

# A Safety-Aware Shared Autonomy Framework with BarrierIK Using Control Barrier Functions

Berk Guler<sup>\*1,6</sup>, Kay Pompetzki<sup>\*1</sup>, Yuanzheng Sun<sup>1</sup>, Simon Manschitz<sup>6</sup>, Jan Peters<sup>1-5</sup>

**Abstract**—Shared autonomy blends operator intent with autonomous assistance. In cluttered environments, linear blending can produce unsafe commands even when each source is individually collision-free. Many existing approaches model obstacle avoidance through potentials or cost terms, which only enforce safety as a soft constraint. In contrast, safety-critical control requires hard guarantees. We investigate the use of control barrier functions (CBFs) at the inverse kinematics (IK) layer of shared autonomy, targeting post-blend safety while preserving task performance. Our approach is evaluated in simulation on representative cluttered environments and in a VR teleoperation study comparing pure teleoperation with shared autonomy. Across conditions, employing CBFs at the IK layer reduces violation time and increases minimum clearance while maintaining task performance. In the user study, participants reported higher perceived safety and trust, lower interference, and an overall preference for shared autonomy with our safety filter. Additional materials available at BarrierIK.

## I. INTRODUCTION

In teleoperation, shared autonomy combines operator input with autonomous assistance to improve task performance [1]. Two common approaches are (i) blending methods, where operator and autonomy commands are linearly blended against each other [2], [3], [4], [5], [6], [7], and (ii) policy methods, where operator input is treated as an observation to weight among predefined or learned assistance policies [8], [9], [10], [11], [12]. These paradigms improve efficiency and reduce workload, but they typically arbitrate in task space without enforcing constraints. As a result, even if each command is individually safe, the blended action may still violate safety requirements such as collision avoidance [2], [5], [4], [13], [6].

Several approaches have been proposed to address constraint satisfaction in teleoperation. Many integrate collision avoidance into the objective function, for example, through artificial potential fields [14], [15], [16], [17], [18] or cost terms. Others enforce feasibility by filtering operator commands through collision-aware inverse kinematics (IK) [19], [20]. While successful, these approaches treat safety as a soft constraint, which allows violations when task objectives dominate. In addition, modeling safety as an extra cost term

<sup>\*</sup>Equal contribution; <sup>1</sup>Technical University of Darmstadt <sup>2</sup>German Research Center for Artificial Intelligence (DFKI) <sup>3</sup>hessian.AI <sup>4</sup>Robotics Institute Germany (RIG) <sup>5</sup>Centre for Cognitive Science <sup>6</sup>Honda Research Institute Europe GmbH; Corresponding author: berk.gueler@tu-darmstadt.de

This work was supported by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG) and the DEMETER project (Grant no.: 01DR25003), the Honda Research Institute EU, Germany, the EU’s Horizon Europe project ARISE (Grant no.: 101135959)

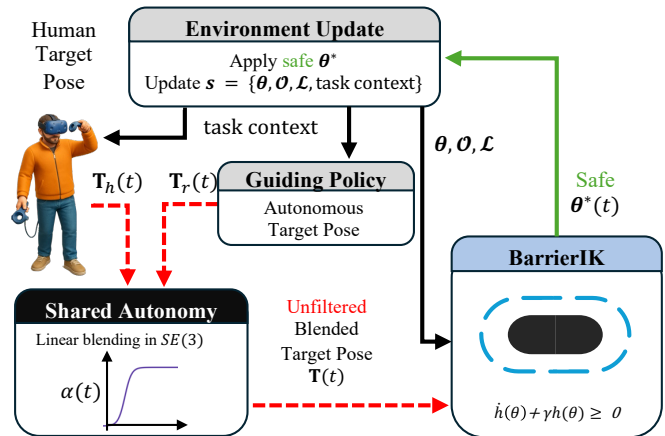


Fig. 1: Pipeline with CBF-based safety filtering *after* arbitration. The operator specifies a target pose  $T_h(t)$ ; the guiding policy outputs  $T_r(t)$  from the current state  $s(t) = \{\theta(t), \mathcal{O}(t), \mathcal{L}(t), \text{task context}\}$ , where  $\theta$  is the joint configuration,  $\mathcal{O}$  and  $\mathcal{L}$  are obstacle and robot link capsules, and “task context” includes task-specific info (e.g., rendered views). These unfiltered poses are blended in  $SE(3)$  via arbitration weight  $\alpha(t)$  to form  $T(t)$  (red dashed), which may violate safety margins. *BarrierIK* receives  $T(t)$  and enforces CBF conditions to compute a safe joint command  $\theta^*(t)$  (green), which is applied to the robot. The updated state  $s(t)$  is fed back to the policy and user.

can lead to conflicting behaviors and carries the risk of lowering user satisfaction.

A promising alternative is the use of control barrier functions (CBFs) [21], [22], [23]. CBFs are designed to ensure forward invariance of safety sets, offering a framework for formal constraints such as collision avoidance. Unlike cost-based methods, they do not compromise between safety and task performance. Instead, they morph potentially unsafe commands onto the safe set via constrained quadratic programming. CBFs have been successfully applied and teleoperation [24], [25], [26], [27], but their application in shared control and shared autonomy remains limited [28]. Advances in operational-space formulations now provide efficient scalability [29], motivating our approach to strictly enforce feasibility and aim for collision-free motion in high-DOF manipulation tasks.

We therefore propose BarrierIK, a CBF-constrained inverse-kinematics solver placed after policy blending. We hypothesize that projecting blended commands into the safety set, rather than treating safety as a soft objective, preserves task performance while increasing constraint satisfaction and maintaining user satisfaction. The pipeline

follows the shared-autonomy order: (i) blend the task-space command, then (ii) target safety at the IK layer. We evaluate BarrierIK in simulation on cluttered manipulation scenarios with dynamic and static obstacles, as well as in a VR teleoperation study with ten participants. In all tested scenarios, BarrierIK reduces violation time and increases clearance while maintaining task performance. The user study further indicates a consistent trend supporting our hypothesis: BarrierIK as a safety filter improves the safety and keeps performance in shared autonomy while preserving user satisfaction.

This paper makes three contributions. First, we present a shared-autonomy architecture that blends task-space commands and then addresses post-blend safety at the IK layer, handling failure modes of command-space blending in cluttered, nonconvex scenes. Second, we introduce BarrierIK, a CBF-constrained IK formulation that treats safety as a hard inequality constraint while remaining compatible with standard IK objectives, including pose tracking, joint limits, and smoothness. Third, we provide a comparative evaluation in simulation (dynamic obstacles and frame/shelf) and in a VR teleoperation study with ten participants, contrasting the proposed approach with IK baselines and quantifying benefits for both safety and task performance.

## II. RELATED WORK

In this section, we review three areas relevant to our scope. First, we discuss shared autonomy methods that incorporate autonomous assistance in teleoperation. Second, we address safety with a focus on collision avoidance. Finally, we review control barrier functions (CBFs), which provide formal safety guarantees as well as a principled way to incorporate safety.

**Shared Autonomy (SA):** aims to enhance operator performance and reduce workload by incorporating autonomous guidance [1]. A common paradigm is linear policy blending, where the final action is a linear combination of the operator’s and autonomy’s commands, based on inferred user intent [2], [5], [3], [4], [13], [7]. The autonomy policy typically take the form of optimization problems [2], [3], [13], movement primitives [11], [30], or encode the behavior from data [4], [6], [7]. The arbitration value controlling the trade-off between operator and autonomy commands depends on the confidence of the goal probability estimate, which is usually derived from the maximum entropy principle [2], [3], [13] or recursive Bayesian inference [31], [7].

Other works focus on policy-based methods, where the operator’s command is treated as an observation that influences predefined or learned assistance policies [8], [9], [10], [12]. These approaches formalize shared autonomy as partially observable Markov decision processes (POMDPs) for goal-aware action selection [8], [9], [10], or arbitrate over movement primitives based on operator commands [32], [12]. Cost functions for these POMDP formulations and movement primitives are typically predefined or learned from data.

Although many arbitration mechanisms exist, linear policy blending remains one of the most widely used methods.

However, it is prone to failure in cluttered or occluded scenes: even when human and autonomy commands are individually safe, their combination may fall outside a nonconvex safe set, leading to unsafe or misaligned actions [33]. These conflicts are exacerbated when multimodality in the joint human–robot distribution is ignored. A natural extension of linear blending to incorporate safety is the use of attracting and repelling fields [14], which add safety considerations but still treat safety as a soft constraint [4], [13], [6]. In navigation-style SA, goal-level blending arbitrates at the goal/distribution level instead of directly in command space, improving responsiveness in clutter [34]. Human factors results also show that making assistance legible (e.g., via haptic or transparent SA) improves agreement and satisfaction compared to opaque blends [35].

In this work, we retain the standard SA pipeline (blending first) but place safety enforcement after the blend at the IK layer. This ensures that post-blend actions prioritize safety and collision-avoidance while remaining consistent with the operator’s intent.

**Safety in teleoperation:** spans multiple notions beyond geometry, including limiting interaction forces and ensuring closed-loop stability (e.g., time-domain passivity), and active constraints/virtual fixtures that restrict motion within task-space envelopes [36], [37]. Here we focus on *environment collision safety* for the manipulator during assisted operation.

**Collision avoidance in teleoperation:** For teleoperated manipulation in *cluttered* environments, collision avoidance is commonly handled at the IK layer. Beyond IK, sampling-based teleoperation optimizers generate collision-aware updates by selecting feasible joint states near the current pose in real time [38]. In this section, we group IK layer methods into (i) optimization-based solvers and (ii) differential/null-space approaches. A canonical null-space strategy leverages operational space control [15], [16] and projects an obstacle-avoidance gradient into the Jacobian’s null space so the end-effector task is preserved while shaping self-motion [39]. However, such methods require robot redundancy and are limited by their hierarchical nature, where one task (e.g., end-effector motion) must be prioritized over others (e.g., collision avoidance).

Among optimization-based IK methods, CollisionIK augments a multi-objective IK with environment-distance penalties and evaluates a pruned active set of nearest link–obstacle pairs each step for tractability [19]. This yields smooth objectives and real-time rates, but safety remains *tradeable*: in narrow passages, the solver can accept transient violations to reduce tracking error, and behavior depends on active-pair selection (and weight tuning). Hard-constraint variants enforce minimum-distance inequalities directly in QP-based IK solver and report real-time collision-free updates when constraints are feasible [40]. These turn safety into a constraint (not a cost) but require reliable signed distances and careful aggregation across many contacts. When the task is infeasible at the chosen margin, they must either relax constraints or deviate substantially from the user’s command. Differential IK methods that do not rely on null space have

also been proposed for tight, cluttered workspaces (e.g., DawnIK [41]); their myopic linearization supports high-rate control but offers no forward-invariance guarantee and can be sensitive to step-size/regularization choices. We therefore adopt an optimization-based IK formulation to support both redundant and non-redundant manipulators.

**Control barrier functions (CBFs):** provide sufficient conditions for forward invariance of a safe set and are typically enforced via minimal-deviation quadratic programs [21], [22], [42]. In teleoperation, CBF filters have been used to guarantee geometric and interaction constraints while staying close to user input [24], [25], [26], [27]. For example, QP filters are employed to constrain interaction forces in robotic surgery [27]. In shared autonomy, works such as [28] address safety using barrier pairs, but are primarily focused on low-dimensional robotic navigation tasks. Recent developments extend CBFs to task-consistent operational-space control, providing kinematic and dynamic formulations that scale to hundreds or even thousands of simultaneous safety constraints [29]. These advancements motivate our work, where we introduce a post-blend safety filter that intercepts the blended user-autonomy command and resolves it at the IK layer. In this approach, we retain an optimization-based IK formulation while enforcing safety through CBF-style inequalities, treating safety as a hard inequality constraint after blending.

### III. A SAFETY AWARE SHARED AUTONOMY FRAMEWORK

In this section, we detail our real-time teleoperation framework. The system operates in two stages, as depicted in Figure 1: a high-level policy blending module combines user and autonomous motion commands via linear interpolation to generate a nominal Cartesian target. This command is then passed to an IK solver that ensures feasibility and safety with respect to kinematic and environmental constraints. We describe the policy blending strategy in Section III, followed by details on three IK solvers—Baseline N (N), Baseline P (P), and our novel Control Barrier Function-based BarrierIK (B).

**Shared Autonomy:** We assume the human operator provides Cartesian pose commands at each time step  $t$ , represented as a transformation  $\mathbf{T}_h(t) \in SE(3)$ , consisting of a position  $\mathbf{x}_h(t) \in \mathbb{R}^3$  and an orientation  $\mathbf{q}_h(t) \in \mathbb{H}$  (a unit quaternion). Simultaneously, the autonomous system proposes a reference pose  $\mathbf{T}_r(t) = (\mathbf{x}_r(t), \mathbf{q}_r(t))$ , derived from a grasp planner. In our experiments, these poses are heuristically defined based on geometric primitives associated with the target objects; however, the framework is agnostic to the grasp planner and can accommodate more sophisticated methods.

The final blended pose  $\mathbf{T}(t) = (\mathbf{x}(t), \mathbf{q}(t))$  is computed by interpolating both position and orientation. The position is blended linearly  $\mathbf{x}(t) = (1 - \alpha_t) \mathbf{x}_h(t) + \alpha_t \mathbf{x}_r(t)$ , where  $\alpha_t \in [0, 1]$  is the arbitration coefficient that modulates the influence of the autonomous system. The

orientation is blended using spherical linear interpolation (SLERP)  $\mathbf{q}(t) = \text{slerp}(\mathbf{q}_h(t), \mathbf{q}_r(t), \alpha_t)$ . To avoid discontinuities caused by antipodal quaternions, we enforce a positive dot product between  $\mathbf{q}_h(t)$  and  $\mathbf{q}_r(t)$  prior to interpolation. The resulting blended pose  $\mathbf{T}(t)$  is passed to the downstream IK solvers to generate a safe and executable robot configuration.

The arbitration coefficient  $\alpha_t$  can be fixed or adapted dynamically based on the disagreement between the user and robot commands. Following works like [3], we employ a sigmoid function for  $\alpha_t$  over the position-space disagreement  $\alpha_t = \sigma\left(p\left(\frac{\|\mathbf{x}_h(t) - \mathbf{x}_r(t)\|_2}{s} + b\right)\right)$ , where  $\sigma(\cdot)$  denotes the logistic function,  $p$  is a slope parameter,  $s$  is a scaling factor, and  $b$  is a bias term. This arbitration strategy reduces autonomy when human and robot intent diverge, and increases it when they align—encouraging fluid collaboration. The arbitration function follows the approach introduced by Muelling et al. [3].

**Inverse Kinematics:** Given a desired end-effector pose  $\mathbf{T}(t) \in SE(3)$ , we implemented a non-linear, optimization-based inverse kinematics (IK) solver to compute joint configurations  $\boldsymbol{\theta} \in \mathbb{R}^n$ , where  $n$  is the number of degrees of freedom (DoF). Our implementation builds on prior work [43], which enables smooth, feasible motions while accounting for pose tracking and self-collision avoidance by ensuring smoothness of motion. We re-implemented this framework in JAX to enable high-performance automatic differentiation for forward kinematics, self-collision checking, and differentiable collision detection algorithms for the IK solvers [44], [45], [46].

The IK problem is posed as a constrained optimization

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg \min_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) \\ \text{s.t. } c_k(\boldsymbol{\theta}) &\leq 0, \quad k = 1, \dots, K, \quad l_i \leq \theta_i \leq u_i \quad \forall i. \end{aligned} \quad (1)$$

Here,  $c_k(\boldsymbol{\theta})$  are inequality constraints (e.g., for manipulability or CBF constraint for BarrierIK), and  $[l_i, u_i]$  denote the joint limits. The objective function  $\mathcal{J}(\boldsymbol{\theta})$  is defined as a weighted sum of differentiable task-specific terms  $\mathcal{J}(\boldsymbol{\theta}) = \sum_{m=1}^M w_m g_m(F_m(\boldsymbol{\theta}))$ , where  $w_m \in \mathbb{R}^+$  is the weight,  $F_m(\boldsymbol{\theta})$  is a differentiable task feature, and  $g_m(\cdot)$  is a scalar penalty function of the  $m$ -th objective. We solve equation 2 using Sequential Least Squares Programming (SLSQP), an SQP method that, at each iteration, linearizes the constraints and minimizes a quadratic model of the Lagrangian, yielding a sequence of QP subproblems [47]. Because SLSQP relies on local linearizations, it enforces constraint feasibility only up to a numerical tolerance, providing approximate rather than continuous-time safety guarantees.

Following [43], the task features  $F(\boldsymbol{\theta})$  includes (i) end-effector tracking that penalizes position and orientation errors relative to  $\mathbf{T}(t)$ , (ii) motion-smoothness regularizers on joint velocities/accelerations/jerks and on Cartesian velocity, and (iii) a self-collision avoidance term that penalizes proximity between robot links using an ANN predictor of link-pair distances conditioned on  $\boldsymbol{\theta}$ . In our JAX implementation, this

network remains fully differentiable through the optimizer steps. In addition, we impose a manipulability constraint  $c_m(\theta)$  that keeps the Jacobian well-conditioned by preventing the smallest singular value from falling below a small threshold and capping the condition number at each step.

We use this IK solver in the following sections as a baseline, referred to as **Baseline N**. It includes the above-mentioned objectives and constraints to comprise the tracking, smoothness, and self-collision avoidance together with the manipulability constraint, but does not model obstacles.

**BarrierIK: Control Barrier Function-Based Environment-Aware Inverse Kinematics:** We introduce *BarrierIK*, an IK formulation based on Control Barrier Functions (CBFs), to target safety during teleoperation, by preserving the objectives and manipulability constraints as defined in Baseline N.

For each obstacle  $o \in \mathcal{O}$  we define the control barrier function  $h_o(\theta) = \phi_o(\theta) - \epsilon$  where  $\phi_o$  is the minimum signed robot–obstacle distance and  $\epsilon$  is the safety margin. In practice, we identify the active link–obstacle pair that realizes  $\phi_o$  (i.e., the closest pair) and compute its gradient via automatic differentiation [46].

We enforce a discrete-time CBF condition of the form  $\nabla h_o(\theta)^\top \Delta\theta + \mathcal{K}(h_o(\theta)) \geq 0$ , with extended class- $\mathcal{K}$  function  $\mathcal{K}(h) = \gamma h + \beta h^3$  where  $\gamma, \beta > 0$  are class- $\mathcal{K}$  coefficients controlling convergence to the safe set boundary. This formulation adopts a position-based update rule ( $\Delta\theta$ ) rather than a velocity-controlled formulation; while this deviates from classical continuous-time CBFs, such a discretized version suits our position-based control loop as a design choice.

To avoid discontinuities in constraint switching and to mitigate underflow or overflow issues, we aggregate all CBF constraints using a temperature-scaled soft-max (log-sum-exp) approximation

$$c_{\text{CBF}}(\theta) = \frac{1}{T} \log \sum_{o \in \mathcal{O}} \exp\left(T[-\dot{h}_o(\theta) - \mathcal{K}(h_o(\theta))]\right),$$

where  $\dot{h}_o(\theta) = \nabla h_o(\theta)^\top \Delta\theta$ . The temperature parameter  $T$  controls the max-approximation (higher  $T \Rightarrow$  closer to max).

The optimization problem of the *BarrierIK*

$$\begin{aligned} \theta^* &= \arg \min_{\theta} \mathcal{J}(\theta) \\ \text{s.t. } c_m(\theta) &\leq 0, \quad c_{\text{CBF}}(\theta) \leq 0, \quad l_i \leq \theta_i \leq u_i \quad \forall i, \end{aligned} \quad (2)$$

minimizes the same objective and retains the manipulability constraints as defined in Baseline N, along with the additional control–barrier inequality  $c_{\text{CBF}}(\theta) \leq 0$ . The CBF constraint enforces a stepwise safety margin while tracking and promotes forward invariance of the safe set under sufficiently small updates.

Instead of imposing obstacle avoidance as a hard constraint in *BarrierIK*, we employ it as a soft constraint inside the objective referred to as **Baseline P**. Following the proximity-penalty formulation in [19], it incorporates environment awareness by penalizing proximity between the robot and surrounding obstacles. For each obstacle  $o \in \mathcal{O}$

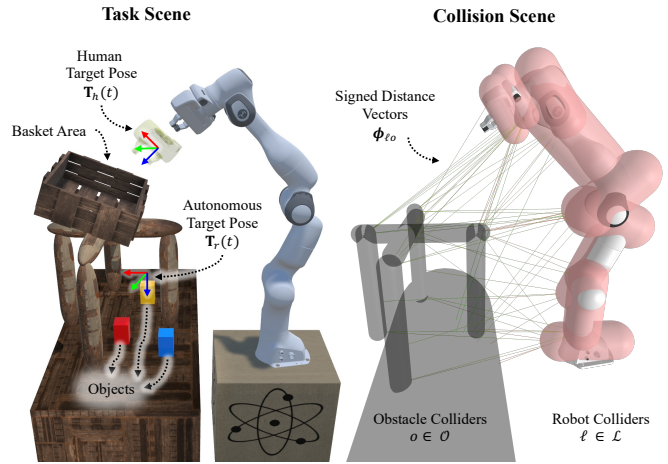


Fig. 2: Simulated task and collision-evaluation setup. **Left:** scene with human target  $\mathbf{T}_h(t)$  and autonomous reference  $\mathbf{T}_r(t)$ . **Right:** robot link (red) and obstacle (black) convex-capsule colliders; for each link  $\ell \in \mathcal{L}$  and obstacle  $o \in \mathcal{O}$  we compute signed distances  $\phi_{\ell o}(\theta)$  and signed distance vectors used by Baseline P and *BarrierIK* (colors encode magnitude).

and each robot link  $\ell \in \mathcal{L}$ , we compute the signed distance  $\phi_{\ell o}(\theta)$  between the link and the obstacle, and define the corresponding collision objective as  $\chi_{\ell o}(\theta) = w_{\text{safe}}(\phi_{\ell o}^2(\theta) + \delta)^{-1}$ , where  $w_{\text{safe}} = (5\epsilon)^2$  is a scaling factor based on the scalar safety margin  $\epsilon$ , and  $\delta$  is a small regularization constant to avoid division by zero. The final objective of Baseline P

$$\mathcal{J}_P(\theta) = \mathcal{J}_N(\theta) + w_{\text{col}} \cdot g_{\text{col}}(F_{\text{col}}(\theta)),$$

augments the Baseline N objective  $\mathcal{J}_N(\theta)$ , which combines end-effector tracking, motion smoothness, and self-collision avoidance by adding a collision penalty over all link–obstacle pairs  $F_{\text{col}}(\theta) = \sum_{\ell \in \mathcal{L}} \sum_{o \in \mathcal{O}} \chi_{\ell o}(\theta)$ , where each  $\chi_{\ell o}(\theta)$  penalizes proximity between link  $\ell$  and obstacle  $o$ . Also, manipulability constrained preserved  $c_m$  as defined in Section III.

The scalar weight  $w_{\text{col}}$  controls the relative importance of obstacle avoidance in the overall cost. This soft-penalty formulation allows the robot to trade off safety against task objectives in a smooth and differentiable manner, but it does not guarantee forward invariance or strict constraint satisfaction.

## IV. EXPERIMENTS

We evaluate the CBF-based safety layer in autonomous mode and teleoperation to test whether **BarrierIK** improves safety, performance, and user experience over two baselines: **N** (no environment collision handling) and **P** (soft penalty-based collision avoidance). We report collisions, clearance, tracking, and smoothness, marking significant B vs. P differences. To model the environment and robot links, we adopt a differentiable collision checking algorithm for capsule primitives [46], as illustrated in Figure 2.

### *Autonomous Evaluation Tasks and Environments:*

We evaluate *BarrierIK* in cluttered environments to assess both performance and generalizability. First, we perform autonomous rollouts on two complementary tasks—(i) a

TABLE I: Autonomous trajectory evaluation (mean  $\pm$  s.d.). For *Min. clearance*, higher is better (negative  $\Rightarrow$  penetration). For all other metrics, lower is better. **Bold** indicates a statistically significant B vs. P difference (N serves only as a reference; excluded from the significance analysis test).

Task	Solver	#Collisions	Min. clearance [m]	Violation time [%]	Pos. err. [m]	Ori. err. [°]	Task jerk [m/s <sup>3</sup> ]	Joint jerk [rad/s <sup>3</sup> ]
Shelf	N	44.13 $\pm$ 9.74	-0.162 $\pm$ 0.016	24.52 $\pm$ 1.19	0.020 $\pm$ 0.001	4.15 $\pm$ 0.07	4.197 $\pm$ 0.354	48.15 $\pm$ 2.89
	P	19.30 $\pm$ 5.85	-0.171 $\pm$ 0.017	4.26 $\pm$ 5.22	0.049 $\pm$ 0.018	<b>4.50 <math>\pm</math> 0.12</b>	6.518 $\pm$ 0.738	<b>57.22 <math>\pm</math> 3.35</b>
	<b>B</b>	16.80 $\pm$ 4.31	<b>-0.153 <math>\pm</math> 0.020</b>	<b>0.53 <math>\pm</math> 0.19</b>	0.043 $\pm$ 0.009	4.76 $\pm$ 0.49	<b>6.037 <math>\pm</math> 0.496</b>	65.89 $\pm$ 6.22
Dynamic	N	13.43 $\pm$ 3.30	-0.091 $\pm$ 0.025	56.25 $\pm$ 16.62	0.002 $\pm$ 0.004	7.26 $\pm$ 0.13	2.835 $\pm$ 0.437	45.55 $\pm$ 3.51
	P	1.55 $\pm$ 1.80	-0.033 $\pm$ 0.022	4.04 $\pm$ 9.17	0.023 $\pm$ 0.015	<b>7.35 <math>\pm</math> 0.25</b>	<b>3.219 <math>\pm</math> 0.911</b>	<b>33.62 <math>\pm</math> 6.25</b>
	<b>B</b>	2.05 $\pm$ 3.74	-0.024 $\pm$ 0.022	<b>0.84 <math>\pm</math> 1.92</b>	<b>0.017 <math>\pm</math> 0.010</b>	8.13 $\pm$ 1.32	5.949 $\pm$ 1.927	54.47 $\pm$ 4.92

N = Baseline N (no collision avoidance); P = Baseline P (objective-based collision avoidance); **B** = **BarrierIK (Ours)** (CBF-based collision avoidance)

shelf/frame scene with tight clearances, and (ii) a dynamic-obstacle scene with prescribed obstacle motions—each designed to stress different aspects of behavior such as minimum clearance, violation time, pose tracking, and smoothness. We then test BarrierIK in a human-in-the-loop teleoperation setup. All controllers share the same non-environment objectives; BarrierIK differs only by enforcing CBF safety at the IK layer, and the same safety margin  $\epsilon$ .

In the **Dynamic Obstacles (DO)** task, three cylindrical obstacles oscillate along orthogonal front-back, left-right, and vertical axes with distinct phases (capped at 0.025 m/s), occasionally approaching the tool. The end-effector remains near center with primarily rotational adjustments, mimicking camera/tool reorientation and stressing responsiveness to moving constraints. The **Frame / Shelf (FS)** task contains a rectangular shelf-like frame with four large “windows” separated by vertical cylindrical bars, simulating a tool insertion task. The end-effector must pass through each opening in sequence with varying orientations. Transitions between windows are deliberately challenging: the reference trajectory lightly penetrates the frame edges, exposing whether controllers can maintain clearance without sacrificing tracking fidelity. For each task, we specify waypoints and generate a time-parameterized end-effector trajectory. All IK solvers track the identical predefined trajectory.

We conducted a **VR-based teleoperation user study** to evaluate the proposed shared autonomy framework with a 7-DoF arm controlled via HTC Vive handheld controllers in Unity3D, as depicted in Figure 2. The task was to pick three color-coded cubes and place them in a basket within clutter. The task involved picking up three color-coded cubes and placing them into a designated basket. The environment was deliberately cluttered, requiring precise navigation around obstacles to complete the task. Ten participants (6 male, 3 female, 1 non-binary; mean age:  $27.9 \pm 3.14$ ) completed trials under six configurations: (i) Baseline N without shared autonomy (N), (ii) Baseline P without shared autonomy (P), (iii) BarrierIK without shared autonomy (B), (iv) Baseline N with shared autonomy (SA-N), (v) Baseline P with shared autonomy (SA-P), and (vi) BarrierIK with shared autonomy (SA-B). Participation was voluntary, informed consent was obtained beforehand, and all data were collected and analyzed anonymously without collecting any personally identifiable information. Each participant completed five zero-shot trials per condition without prior task-specific training. A 3-minute familiarization phase preceded the trials. Condition

order was randomized per user to avoid learning effects. Each trial had a 3-minute time limit and a maximum of five collisions; exceeding either threshold resulted in failure. Trials were also marked as failed if objects were dropped outside the workspace.

The complete pipeline—including simulation, shared autonomy, and IK solving—ran at 90 Hz to match the VR headset refresh rate. Although the computational backend supported higher frequencies, rendering was the bottleneck. In 100 Hz-capped testing, B ran at 98.2 Hz ( $\pm 7.91$ ), P at 99.0 Hz ( $\pm 3.20$ ), and N at a fixed 100 Hz (due to implementation constraints). All computations ran on CPU (Intel® Core™ i7-14700F); GPU acceleration was disabled for compatibility with SteamVR and JAX, which lacks stable CUDA support on Windows.

**Evaluation Metrics:** We report objective and subjective metrics across autonomous rollouts and teleoperation.

In the **autonomous rollouts setting**, we evaluate the performance of three IK solvers—Baseline N (N), Baseline P (P), and BarrierIK (B)—across two challenging tasks. All controllers share the same non-environment objectives; differences emerge only from how they handle obstacle avoidance. We report the following metrics, averaged across 40 trials with randomized seeds: (i) *Number of Collisions*: Total link-obstacle collisions over the trajectory; (ii) *Minimum Clearance*: The smallest signed distance  $\min_{\ell,o} \phi_{\ell o}(\theta)$  between any robot link  $\ell$  and any obstacle  $o$ ; (iii) *Violation Time Percentage*: The percentage of timesteps for which any  $\phi_{\ell o} < 0$ , indicating penetration; (iv) *Position Error and Orientation Error*: The average Euclidean and rotational (quaternion) error between the executed and reference end-effector trajectory; (v) *Task Jerk*: The time-averaged magnitude of third-order derivatives of end-effector position, reflecting Cartesian motion smoothness; (vi) *Joint Jerk*: The time-averaged magnitude of joint-space jerk  $\ddot{\theta}(t)$ , indicating low-level actuation smoothness. For evaluation, collision checking is performed using the HPP-FCL [48], a high-fidelity geometric proximity library capable of computing signed distances between convex capsule models.

In the **teleoperation experiments**, participants teleoperate the robot in VR. We evaluate task efficiency and safety across the three solvers—Baseline N, Baseline P, and BarrierIK—each with and without shared autonomy.

As objective metric we report (i) *Success rate*: percentage of trials placing all three objects without breaching time or collision limits; (ii) *Completion time* (successful trials

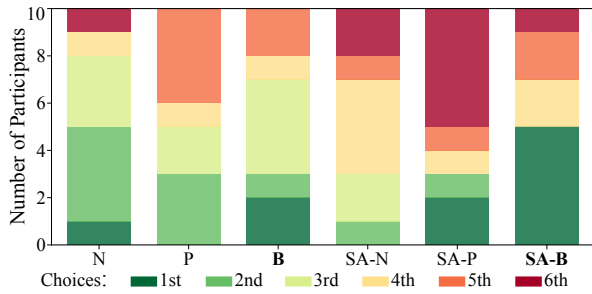


Fig. 3: The stacked bar chart illustrates the number of participants who ranked each controller configuration from 1st to 6th choice. Each color segment within a bar represents the count of users selecting that configuration at the respective rank.

only), measured from first human movement  $t_i$  to final placement  $t_f$ ; (iii) *Collision count* from Unity PhysX. Internally, solvers use JAX-based differentiable collision models, but Unity’s engine provides real-time contact detection and user feedback/visualization. We omit position/orientation tracking errors and jerk in teleoperation because human variability makes these low-level metrics less diagnostic for comparison.

To assess perceived usability based on subjective metrics, we administered a modified NASA Task Load Index (TLX) [49] after each configuration, in which all dimensions were inverted so that higher scores indicate more favorable outcomes. The inverted scales included physical (I-PD), temporal (I-TD), and mental demand (I-MD), perceived effort (I-EFF), frustration (I-FL), and perceived performance (I-PER). We further extended the TLX with dimensions specific to shared autonomy, namely perceived control level (CL), assistance level (AL), and safety (SL). These ratings were visualized as polar plots, where larger areas represent a more favorable user experience. Finally, participants provided a ranked list of the six configurations from most to least preferred after completing all trials.

## V. RESULTS

This section presents our experimental findings across autonomous and teleoperation settings. We evaluate BarrierIK against baseline methods using standard safety and performance metrics, and report both quantitative results and user study outcomes.

**Autonomous Trajectory - Evaluation Results:** Across both scenes, **B** achieves markedly lower *violation time* and less negative *minimum clearance* than **P**, indicating that when contacts do occur they are *shorter and shallower*. The residual violation time for **B** results from discrete-time position updates and SLSQP’s local linearization, which can lead to transient numerical penetrations between control steps. Practically, the controller does not linger near the boundary but returns to the safe set promptly—i.e., it *recovers* from adverse states rather than getting *stuck* at the constraint.

**Teleoperation - Subjective Results:** Participants split into two clear archetypes: a group favoring **N** (no assistance) and another preferring **SA-B** (shared autonomy with BarrierIK) (Fig. 3). These configurations sit at opposite ends of the autonomy spectrum, yet both deliver favorable

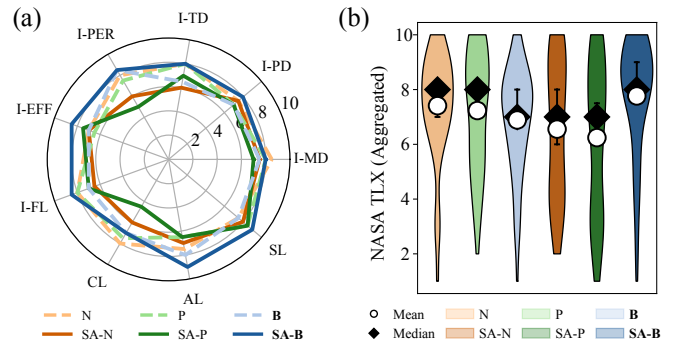


Fig. 4: Subjective workload and usability across teleoperation configurations. **(a)** Polar (“radar”) plot of modified NASA-TLX dimensions—physical demand (I-PD), temporal demand (I-TD), mental demand (I-MD), performance (I-PER), effort (I-EFF), frustration (I-FL), control level (CL), assistance level (AL), and safety level (SL)—with “I” denoting inverted scales so that larger values correspond to more favorable assessments. **(b)** Violin plots of participants’ aggregated NASA-TLX scores for each configuration (mean  $\circ$ , median  $\blacklozenge$ ), where lower workload (higher support) appears toward the top of the scale.

experiences, indicating that perceived quality depends as much on *how* assistance interacts with the operator as on *how much* assistance is provided.

On the inverted NASA-TLX radar (Fig. 4a), **N** and **SA-B** cover the largest areas: *lower perceived mental, physical, temporal demand, effort, and frustration*, together with higher *control level, assistance level, and self-reported safety level*. Notably, **SA-B** concentrates responses near the favorable region, suggesting that constraint-aware assistance can reduce workload without eroding agency, consistent with high CL alongside high AL and SL. In contrast, intermediate shared-autonomy modes (**SA-N**, **SA-P**) exhibit smaller, more dispersed footprints, reflecting mixed impressions: when assistance deviates from user intent, operators report reduced control and lower perceived support.

To summarize across dimensions, we aggregated inverted workload scores with CL/AL/SL into a composite measure (Fig. 4). **SA-B** achieved the highest average score, followed by **N**. The **SA-B** distribution also appears tighter, suggesting more consistent relief in perceived workload across participants. With  $n=10$ , a Friedman test with Nemenyi post-hoc found no significant pairwise differences (all  $p>0.05$ ), and therefore the observed patterns should be interpreted as indicative trends rather than conclusive effects. Nevertheless, the results indicate a coherent tendency: operators either prefer the transparency and predictability of **N** or the consistent, constraint-aware guidance of **SA-B**.

We interpret the results as follows: Participants accept help if the system prioritizes safety and they can retain control. Employing CBF at the IK layer guides actions to morph within safe and feasible regions rather than overruling the operator. Accordingly, **SA-B** scores better on CL (control), AL (assistance), and SL (safety) than **SA-N/SA-P**.

**Teleoperation - Objective Results:** For *success rate*, results cluster around  $\sim 80\%$  for most configurations; **SA-P**

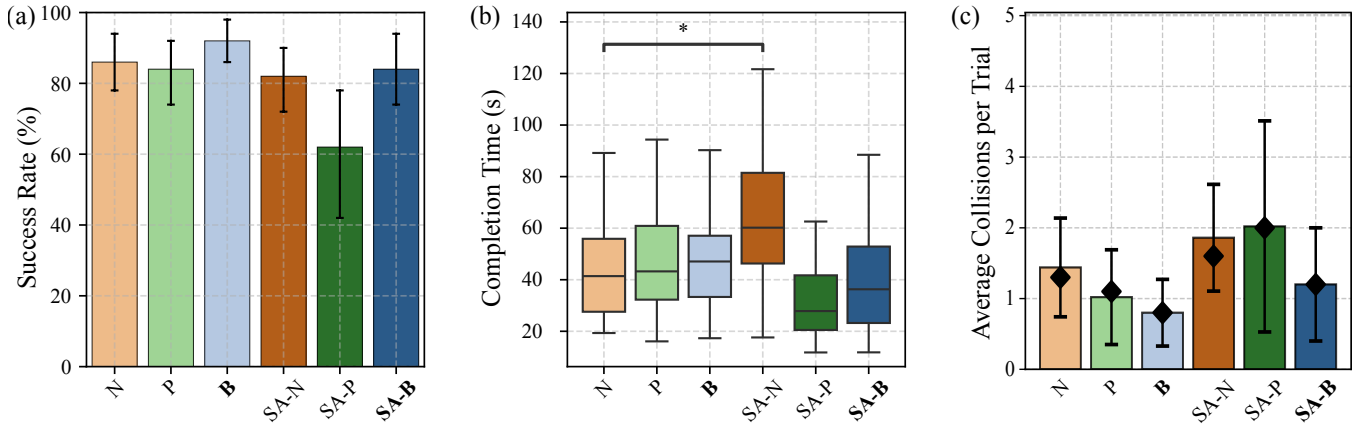


Fig. 5: Objective evaluation across six system configurations. Subplots (a)–(c) report performance and safety metrics: success rate (higher is better), task completion time (lower is better, measured only for successful trials), number of collisions (lower is better). ♦ indicates the median, asterisks indicate statistical significance: \*  $p < .05$ , \*\*  $p < .01$ , \*\*\*  $p < .001$ .

is the exception with a lower average success rate and higher variability (Fig. 5a). The non-assisted **B** configuration trends toward a higher median success rate with a tighter interquartile range; however, pairwise differences are not statistically significant at our sample size. In terms of *completion time* (successful trials only), **SA-P** appears faster on successful trials (Fig. 5b), but this reflects the exclusion of failed runs rather than an inherent efficiency advantage. For *collisions*, unstructured assistance tends to increase contacts in cluttered environments. **SA-N** and **SA-P** show higher collision counts than their non-SA counterparts (Fig. 5c). Among shared-autonomy modes, **SA-B** reduces collisions, while **B** (no SA) remains best overall on raw collision count.

## VI. CONCLUSION AND FUTURE WORK

We presented a hierarchical shared-autonomy framework that separates goal-directed guidance from IK-level safety enforcement. The safety layer, **BarrierIK (B)**, integrates control barrier functions with an optimization-based IK to systematically address safety *after* blending human and autonomous inputs. Our evaluation comprised (i) autonomous rollouts in representative cluttered scenes and (ii) a VR teleoperation study with and without shared autonomy. In autonomous rollouts, **B** consistently reduced *violation time* and yielded *less-negative minimum clearance* compared to the cost-based baseline **P**. These findings indicate shorter, shallower boundary contacts and more reliable recovery to the safe set. In teleoperation, assistance *without* an IK-level safety filter (**SA-N**, **SA-P**) tended to degrade outcomes in clutter (e.g., lower success, higher collisions), whereas our method **SA-B** recovered safety and maintained user acceptance, producing the most balanced overall profile when considering both subjective and objective criteria. We refrain from pairwise significance claims on collision counts at our sample size, and interpret these patterns as consistent trends that align with our predefined hypothesis that projecting blended commands from CBFs into the safety set preserves task performance while increasing constraint satisfaction and maintaining user satisfaction.

**Limitations:** Treating safety as a hard constraint can induce detours; in our study, this appears as a higher average

joint jerk for **SA-B**. Although violation time and clearance are reduced, this safety–smoothness trade-off may be undesirable for contact-sensitive tasks. Our CBF implementation is discrete and, in dynamic scenes, does not incorporate obstacle-velocity estimates; the resulting safe set can be conservative when obstacles move. Design choices for the barrier (e.g., class- $\mathcal{K}$  functions and shaping) materially influence intervention behavior and were not tuned per-user or per-scene. Finally, while CBFs are typically posed in velocity/torque space, our teleoperation stack runs in position space; we therefore approximate a task-space velocity via a temporal-difference between the current end-effector pose and the shared-autonomy target. This myopic, step-wise approximation enables IK layer CBF enforcement at real-time teleoperation rates but may miss longer-horizon structure.

**Future work:** We plan to deploy on physical hardware and directly measure contact dwell to quantify recovery, incorporate velocity observers to update dynamic CBF constraints online, and study the impact of class- $\mathcal{K}$  choices and adaptive shaping on user acceptance and intervention rates with a more extensive user study. Overall, the results show that an optimization-based IK solver can be equipped with CBF constraints to provide *myopic safety awareness* at teleoperation rates, yielding intent-preserving, constraint-aware assistance in cluttered environments.

## REFERENCES

- [1] M. Selvaggio, M. Cagnetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano, “Autonomy in physical human-robot interaction: A brief survey,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7989–7996, 2021.
- [2] 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.
- [3] K. Muelling, A. Venkatraman, J.-S. Valois, J. E. Downey, J. Weiss, S. Javdani, M. Hebert, A. B. Schwartz, J. L. Collinger, and J. A. Bagnell, “Autonomy infused teleoperation with application to brain computer interface controlled manipulation,” *Autonomous Robots*, vol. 41, pp. 1401–1422, 2017.
- [4] H. J. Jeon, D. P. Losey, and D. Sadigh, “Shared autonomy with learned latent actions,” *arXiv preprint arXiv:2005.03210*, 2020.
- [5] D. Gopinath, S. Jain, and B. D. Argall, “Human-in-the-loop optimization of shared autonomy in assistive robotics,” *IEEE robotics and automation letters*, vol. 2, no. 1, pp. 247–254, 2016.

- [6] P. Song, P. Li, E. Aertbeliën, and R. Detry, “Robot trajectron: Trajectory prediction-based shared control for robot manipulation,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5585–5591.
- [7] U. Atan, V. R. Bharadwaj, and C. Jiang, “A probabilistic inference approach for skill-based shared autonomy in assistive robotic manipulation,” *IEEE Robotics and Automation Letters*, vol. 10, no. 9, pp. 9518–9525, 2025.
- [8] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, “Shared autonomy via hindsight optimization,” *Robotics science and systems: online proceedings*, vol. 2015, pp. 10–15 607, 2015.
- [9] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, “Shared autonomy via hindsight optimization for teleoperation and teaming,” *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 717–742, 2018.
- [10] S. Nikolaidis, Y. X. Zhu, D. Hsu, and S. Srinivasa, “Human-robot mutual adaptation in shared autonomy,” in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 2017, pp. 294–302.
- [11] M. Khoramshahi and A. Billard, “A dynamical system approach to task-adaptation in physical human–robot interaction,” *Autonomous Robots*, vol. 43, pp. 927–946, 2019.
- [12] M. Allenspach, M. Pantic, R. Girod, L. Ott, and R. Siegwart, “Task adaptation in industrial human-robot interaction: Leveraging riemannian motion policies,” *arXiv preprint arXiv:2406.17333*, 2024.
- [13] A. Gottardi, S. Tortora, E. Tosello, and E. Menegatti, “Shared control in robot teleoperation with improved potential fields,” *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 3, pp. 410–422, 2022.
- [14] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.
- [15] —, “A unified approach for motion and force control of robot manipulators: The operational space formulation,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 2003.
- [16] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, “Operational space control: A theoretical and empirical comparison,” *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.
- [17] N. D. Ratliff, J. Issac, D. Kappler, S. Birchfield, and D. Fox, “Riemannian motion policies,” *arXiv preprint arXiv:1801.02854*, 2018.
- [18] C.-A. Cheng, M. Mukadam, J. Issac, S. Birchfield, D. Fox, B. Boots, and N. Ratliff, “Rmp flow: A computational graph for automatic motion policy generation,” in *Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics 13*. Springer, 2020, pp. 441–457.
- [19] D. Rakita, H. Shi, B. Mutlu, and M. Gleicher, “Collisionnik: A per-instant pose optimization method for generating robot motions with environment collision avoidance,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9995–10001.
- [20] Y. Wang, P. Praveena, D. Rakita, and M. Gleicher, “Rangedik: An optimization-based robot motion generation method for ranged-goal tasks,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9700–9706.
- [21] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in *53rd IEEE conference on decision and control*. IEEE, 2014, pp. 6271–6278.
- [22] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [23] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *2019 18th European control conference (ECC)*. Ieee, 2019, pp. 3420–3431.
- [24] B. Xu and K. Sreenath, “Safe teleoperation of dynamic uavs through control barrier functions,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7848–7855.
- [25] D. Zhang, R. Tron, and R. P. Khurshid, “Haptic feedback improves human-robot agreement and user satisfaction in shared-autonomy teleoperation,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3306–3312.
- [26] D. Zhang, G. Yang, and R. P. Khurshid, “Haptic teleoperation of uavs through control barrier functions,” *IEEE Transactions on Haptics*, vol. 13, no. 1, pp. 109–115, Jan 2020.
- [27] W. Qin, H. Yi, Z. Fan, and J. Zhao, “Haptic shared control framework with interaction force constraint based on control barrier function for teleoperation,” *Sensors*, vol. 25, no. 2, 2025.
- [28] B. He, M. Ghasemi, U. Topcu, and L. Sentis, “A barrier pair method for safe human-robot shared autonomy,” in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 2854–2861.
- [29] D. Morton and M. Pavone, “Safe, task-consistent manipulation with operational space control barrier functions,” 2025.
- [30] G. Maeda, “Blending primitive policies in shared control for assisted teleoperation,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9332–9338.
- [31] S. Jain and B. Argall, “Probabilistic human intent recognition for shared autonomy in assistive robotics,” *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 9, no. 1, pp. 1–23, 2019.
- [32] M. Ewerton, O. Arenz, and J. Peters, “Assisted teleoperation in changing environments with a mixture of virtual guides,” *Advanced Robotics*, vol. 34, no. 18, pp. 1157–1170, 2020.
- [33] P. Trautman, “Assistive planning in complex, dynamic environments: A probabilistic approach,” in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 3072–3078.
- [34] Y.-S. Jiang, G. Warnell, and P. Stone, “Goal blending for responsive shared autonomy in a navigating vehicle,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, pp. 5939–5947, May 2021.
- [35] M. C. Fontaine and S. Nikolaidis, “A quality diversity approach to automatically generating human-robot interaction scenarios in shared autonomy,” *CoRR*, vol. abs/2012.04283, 2020.
- [36] S. A. Bowyer, B. L. Davies, and F. Rodriguez y Baena, “Active constraints/virtual fixtures: A survey,” *Trans. Rob.*, vol. 30, no. 1, p. 138–157, Feb. 2014.
- [37] J.-H. Ryu, D.-S. Kwon, and B. Hannaford, “Stable teleoperation with time-domain passivity control,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 365–373, 2004.
- [38] S. Manschitz, B. Güler, W. Ma, and D. Ruiken, “Sampling-based grasp and collision prediction for assisted teleoperation,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.
- [39] A. A. Maciejewski and C. A. Klein, “Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments,” *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.
- [40] J. Ashkanazy, A. Spalter, J. Hays, L. Hiatt, R. Leontie, and C. G. Henshaw, “Collision-free inverse kinematics through qp optimization (ikinqp),” 2023.
- [41] S. Marangoz, R. Menon, N. Dengler, and M. Bennewitz, “Dawnik: Decentralized collision-aware inverse kinematics solver for heterogeneous multi-arm systems,” 2023.
- [42] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” *CoRR*, vol. abs/1903.11199, 2019.
- [43] D. Rakita, B. Mutlu, and M. Gleicher, “Relaxedik: Real-time synthesis of accurate and feasible robot arm motion,” in *Robotics: Science and Systems*, vol. 14. Pittsburgh, PA, 2018, pp. 26–30.
- [44] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, “JAX: composable transformations of Python+NumPy programs,” 2018.
- [45] F. L. Ferretti, D. Ferigo, C. Sartore, A. Croci, O. G. Younis, S. Traversaro, and D. Pucci, “Hardware-accelerated morphology optimization via physically consistent differentiable simulation,” 2025.
- [46] K. Tracy, T. A. Howell, and Z. Manchester, “Diffpills: Differentiable collision detection for capsules and padded polygons,” 2022.
- [47] D. Kraft, “Algorithm 733: TOMP—fortran modules for optimal control calculations,” *ACM Transactions on Mathematical Software*, vol. 20, pp. 262–281, 1994.
- [48] J. Pan, S. Chitta, D. Manocha, F. Lamiraux, J. Mirabel, J. Carpentier, L. Montaut, *et al.*, “Coal: an extension of the flexible collision library,” <https://github.com/coal-library/coal>, 2015–2024.
- [49] S. G. Hart and L. E. Staveland, “Development of nasa-tlx (task load index): Results of empirical and theoretical research,” in *Advances in psychology*. Elsevier, 1988, vol. 52, pp. 139–183.