2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

A Model Predictive Control Approach to Blending in Shared Control

Elio Jabbour[1], Margot Vulliez[1], Célestin Préault[2], Vincent Padois[1]

Abstract— **Shared control aims at assisting human operators using robots in physically and cognitively demanding tasks which cannot be automated as they require human expertise and deliberative abilities. Sharing control for a given task typically involves blending algorithms that combine human control inputs and (pre)planned assistance trajectories. Conventional blending techniques, such as Linear Blending, compute a combined output but neither guarantee the feasibility of the blended motion nor the optimality of the combined decision. In the context of teleoperation, this paper presents a formulation where blending is defined as a constrained optimal control problem. Model Predictive Control is used to determine a feasible blended trajectory through a receding horizon constrained optimization. The proposed method is evaluated in a 13-participant pick and place teleoperation study and compared to Linear Blending and unassisted Teleoperation. The experimental results demonstrate the superiority of the proposed shared control framework in terms of safety, performance as well as physical and cognitive comfort.**

I. INTRODUCTION

Autonomous robots, while highly effective in speed and precision, often fall short in handling complex and dynamic scenarios [1]. Teleoperation allows human experts to perform tasks remotely in dangerous or inaccessible environments, such as in nuclear decommissioning [2]. Despite its advantages, pure teleoperation can be complex and lead to cognitive overload, as the operator must continuously process and respond to various sensory inputs [3]. Improving teleoperation thus requires a teaming approach that leverages the strengths of both the human and the robot.

Shared control frameworks aim at addressing this challenge by dynamically allocating control authority between the human and the robot so that they collaborate effectively [4]. This concept is first formalized by Sheridan as a continuum from full human operation to full automation [1]. Combining control inputs from human operators with those generated by an autonomous system typically involves blending algorithms that combine inputs from each source [5] or probabilistic [6] and game-theoretic approaches [7] that model the interaction as a dynamic, interdependent relationship.

Shared control frameworks are generally implemented as modular architectures composed of several components as exemplified for the proposed approach in Fig. 1. These architectures include a module to acquire **human input** (e.g., from a teleoperation device); an **assistance system** that generates an autonomous and goal-oriented trajectory; and a dynamic **authority allocation** module that adjusts the control balance, β, between the two agents. These human, assistance, and authority inputs are processed by a **blending controller**, which merges them into a unified command for the robot. To maintain the relevance of the assistance, most systems also include a module of **intent recognition** to infer the operator's goal based on their actions.

Blending can be a source of instability, operator frustration, and safety compromises [5]. This blending problem can be explained by four primary limitations that conventional shared controllers fail to address: abrupt shifts in control authority that introduce dynamic transients [8]; discontinuities in the assistance command affecting the robot motion [4]; inability to guarantee compliance with system constraints post blending [9]; and direct transmission of stochastic, high-frequency signals from the human input motion [10].

To address these limitations, this work investigates the formulation of the blending problem using a Model Predictive Control (MPC) approach. The proposed method is evaluated in comparison to pure teleoperation and linear blending in a pick-and-place user experiment. This study demonstrates that the MPC-based approach outperforms traditional methods in terms of safety, feasibility, repeatability, and human effort reduction. This systematic evaluation of the proposed approach through multiple metrics covering all aspects of performance also constitutes one of the contributions of this work. The paper is organized as follows: a review of the literature of shared control methods is given in section II, main components of standard shared control architecture are described in section III, the formulation of the proposed MPC-based blending controller (MPC-B) is presented in section III-C, the experimental set-up for the user study is described in section IV, the results are given and analyzed in section V. Conclusions and future works directions end the paper in section VI.

II. RELATED WORK AND CONTRIBUTION

The state-of-the-art in blending control aims at addressing several design principles: **verifiable safety**, **predictive foresight** to avoid reactive decisions, **computational tractability** for real-time application and **architectural decoupling** to adapt to different task/setups.

Basic instantaneous Linear Blending (LB) does not provide any of these features [5]. This conventional approach presents significant safety challenges, especially when human and assistance inputs conflict, as it can produce a meaningless and potentially unsafe median trajectory. Furthermore, LB is not designed to consider a priori constraints on the task, the robot's workspace, or its physical capabilities. It also lacks a built-in mechanism to ensure smooth transitions between

This work was supported by the ANR ASAP-HRC (ANR-21-CE10-0001) and the AAPR NA Perception-HRI (AAPR2022-2021-16819910) programs.

[1]Elio Jabbour, Margot Vulliez, and Vincent Padois are with Inria, Auctus team, Bordeaux, France. firstname.name@inria.fr

[2]Célestin Préault is with CESI Lineact, France. cpreault@cesi.fr

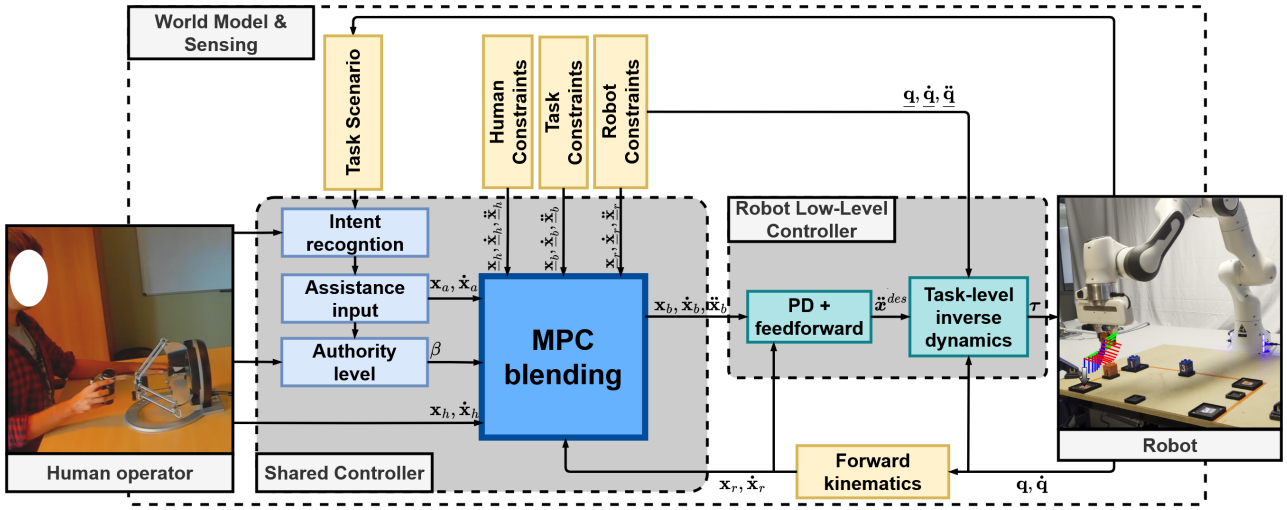

Fig. 1. Overview of the proposed modular control architecture. The system features a human operator controlling a Franka Emika Panda robot for a pick-and-place task. The high-level **Shared Controller** (left block) is centered around the **MPC blending** module. This module prospectively computes a safe and smooth blended trajectory in task-space (x_b, $\dot{x}_b$, $\ddot{x}_b$) by optimally blending the operator's input (x_h, $\dot{x}_h$) with an autonomous **Assistance** input (x_a, $\dot{x}_a$). The fusion is weighted by the **Authority** level (β) and is explicitly bounded by minimum and maximum safety limits (e.g., $\underline{x}_b$, $\underline{x}_h$, $\underline{x}_r$) defined by the **Human, Task, and Robot Constraints**. This feasible task-space command is then tracked by the **Robot Low-Level Controller** (right block), which calculates the required joint torques (τ) via inverse dynamics.

control inputs, which is problematic when authority shifts abruptly or when the operator's movements are noisy. Yet, some approaches try to tackle this problem. This section reviews shared-control approaches that can meet some of the design principles required to solve the blending problem.

Among these, game theory is employed to model human-robot interaction as a dynamic game between rational agents, where the blended action is computed by solving for an equilibrium strategy [7]. This approach provides a model of strategic co-adaptation, wherein the robot can anticipate and leverage the human's response to its own actions [11]. However, its application is limited by two assumptions. First, it models the human as a rational optimizer, a premise often violated in practice as human behavior systematically deviates from equilibrium due to cognitive biases [12]. Second, its formulation requires solving coupled Hamilton-Jacobi-Bellman (HJB) equations, which is computationally intractable for a high-frequency real-time robotic control [13].

To provide verifiable safety guarantees, a second approach tries to solve these problems, by formulating the control as a reactive optimization problem. These corresponding controllers are often implemented as a Quadratic Program (QP) that generates a final safe command for the robot [14]. In such approaches, the human's input is considered as the desired goal of the optimization. The QP then computes a new control command that minimally deviates from the operator's desired trajectory while strictly satisfying safety constraints, encoded using Control Barrier Functions . For instance, recent work has demonstrated how these barrier functions can be learned instantaneously from local sensor data to ensure safety in previously unknown environments [15]. This ensures that the command sent to the robot is feasible and safe, rather than simply filtering the human's input. This reactive controller provides instantaneous safety with high computational efficiency. However, relying solely on the current state without prediction, these controllers are short-sighted. This limitation has been identified, and solutions involving the integration of these methods with Model Predictive Control (MPC) have been proposed to enable predictive capabilities [16].

It is worth noting that these reactive optimization approaches are only applied to constraining or filtering the robot motion in direct teleoperation. Extending them to blending would require rewriting the optimization problem to include inputs from an autonomous agent and, therefore, provide assistance to the human.

A third data-driven approach uses techniques like Reinforcement Learning (RL) to learn a shared control policy from experience, thus bypassing the need to explicitly model the system dynamics or human intent. These methods can learn either an end-to-end robot control policy, which maps environmental observations and human input directly to motor commands [17], or a dedicated blending policy that arbitrates between the human and an assistance system [5]. However, the learning process demands a large amount of interaction data, and its reliance on exploration can be unsafe, as constraints are typically learned only by violating them [18]. Furthermore, the resulting policies are often encoded in neural networks, which prevents the formal verification of safety and, like reactive methods, can lack the prediction needed to avoid future constraint violations.

The literature reveals a trade-off among three competing paradigms. One option includes strategic controllers where the blended action is modeled as an optimal equilibrium strategy, but these are limited by unrealistic assumptions about human behavior and high computational demands (Game-Theoretic). The alternatives are safe but short-sighted approaches that often lack a principled blending mechanism, focusing instead on reactively filtering a single user input (Reactive Optimization),

or adaptive methods where blending is learned implicitly but without formal safety guarantees (Learning-based). This analysis motivates the need for a unified blending framework that can integrate verifiable safety and predictive foresight within a computationally tractable and a modular architecture, thereby providing a more robust foundation for shared control.
Several recent works use MPC to combine human commands with autonomous references. For example, MPC has been used as a baseline to mix a driver's joystick commands with a reference trajectory for an unmanned ground vehicle [19], and shared-autonomy MPC formulations penalize deviation from both operator commands and autonomous goals [20]. Sampling-based MPC approaches select safe, minimally-intervening inputs by evaluating many candidate trajectories rather than blending with a separate assistance signal [21]. Other systems treat blending as an external, post-hoc step: an optimal path is generated by MPC and then combined with driver input via a linear blending law outside the optimizer [22]. Shared haptic steering work uses dynamic weighting to allocate control between driver and automation [23], and some approaches inject detailed human cognitive/neuromuscular models into the prediction loop to improve haptic intuitiveness [24]. These methods share the multi-objective idea, but differ in where and how blending is realized (inside the optimizer, by sampling, or outside it), and in whether they explicitly enforce constraint compliance for the blended command.
The central contribution is MPC-B, a model-predictive blending framework that embeds a continuous authority parameter within a receding-horizon optimizer to combine operator and assistance objectives at each step. Blending is performed intrinsically within the constrained optimization, ensuring that the resulting command is constraint-compliant by design. The framework is implemented and validated in a teleoperation pick-and-place study, where MPC-B is compared to linear blending and unassisted teleoperation, demonstrating reduced operator effort, improved task performance, and significantly fewer kinematic violations.

III. BLENDING CONTROLLERS

The proposed MPC-B framework is implemented within a modular, cascaded control architecture, shown in Fig. 1. This structure separates high-level blending trajectory planning from low-level torque robot control. It allows solving the blending planning problem over a receding horizon at 1000Hz. This section details the blending components of the shared control architecture.

A. Main Components in Shared Control

Shared control in teleoperation consists of several components described in this section: human input, assistance system, authority level distribution, and blending controller.

In teleoperation, the user interacts with a joystick which captures the human position ${}^{H}\mathbf{x}_h \in \mathbb{R}^3$ for the robot to track, which is measured in the device base frame $\{H\}$. The transformation of the user input position to the robot frame is given by:

$$ {}^{R}\mathbf{x}_h = s\mathbf{R}_{RH}({}^{H}\mathbf{x}_h - {}^{H}\mathbf{x}_{hi}) + {}^{R}\mathbf{x}_{ri} \tag{1}$$

Here, H and R denote the teleoperation device and robot frames, respectively. ${}^{H}\mathbf{x}_{hi}$ and ${}^{R}\mathbf{x}_{ri} \in \mathbb{R}^3$ are the positions of the device and robot workspace centers. $s \in \mathbb{R}^+$ is a scaling factor, and $\mathbf{R}_{RH} \in \mathbb{R}^{3\times3}$ is the rotation matrix expressing the orientation of the haptic frame in the robot frame, for readability, the R-frame upper index is omitted in the equations for the rest of the section. The same transformation can be applied to the human velocity ${}^{H}\dot{\mathbf{x}}_h \in \mathbb{R}^3$. The position $\mathbf{x}_h$ and velocity $\dot{\mathbf{x}}_h$ are used in the shared controller as human inputs[1].

The assistance trajectory $[\mathbf{x}_a, \dot{\mathbf{x}}_a] \in \mathbb{R}^3$ is often learned by demonstrations [26] or preplanned [27]. It reflects the motion the robot would do if performing the task fully autonomously. Detecting the human's intention to align the assistance with the operator's goals is crucial, to avoid conflicting behaviors. A review on intent detection in shared control is given in [10].

The authority level, defined as a scalar $\beta \in [0, 1]$, allocates the control between the operator ($\beta = 1$) and the assistance system ($\beta = 0$). It can be modulated by specific factors such as the operator fatigue [28] or the robot reliability [29]. The expertise level of the operator can be learned, to provide less assistance as expertise grows [30]. Task proximity is another metric that can be used to shift the authority when precision is demanded [22].

This work assumes that both the human intent and the assistance trajectory are provided, with the focus placed on the blending controller. The objective in the subsequent sections is to compute the blending position and velocity, denoted as $[\mathbf{x}_b, \dot{\mathbf{x}}_b]$, using the function $f(\mathbf{x}_a, \mathbf{x}_h, \dot{\mathbf{x}}_a, \dot{\mathbf{x}}_h, \beta) \in \mathbb{R}^3$. This blended output is then used as the input to the robot low-level controller.

B. Linear Blending (LB)

A conventional approach to blend human operator and autonomous agent control inputs is linear blending, here illustrated at the position level:

$$\mathbf{x}_b = \beta\mathbf{x}_h + (1 - \beta)\mathbf{x}_a \tag{2}$$

As detailed in Section II, this method presents significant challenges related to safety, constraint compliance, and motion smoothness.

C. Model Predictive Control Blending (MPC-B)

These limitations can be addressed by formulating the blending mechanism as a constrained optimal control problem solved using Model Predictive Control (MPC). By utilizing a model of the system's dynamics, MPC predicts the effects of control actions over a receding horizon, enabling the computation of a locally optimal control strategy that respects predefined constraints.

Let $\mathbf{X}_b = \{\mathbf{x}_b, \dot{\mathbf{x}}_b\} \in \mathbb{R}^6$ be the desired robot end-effector state resulting from blending, and let the current robot task-space state be $\mathbf{X}_r = \{\mathbf{x}_r, \dot{\mathbf{x}}_r\} \in \mathbb{R}^6$. Blending is formulated as a minimization problem with two potentially conflicting objectives: tracking the human operator input $\mathbf{X}_h = \{\mathbf{x}_h, \dot{\mathbf{x}}_h\} \in \mathbb{R}^6$ and the autonomous agent input $\mathbf{X}_a = \{\mathbf{x}_a, \dot{\mathbf{x}}_a\} \in \mathbb{R}^6$. These

[1] In this work, the controller is formulated at the translation level. Extension of linear MPC to SE(3) is possible but is beyond the scope of this paper [25].

inputs are used as references for the whole prediction horizon. Considering a control variable $\mathbf{u}$ at the task space acceleration level, the proposed MPC cost function is defined as follows:

$$J = \sum_{k=1}^{H} \left[\|\mathbf{X}_{b,k} - \mathbf{X}_{h,k}\|^2_{\mathbf{Q}_h} + \|\mathbf{X}_{b,k} - \mathbf{X}_{a,k}\|^2_{\mathbf{Q}_a} + \|\mathbf{u}_k\|^2_{\mathbf{R}} \right] \tag{3}$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ is a positive definite matrix that penalizes the control effort with γ... The k index signifies the state $\mathbf{X}_{b,k}$ or control input $\mathbf{u}_k$ at the k-th step of the prediction horizon of length H. Similarly, $\mathbf{X}_{h,k}$ and $\mathbf{X}_{a,k}$ are the human and assistance inputs at step k. $\mathbf{Q}_h, \mathbf{Q}_a \in \mathbb{R}^{6\times6}$ are symmetric positive-semidefinite diagonal matrices that weight the tracking errors to the human and assistance inputs, respectively, according to the authority level β. These matrices are the product of two diagonal blocks that weight position and velocity errors separately and their structure is defined as:

$$\mathbf{Q}_h = \begin{bmatrix} \beta\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & k_{vh}\beta\mathbf{I}_{3\times3} \end{bmatrix}$$

$$\mathbf{Q}_a = \begin{bmatrix} (1-\beta)\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & k_{va}(1-\beta)\mathbf{I}_{3\times3} \end{bmatrix}$$

The dimensionless gains k_{vh} and k_{va} allow for tuning the controller's behavior, prioritizing velocity tracking (higher k_v) for responsiveness or position tracking (lower k_v) for goal-oriented precision. The specific values used for these gains and other key controller parameters are listed in Table I.

The system's evolution is approximated by a discrete-time linear model derived from a double integrator with a zero-order hold on the acceleration input:

$$\mathbf{X}_{bk+1} = \mathbf{A}\mathbf{X}_{bk} + \mathbf{B}\mathbf{u}_k \tag{4}$$

where $\mathbf{A} \in \mathbb{R}^{6\times6}$ and $\mathbf{B} \in \mathbb{R}^{6\times3}$ are the state-space matrices. Propagating this model over a prediction horizon of H steps yields:

$$\bar{\mathbf{X}}_b = \bar{\mathbf{A}}\mathbf{X}_{b0} + \bar{\mathbf{B}}\bar{\mathbf{U}} \tag{5}$$

$\bar{\mathbf{A}} \in \mathbb{R}^{6H\times6}$, $\bar{\mathbf{B}} \in \mathbb{R}^{6H\times3H}$ are the propagated system matrices and for a prediction horizon of H steps with time increment δt, the total duration is $T = H\delta t$. H is set to ensure that the optimization window is sufficiently long to anticipate constraints given the robot's braking capabilities. The stacked state and control input vectors are defined as follows, for $k = 1, ..., H$:

$$\bar{\mathbf{U}} = \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_{k+H-1} \end{bmatrix} \in \mathbb{R}^{3\times(H-1)} \quad \bar{\mathbf{X}}_b = \begin{bmatrix} \mathbf{X}_{b,k} \\ \mathbf{X}_{b,k+1} \\ \vdots \\ \mathbf{X}_{b,k+H} \end{bmatrix} \in \mathbb{R}^{6\times H}$$

One notes that $\mathbf{X}_{b0} = \mathbf{X}_r$ is the current robot state (position and velocity) while the sensor feedback ensures that this initial state is accurately represented.

D. Constraint Formulation

As illustrated in Fig. 1, the MPC incorporates constraints from three sources:

1) *Robot Constraints:* Cartesian velocity and acceleration limits are set based on the manipulator's manufacturer specifications for safe operation.
2) *Task Constraints:* Workspace position limits are defined as the intersection between the robot physical workspace limit and eventual task-related safety boundaries.
3) *Human Input Constraints:* The MPC enforces limits on the blended command's velocity and acceleration to filter out excessive variation or tremor originating from the human input.

These are expressed as linear inequalities over the horizon:

$$\bar{\mathbf{X}}_{b,min} \leq \bar{\mathbf{X}}_b \leq \bar{\mathbf{X}}_{b,max} \quad \text{and} \quad \bar{\mathbf{U}}_{min} \leq \bar{\mathbf{U}} \leq \bar{\mathbf{U}}_{max} \tag{6}$$

This leads to the constrained quadratic program (QP) solved at each time step:

$$\bar{\mathbf{U}}^\star = \arg\min_{\bar{\mathbf{U}}} \frac{1}{2}\bar{\mathbf{U}}^T\mathbf{H}\bar{\mathbf{U}} + \mathbf{F}^T\bar{\mathbf{U}} \tag{7}$$

subject to the constraints derived from equation (6). The Hessian matrix $\mathbf{H} \in \mathbb{R}^{(H-1)\times(H-1)}$ and gradient vector $\mathbf{F} \in \mathbb{R}^{(H-1)\times1}$ are derived by substituting equation (5) into the cost function equation (3) [31]. In accordance with the receding horizon principle, only the first element of the optimal control sequence $\bar{\mathbf{U}}^\star$ is applied to the robot, and the optimization is repeated at the next time step.

TABLE I
CONTROL PARAMETERS FOR THE SYSTEM

Parameter	Value	Parameter	Value
K_p (s^{-2})	250	K_d (s^{-1})	30
H (–)	10	δ (ms)	40
K_{va} (–)	1×10^{-2}	K_{vh} (–)	1×10^{-3}
K_{va} (TC) (–)	1000	K_{vh} (TC) (–)	1000
γ (–)	1×10^{-5}	γ (TC) (–)	1×10^{-9}
$\dot{x}^{min_b,max_b}$ (m/s)	$\{-0.3, 0.3\}$	$\ddot{x}^{min_b,max_b}$ (m/s^2)	$\{-7.0, 7.0\}$

IV. USER STUDY

To validate the performance of the proposed MPC-B framework against conventional Linear Blending (LB) and unassisted Teleoperation (T), a human-in-the-loop user study is conducted. This section details the experimental protocol, including the task and control modalities, the metrics used for evaluation, the study's core hypotheses, and the physical apparatus used.

A. Experimental Protocol

1) Task Description: The experiment is composed of a sequential pick-and-place task of four cube-shaped objects as depicted on the "Robot" block in figure 1. Each object is placed at a distinct location within the robot's workspace, and the participants are asked to pick-and-place them in order from 1 to 4 into given areas. A trial is defined as successfully picking and placing all four cubes sequentially. The grasping and releasing actions of the gripper are automated, allowing the operator to focus exclusively on the teleoperation of movement phases.

2) Design Rationale: The experimental protocol is structured to systematically probe the performance, human comfort, and safety compliance of shared controllers.

First, to test constraint-handling behavior, all target positions are placed near the hard boundaries of the workspace. This ensures that the terminal phase of each sub-task (point to point motion) creates an unavoidable trade-off between the operator's

primary objective (reaching the target) and the system's safety imperative (respecting workspace limits).

Second, to isolate the blending logic, the assistance signal for the shared control modalities is deliberately simplified: the assistance input $\mathbf{x}_a$ is set equal to the current target position $\mathbf{x}_t$ at each time step, being either the object to pick or place position. This ensures that any observed differences between LB and MPC-B are attributable solely to the blending mechanism itself, rather than an intent-inference module.

3) Control Modalities and Procedure: The study employs a within-subjects design where each of the participants experiences all three control modalities: Teleoperation (T), Linear Blending (LB), and Model Predictive Control with Blending (MPC-B). To mitigate learning effects, the sequence of the presented modalities is counterbalanced across the participant pool using a Latin square design [32]. The procedure for each participant is standardized, beginning with a training phase for each controller, followed by six experimental trials per modality, each trial is the completion of a sequence of the four pick-and-place tasks.

For all three modalities, a low-level, QP-based torque controller tracks the desired trajectories and reactively enforces the robot's joint capability and torque limits for workspace safety. This is a critical role of the low-level controller: the high-level commands for the T and LB modalities do not account for constraints, leaving safety enforcement entirely to the low-level controller in a reactive manner. The desired Cartesian acceleration $\ddot{\mathbf{x}}_{des}$ is computed via a standard Proportional-derivative (PD) tracking law: $\ddot{\mathbf{x}}_{des} = \mathbf{K}_p(\mathbf{x}_{des} - \mathbf{x}_r) + \mathbf{K}_d(\dot{\mathbf{x}}_{des} - \dot{\mathbf{x}}_r) + \ddot{\mathbf{x}}_{ff}$, where $\mathbf{K}_p$ and $\mathbf{K}_d$ are the proportional and derivative gains, respectively, and $\ddot{\mathbf{x}}_{ff}$ is a feedforward term. $(\mathbf{x}_r, \dot{\mathbf{x}}_r)$ is the current robot state and $(\mathbf{x}_{des}, \dot{\mathbf{x}}_{des})$ are the desired blended position and velocity that vary by the following control modalities:

1) *Teleoperation (T):* The operator has full, unassisted control over the robot. The desired states are set as $\mathbf{x}_{des} = \mathbf{x}_h$ and $\dot{\mathbf{x}}_{des} = \dot{\mathbf{x}}_h$.
2) *Linear Blending (LB):* The operator's command is blended with the assistance signal using the conventional linear law from Equation (2). The desired states are set as $\mathbf{x}_{des} = \mathbf{x}_b$ and $\dot{\mathbf{x}}_{des} = \dot{\mathbf{x}}_b$.
3) *MPC-B:* The operator's command is blended with the assistance using the proposed predictive framework in sec(III-C). The desired states are set as $\mathbf{x}_{des} = \mathbf{x}_b$ and $\dot{\mathbf{x}}_{des} = \dot{\mathbf{x}}_b$ and the feedforward term is set as $\ddot{\mathbf{x}}_{ff} = \mathbf{u}_0^\star$, the first element of the optimal control sequence.

This provides a direct comparison between ensuring proactive safety guarantees from the predictive planning (MPC-B) or only enforcing reactive safety checks at the low-level control for (LB) and (T).

For the shared control modalities (LB and MPC-B), the authority level, β is computed as a function of the robot's Euclidean distance to the target $\mathbf{x}_t$, governed by a sigmoid function as in [22]. This grants more control to the assistance system ($\beta \to 0$) for precise positioning near the goal while returning control to the human ($\beta \to 1$) when far from it.

The specific parameters for all three control modalities are kept constant throughout the experiment and are summarized in Table I, where TC stands for terminal cost.

TABLE II
EVALUATION METRICS FOR USER STUDY

Metric (Unit)	Formulation and Description
H1: Human Factors	
Workload	NASA-TLX
Effort (J)	$\sum_{i=1}^{N} F_h^T \dot{x}_{h,i} \Delta t$
H2: Task Performance	
Path Length (m)	$\sum_{i=1}^{N} \|x_{r,i+1} - x_{r,i}\|_2$
Task Error (mm)	$\frac{1}{M}\sum_{i=1}^{M} \|x_{r,i} - x_{t,i}\|_2$
Task Failures	$\sum_{i=1}^{C} c_i$
H3: Interaction Quality	
Agreement	$\frac{1}{N}\sum_{i=1}^{N} \frac{\dot{x}_{h,i}^T \dot{x}_{a,i}}{\|\dot{x}_{h,i}\|\|\dot{x}_{a,i}\|}$
Assistance (%)	$100\% \times \frac{1}{N}\sum_{i=1}^{N}(1-\beta_i)$
Repeatability	Avg. Dynamic Time Warping dist. between trials
Smoothness (m/s^3)	$\sum_{i=1}^{N} \frac{1}{T}\|\dddot{x}_{b,i}\|\Delta t$
H4: Safety	
Violations (%)	$(T_{violation}/T) \times 100\%$

B. Evaluation Metrics

To conduct a comprehensive evaluation of the controllers, objective performance data and subjective user feedback are collected. The metrics are based on standard methods from the human-robot interaction literature and are summarized in Table II.

Task performance is evaluated by Completion Time, Path Length (the distance traversed by the end-effector $\mathbf{x}_r$ [33]), and Task Error, defined as the mean distance between the end-effector position $\mathbf{x}_{r,i}$ and the target location $\mathbf{x}_{t,i}$ across M targets. Task Failures is captured as the total count C of collisions with the cubes.

Operator experience is assessed using the standardized NASA-TLX survey for perceived cognitive Workload [34] and by computing mechanical Effort, the work done by the operator's force $\mathbf{F}_h$ over their commanded velocity $\dot{\mathbf{x}}_h$ over N time steps of duration Δt [35].

Interaction quality is characterized by Agreement, measured as the cosine similarity between the human velocity command $\dot{\mathbf{x}}_h$ and the assistance velocity $\dot{\mathbf{x}}_a$ [21], and the Level of Assistance, the average percentage of control authority given to the autonomous system through β_k [36]. Trajectory Repeatability is quantified via Dynamic Time Warping (DTW), which compares end-effector motion profiles of the same subtask across trials while accounting for temporal variations [37]. Smoothness is expressed as the average jerk $\dddot{\mathbf{x}}_b$ over the task duration T [38].

Finally, safety is evaluated by the percentage of time the system operates outside its predefined kinematic limits. These constraint Violations are calculated as the ratio of time spent in violation, $T_{violation}$, to the total task time, T [39].

C. Hypotheses

The following operational hypotheses are formulated to test the central claim that a predictive, constraint-aware blending framework (MPC-B) yields quantifiable improvements over

both reactive Linear Blending (LB) and unassisted Teleoperation (T).

H1 (Human Factors): MPC-B reduces the operator's physical Effort and perceived Workload compared to LB and T.

H2 (Task Performance): MPC-B yields superior task performance, resulting in the fastest Completion Times, shortest Path Lengths, fewest Task Failures and lower Task Errors.

H3 (Interaction Quality): MPC-B's predictive nature leads to a higher quality of interaction, evidenced by greater human-assistance Agreement, improved trajectory Repeatability, smoother motion (Smoothness) and a higher percentage of Assistance compared to LB.

H4 (Safety): MPC-B demonstrates superior safety, with significantly fewer kinematic Constraint Violations compared to LB and T modalities.

D. Participants and Experimental Setup

A cohort of 13 volunteers (9 male, 4 female; age range 22-31) participated in the study. They all have a background in robotics but are not familiar with shared-controlled teleoperation. The experimental hardware comprises a 7-DOF Franka Emika Panda manipulator as the remote agent and a Force Dimension Omega.7 haptic device as the master interface. Haptic feedback is disabled to focus the study on the effect of the proposed shared controller on the user behavior. The only feedback provided to the operator is a direct view of the working area of the robot.

V. RESULTS

A summary of the results and statistical analysis is presented in Table III. Our analysis evaluates the hypotheses concerning human factors, task performance, interaction quality, and safety. Due to the within-subjects design, a repeated-measures ANOVA or a non-parametric Friedman test is used, followed by post-hoc pairwise comparisons with Bonferroni correction. The choice of statistical test is dictated by the data's adherence to the normality assumption, with descriptive statistics reported (Tab. III) accordingly: mean $\pm$ (SD) for parametric data (ANOVA) and the more robust median (IQR) for non-parametric data (Friedman). The significance level is set at $p < 0.05$.

TABLE III
STATISTICAL ANALYSIS OF KEY PERFORMANCE METRICS
(N=13)

	Modality			Stats	
Metric	**Teleop (T)**	**LB**	**MPC-B**	**Test Stat.**	**MPC-LB(p)**
H1: Human Factors					
Effort (J)	13.95 (4.33)	8.90 (1.85)	**5.86 (1.72)**	$\chi^2(2)$=22.15	**0.001**
NASA-TLX	9.33 (3.17)	6.33 (3.00)	**3.83 (1.67)**	$\chi^2(2)$=18.62	**0.021**
H2: Task Performance					
Time (s)	26.06 $\pm$ 5.96	16.16 $\pm$ 2.40	**13.98 $\pm$ 1.63**	$F_{2,24}$=56.34	**0.011**
Path (m)	4.96 $\pm$ 0.71	4.11 $\pm$ 0.41	**3.60 $\pm$ 0.17**	$F_{2,24}$=25.42	**.007**
Failures (%)	0.83 (0.67)	0.33 (0.33)	**0.00 (0.17)**	$\chi^2(2)$=15.48	**0.033**
H3: Interaction Quality					
Agreement	0.16 $\pm$ 0.03	0.19 $\pm$ 0.05	**0.27 $\pm$ 0.04**	$F_{2,24}$=56.35	**<0.001**
Jerk (m/s^3)	272.2 $\pm$ 21.8	516.9 $\pm$ 52.4	**280.6 $\pm$ 20.8**	$F_{2,24}$=294.2	**<0.001**
Repeat. (DTW)	0.38 (0.13)	0.44 (0.16)	**0.31 (0.04)**	$\chi^2(2)$=11.69	**0.005**
Assist. (%)	–	40.59 $\pm$ 6.99	43.28 $\pm$ 1.92	t_{12}= $-$ 1.35	0.203
H4: Safety (Violations %)					
Velocity	29.97 $\pm$ 8.78	60.88 $\pm$ 8.43	**2.23 $\pm$ 1.04**	$F_{2,24}$=258.0	**<0.001**
Position	30.12 $\pm$ 15.25	8.96 $\pm$ 5.90	**0.65 $\pm$ 0.46**	$F_{2,24}$=39.50	**<0.001**
Accel.	29.69 (12.68)	58.84 (17.14)	**0.36 (0.62)**	$\chi^2(2)$=26.00	**<0.001**

Format: Mean $\pm$ SD (ANOVA) or Median (IQR) (Friedman). **Test Stats:** F_{df} (ANOVA), $\chi^2(df)$ (Friedman), t_{df} (t-test). Pairwise p-values utilize Bonferroni correction.

A. H1: Human Factors

The MPC-B framework reduces the workload imposed on the operator. The measured physical effort is lowest for MPC-B, followed by LB, with unassisted Teleoperation being the most demanding (Friedman test, $\chi^2(2) = 22.15$, $p < .001$). Subjectively, operators perceived the lowest overall workload with MPC-B, primarily driven by significant reductions in reported Mental Demand and Effort compared to the other modalities (NASA-TLX, $\chi^2(2) = 18.62$, $p < .001$).

B. H2: Task Performance

Shared control improves task efficiency, with MPC-B proving the most effective. The mean task completion time is lowest for MPC-B ($F_{2,24} = 56.34, p < .001$), which is faster than both LB ($p = .011$) and T ($p < .001$). Similarly, MPC-B enables operators to execute the task with the shortest path length (ANOVA, $F_{2,24} = 25.42$, $p < .001$). Both shared control modalities improve precision over teleoperation. MPC-B is the most effective modality at improving task success, showing a statistically significant reduction in task failures (collisions) ($\chi^2(2) = 15.48, p < .001$) compared to both LB ($p = .033$) and T ($p = .007$).

C. H3: Quality of Interaction

The predictive nature and constraint enforcement of the MPC-B controller lead to a higher quality of interaction. Human-assistance agreement is significantly higher for MPC-B ($F_{2,24} = 56.35, p < .001$) compared to LB, indicating less conflictual interaction. Furthermore, trajectories executed with MPC-B present a lower variability, showing significantly higher repeatability ($\chi^2(2) = 11.69, p = .003$) than both LB ($p = .005$) and T ($p = .014$) according to the Dynamic Time Warping (DTW) analysis. In terms of motion smoothness, MPC-B also demonstrates a clear advantage over conventional blending, producing trajectories with significantly less jerk ($F_{2,24} = 294.2, p < .001$) than LB. The level of assistance does not show a significant difference between the shared control modalities (LB and MPC-B) ($t_{12} = -1.35, p = .203$).

D. H4: Safety and Constraint Compliance

The most pronounced difference between the shared controllers concerns safety compliance, validating hypothesis **H4**. The aggregated data in Table III shows a near-total elimination of kinematic constraint violations for the MPC-B framework. The predictive controller has statistically fewer constraint violations than LB and T across position ($F_{2,24} = 39.50$), velocity ($F_{2,24} = 258.0$), and acceleration ($\chi^2(2) = 26.00$) ($p < .001$ for all comparisons).

Fig. 2 shows the input/output trajectories of a single, representative motion from the user study, revealing the behaviors under the different control modalities. The plots highlight the difference between proactive and reactive safety. For Teleoperation and LB, the robot velocity and acceleration profiles are noisy and frequently breach the constraint boundaries (dashed lines). These breaches represent instances where the high-level

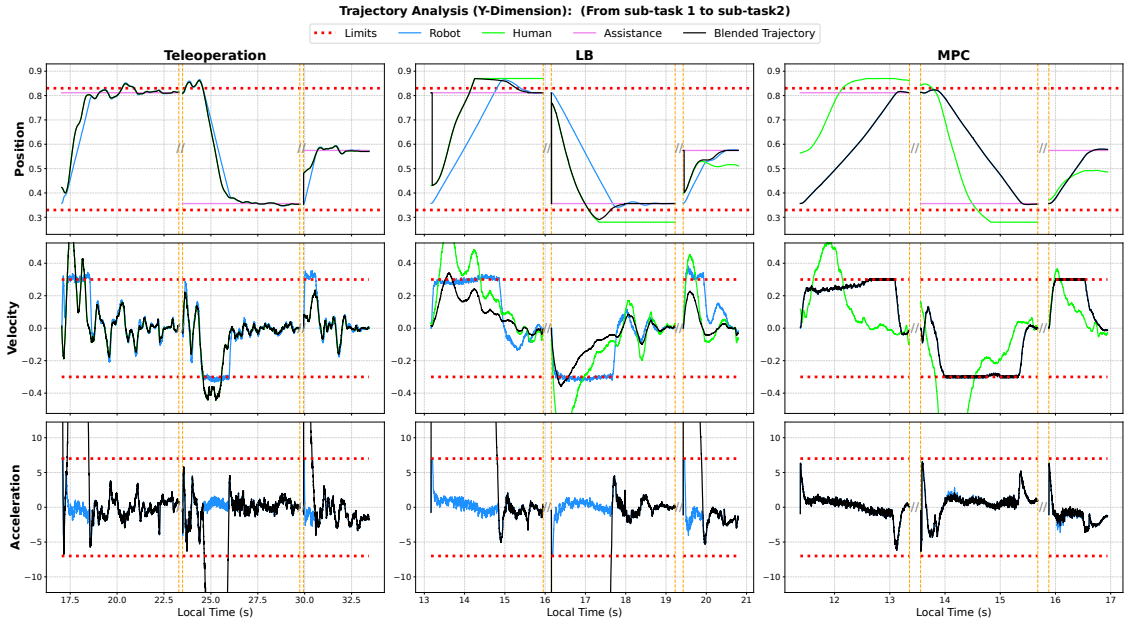

Fig. 2. Time evolution of the end-effector Y-position, velocity and acceleration for a representative trial under the three control modalities: (left) Teleoperation, (center) Linear Blending, (right) MPC-B. Each subplot shows local time on the horizontal axis and the corresponding state on the vertical axis. Gaps between orange segments indicate unplotted intermediate actions such as automated grasp or release phases that are not a zone of interest. The kinematic limits are represented by the red horizontal dot lines. Human (green) and assistance (purple) inputs are shown on each plot. The desired trajectory is plotted in black and shows the output behavior from the blending controllers. The current robot trajectory is given in blue, and is the output of the low-level controller that either tracks the human (in T) or the blending command (in LB and MCP-B).

commands are unsafe, as shown on the blending trajectory output from LB, and saturated by the low-level controller. This constant reactive filtering is what produces the noisy, clipped appearance of the acceleration signal. In contrast, the trajectory generated by MPC-B is different; its acceleration profile is smooth within the specified limits because the high-level command is already feasible. This demonstrates that the controller is not merely filtering an unsafe command, but is generating a new, proactively safe plan from the outset.

The stark differences in controller behavior can be seen in the Teleoperation modality (left column) where the position plot reveals a characteristic overshoot of the target, followed by corrective motions from the human operator. This reactive, trial-and-error approach, which places the full burden of correction on the human, directly explains the longer task completion time (26.06 s) and a 27.4% greater path length observed in the quantitative results compared to MPC-B.

The Linear Blending modality (center column) illustrates the effect of unconstrained blending. The resulting blended trajectory and its high-frequency acceleration signal demonstrate this by exhibiting a clipped appearance of the acceleration profile which is a direct result of the low-level controller reactively saturating unsafe high-level commands.

Conversely, the MPC-B controller (right) generates a proactively feasible plan. Its direct, non-overshooting position trajectory corresponds to the shortest path length (3.60 m). The smooth velocity curve shows deceleration that anticipates the position limit, rather than reacting to it. As a result, the acceleration profile remains well within the constraint boundaries. This proactive constraint satisfaction minimized kinematic violations (reduced to less than 2.3% across all metrics) and corresponds to a 46.3% reduction in completion time and a 58% reduction in physical operator effort compared to teleoperation. These plots factually demonstrate that the proactive, constraint-aware planning of MPC-B produces a stable and efficient motion profile, in contrast to the other modalities, which rely on low-level controllers to reactively enforce safety constraints after receiving potentially unsafe commands.

VI. CONCLUSION

This paper introduces a model predictive control-based blending, that is designed to overcome key shortcomings of conventional shared control methods by formulating blending as a constrained optimal control problem. This approach unifies trajectory planning and authority arbitration to proactively ensure safety and smoothness of the shared human-assistance trajectory. A 13-participant user study shows that the proposed, predictive and constraint-aware method offers a safer, more efficient, and more repeatable solution than both standard teleoperation and conventional Linear Blending with reactive safety enforcement. The results show that MPC-B eliminates kinematic constraint violations ($p < .001$). This validates that unifying trajectory generation with arbitration proactively resolves conflicts that standard blending can only react to. Future work will also focus on developing adaptive models that dynamically adjust the authority level based on contextual factors. This includes integrating online learning algorithms to

personalize assistance according to user skill, fatigue, or intent, and investigating hybrid approaches that combine predictive blending with data-driven policy adaptation. Finally, while the assistance system operates with reliable knowledge of the environment in this study, future research will explore scenarios with higher levels of uncertainty, where degraded assistance may require the user to assume greater control.

REFERENCES

[1] T. B. Sheridan, *Telerobotics, automation, and human supervisory control*. Cambridge, MA, USA: MIT Press, 1992.

[2] C. A. Contreras, M. Chiou, A. Rastegarpanah, M. Szulik, and R. Stolkin, "Probabilistic human intent prediction for mobile manipulation: An evaluation with human-inspired constraints," 2025.

[3] M. Goodrich and A. Schultz, "Human-robot interaction: A survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, pp. 203–275, January 2007.

[4] D. Abbink, T. Carlson, M. Mulder, J. de Winter, F. Aminravan, T. Gibo, and E. Boer, "A topology of shared control systems-finding common ground in diversity," *IEEE Transactions on Human-Machine Systems*, vol. 48, pp. 509–525, September 2018.

[5] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[6] S. Jain and B. Argall, "Recursive bayesian human intent recognition in shared-control robotics," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3905–3912.

[7] D. Sadigh, N. Landolfi, S. Sastry, S. Seshia, and A. Dragan, "Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state," *Autonomous Robots*, vol. 42, October 2018.

[8] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?" *Cogn. Technol. Work*, vol. 14, no. 1, pp. 19–28, 2012.

[9] S. Bustamante, I. Rodríguez, G. Quere, P. Lehner, M. Iskandar, D. Leidner, A. Dömel, A. Albu-Schäffer, J. Vogel, and F. Stulp, "Feasibility checking and constraint refinement for shared control in assistive robotics," *IEEE Robotics and Automation Letters*, vol. 9, no. 9, pp. 8019–8026, 2024.

[10] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human-robot interaction," *Applied Mechanics Reviews*, vol. 70, no. 1, p. 010804, 2018.

[11] S. Nikolaidis, S. Nath, A. D. Procaccia, and S. S. Srinivasa, "Game-theoretic modeling of human adaptation in human-robot collaboration," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 2017, pp. 323–331.

[12] R. Tian, L. Sun, M. Tomizuka, and D. Isele, "Anytime game-theoretic planning with active reasoning about humans' latent states for human-centered robots," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 4509–4515.

[13] J. Tan, S. Xue, H. Cao, and S. S. Ge, "Human-ai interactive optimized shared control," *Journal of Automation and Intelligence*, 2025.

[14] D. Lee, D. Ko, W. K. Chung, and K. Kim, "Maximal manipulation framework using quadratic programming for a teleoperated robotic system with articulated bodies," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9339–9345.

[15] C. Li, Z. Zhang, N. Ahmed, Q. Liu, F. Liu, and M. Buss, "Safe feedback motion planning in unknown environments: An instantaneous local control barrier function approach," *Journal of Intelligent & Robotic Systems*, vol. 109, October 2023.

[16] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete time control barrier function," in *2021 American Control Conference (ACC)*, 2021, pp. 3882–3889.

[17] S. Reddy, A. D. Dragan, and S. Levine, "Shared autonomy via deep reinforcement learning," in *Robotics: Science and Systems XIV*, 2018.

[18] A. Padalkar, G. Quere, F. Steinmetz, A. Raffin, M. Nieuwenhuisen, J. Silvério, and F. Stulp, "Guiding reinforcement learning with shared control templates," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 531–11 537.

[19] C. M. E. Bou, M. Focchi, M. R. Chang, M. Camurri, and K. D. von Ellenrieder, "Smooth human-robot shared control for autonomous orchard monitoring with ugvs," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 13 603–13 620, 2025.

[20] R. Lima, S. Saha, V. Vakharia, V. Vatsal, and D. K. Das, "Augmenting robot teleoperation with shared autonomy via model predictive control," Preprint, pp. 42–48, November 2024.

[21] A. Broad, T. Murphey, and B. Argall, "Learning models for shared control of human-machine systems with unknown dynamics," 2018.

[22] M. Yue, C. Fang, H. Zhang, and J. Shangguan, "Adaptive authority allocation-based driver-automation shared control for autonomous vehicles," *Accident Analysis & Prevention*, vol. 160, 2021.

[23] W. Guo, X. Song, H. Cao, S. Zhao, B. Yi, and J. Wang, "Human-centered driving authority allocation for driver-automation shared control: A two-layer game-theoretic approach," *Physica A: Statistical Mechanics and its Applications*, vol. 626, no. C, 2023.

[24] A. M. R. Lazcano, T. Niu, X. C. Akutain, D. Cole, and B. Shyrokau, "Mpc-based haptic shared steering system: A driver modeling approach for symbiotic driving," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 3, pp. 1201–1211, 2021.

[25] N. T. Alberto, "Predictive control of collaborative robots in dynamic contexts," Theses, Université Bordeaux, October 2023.

[26] M. J. Zeestraten, I. Havoutis, and S. Calinon, "Programming by demonstration for shared control with an application in teleoperation," *Robotics and Automation Letters*, vol. 3, no. 3, pp. 1848–1855, 2018.

[27] A. Hansson and M. Servin, "Semi-autonomous shared control of large-scale manipulator arms," *Control Engineering Practice*, vol. 18, no. 9, pp. 1069–1076, 2010.

[28] L. Peternel, N. Tsagarakis, D. Caldwell, and A. Ajoudani, "Robot adaptation to human physical fatigue in human-robot co-manipulation," *Autonomous Robots*, vol. 42, pp. 1011–1021, 2018.

[29] R. Balachandran, H. Mishra, M. Cappelli, B. Weber, C. Secchi, C. Ott, and A. Albu-Schaeffer, "Adaptive authority allocation in shared control of robots using bayesian filters," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 11 298–11 304.

[30] L. Milliken and G. A. Hollinger, "Modeling user expertise for choosing levels of shared autonomy," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2017, pp. 2285–2291.

[31] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[32] D. C. Montgomery, *Design and Analysis of Experiments*, 10th ed. John Wiley & Sons, 2019.

[33] D. Zhang, Z. Wu, J. Chen, R. Zhu, A. Munawar, B. Xiao, Y. Guan, H. Su, W. Hong, Y. Guo, G. S. Fischer, B. Lo, and G.-Z. Yang, "Human-robot shared control for surgical robot based on context-aware sim-to-real adaptation," 2022.

[34] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Human Mental Workload*, ser. Advances in Psychology, P. A. Hancock and N. Meshkati, Eds. North-Holland, 1988, vol. 52, pp. 139–183.

[35] M. Khoramshahi and A. Billard, "A dynamical system approach to task-adaptation in physical human-robot interaction," *Auton. Robots*, vol. 43, no. 4, pp. 927–946, 2019.

[36] R. M. Aronson and E. S. Short, "Intentional user adaptation to shared control assistance," in *2024 19th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2024, pp. 4–12.

[37] L. Gong, B. Chen, W. Xu, C. Liu, X. Li, Z. Zhao, and L. Zhao, "Motion similarity evaluation between human and a tri-co robot during real-time imitation with a trajectory dynamic time warping model," *Sensors*, vol. 22, no. 5, 2022.

[38] L. Bahar, Y. Sharon, and I. Nisky, "Surgeon-centered analysis of robot-assisted needle driving under different force feedback conditions," *Frontiers in Neurorobotics*, vol. 13, 2020.

[39] B. Xu and K. Sreenath, "Safe teleoperation of dynamic uavs through control barrier functions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7848–7855.