation of the proactive assistance framework as a Partially Observable Markov Decision Process (POMDP) [17].

A. Real-Time Human Action Completion Estimation

The first component of the PACE framework is the real-time estimation of human action completion percentage. While this challenge has received limited attention in the HRC literature [14], [18], it is critical for accurately tracking task progress and ensuring the robot provides assistance at the most appropriate moments. In this section, we introduce a novel version of Open-End Dynamic Time Warping (OE-DTW) [13], incorporating a correlation-based distance metric that is more effective for capturing patterns in human hand movements.

Dynamic Time Warping (DTW) [19] is a well-established algorithm for time-series alignment, widely used in applications such as speech recognition and gesture analysis. It aligns a signal to a reference by minimizing the cumulative Euclidean distance between corresponding samples, enabling robust temporal alignment even in the presence of nonlinear time distortions. The open-ended variant of DTW, i.e., OE-DTW, has already been applied to human task progress monitoring [18]. However, we found that standard OE-DTW lacks regularization, often producing unrealistic warping paths when applied to human motion, where trajectories can vary significantly in shape and amplitude. To address these challenges, Soft-DTW [20] introduces a differentiable soft-minimum operator, which smooths the alignment cost by weighing all possible warping paths. This approach has demonstrated superior performance in tasks such as time-series clustering and temporal signal matching, offering greater robustness to variations in position and speed.

Building upon these methods, we combine the open-ended and soft versions of DTW to develop a more robust approach, as detailed in Algorithm 1. In this algorithm, δ denotes a distance function between signal points, typically the Euclidean distance in DTW, and $\min^{\gamma}$ represents the soft-minimum operator from [20]. Algorithm 1 computes the *phase* of a query signal relative to a reference signal, defined as the percentage of the reference trajectory matched up to the current timestep. Given an alignment π that maps each index i of the query signal with an index j^* of the reference trajectory, the phase at timestep i is computed as:

$$\tau_i = \frac{\pi(i)}{n-1}, \tag{2}$$

where n is the length of the reference trajectory. The phase τ_i quantifies the completion percentage as a value in $[0,1]$. We refer to the Open-end Soft-DTW algorithm as OS-DTW$_{\text{EU}}$, where EU specifies the use of the Euclidean distance.

However, the Euclidean distance employed in most DTW algorithms fails to align signals that have both local shape variations and substantial shifts in absolute positions, often present in complex motion patterns.

To overcome these limitations, we introduce a novel correlation-based metric called Windowed-Pearson (WP) distance, which normalizes amplitude differences over windows during alignment. This results in an online-capable method that directly compares trajectory shapes through local correlation analysis, while preserving DTW's temporal elasticity. We formally define the Windowed-Pearson distance between two signal samples as:

$$\delta_{\text{WP}}^{w}(\mathbf{p}_i, \mathbf{q}_j) := \sum_{k=0}^{d-1} \left(1 - \frac{\operatorname{Cov}\left(\mathbf{p}_{i-w+1:i,k}, \mathbf{q}_{j-w+1:j,k}\right)}{\sqrt{\operatorname{Var}\left(\mathbf{p}_{i-w+1:i,k}\right) \operatorname{Var}\left(\mathbf{q}_{j-w+1:j,k}\right)}} \right) \tag{3}$$

where w represents the window size, and $\mathbf{p}_{i:j,k}$ denotes a subsequence of $\mathbf{p}$ along the k-th dimension, spanning from index i to j. The same notation applies to $\mathbf{q}$. We refer to the combination of Open-end Soft-DTW with the WP distance as OS-DTW$_{\text{WP}}$.

This proposed method depends on two parameters, the smoothing factor γ and the window size w. Rather than tuning these parameters manually, we propose to optimize them automatically. Specifically, we employ Bayesian optimization to minimize the mean squared error between the phase τ_i estimated by OS-DTW$_{\text{WP}}$ and the phase corresponding to a linear progression $\bar{\tau}_i = \frac{i}{m-1}$ computed a posteriori, where m is the length of the query trajectory.

This optimization is possible thanks to the training trajectories defined in Section III. For each action, one trajectory is selected as the reference, while the remaining ones are used to tune the parameters.

Algorithm 1 Open-end Soft-DTW

Inputs:
- Query signal $\mathbf{p} = [\mathbf{p}_0, \ldots, \mathbf{p}_{m-1}] \in \mathbb{R}^{m \times d}$
- Reference signal $\mathbf{q} = [\mathbf{q}_0, \ldots, \mathbf{q}_{n-1}] \in \mathbb{R}^{n \times d}$
- Distance $\delta(\cdot,\cdot)$
- Smoothing parameter $\gamma \geq 0$

Output:
- Phase $\boldsymbol{\tau} = [\tau_0, \ldots, \tau_{m-1}] \in \mathbb{R}^{m}$ of $\mathbf{p}$ w.r.t. $\mathbf{q}$

Initialize $\mathbf{D} \in \mathbb{R}^{m \times n}$, where $D_{i,j} = \delta(\mathbf{p}_i, \mathbf{q}_j)$
Initialize $\mathbf{R} \in \mathbb{R}^{(m+1)\times(n+1)}$, with $R_{0,0} = 0$, $R_{i,0} = \infty$ for $i \in [1,m]$, and $R_{0,j} = \infty$ for $j \in [1,n]$
for $i = 1$ to m **do**
  for $j = 1$ to n **do**
    $R_{i,j} = D_{i-1,j-1} + \min^{\gamma}(R_{i-1,j}, R_{i,j-1}, R_{i-1,j-1})$
  end for
  $j^* = \arg\min_{j \in [0,n-1]} R_{i,j}$
  $\tau_i = j^*/(n-1)$
end for

B. POMDP

The interaction between the human and the robot is modeled as a finite-horizon episodic POMDP. In this framework, the robot acts as an agent that makes binary decisions between each robot-task action—whether to assist the human or not—while the human is treated as part of the environment. The POMDP is formally defined as a tuple (S, A, T, R, Ω, O), where S is the state space, $A = \{0,1\}$ is the set of *policy* actions (with 0 and 1 representing *do not assist* and *assist*, respectively), $T(s,a,s')$ is the state transition function, $R(s,a,s')$ is the reward function, Ω is the observation space, and $O(s)$ is the observation function. Note that the *policy* action $a \in A$ should not be confused with the task actions representing the robot operations introduced in Section III.

Each element of the state space S is defined as $s = (\mathrm{a}_i^R, \mathrm{a}_j^H, \mathrm{a}_l^J, \Delta_{start}^H, \mathbf{y}^H, \Delta_{idle}^R, \Delta_{idle}^H)$, where: a_i^R denotes the last robot-task action, a_j^H is the current human action, a_l^J represents the joint action that human and robot should perform next, Δ_{start}^H is the elapsed time from the start of the current human action a_j^H, $\mathbf{y}^H$ is a vector representing the observed human hand trajectory from the start of the current human action, Δ^R_{idle} and Δ^H_{idle} are the waiting times of robot and human observed during the last transition.

The transition function $T(s,a,s') := P(s' \mid a,s)$ describes the probability of transitioning from state s to state $s' = (\mathrm{a}^R_{i'}, \mathrm{a}^H_{j'}, \mathrm{a}^J_{l'}, \Delta'^H_{start}, \mathbf{y}'^H, \Delta'^R_{idle}, \Delta'^H_{idle})$. The state variables evolve as follows:

$$\mathrm{a}^R_{i'} = \begin{cases} \mathrm{a}^R_{(i+1) \bmod M} & a=0 \\ \mathrm{a}^R_i & a=1 \end{cases}$$
$$\mathrm{a}^J_{l'} = \begin{cases} \mathrm{a}^J_l & a=0 \\ \mathrm{a}^J_{l+1} & a=1, \end{cases}$$

while the remaining state variables are directly observed. In Section IV-C, we describe the model used to simulate the evolution of these quantities.

The reward function $R(s,a,s')$ is designed to minimize the total cost introduced in Equation (1), thus:

$$R(s,a,s') := -\Delta^R_{idle} - \lambda \Delta^H_{idle}. \tag{4}$$

The observation function is defined as $O(s) := (a^R_i, a^H_j, \Delta^H_{start}, \tau^j(\mathbf{y}^H))$, where $\tau^j(\mathbf{y}^H)$ represents the phase of the human action a^H_j, computed from the observed human hand trajectory $\mathbf{y}^H$ using OS-DTW$_{\mathrm{WP}}$. Namely, the observations consist of the last robot-task action, the current human action, the elapsed time from the start of the current human action, and the estimated human action completion percentage. The definition of the the observation space Ω follows accordingly.

C. Simulation and Training

Training an RL algorithm directly on the robot on a real-world environment is costly, time consuming and prone to major failures. Therefore, we propose to create a simulated environment that models the problem described in Section III leveraging limited real world demonstrations. This allows us to utilize online and on-policy algorithms to solve the POMDP described in Section IV-B.

We adopt Proximal Policy Optimization (PPO) [21] due to its native support for state representations that encompass both discrete and continuous spaces, as well as its ability to handle discrete action spaces, and robustness in highly stochastic environments. Specifically, PPO's clipped surrogate objective ensures stable policy updates, while its on-policy advantage estimation mitigates the high variance typically encountered in real-world human–robot interactions. This balance of simplicity, sample efficiency, and performance makes PPO particularly well-suited for human–robot collaboration, where data collection is costly.

To model the collaborative task, we assume the duration of each action follows a Gaussian distribution, and estimate them from demonstration data. Specifically, $\Delta^X_k \sim N(\mu_{X_k}, \sigma^2_{X_k})$, where $X \in \{H,R,P,E\}$ corresponds to human, robot-task, preparatory, and homing actions, respectively.

At the beginning of each episode, we sample from these distributions the durations human actions $\{\Delta^H_j\}^N_{j=1}$, preparatory actions $\{\Delta^P_l\}^L_{l=1}$, and homing actions $\{\Delta^E_l\}^L_{l=1}$. Then, one trajectory $\tilde{\mathbf{y}}_j$ is sampled from the set of demonstrations Y_j for each *non-joint* action a^H_j.

Moreover, to avoid overfitting on the training data, we linearly rescale the time axis of each trajectory $\tilde{\mathbf{y}}_j$ to align with each sampled duration Δ^H_j. As a result, each new trajectory represents either a compressed or stretched version of an actual demonstration. We found this augmentation essential for ensuring robustness and improving the policy's generalization capabilities.

By employing these quantities, we model the transitions of the POMDP defined in Section IV-B as:

$$\begin{aligned}
\mathrm{a}^H_{j'} &= \beta_{(\mathrm{a}^H_j, \Delta^H_{start})}(\Delta) \\
\Delta'^H_{start} &= \Delta - \Delta^H_{start} - \textstyle\sum_{k=j}^{j'-1} \Delta^H_k \\
\mathbf{y}'^H &= \tilde{\mathbf{y}}_{j'}(0 : \Delta'^H_{start}) \\
\Delta'^R_{idle} &= \begin{cases} 0 & a=0 \\ \max\left\{0, \sum_{k=j}^{\alpha(l)-1} \Delta^H_k - \Delta^H_{start} - \Delta^P_l \right\} & a=1 \end{cases} \\
\Delta'^H_{idle} &= \begin{cases} \max\left\{0, \Delta^R + \Delta^H_{start} - \sum_{k=j}^{\alpha(l)-1} \Delta^H_k \right\} & a=0 \\ \max\left\{0, \Delta^P_l + \Delta^H_{start} - \sum_{k=j}^{\alpha(l)-1} \Delta^H_k \right\} & a=1 \end{cases}
\end{aligned}$$

$\Delta^R \sim N\left(\mu_{R_{i'}}, \sigma^2_{R_{i'}}\right)$ is the duration of the robot-task $\mathrm{a}^R_{i'}$. Δ is the duration of the transition:

$$\Delta = \begin{cases} \Delta^R & a=0 \\ \Delta^P_l + \Delta^H_{\alpha(l)} + \Delta^E_l & a=1. \end{cases}$$

β is a function that, given the current human action a^H_j and its elapsed time Δ^H_{start}, returns the ongoing human action after a time Δ, namely,

$$\beta_{(\mathrm{a}^H_j, \Delta^H_{start})}(\Delta) := \underset{\mathrm{a}^H_{j'}}{\operatorname{argmin}} \left\{ j' \geq j \,\middle|\, \Delta \leq \textstyle\sum_{k=j}^{j'} \Delta^H_k \right\}.$$

V. Experiments and Results

The PACE framework was validated in a real-world scenario through a pilot study involving the collaborative assembly of an IKEA wooden chair[1]. In this assembly process, the human operator performed tasks requiring fine manual dexterity, such as screwing and positioning parts, while the robot acted as a smart assistant. The robot provided support at various stages of the assembly, including carrying large components and passing tools, such as Allen keys, at the appropriate moments for screwing operations.

A. Experimental Setup

The experimental setup consists of the robotic workcell shown in Fig. 4, including a Franka Emika Panda robot[2] and three main working areas: a *sorting table* for the robot task, a *warehouse table* where the components to be assembled are stored, and an *assembly table* where the collaborative assembly process takes place. An RGB-D camera monitors the area around each table using April-Tag markers [22] to locate the chair components. The robot was programmed

[1] https://www.ikea.com/us/en/p/ivar-chair-pine-90263902/
[2] https://robodk.com/robot/Franka/Emika-Panda

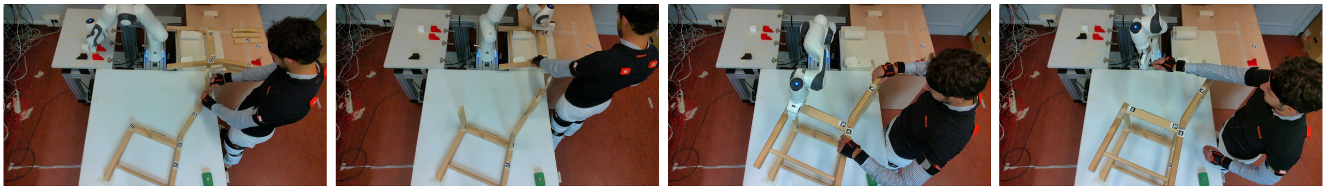

Fig. 3. Main steps of the collaborative assembly process: a) the human places the rails while the robot sorts cubes; b-c) human and robot work together to carry and position the side of the chair; d) the robot hands an Allen key to the human.

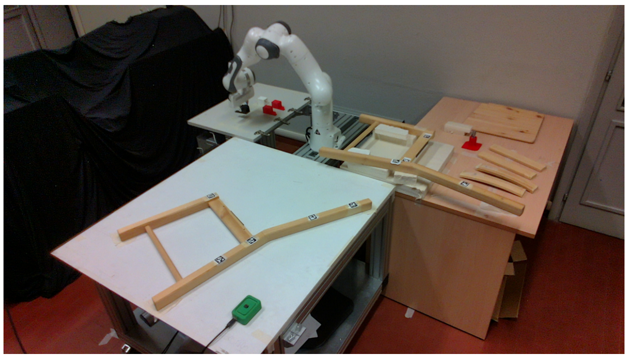

Fig. 4. Experimental setup for the wooden chair assembly process.

using ROS[3] and MoveIt[4]. Participants were equipped with the Xsens MVN Awinda motion capture system [23] which recorded the position of their right hand at a sampling rate of 10 Hz. Alternative tracking gloves such as Rokoko[5], HaptX[6], could be employed.

B. Task Description

The robot task consists in a series of cube sorting operations (a_i^R). The assembly process, illustrated in Fig. 3, is outlined as follows: (i) the human connects 4 rails to the right side of the chair (a_1^H); (ii) the human and robot collaboratively transport the left side from the warehouse area and place it on top of the rails (a_1^J); (iii) the human adjusts the top chair sides and places 3 screws on the left side (a_3^H); (iv) the robot hands an Allen key to the human (a_2^J); (v) the human uses such key to tighten 2 screws (a_5^H); (vi) the robot hands over a second key for the human to tighten the remaining screw (a_3^J).

For reference, the average durations of the non-joint human actions described above were approximately 22, 18, and 40, seconds respectively. The robot preparatory actions a_l^P for the following joint actions took on average 11 seconds, 8 seconds, and 9 seconds, respectively. Each robot cube sorting operation, represented by the action a_i^R, had a duration of approximately 8 seconds.

C. Data Collection and Training Procedure

We collected data from 5 subjects, with each subject performing the assembly task 4 times. Additionally, one of the subjects provided an extra demonstration to generate the references for the OS-DTW$_{\text{WP}}$ algorithm. During the data collection, users explicitly requested assistance from the robot by pressing a button, shown in Fig. 4. We implemented the POMDP described in Section IV-B as a custom Gymnasium environment [24] and used the Stable-Baselines3 library [25] for policy training. Out of the 4 demonstrations per subject, 3 were used for training and 1 for validation. Finally, motivated by the quantitative studies in [26], [27], our experiments assume that the cost of operating a robot is roughly one-third that of human labor. Consequently, we set the weighting parameter λ in Equations (1) and (4) equal to 3.

[3] https://www.ros.org/
[4] https://moveit.ai/
[5] https://www.rokoko.com/products/smartgloves
[6] https://haptx.com

D. User Study and Experiment Design

The experiments involved 12 volunteers (5 women and 7 men) aged 24 to 28, including two individuals who also participated as training subjects. Participants were first briefed on the assembly task and the robot's action capabilities. Then, they assembled the wooden chair collaborating with the robot controlled by three different methods: (i) *PACE*, our proposed method, incorporating phase estimation via OS-DTW$_{\text{WP}}$; (ii) *PACE w/o phase*, an ablation method that excludes the phase from the observations provided to the policy; (iii) *explicit query*, a baseline system in which the human operator explicitly requests robot assistance by pressing a button after completing each action.

Each participant experienced all three methods in a randomized, unknown order, completing two trials per method. Participants were not informed in advance about the differences between *PACE* and *PACE w/o phase*. After completing each set of trials, they filled out the NASA Task Load Index survey [28], and a custom 5-point Likert scale questionnaire.

E. Results

Our goal is to evaluate whether proactive robot policies can reduce assembly downtime and improve user experience. To investigate this, we compared the three methods based on robot and user waiting times. For each experiment, we recorded the robot's idle times and captured video of the assembly process. These videos were analyzed to annotate participant waiting times before each joint action. Quantitative results are summarized in Table I. The reported quantities are the averages computed from all participants' trials. Columns $A1$, $A2$, and $A3$ show the idle times with respect to the first, second, and third joint actions. The difference between the performances of *PACE* and *PACE w/o phase* is statistically significant ($p = 0.007$), computed using a pairwise Wilcoxon signed-rank test on the average cost per subject.

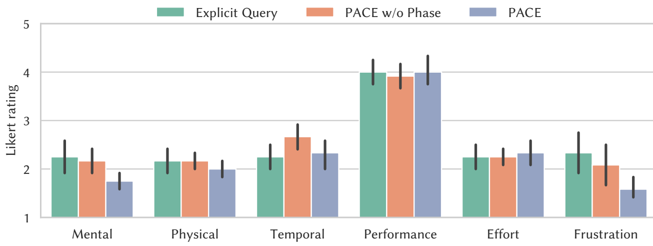

Fig. 5. NASA-TLX [29] findings for subjective measures on a 5-point scale. Plot shows means and 75% confidence intervals of ratings.

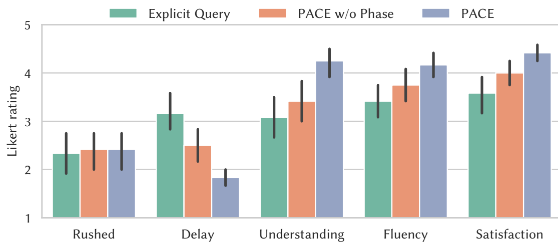

Fig. 6. Findings for subjective measures on a 5-point scale ranging from *Strongly Disagree* to *Strongly Agree*. Plot shows means and 75% confidence intervals of ratings. The questions are the following. **Rushed**: *I felt rushed by the robot's action*. **Delay**: *I felt the robot took too long to provide assistance*. **Understanding**: *I felt the robot had a good understanding of the task*. **Fluency**: *The robot and I collaborated fluently*. **Satisfaction**: *I feel satisfied by the performance of the system*.

As expected, participants experienced the longest waiting times with the *explicit query* method. This discomfort is reflected in survey results, where participants reported that the robot took too long to provide assistance (see Fig. 6). Additionally, Table I shows that *PACE* reduces robot idle time by more than half compared to *PACE w/o phase*, without significantly increasing human waiting time. Note that the baseline *explicit query* exhibits zero idle time by design, as the robot is manually activated by the participant.

PACE w/o phase also outperforms the other methods in subjective measures, as reported in Fig. 6. Users reported higher levels of fluency, understanding, and overall satisfaction, indicating that the method adapts well to individual participant pacing. Furthermore, Fig. 5 shows that a proactive robot operating autonomously does not increase mental strain or the overall Task Load Index. Notably, five out of twelve participants explicitly stated in the questionnaire's open comment section that they preferred the system monitoring their task progress, with many appreciating that assistance was provided only as they neared the end of their action.

TABLE I

EXPERIMENTAL RESULTS ON HUMAN AND ROBOT IDLE TIMES

Method	Robot Idle Time [s]				Human Idle Time [s]				Cost
	A1	A2	A3	Total	A1	A2	A3	Total	($\lambda = 3$)
Explicit query	**0.0**	**0.0**	**0.0**	**0.0**	14.00	11.28	12.15	37.43	112.3
PACE w/o phase	1.56	3.48	11.64	16.68	1.79	1.02	**1.06**	**3.87**	28.29
PACE (ours)	1.93	2.65	1.28	5.86	**1.20**	**0.56**	2.56	4.32	**18.81**

F. Ablation Results on Progress Estimation

As detailed in Section IV-A, OS-DTW$_{\mathrm{WP}}$'s parameters were optimized against a linear phase evolution computed a

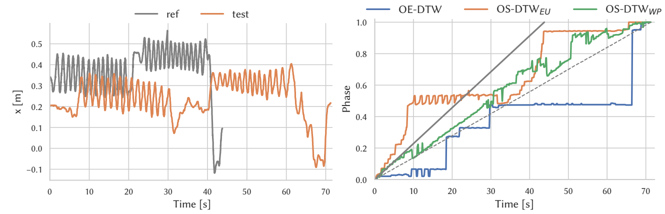

Fig. 7. Illustrative example of the estimated phases for a single trajectory of the screwing task. On the left are reported the reference and query signals along the x-axis. On the right the estimated phases, computed using OE-DTW, OS-DTW$_{\mathrm{EU}}$, OS-DTW$_{\mathrm{WP}}$.

posteriori. Table II reports the average mean squared errors (MSE) of OS-DTW$_{\mathrm{WP}}$, OS-DTW$_{\mathrm{EU}}$, and OE-DTW across the three non-joint human actions over the training data. While OS-DTW$_{\mathrm{WP}}$ achieves superior performance overall, OS-DTW$_{\mathrm{EU}}$ performs adequately for placing actions but struggles with screwing. For reference, in Fig. 7 we also report an illustrative example showing the estimated completion percentage during a screwing experiment.

These results highlight the effectiveness of the windowed-Pearson (WP) distance as a local measure for human-hand trajectories. Unlike the Euclidean distance, the WP distance is invariant to mean shifts, making it robust to the variability inherent in human task execution.

TABLE II

AVERAGE PHASE MSE ON THE TRAINING DATA

Task	OE-DTW	OS-DTW$_{\mathrm{EU}}$	OS-DTW$_{\mathrm{WP}}$
Rail Placing	0.100	0.012	**0.002**
Screw Placing	0.091	0.013	**0.012**
Screwing	0.168	0.052	**0.006**

VI. Conclusions

In this work, we introduced the Proactive Assistance through action-Completion Estimation (PACE) framework, which leverages reinforcement learning and real-time human progress monitoring to improve robotic assistance in collaborative tasks. PACE addresses variability in human execution pace through OS-DTW$_{\mathrm{WP}}$, a novel Dynamic Time Warping algorithm that incorporates local correlation-based distances for robust real-time action completion estimation.

Our experiments with human participants demonstrated that a robot using PACE can reduce idle times by more than half, with participants highlighting its timely and adaptive support. These results confirm that PACE not only enhances collaborative efficiency but also improves user experience, paving the way for more intuitive and effective human-robot interactions in assembly task.

Future work could extend PACE applications to more complex scenarios beyond industrial assembly, such as home robotics or collaborative cooking, while also enhancing its flexibility to handle more adaptable assembly processes. Additionally, further studies could expand OS-DTW$_{\mathrm{WP}}$'s applicability by enabling estimation of action completion times, increasing its utility and generalizability.

References

[1] E. Matheson, R. Minto, E. G. Zampieri, M. Faccio, and G. Rosati, "Human–robot collaboration in manufacturing applications: A review," *Robotics*, vol. 8, no. 4, p. 100, 2019.

[2] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[3] G. Hoffman and W. Ju, "Designing robots with movement in mind," *J. Hum.-Robot Interact.*, vol. 3, no. 1, p. 91–122, Feb. 2014. [Online]. Available: https://doi.org/10.5898/JHRI.3.1.Hoffman

[4] 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.

[5] S. M. Rahman, B. Sadrfaridpour, and Y. Wang, "Trust-based optimal subtask allocation and model predictive control for human-robot collaborative assembly in manufacturing," in *Dynamic systems and control conference*, vol. 57250. American Society of Mechanical Engineers, 2015, p. V002T32A004.

[6] G. Giacomuzzo, M. Terreran, S. Jain, and D. Romeres, "Decaf: a discrete-event based collaborative human-robot framework for furniture assembly," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 7085–7091.

[7] A. Gottardi, M. Terreran, C. Frommel, M. Schoenheits, N. Castaman, S. Ghidoni, and E. Menegatti, "Dynamic human-aware task planner for human-robot collaboration in industrial scenario," in *2023 European Conference on Mobile Robots (ECMR)*. IEEE, 2023, pp. 1–8.

[8] Y. Cheng, L. Sun, C. Liu, and M. Tomizuka, "Towards efficient human-robot collaboration with robust plan recognition and trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2602–2609, 2020.

[9] 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.

[10] K. Ramachandruni, C. Kent, and S. Chernova, "Uhtp: A user-aware hierarchical task planning framework for communication-free, mutually-adaptive human-robot collaboration," *ACM Transactions on Human-Robot Interaction*, 2023.

[11] P. A. Lasota and J. A. Shah, "Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration," *Human factors*, vol. 57, no. 1, pp. 21–33, 2015.

[12] R. Maderna, M. Ciliberto, A. Maria Zanchettin, and P. Rocco, "Robust real-time monitoring of human task advancement for collaborative robotics applications," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 11 094–11 100.

[13] P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli, "Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation," *Artificial intelligence in medicine*, vol. 45, no. 1, pp. 11–34, 2009.

[14] Y. Cheng and M. Tomizuka, "Long-term trajectory prediction of the human hand and duration estimation of the human action," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 247–254, 2021.

[15] J. Leu, Y. Cheng, M. Tomizuka, and C. Liu, "Robust task planning for assembly lines with human-robot collaboration," in *Proceedings of the International Symposium on Flexible Automation 2022 International Symposium on Flexible Automation*. The Institute of Systems, Control and Information Engineers, 2022, pp. 188–195.

[16] E. A. Lee and S. A. Seshia, *Introduction to embedded systems: A cyber-physical systems approach*. MIT press, 2017.

[17] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000437029800023X

[18] R. Maderna, P. Lanfredini, A. M. Zanchettin, and P. Rocco, "Real-time monitoring of human task advancement," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 433–440.

[19] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[20] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," in *International conference on machine learning*. PMLR, 2017, pp. 894–903.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[22] D. Malyuta, C. Brommer, D. Hentzen, T. Stastny, R. Siegwart, and R. Brockers, "Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition," *Journal of Field Robotics*, p. arXiv:1908.06381, Aug. 2019.

[23] M. Schepers, M. Giuberti, G. Bellusci *et al.*, "Xsens mvn: Consistent tracking of human motion using inertial sensing," *Xsens Technol*, vol. 1, no. 8, pp. 1–8, 2018.

[24] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. J. S. Tan, and O. G. Younis, "Gymnasium (v1.0.0a2)," 2024. [Online]. Available: https://doi.org/10.5281/zenodo.11232524

[25] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[26] M. Decker, M. Fischer, and I. Ott, "Service robotics and human labor: A first technology assessment of substitution and cooperation," *Robotics and Autonomous Systems*, vol. 87, pp. 348–354, 2017.

[27] C. Brosque and M. Fischer, "Safety, quality, schedule, and cost impacts of ten construction robots," *Construction robotics*, vol. 6, no. 2, pp. 163–186, 2022.

[28] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Human mental workload*, vol. 1, no. 3, pp. 139–183, 1988.

[29] S. G. Hart, "Nasa task load index (tlx)," 1986.