Human Monitoring with Correlation-based Dynamic Time Warping for Collaborative Assembly

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

TR2025-067 May 28, 2025

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. Central to PACE is OS- DTWWP, a novel open-ended Dynamic Time Warping (DTW) algorithm for real-time human action completion estimation, which incorporates a Windowed-Pearson (WP) correlation- based distance metric to dynamically track 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) - Workshop on The Future of Intelligent Manufacturing: From innovation to implementation 2025

Human Monitoring with Correlation-based Dynamic Time Warping for Collaborative Assembly

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. Central to PACE is OS-DTW$_{\text{WP}}$, a novel open-ended Dynamic Time Warping (DTW) algorithm for real-time human action completion estimation, which incorporates a Windowed-Pearson (WP) correlation-based distance metric to dynamically track 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.**

I. Introduction

Recent advancements in robotics and artificial intelligence have significantly increased the use of robots alongside humans in industrial settings [1]. Assembly tasks provide an ideal scenario for human-robot collaboration (HRC), leveraging robot precision and speed alongside the dexterity and adaptability of humans. This collaboration can enhance productivity and improve work quality by assigning repetitive, physically demanding tasks to robots [2].

However, effective human-robot collaboration requires careful task planning and assignment, which is challenging due to the inherent variability in human performance. For example, different operators may take varying amounts of time to complete the same task, even when following a predefined sequence. Mutual understanding is therefore crucial, and timing plays a key role in shaping interactions, as humans are especially sensitive to timing and smoothness [3].

Our work focuses on enabling the robot to proactively support the human operator, minimizing idle times and reducing human effort. Two primary challenges must be addressed: (i) accurately perceiving and predicting human actions and their temporal progression, and (ii) planning the robot's actions to synchronize with the human's needs.

To tackle these challenges, we first estimate human action completion using a real-time adaptation of Dynamic Time Warping (DTW), which accommodates variations in human hand movements. We then leverage data collected from demonstrations to train a reinforcement learning (RL) policy that reduces idle times for both human and robot.

To our knowledge, only two studies have explored real-time monitoring of human movements at the action level. The first [4] employs Open-End Dynamic Time Warping (OE-DTW) [5] to estimate task progression and dynamically re-plan robot actions. The second [6] proposes a Sigma log-normal model of human movements, outperforming DTW for action completion estimation; the same approach is also used for task planning in a collaborative assembly application [7]. However, OE-DTW alone is not robust to the variability of human movements. We therefore introduce a new DTW-based algorithm capable of overcoming these limitations and integrate it into an RL framework that directly optimizes robot performance without explicitly estimating action completion times.

Our work offers the following key contributions:

- We present OS-DTW$_{\text{WP}}$, 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$_{\text{WP}}$, minimizing waiting times and ensuring seamless human-robot synchronization.
- We validate the PACE framework with real-world experiments involving a chair assembly task and human participants.

II. Proactive Assistance Through Action-Completion Estimation Framework

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.

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

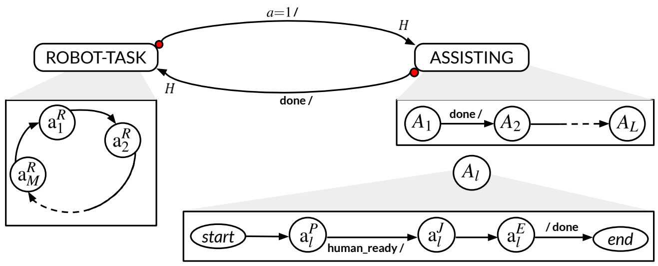

Fig. 1. Hierarchical state machine depicting the collaborative task from the robot perspective. The robot transitions from *ROBOT-TASK* to *ASSISTING* in between states a_i^R if $a = 1$. Once an (*assist*) action A_i is completed, the robot goes back to its task. The state machine follows the conventions as in [8]. Each transition is labeled with *guard* / *effect*. A transition is taken on a reaction only if the *guard* holds true.

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 $\{a_l^P\}_{l=1}^L$ and $\{a_l^E\}_{l=1}^L$, respectively. A depiction of the collaborative task in the form of a hierarchical state machine is provided in Fig. 1.

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, i.e., the time each agent waits for the other before starting the joint actions. We define the cost function:

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

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.

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 [6], [9], 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) [5], incorporating a correlation-based distance metric effective for capturing patterns in human hand movements.

Dynamic Time Warping (DTW)[10] is a time-series alignment algorithm that aligns a signal to a reference by minimizing cumulative Euclidean distances. Its open-ended variant, OE-DTW, has been used for human task progress monitoring[9], but lacks regularization, often resulting in unrealistic warping paths for human motion. Soft-DTW [11] alleviates these issues by introducing a differentiable soft-minimum operator that weighs all possible warping paths, improving robustness in tasks such as time-series clustering and temporal signal matching.

Building on these methods, we combine open-ended and soft DTW to develop a more robust alignment approach (Algorithm 1). In this algorithm, δ is the pointwise distance function (commonly Euclidean), and $\min^\gamma$ denotes the soft-minimum operator from [11]. Algorithm 1 also computes the *phase* of a query signal relative to a reference, defined as the fraction of the reference trajectory matched at each timestep. Given an alignment function π that associates each query index i with a reference index j^*, the phase at timestep i is:

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

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. 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 WP distance between two signal samples as:

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

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$_{\text{WP}}$.

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 metric $\delta(\cdot,\cdot)$; 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

Human-robot interaction is modeled as a finite-horizon episodic POMDP. The robot, acting as the agent, makes binary decisions (assist or not) between each robot-task action, while the human is treated as part of the environment. The POMDP is formally defined as a tuple (S, A, T, R, Ω, O),

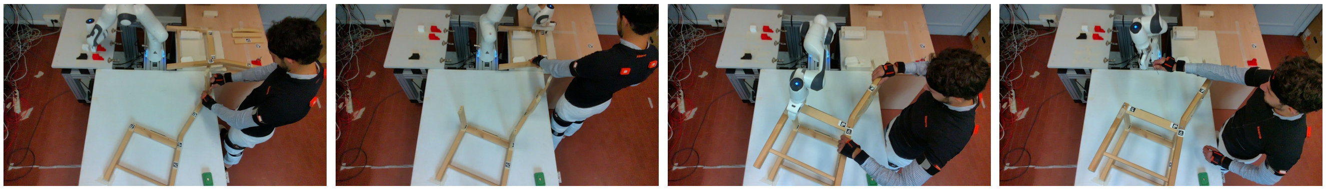

Fig. 2. Main steps of the collaborative assembly process: a) the human places the rails while the robot sorts cubes; b-c) human and robot work together to carry and position the side of the chair; d) the robot hands an Allen key to the human.

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 II.

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_{idle}^R and Δ_{idle}^H 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 $s' = (\mathrm{a}_{i'}^R, \mathrm{a}_{j'}^H, \mathrm{a}_{l'}^J, \Delta_{start}'^H, \mathbf{y}'^H, \Delta_{idle}'^R, \Delta_{idle}'^H)$.

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

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

The observation function is defined as $O(s) := (a_i^R, a_j^H, \Delta_{start}^H, \tau^j(\mathbf{y}^H))$, where $\tau^j(\mathbf{y}^H)$ represents the phase of the human action a_j^H, computed from the observed human hand trajectory $\mathbf{y}^H$ using OS-DTW$_{\mathrm{WP}}$. 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 phase.

C. Simulation and Training

We built a simulated environment, to avoid the issues of training RL algorithms directly on real-world robots, but informed by limited real-world demonstrations, to model the problem in Section II and enable online, on-policy algorithms to solve the POMDP in Section II-B. We adopt Proximal Policy Optimization (PPO) [12] for its handling of discrete actions and robustness in stochastic environments.

To model the collaborative task, we assume the duration of each action follows a Gaussian distribution, and estimate them from demonstration data. Specifically, $\Delta_k^X \sim N(\mu_{X_k}, \sigma_{X_k}^2)$, 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 $\{\Delta_j^H\}_{j=1}^N$, preparatory actions $\{\Delta_l^P\}_{l=1}^L$, and homing actions $\{\Delta_l^E\}_{l=1}^L$. Then, one trajectory $\tilde{\mathbf{y}}_j$ is sampled from the set of demonstrations Y_j for each *non-joint* action a_j^H. 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_j^H. 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.

III. 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. 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 experimental setup consists of a Franka Emika Panda robot 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. Participants were equipped with the Xsens MVN Awinda motion capture system [13] which recorded the position of their right hand at a sampling rate of 10 Hz.

A. Task Description

The robot task involves cube sorting operations (a_i^R). The assembly process, illustrated in Fig. 2, 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.

B. 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$_{\mathrm{WP}}$ algorithm. During the data collection, users explicitly requested assistance from the robot by pressing a button. We implemented the POMDP described in Section II-B as a custom Gymnasium environment [14] and used the Stable-Baselines3 library [15] 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 [16], [17], 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.

C. 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 [18], and a custom 5-point Likert scale questionnaire.

D. 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 and human's idle times. 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.

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. 4). 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. 4. Users reported higher levels of fluency, understanding, and overall satisfaction, indicating that the method adapts well to individual participant pacing. Furthermore, Fig. 3 shows that a proactive

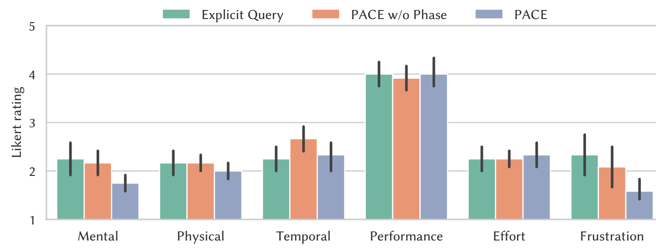

Fig. 3. NASA-TLX [19] findings for subjective measures on a 5-point scale. Plot shows means and 75% confidence intervals of ratings.

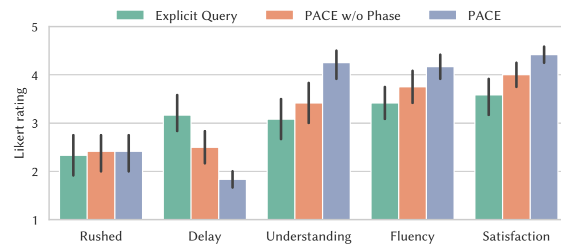

Fig. 4. Findings for subjective measures on a 5-point scale ranging from *Strongly Disagree* to *Strongly Agree*. Plot shows means and 75% confidence intervals of ratings. The questions are the following. **Rushed**: *I felt rushed by the robot's action*. **Delay**: *I felt the robot took too long to 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.*

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**

IV. 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$_{\text{WP}}$, 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.

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] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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

[15] 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

[16] 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.

[17] 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.

[18] 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.

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