

Adaptive Human-Robot Collaborative Painting Combining Preference-Based Optimization and Dynamic Motion Primitives

C. Cella¹, M. Ristic¹, M. Faroni¹, A. M. Zanchettin¹, and P. Rocco¹

Abstract—This work presents a human-centered collaborative framework that integrates Preference-Based Optimization (PBO) and Dynamic Movement Primitives (DMPs) to optimize robot-assisted tasks such as painting. The system allows the operator to perform the process while the robot adapts its behavior in real-time, dynamically adjusting the orientation of the piece in order to match the orientation of the operator’s hand. The PBO framework leverages the GLISp algorithm to iteratively refine control parameters such as execution time, robot responsiveness, and rotation amplification through human feedback. Moreover, DMPs have been modified to enhance the reactive behavior of the robot and its adaptability to ergonomic requirements. The method was validated with a heterogeneous group of participants executing painting tasks. The results show that our strategy effectively reduces operator effort while optimizing process outcomes.

Index Terms—human-robot collaboration, human factors and human-in-the-loop.

I. INTRODUCTION

THE Industry 5.0 paradigm has advanced industrial robotic systems for collaborative processes, enhancing human performance by combining operator dexterity with robotic repeatability [1]. Among these, a key subset involves *non-contact* assistive tasks, where robots support operators without direct physical interaction. Such frameworks improve efficiency and ergonomics leveraging *leader-follower* paradigms, where the robot trajectories are optimized to ensure adaptive responsiveness to human intent in industrial environments [2].

In this context, spray painting is one of the most critical industrial applications. In sectors like automotive and furniture manufacturing, the quality of the painted surface is essential, influencing both aesthetics and structural integrity [3]. Although automated solutions exist, human expertise remains indispensable, particularly for irregularly shaped or cumbersome components. On the other side, prolonged manual operations increase fatigue and ergonomic strain, reducing efficiency over time [4]. The most critical aspect related to traditional spray

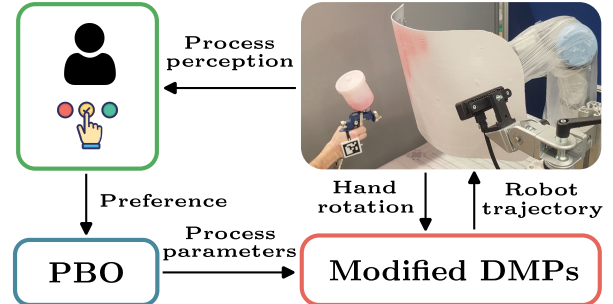


Fig. 1. Sketch of the proposed approach applied to collaborative painting.

painting setups is the lack of metrics accounting for operator preferences and feedback, which limits process flexibility and hinders effective robotic assistance across diverse production scenarios. The balancing of productivity and ergonomics should transform the robot into an *assistive* partner, capable of adjusting its actions to support the operator’s needs during the process, rather than being a passive tool.

To address this problem, we propose a *human-centric* robotic assistance framework for non-contact industrial operations. Although applicable to various processes, spray painting is our primary focus. To this purpose, consider the scenario in Fig. 1: the robot holds the workpiece while the operator is painting it. The robot shall assist the operator by adjusting the workpiece pose online, in order to reduce human effort and enhance flexibility. Our method is based on a combination of *Preference-Based Optimization* (PBO) and *Dynamic Movement Primitives* (DMPs). While DMPs provide a convenient framework for adaptive trajectory generation, they do not explicitly account for subjective user preferences, which are central in Human-Robot Collaboration (HRC). For this purpose, we integrate DMPs-based motion generation with PBO, enabling adjustments of the robot behavior based on qualitative operator feedback. This approach allows for real-time tuning of robot responsiveness (red box in Fig. 1), while the GLISp algorithm [5] is employed in between experiments to optimize the set of process parameters from qualitative feedback (green and blue boxes). Results on a real-world painting setup show that our approach increases ergonomics and the operator’s satisfaction, without jeopardizing the process quality.

II. RELATED WORK AND CONTRIBUTIONS

Assistive robotic tasks in industrial settings require robots to adapt their behavior to human preferences, rather than

Manuscript received 3 May 2025; Revised 25 November 2025; Accepted 16 March 2026.

This paper was recommended for publication by Editor Haoyong Yu upon evaluation of the Associate Editor and Reviewers’ comments. This work was supported by MICS (Made in Italy – Circular and Sustainable) Extended Partnership and received funding from Next-Generation EU (Italian PNRR – M4 C2, Invest 1.3 – D.D. 1551.11-10-2022, PE00000004). CUP MICS D43C22003120001

¹The authors are with Politecnico di Milano, 20133, Milano, Italy (e-mail: christian.cella@polimi.it).

Digital Object Identifier (DOI): see top of this page.

executing predefined trajectories [6]. In these cases, ensuring seamless collaboration requires balancing human comfort and process efficiency, a challenge that is widely emphasized in ergonomics-oriented HRC strategies, where the goal is to adapt the robot's behavior as a function of the operator's perceived comfort [7, 8]. In accordance with this vision, recent advances integrate *preference-based learning* to personalize the robot actions [9], allowing them to adapt their behavior based on *qualitative* human feedback.

Among the industrial applications that could benefit the most from these approaches, robotic painting proposes unique challenges, due to its strict quality requirements and variability in task conditions. The adoption of industrial robots in painting improves productivity and surface quality, particularly in the automotive and furniture sectors [3]. Although recent research has introduced collision-free trajectory planning [10], these approaches often overlook human preferences and key aspects in HRC. Indeed, most trajectory generation algorithms focus on geometric optimization while overlooking real-time adaptation to human input and the integration of qualitative feedback, limiting their flexibility and effectiveness in HRC industrial applications [11].

A promising solution is offered by Preference-Based Optimization (PBO) [5], which accounts for the operator's preferences with the aim of optimizing the robotic behavior. PBO has been successfully applied in HRC scenarios to fine-tune control parameters based on user preferences [12]. Recent works focused on ergonomic optimization, such as fostering operator trust [13] and integrating ergonomic assessments to enhance comfort and productivity in object-handling tasks [14]. From an industrial perspective, PBO has also been employed to optimize velocity planners in sealing operations, refining motion control based on human preferences [15].

While *preference learning* effectively captures human intent, an additional challenge in HRC applications is represented by the mapping of these preferences into executable motion commands. This is where Dynamic Movement Primitives (DMPs) provide an effective solution. DMPs enable smooth, adaptive trajectory generation, responding to external changes through a dynamical system formulation [16]. As highlighted in [17], many extensions have been introduced over the years, ranging from online adaptation mechanisms to formulations that incorporate *coupling terms* for reactive behavior. Within this family of improvements, recent works have proposed adaptive coefficients to modulate execution time and end-position behaviors in collaborative handover tasks [18, 19]. The main drawback of these DMPs-based works concerns the type of applications for which they are engineered, where the main interest is to adapt the process to the operator, rather than enhancing the cooperation between the human and the robot based on qualitative feedback coming from the subjective perception of the process. On the other hand, PBO-based works mainly focus on the estimation of control parameters in physical interaction, and they do not straightforwardly apply to motion planning for non-contact collaborative tasks.

In this regard, the integration of DMPs and PBO is an underexplored, yet promising, direction for tasks requiring adaptability and subjective optimization. This work aims at

filling this gap with the following contributions:

- We developed a *human-centric* collaborative painting application that redefines the role of the robot, from a primary executor to an adaptive assistant. Unlike previous approaches, such as [20] and [3], which primarily focus on autonomous task execution, our pipeline is specifically designed to support the operator during the manipulation of cumbersome components. Additionally, despite both the method presented in [4] and ours are applicable in the presence of *extended* surfaces, we adapt the robot motion through tunable parameters and preference-driven adjustments, enabling a more flexible and user-tailored collaborative painting experience.
- We introduced significant modifications to the standard DMPs framework, incorporating rotational amplification and responsiveness adjustments. Taking inspiration from the works dealing with collaborative handovers [18, 19], which primarily address spatial and temporal synchronization between the agents, we introduce instead modifications to the DMPs equations that enable the robot to dynamically amplify the operator's hand rotations, while preserving smooth trajectory generation.

III. PROBLEM STATEMENT

Painting tasks can cause fatigue and ergonomic strain on the operator, reducing efficiency over time, especially when dealing with cumbersome components. HRC painting offers a way to mitigate such side effects by employing a robot to assist the operator during the task. In this context, we consider a scenario in which the operator is responsible for painting a workpiece, while the robot holds and maneuvers the component to ease the operator's actions. Our goal is to devise a motion planning strategy for the robotic assistant that incorporates the operator's preferences and helps increasing the ergonomics of the painting process. Since these preferences are not easily captured through an analytical objective function, we propose to address them by leveraging explicit feedback from the operator after completing the task. The proposed approach must meet a set of requirements that consider both ergonomics and the overall effectiveness of the operation.

Requirement 1: Small changes in orientations of the operator's hand should be associated to *scaled-up* rotations of the piece to be painted. At the same time, the movements should comply with the robot's physical limits.

Requirement 2: To obtain an acceptable surface painting, the robot should be *reactive* in tracking changes in the operator's hand orientation, favoring the regions of the component that the human wishes to paint.

Requirement 3: The robot behavior should resemble that of a second operator holding the workpiece and adapting to the first operator's movements. Hence, the trajectory should be generated so that the robot moves in a *human-like* style.

IV. METHOD

Our approach can be summarized as follows. We propose a modified version of DMPs to generate the robot's trajectories, where the modification introduces a set of hyperparameters

that influence the quality and comfort of the collaborative operation. Since manually tuning these parameters to suit each operator's preferences is impractical, we rely on the GLISp algorithm [5] that implements PBO. This method iteratively adjusts the parameters based on the operator's feedback, ultimately allowing the system to be fine-tuned to individual preferences. This Section describes the DMPs-based motion generation (IV-A), our changes to the standard DMPs (IV-B), and the PBO framework (IV-C) we adopted.

A. Generation of the robot trajectories

Dynamic Movement Primitives (DMPs) allow to encode trajectories as solutions of a second-order dynamical system, modulated by a non-linear forcing term learned from a single demonstration. Their inherent flexibility allows online modulation of both timing and shape of the trajectory, making them particularly suitable for applications where operational parameters vary during the execution, and the trajectories must be recomputed at run-time.

The standard formulation of DMPs [16] does not suit well to the requirements described in Section III, since they do not depend on the process parameters acquired from the environment. For this reason, we resort to a variant of the DMPs formulation. First, we use a decoupled set of equations to generate the robot reference trajectories, for Cartesian positions and orientation. This strategy enables the robot to dynamically respond to the operator's actions and adjust the orientation of the piece to be painted accordingly. Because the reference trajectories are taught through teleoperation, the Cartesian position coordinates $\{x, y, z\}$ inherently capture *human-like* behavior (Requirement 3). The decoupled set of equations we implemented is as follows:

$$\begin{aligned} \dot{\mathbf{z}}/\bar{\tau} &= \alpha_z (\beta_z (\mathbf{g}_p - \mathbf{p}) - \mathbf{z}) + \mathbf{f}_p(s) \\ \dot{\mathbf{p}}/\bar{\tau} &= \mathbf{z} \\ \dot{\boldsymbol{\eta}}/\bar{\tau} &= \alpha_z (\beta_z 2 \log(\bar{\mathbf{g}}_o \otimes \mathbf{q}^*) - \boldsymbol{\eta}) + \mathbf{f}_o(s) + \boldsymbol{\gamma}_s \\ \dot{\mathbf{q}}/\bar{\tau} &= \frac{1}{2} \boldsymbol{\eta} \otimes \mathbf{q} \\ \dot{s}/\bar{\tau} &= -\alpha_s s \end{aligned} \quad (1)$$

where \mathbf{z} is the *transformation variable* acting as a scaled velocity, while s is the *phase variable*. We denote by $\mathbf{p} = [p_x, p_y, p_z]^T$ the Cartesian position of the robot, while the Cartesian goal \mathbf{g}_p is fixed. The orientation equations leverage quaternion-based DMPs [17], [21]: $\mathbf{q} = [q_x, q_y, q_z, q_w]^T$ is the quaternion representing the robot orientation in time; \mathbf{q}^* is the quaternion conjugate, and " \otimes " is the quaternion product. $\boldsymbol{\eta}$ is the *augmented* angular velocity $\boldsymbol{\omega}$, with null scalar part ($\boldsymbol{\eta} = [\omega_x, \omega_y, \omega_z, 0]^T$), while $\log(\mathbf{q}) : \mathbb{S}^3 \rightarrow \mathbb{R}^3$ is the quaternion *logarithmic mapping* that allows to extract the rotation $\mathbf{r} \in \mathbb{R}^3$ (*Rodrigues* representation) from \mathbf{q} [22]. Coefficients α_z , β_z and α_s are selected to obtain a *critically damped* behavior (see Table I), while the weights \mathbf{w}_p and \mathbf{w}_o of the non-linear forcing terms $\mathbf{f}_p(s)$ and $\mathbf{f}_o(s)$ are learned leveraging *locally weighted regression* (LWR) exploiting the general definition presented in [16]. Finally, we define with $\mathbf{y} = [\mathbf{p}, \mathbf{r}] \in \mathbb{R}^6$ the complete state of the system and with $\dot{\mathbf{y}}, \ddot{\mathbf{y}}$ the velocity and the acceleration, respectively. To account

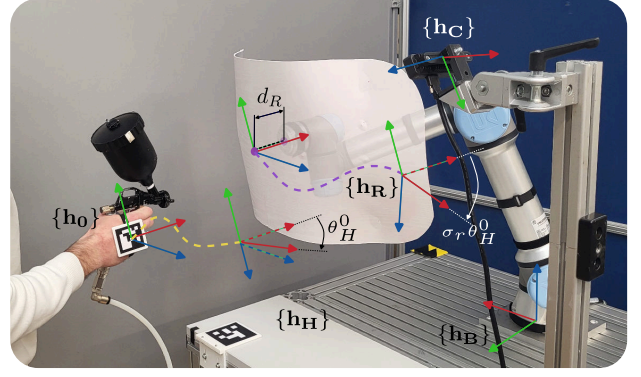


Fig. 2. Setup used for painting. The yellow and purple dashed lines represent the human and robot trajectories, respectively.

for the modifications described in Section III we introduced the corrective terms $\boldsymbol{\gamma}_s$, $\bar{\mathbf{g}}_o$ and $\bar{\tau}$, detailed in the following.

B. Explanation of the DMPs corrective terms

Requirement 1 is related to the *feasibility* of the task. Especially for cumbersome pieces, confining the range of motion of the operator's hand in a limited space leads to reduced efforts during the painting process [4]. To address this requirement in a systematic way, we introduced the reference frames in Fig. 2, all expressed with respect to the robot base frame $\{h_B\}$. We denoted by $\{h_0\}$ the reference frame associated to the operator's hand in the first time instant of the process, which is fixed with respect to the robot base. During the execution of the task, the orientation of the human hand, expressed with $\{h_H\}$, is tracked by the camera whose frame is denoted as $\{h_C\}$. Based on this, the relative rotation of the hand with respect to $\{h_0\}$ can be computed in terms of rotation matrices as follows:

$$\mathbf{R}_H^0 = (\mathbf{R}_0^B)^{-1} \mathbf{R}_H^B \quad (2)$$

Equation (2) can easily be transformed to $\mathbf{r}_H^0 = \theta_H^0 \mathbf{u}_H^0$ leveraging the *Rodrigues* notation (θ_H^0 and \mathbf{u}_H^0 can be computed as described in [22]). To address Requirement 1 we compute the *scaled* rotation as follows:

$$\bar{\mathbf{r}}_H^0 = \sigma_r \theta_H^0 \mathbf{u}_H^0, \quad \text{with } \sigma_r = 1 + \frac{k_m - 1}{1 + e^{-a_d(\|\mathbf{r}_H^0\| - \delta_d)}} \quad (3)$$

where σ_r represents a decreasing sigmoidal function that can take values between k_m , that is a tunable parameter, and 1. Parameters a_d and δ_d are the steepness and the center of the sigmoidal, respectively. In this way, after computing the equivalent matrix form of equation (3) ($\bar{\mathbf{R}}_H^0$), it is possible to invert equation (2) to find $\bar{\mathbf{R}}_H^B$. Finally, to keep consistency with equation (1), $\bar{\mathbf{R}}_H^B$ is transformed into $\bar{\mathbf{g}}_o$, which represents the scaled absolute rotation of the operator's hand that should be reached by the robot. Referring to Fig. 2, the robot is supposed to rotate by the *scaled* quantity $\sigma_r \theta_H^0$ as a response to a rotation of θ_H^0 of $\{h_H\}$. For how we engineered our framework, the $\{x, y, z\}$ Cartesian coordinates of the human hand do not influence the performance, since our main goal is to reduce the operator's hand range of motion. Despite DMPs

would also allow to implement additional modifications, we decided to adjust the robot's behavior only in terms of orientation, allowing the operator to freely modulate their Cartesian distance to the workpiece. As shown in Section V, this choice does not preclude the possibility of applying the framework with operators characterized by heterogeneous heights, even though the z Cartesian coordinates are not directly considered.

Requirement 2 concerns the *responsiveness* of the robot with respect to changes in the orientation of the goal (*i.e.*, the human's hand). To account for this, we introduced

$$\gamma_s = \alpha_z k_s \sigma_s \boldsymbol{\eta}, \quad \text{with } \sigma_s = \frac{1}{1 + e^{a_s(d(\bar{\mathbf{g}}_o, \mathbf{q}) - \delta_s)}} \quad (4)$$

where k_s is a tunable scaling factor, while σ_s is a sigmoidal function centered in δ_s and with steepness a_s . The term $d(\bar{\mathbf{g}}_o, \mathbf{q}) \in \mathbb{S}^3$ is defined as in [21] and represents a *distance* metric valid for rotations. The rationale behind γ_s is the following: in case the relative rotation between human and robot is above a limit (δ_s), the robot should quickly *react* to this mismatch, trying to catch up with the orientation of the operator (possibly scaled according to equation (3)). To fully exploit this behavior, the product $k_s \sigma_s$ should be as close as possible to 1: in this case, as visible from equation (1), the term γ_s would be equal and opposite to $\alpha_z \boldsymbol{\eta}$, removing the damping from the system and increasing the responsiveness. For this reason though, an upper bound on k_s needs to be selected to avoid that the system becomes unstable.

The standard DMPs formulation [16] explicitly uses the trajectory execution time, τ . On the contrary, we use $\bar{\tau} = 1/\tau$, where $\bar{\tau}$ can be regarded as a time scaling factor, ranging from 0 (infinite execution time) to 1 (execution time equal to 1 second). The reason is that the PBO algorithm we exploited (see Section IV-C) is a learning-based iterative scheme and, to be effective, the design variables should have similar variational ranges, to balance their influence during the process.

Based on $\bar{\tau}$ and the scaling factors k_m and k_s introduced with equations (3) and (4), a set of tunable parameters emerges that directly influence the robot's responsiveness and its adaptive behavior. To optimize these parameters according to individual user preferences, we integrate the PBO framework presented in Section IV-C.

C. Optimization of the process parameters

In the context of HRC painting, the underlying *objective function* representing the process is difficult to model explicitly, considering the complexity and high dimensionality of the design variables involved. By defining $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ as the set of parameters and $J(\boldsymbol{\theta}) : \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}$ as the objective function, we can formalize the general initial problem as follows:

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \\ \text{s.t. } &\boldsymbol{\theta}_l \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_u \\ &\boldsymbol{\theta} \in \Gamma \end{aligned} \quad (5)$$

where $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_u$ are the lower and upper limits on the domain, while $\Gamma = \{\boldsymbol{\theta} \in \mathbb{R}^{n_\theta} : g(\boldsymbol{\theta}) \leq 0\}$ is the set of q inequalities $g : \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^q$ constraining the problem.

To overcome the complexity of modeling J , we solve (5) through PBO by using the GLISp algorithm [5]. The PBO paradigm works as follows: it samples a candidate vector, and a new test implementing such parameters is executed. At the end of the test, a subject is asked to evaluate whether the last execution was *better*, *as good as*, or *worse* than the previous execution. The procedure is repeated for N tests, provided that this number guarantees convergence. Consequently, the problem of minimizing an objective function J can be interpreted as finding $\boldsymbol{\theta}^*$ such that $\pi(\boldsymbol{\theta}^*, \boldsymbol{\theta}) \leq 0, \forall \boldsymbol{\theta} \in \Gamma, \boldsymbol{\theta}_l \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_u$ after defining the following *preference function* $\pi : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_\theta} \rightarrow \{-1, 0, 1\}$:

$$\pi(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \begin{cases} -1 & \text{if } \boldsymbol{\theta}_i \text{ is "better" than } \boldsymbol{\theta}_j \\ 0 & \text{if } \boldsymbol{\theta}_i \text{ is "as good" as } \boldsymbol{\theta}_j \\ 1 & \text{if } \boldsymbol{\theta}_i \text{ is "worse" than } \boldsymbol{\theta}_j \end{cases} \quad (6)$$

Consider a set of $N \geq 2$ samples, $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_N\}$, with $\boldsymbol{\theta}_i, \boldsymbol{\theta}_j \in \mathbb{R}^{n_\theta}$ and such that $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}_j \forall i \neq j$, with $i, j = 1, \dots, N$. Based on Θ , a *preference vector* $\mathbf{B} = [b_1, \dots, b_M]^\top \in \{-1, 0, 1\}^M$ can be created through equation (6). Here, M is the number of preferences and such that $1 \leq M \leq \binom{N}{2}$. Each element $b_h = \pi(\boldsymbol{\theta}_{i(h)}, \boldsymbol{\theta}_{j(h)})$, with $h \in \{1, \dots, M\}$, $i(h), j(h) \in \{1, \dots, N\}$ and $i(h) \neq j(h)$, encodes the preference with respect to the experiment h .

Since the underlying function J is unknown, we substitute it with a surrogate function $\hat{J} : \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}$ featured by a preference vector $\hat{\boldsymbol{\pi}}$, defined in the same way as equation (6) and such that $\pi(\boldsymbol{\theta}_{i(h)}, \boldsymbol{\theta}_{j(h)}) = \hat{\boldsymbol{\pi}}(\boldsymbol{\theta}_{i(h)}, \boldsymbol{\theta}_{j(h)})$, $\forall h \in \{1, \dots, M\}$. Based on the work proposed in [23], we defined the surrogate \hat{J} as the following *radial basis function* (RBF):

$$\hat{J}(\boldsymbol{\theta}) = \sum_{i=1}^N \beta_i \phi(\epsilon d_e(\boldsymbol{\theta}, \boldsymbol{\theta}_i)) \quad (7)$$

where $\epsilon > 0$ is a scalar hyper-parameter, $d_e(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2$, $\boldsymbol{\theta}_i, \boldsymbol{\theta}_j \in \mathbb{R}^{n_\theta}$ represents the Euclidean distance, while $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is the actual RBF, that we defined as $\phi(\epsilon d) = \frac{1}{1 + (\epsilon d)^2}$ (*inverse quadratic* definition). The vector $\boldsymbol{\beta} = [\beta_1, \dots, \beta_N]^\top$ contains the coefficients β_i appearing in equation (7), which can be computed by solving the convex optimization problem proposed in [5].

To enhance performances, we introduced a *recalibration* strategy for the hyper-parameter ϵ , similar to the one implemented in [13]. For this purpose, we define a sequence of L elements $\boldsymbol{\alpha} = \{10^{-1 + \frac{1}{5}(l-1)}\}_{l=1}^{L=10}$ and, when the recalibration is triggered, L new values of ϵ are computed (denoted as $\bar{\epsilon}$): the selected one (ϵ^*), minimizes the *leave-one-out cross-validation error*. By defining with \bar{N} the number of initial samples for the training phase, the set containing the indices of the experiments requiring recalibration can be computed as

$$\mathcal{I} = \{\bar{N}, \bar{N} + \lfloor \frac{N - \bar{N}}{4} \rfloor, \bar{N} + \lfloor \frac{N - \bar{N}}{2} \rfloor, \bar{N} + \lfloor \frac{3(N - \bar{N})}{4} \rfloor\} \quad (8)$$

For an effective sampling of the search space, a surrogate function \hat{J} is usually not enough. To overcome this issue, an acquisition function $a(\boldsymbol{\theta})$ can be used, provided that the prob-

lem proposed in equation (5) is reformulated as an iterative scheme by which new candidates are proposed:

$$\theta_{k+1} = \arg \min_{\substack{\theta \in \Gamma \\ \theta_l \leq \theta \leq \theta_u}} a(\theta) \quad (9)$$

where k represents the current iteration. In our work, we defined the acquisition function as:

$$a(\theta) = \frac{\hat{J}(\theta)}{\max_i(\hat{J}(\theta_i)) - \min_i(\hat{J}(\theta_i))} - \delta_E z_N(\theta) \quad (10)$$

where θ is the space limited by θ_l and θ_u , θ_i is a generic sample in the domain, $\delta_E \geq 0$ is the *exploration parameter*, while $z_N(\theta)$ is the exploration function, defined as in [24].

Alg. 1 shows the complete procedure. In steps 5, 9 and 14, we denote by $\pi^{-1}(\mathbf{B}(k))$ the operation that allows to retrieve the specific element $\Theta(k)$ that resulted in $\mathbf{B}(k)$.

V. EXPERIMENTAL VALIDATION AND USE CASE

We consider a Universal Robots UR5e cobot, equipped with a 3D-printed support to keep the piece at $d_R = 50$ mm from the flange (see Fig. 2). All algorithms ran on a laptop with 16 GB of RAM and with a Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz. We used ROS Noetic to handle the communication protocols and to collect the data (the sets \mathcal{X}_i and \mathcal{Z}_i described in Subsection V-A) coming from a OAK-D Pro camera at 30 Hz. Table I shows the selected parameters.

The experiments consist of $N_p = 15$ tests, each performed by a different participant, denoted as P_i , with $i \in \{1, \dots, 15\}$. The set was composed of 8 men and 7 women among PhD and master's students, with an average age of 25.2 years with standard deviation of ± 0.98 years, and an average height of 180.8 cm with standard deviation of ± 11.26 cm. As shown in Fig. 3a, the baseline trajectory taught with teleoperation (applying minor modifications to the work presented in [25]) presents a *c-shaped* profile, and is kept the same for all participants to obtain comparable results. We preferred this strategy over more intuitive ones, such as tracking the pose of a marker on the workpiece, because it provides more stable and reliable pose estimates, avoiding the noise and flickering that can affect vision-based measurements. The painting strategy can be seen on the right side of Fig. 3: as the robot moves from point \mathbf{p}_A to point \mathbf{p}_B (Fig. 3b), the operators should paint the part of the piece on their left (yellow); the other part of the piece (red) should be painted as the robot moves from point \mathbf{p}_B to point \mathbf{p}_C (Fig. 3c). Moreover, the component we used in the process is characterized by an extended surface (≈ 1 m²) and is featured with different curvature radii, which entail challenging painting conditions for the operators and a meaningful testbed for our framework. We substituted the real paint with natural coloring for food, which is non-toxic and does not require specific ventilation devices, but allowed us to perform *visual inspection* of the results considering the percentage painted surface as the main reference parameter [26]. A video of the experiments is available at this link¹.

The experiments consisted of two tests. In the first one, each participant was asked to paint the piece while the robot

Algorithm 1 Complete painting process (PBO + DMPs)

```

1: Given:  $N, \bar{N}, n_\theta, \mathcal{I}, \alpha, \theta_l \leq \theta \leq \theta_u$ 
2: Initialize:  $\mathbf{B} \leftarrow \mathbf{0}, \beta \leftarrow \mathbf{0}, \Theta \leftarrow \mathbf{0}, \theta^* \leftarrow \mathbf{0}, \epsilon \leftarrow 1$ 
3:  $\Theta(1 : \bar{N}) \leftarrow \text{GenerateInitialSamples}(\theta_l, \theta_u)$ 
4:  $\text{ModifiedDMPs}(\Theta(1), \Theta(2)), \text{ExecPainting}(\Theta(1), \Theta(2))$ 
5:  $\mathbf{B}(1) \leftarrow \pi(\Theta(1), \Theta(2)), \theta^* \leftarrow \pi^{-1}(\mathbf{B}(1))$ 
6: for  $i \leftarrow 2, \dots, N$  do
7:    $\text{ModifiedDMPs}(\Theta(i+1)), \text{ExecPainting}(\Theta(i+1))$ 
8:   if  $i < \bar{N}$  then
9:      $\mathbf{B}(i) \leftarrow \pi(\Theta(i+1), \theta^*), \theta^* \leftarrow \pi^{-1}(\mathbf{B}(i))$ 
10:   else
11:     if  $i \in \mathcal{I}$  ( $\mathcal{I}$  computed with Eq. (8)) then
12:        $\bar{\epsilon} \leftarrow \alpha\epsilon, \epsilon^* \leftarrow \text{Recalibration}(\bar{\epsilon}), \epsilon \leftarrow \epsilon^*$ 
13:     end if
14:      $\mathbf{B}(i) \leftarrow \pi(\Theta(i), \theta^*), \theta^* \leftarrow \pi^{-1}(\mathbf{B}(i))$ 
15:      $\beta \leftarrow \text{SolveConvexProblem}(\mathbf{B}(i))$ 
16:      $\hat{J}(\theta) \leftarrow \sum_{j=1}^i \beta(j)\phi(\epsilon d(\theta, \Theta(j)))$  (Eq. (7))
17:      $a(\theta) \leftarrow \frac{\hat{J}(\theta)}{\max_k(\hat{J}(\theta_k)) - \min_k(\hat{J}(\theta_k))} - \delta_E z_N(\theta)$  (Eq. (10))
18:      $\Theta(i+1) \leftarrow \text{ParticleSwarmOptimization}(a(\theta))$  (Eq. (9))
19:   end if
20: end for
21: return:  $\theta^*$ 

```

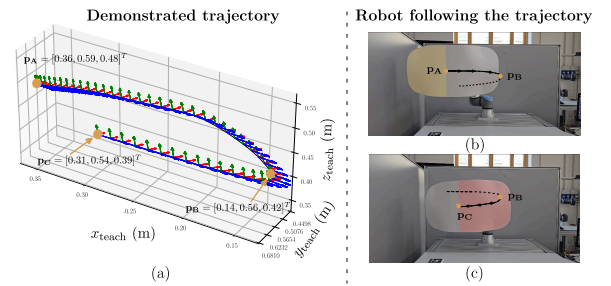


Fig. 3. (a) Demonstrated trajectory, taught offline with teleoperation and associated to $\{\mathbf{h}_R\}$. (b) Robot in \mathbf{p}_A . (c) Robot in \mathbf{p}_C .

was kept still in \mathbf{p}_A . This first test is representative of a non-collaborative baseline, common in current industrial setups. In the second test, we implemented the procedure summarized with Alg. 1. Each complete experiment (N iterations and the static case) lasted approximately 1 hour, due to the setup and cleaning times between experiments.

We aim at optimizing $\theta = [\bar{\tau}, k_s, k_m]^\top$ ($n_\theta = 3$), since they are the main parameters characterizing the robot behavior modeled through equation (1). We set the upper and lower limits as $\theta_u = [0.1, 0.7, 1.5]^\top$ and $\theta_l = [0.02, 0.3, 1.0]^\top$.

TABLE I
USER-DEFINED PARAMETERS FOR DMPs AND PBO.

DMPs							PBO		
a_d	δ_d	a_s	δ_s	α_s	α_z	β_z	\bar{N}	N	δ_E
-10	0.35 rad	-10	0.35 rad	5	25	6.25	6	15	0.5

A. Quantitative metrics for ergonomics assessment

According to the painting strategy, we expect a reduction in the range of motion of the operator's hand. In the literature, several scores have been proposed with the aim of quantifying the overall ergonomics [27], and they are meant to be applicable for a broad class of tasks. For the specific settings we tackle, we preferred to propose two custom indicators that were tailored to collaborative painting. Indeed, in this task the human is mainly required to move their hand along the x and

¹Video of the experiments: <https://youtu.be/0POZMUFvQUE>

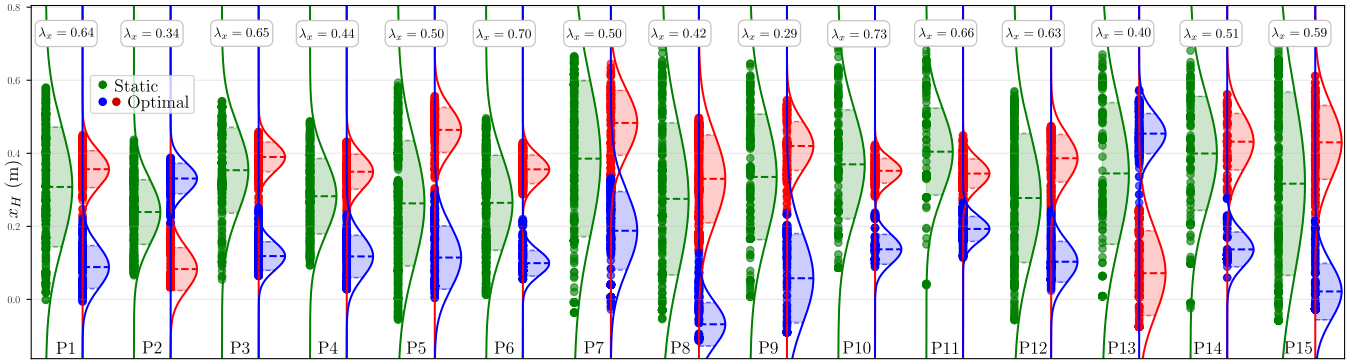


Fig. 5. Results for the *lateral ergonomics*. For each participant P_i , the red and blue curves (depicting the optimal case), are obtained through the *k-means* clustering technique. Despite representing the human hand positions along the x axis (horizontal), the data are reported vertically for readability.

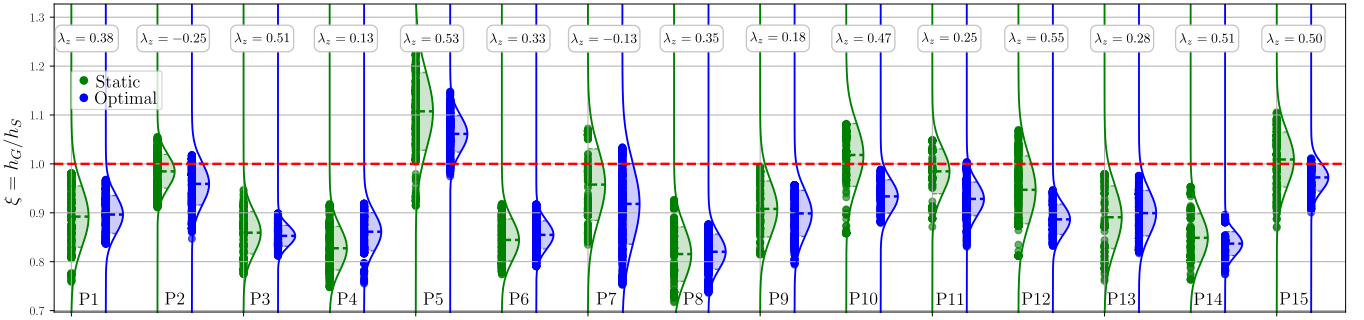


Fig. 6. Results for the *overhead ergonomics*. The dashed red line represents the acceptable limit on ξ , the green area is the data dispersion for the static case, while the blue one represents the optimized situation.

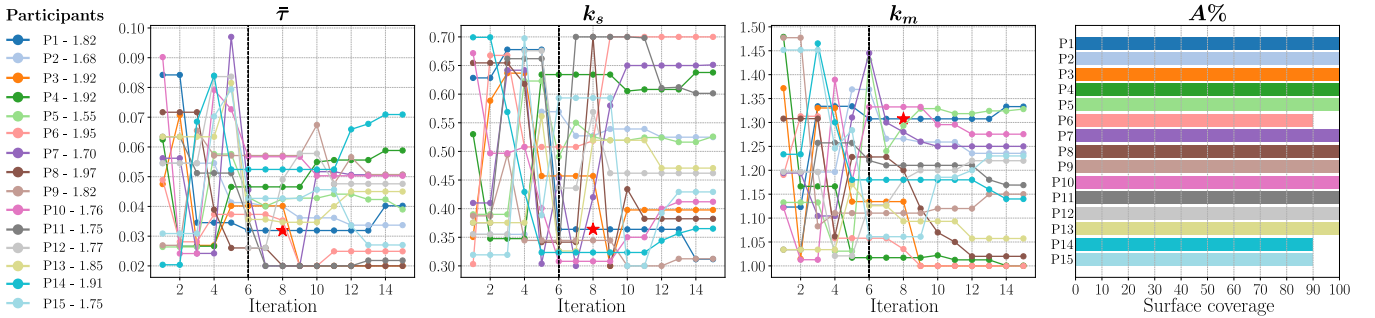


Fig. 7. Trend of the parameters composing θ as the number of iterations increases and the percentage painted surface $A\%$ for the final (optimal) test. The dashed vertical line separates the *training* iterations from the *active learning* stage. The point characterized by a red star in the trends for participant 1 at iteration 8 denotes an infeasible operation. In the legend, we also report the values of $h_{P,i}$ in meters.

proposed by PBO for that iteration and the best set selected until that moment, and they were asked to fill out a questionnaire inspired by the questions presented in [28], adapting and expanding them to suit the specific case under analysis. The proposed questions were as follows:

Question 1: Did the operation last a reasonable amount of time? (Q_1)

Question 2: How would you rate the robot assistance? (Q_2)

Question 3: How would you rate the general comfort of the operation? (Q_3)

The participants were asked to answer with scores ranging from 1 (*not satisfied at all*) to 5 (*completely satisfied*).

It is important to point out that the questionnaire and the related statistical analysis are not meant to be an exhaustive evaluation, but aim at providing an additional assessment of

the workers' perceived impact, in order to better contextualize the obtained results.

E. Analysis of the questionnaire scores

Figure 8 presents the results connected to the questions proposed to each participant, representing the average score $\bar{q}_{i,k} = (\sum_{j=1}^{N_p=6} q_{i,k}^j) / N_p$ for the i th question, given by the j th participant at the k th iteration. From the plot it is possible to notice the validity of the mechanism behind the iterative choice of the parameters: in the first iterations of the *active learning* phase delimited by the dashed vertical lines, the trends present evident fluctuations, that are the direct consequence of testing unexplored regions of the search space, with the higher likelihood of proposing testing conditions that are far from the desired ones. In terms of subjective experience, the candidates

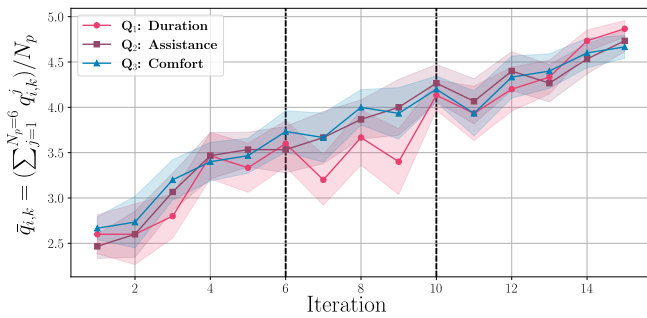


Fig. 8. Trend of the average scores $\bar{q}_{i,k}$. The central region exhibits higher fluctuations, characteristic of the exploration of the search space. The dispersion bands represent the standard error of the mean.

reacted with preferences characterized by higher uncertainty. However, all the three questions present an increasing trend, reaching values close to the maximum. This aspect, coupled with the scores for $A\%$ visible in Fig. 7, allows us to conclude that the framework is generally successful in optimizing the HRC painting application in a holistic sense.

VI. CONCLUSION

This work presented a novel human-centric collaborative framework that integrates Preference-Based Optimization (PBO) with Dynamic Movement Primitives (DMPs) to enhance HRC in industrial painting applications. By redefining the robot's role from a primary executor to an adaptive assistant, our approach allows the system to adjust its behavior based on real-time input from the working environment, significantly reducing the required ranges of motion and enhancing the perceived comfort. The proposed modifications to the DMPs framework proved effective in generating adaptive trajectories that respond to human preferences and ergonomics.

Possible future improvements are the inclusion of the points \mathbf{p}_A and \mathbf{p}_C into the set of design variables θ , in order to attenuate the problems connected to the initial mismatch of the operator's hand with respect to the piece. In addition, to account for slowdowns or fatigue of the operator, additional optimization and tracking modules may be added to correct the value of $\bar{\tau}$ based on modifications of the process conditions.

REFERENCES

- [1] J. M. Güllow, P. Paetzold, and O. Deussen. "Recent developments regarding painting robots for research in automatic painting, artificial creativity, and machine learning". In: *Applied Sciences* 10.10 (2020), p. 3396.
- [2] E. M. Van Zoelen, E. I. Barakova, and M. Rauterberg. "Adaptive leader-follower behavior in human-robot collaboration". In: *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2020, pp. 1259–1265.
- [3] I. W. Muzan, T. Faisal, H. Al-Assadi, and M. Iwan. "Implementation of industrial robot for painting applications". In: *Procedia engineering* 41 (2012), pp. 1329–1335.
- [4] A. M. Zanchettin, E. Lotano, and P. Rocco. "Collaborative robot assistant for the ergonomic manipulation of cumbersome objects". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6729–6734.
- [5] A. Bemporad and D. Piga. "Global optimization based on active preference learning with radial basis functions". In: *Machine Learning* 110.2 (2021), pp. 417–448.
- [6] A. Kupcsik, D. Hsu, and W. S. Lee. "Learning dynamic robot-to-human object handover from human feedback". In: *Robotics Research: Volume 1* (2018), pp. 161–176.
- [7] S. Proia, R. Carli, G. Cavone, and M. Dotoli. "Control techniques for safe, ergonomic, and efficient human-robot collaboration in the digital industry: A survey". In: *IEEE Transactions on Automation Science and Engineering* 19.3 (2021), pp. 1798–1819.
- [8] M. Lorenzini, M. Lagomarsino, L. Fortini, S. Gholami, and A. Ajoudani. "Ergonomic human-robot collaboration in industry: A review". In: *Frontiers in Robotics and AI* 9 (2023), p. 813907.
- [9] A. Castro, F. Silva, and V. Santos. "Trends of human-robot collaboration in industry contexts: Handover, learning, and metrics". In: *Sensors* 21.12 (2021), p. 4113.
- [10] K. Zbiss, A. Kacem, M. Santillo, and A. Mohammadi. "Automatic Optimal Robotic Base Placement for Collaborative Industrial Robotic Car Painting". In: *Applied Sciences* 14.19 (2024), p. 8614.
- [11] P. Schaldenbrand, G. Parmar, J.-Y. Zhu, J. McCann, and J. Oh. "Cofrida: Self-supervised fine-tuning for human-robot co-painting". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2296–2302.
- [12] M. Maccarini, F. Pura, D. Piga, L. Roveda, L. Mantovani, and F. Braghin. "Preference-based optimization of a human-robot collaborative controller". In: *IFAC-PapersOnLine* 55.38 (2022), pp. 7–12.
- [13] G. Campagna, M. Lagomarsino, M. Lorenzini, D. Chrysostomou, M. Rehm, and A. Ajoudani. "Promoting Trust in Industrial Human-Robot Collaboration through Preference-Based Optimization". In: *IEEE Robotics and Automation Letters* (2024).
- [14] M. M. Falerni, V. Pomponi, H. R. Karimi, M. L. Nicora, M. Malosio, L. Roveda, et al. "A framework for human-robot collaboration enhanced by preference learning and ergonomics". In: *Robotics and Computer-Integrated Manufacturing* 89 (2024), p. 102781.
- [15] L. Roveda, B. Maggioni, E. Marescotti, A. A. Shahid, A. M. Zanchettin, A. Bemporad, and D. Piga. "Pairwise preferences-based optimization of a path-based velocity planner in robotic sealing tasks". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6632–6639.
- [16] A. J. Ijspeert, J. Nakanishi, and S. Schaal. "Movement imitation with nonlinear dynamical systems in humanoid robots". In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*. Vol. 2. IEEE, 2002, pp. 1398–1403.
- [17] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel. "Dynamic movement primitives in robotics: A tutorial survey". In: *The International Journal of Robotics Research* 42.13 (2023), pp. 1133–1184.
- [18] F. Iori, G. Perovic, F. Cini, A. Mazzeo, E. Falotico, and M. Controzzi. "DMP-based reactive robot-to-human handover in perturbed scenarios". In: *International Journal of Social Robotics* 15.2 (2023), pp. 233–248.
- [19] G. Perovic, F. Iori, A. Mazzeo, M. Controzzi, and E. Falotico. "Adaptive Robot-Human Handovers with Preference Learning". In: *IEEE Robotics and Automation Letters* (2023).
- [20] K. Zbiss, A. Kacem, M. Santillo, and A. Mohammadi. "Automatic collision-free trajectory generation for collaborative robotic car-painting". In: *IEEE Access* 10 (2022), pp. 9950–9959.
- [21] A. Ude, B. Nemeč, T. Petrić, and J. Morimoto. "Orientation in cartesian space dynamic movement primitives". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2997–3004.
- [22] S. Bruno, S. Lorenzo, V. Luigi, and O. Giuseppe. *Robotics: modelling, planning and control*. 2010.
- [23] H.-M. Gutmann. "A radial basis function method for global optimization". In: *Journal of global optimization* 19.3 (2001), pp. 201–227.
- [24] M. Zhu, D. Piga, and A. Bemporad. "C-GLISp: Preference-based global optimization under unknown constraints with applications to controller calibration". In: *IEEE Transactions on Control Systems Technology* 30.5 (2021), pp. 2176–2187.
- [25] M. C. Welle, N. Ingelhart, M. Lippi, M. Wozniak, A. Gasparri, and D. Kragic. "Quest2ros: An app to facilitate teleoperating robots". In: *7th International Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interactions*. 2024.
- [26] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset, and A. A. Rizzi. "Uniform coverage of automotive surface patches". In: *The International Journal of Robotics Research* 24.11 (2005), pp. 883–898.
- [27] D. Kee. "Systematic comparison of OWAS, RULA, and REBA based on a literature review". In: *International Journal of Environmental Research and Public Health* 19.1 (2022), p. 595.
- [28] P. A. Lasota and J. A. Shah. "Analyzing the effects of human-aware motion planning on close-proximity human-robot collaboration". In: *Human factors* 57.1 (2015), pp. 21–33.