frontiers
in Robotics and AI

ORIGINAL RESEARCH
published: 09 September 2021
doi: 10.3389/frobt.2021.730433

# Motion Polytopes in Virtual Reality for Shared Control in Remote Manipulation Applications

*Mark Zolotas* [1]*, *Murphy Wonsick* [1], *Philip Long* [1,2] *and Taşkın Padır* [1]

[1]Northeastern University, Boston, MA, United States, [2]Irish Manufacturing Research, Dublin, Ireland

In remote applications that mandate human supervision, shared control can prove vital by establishing a harmonious balance between the high-level cognition of a user and the low-level autonomy of a robot. Though in practice, achieving this balance is a challenging endeavor that largely depends on whether the operator effectively interprets the underlying shared control. Inspired by recent works on using immersive technologies to expose the internal shared control, we develop a virtual reality system to visually guide human-in-the-loop manipulation. Our implementation of shared control teleoperation employs end effector manipulability polytopes, which are geometrical constructs that embed joint limit and environmental constraints. These constructs capture a holistic view of the constrained manipulator's motion and can thus be visually represented as feedback for users on their operable space of movement. To assess the efficacy of our proposed approach, we consider a teleoperation task where users manipulate a screwdriver attached to a robotic arm's end effector. A pilot study with prospective operators is first conducted to discern which graphical cues and virtual reality setup are most preferable. Feedback from this study informs the final design of our virtual reality system, which is subsequently evaluated in the actual screwdriver teleoperation experiment. Our experimental findings support the utility of using polytopes for shared control teleoperation, but hint at the need for longer-term studies to garner their full benefits as virtual guides.

**Keywords: human-in-the-loop teleoperation, motion polytopes, shared control, virtual fixtures, virtual reality**

OPEN ACCESS

***Edited by:***
*Yanan Li,*
*University of Sussex, United Kingdom*

***Reviewed by:***
*Mario Selvaggio,*
*University of Naples Federico II, Italy*
*Ziwei Wang,*
*Imperial College London,*
*United Kingdom*

****Correspondence:***
*Mark Zolotas*
*m.zolotas@northeastern.edu*

***Specialty section:***
*This article was submitted to*
*Robotic Control Systems,*
*a section of the journal*
*Frontiers in Robotics and AI*

***Received:*** *24 June 2021*
***Accepted:*** *23 August 2021*
***Published:*** *09 September 2021*

***Citation:***
*Zolotas M, Wonsick M, Long P and Padır T (2021) Motion Polytopes in Virtual Reality for Shared Control in Remote Manipulation Applications. Front. Robot. AI 8:730433. doi: 10.3389/frobt.2021.730433*

## 1 INTRODUCTION

Teleoperation is a well-established robot control method that plays a pivotal role in complex and unpredictable settings where human supervision is necessary. For instance, remote teleoperation is highly desirable in extreme scenarios where the co-presence of a human operator poses unwanted risk, such as tasks performed underground, underwater, or even in space. However, the direct teleoperation of robotic systems is challenging and often places significant cognitive burden on the operator (Tanwani and Calinon, 2017; Xi et al., 2019; Hetrick et al., 2020). This is especially true when handling robots with high degrees-of-freedom, like robotic arms for grasping and manipulation (Losey et al., 2020). In order to alleviate any excess workload exerted on a teleoperator, shared control is typically employed as a means of providing autonomous assistance.

The shared control paradigm is widely applied in any task where a human operator and robot collaborate towards a common goal by *simultaneously* issuing control over a system (Abbink et al., 2018). Many prior works have demonstrated that by engaging in shared control, a human user can apply their expertise and high-level cognition to the teleoperation task, as well as exploit the precision and accuracy of robot autonomy (Dragan and Srinivasa, 2013; Javdani et al., 2015; Jain and Argall, 2019). Despite these benefits, users may find the arbitration process of shared control a frustrating and bewildering experience. A predominant reason for this is the misalignment between a user's intended control policy and the actual robot behavior (Zolotas and Demiris, 2019). Even when the robot either correctly infers a human's goal or knows it *a priori*, the user may not accept the resulting behavior unless administered reassurance through feedback (Dragan and Srinivasa, 2013).

In many instances of shared control, the haptic channel is the selected modality of sensory feedback during this exchange (Abbink et al., 2018; Losey et al., 2018). Indeed, haptic shared control offers numerous benefits, ranging from operator training (O'Malley et al., 2005) to improving teleoperation performance (Selvaggio et al., 2018). Force feedback has also enjoyed success in learning from demonstration applications within shared control (Kucukyilmaz and Demiris, 2018; Zeestraten et al., 2018). Nevertheless, a potential drawback in this medium resides in the limited sensorimotor teaching derived from force feedback. For complex manipulation tasks that necessitate heightened user attention and precision, haptic interfaces by themselves may not provide a sufficiently rich channel to relay information back to users (Losey et al., 2018).

Alternatively, mixed reality headsets are an emerging technology that have recently gained traction in addressing misalignment in shared control (Zolotas and Demiris, 2019; Brooks and Szafir, 2020). These headsets are not without flaws, but in the scope of shared control where users must maintain attention and *actively* participate in the task-at-hand, they hold great promise over other technologies (Sibirtseva et al., 2018). Furthermore, they can supply feedback across multiple modalities, *e.g.,* through visualizations, sounds and haptic vibrations. While these headsets are a suitable medium for exposing the otherwise "black-box" nature of shared control, certain design considerations are crucial to avoid hindering task performance (Zolotas et al., 2018). In particular, the user interface should clearly delineate the range of constraints associated with the remote robot manipulator, including joint limits, proximity to obstacles and singularities. A naive presentation of this information would lead to the teleoperator considering dozens of independent variables at once, risking information overload (Mortimer et al., 2017).

Taking these design considerations into account, we propose a shared control methodology involving on-the-fly generation of *virtual fixtures* for remote teleoperation in virtual reality (VR). Virtual fixtures (Rosenberg, 1992; Bowyer et al., 2014) aim to reduce the cognitive burden on the operator by creating enclosed volumetric zones, inside of which the robot can operate. These zones can be created in numerous ways, *e.g.,* directly from 3D point cloud data (Yamamoto et al., 2012), using shape primitives (Bettini et al., 2004), or manually (Quintero et al., 2017). Although virtual fixtures are typically concerned with the surrounding environment, an optimal system should also incorporate the robot's intrinsic constraints, such as joint limits and performance capacities. In Long et al. (2019), constraints originating from the environment and robot are embedded into geometrical objects, namely polytopes, to generate a series of virtual guides for guarded teleoperation.

Inspired by the prospects of using polytope-generated virtual guides for *both* assistance and feedback, we develop a polytope-based method of shared control. To establish a tightly coupled relationship between polytopes and shared control, we introduce a novel "shrinking" modification over traditional motion polytopes that arbitrates operator inputs in accordance with their estimated intentions. Unlike most virtual fixture methods of shared control relying on haptic guidance to enhance telepresence (O'Malley et al., 2005; Selvaggio et al., 2018; Zeestraten et al., 2018), ours instead adopts a visually immersive experience through VR. Moreover, most teleoperation works in VR present a robot embodied view of the remote environment (Xi et al., 2019; Hetrick et al., 2020; Wonsick and Padir, 2020). However, recent works have also shown the utility of model-based perspectives (Van de Merwe et al., 2019; Wonsick et al., 2021a). Ergo, we additionally explore the potential benefits of a model-based perspective.

We investigate the efficacy of our approach by conducting a human-robot interaction experiment where subjects teleoperate a robotic arm in VR to screw in a set of bolts (shown in **Figure 1**). The task is motivated by the potential advantages of deploying robots to utilize tools under extreme conditions, *e.g.,* nuclear decommissioning (Wonsick et al., 2021b). Prior to this experiment, a pilot user study was also held to designate the most suitable VR interface for the task. Therefore, the main contributions of this paper are: 1) a shared control teleoperation method that relies on virtual fixtures obtained from joint space polytopes to inform the arbitration; 2) a user-informed VR interface for teleoperation with the internal properties of polytopes visualized to graphically aid operators; and 3) a real-world experiment that evaluates how effective the proposed system is when remotely controlling a robotic arm to wield a screwdriver.

The remainder of the paper is organized as follows. **Section 2** describes our shared control system, fusing a simple intention estimation algorithm with polytope-guided teleoperation assistance. Our VR system and an array of visualizations are then detailed in **Section 3**, alongside a pilot user study that helped determine the final interface. In **Section 4**, we present the remote-operated screwdriver experiment and its results. **Sections 5** and **6** reflect on the key insights drawn from our teleoperation experiment and discuss future avenues for research. Closing remarks are then provided in **Section 7**.

## 2 SHARED CONTROL METHODOLOGY

In this section, we introduce a shared control methodology tailored to the target domain of teleoperation. Our method falls under the popular scheme of "predict-then-blend" (Javdani et al., 2015), which consists of two core processes: intention estimation and arbitration. A typical interaction cycle of these processes will have the robot first recognize a

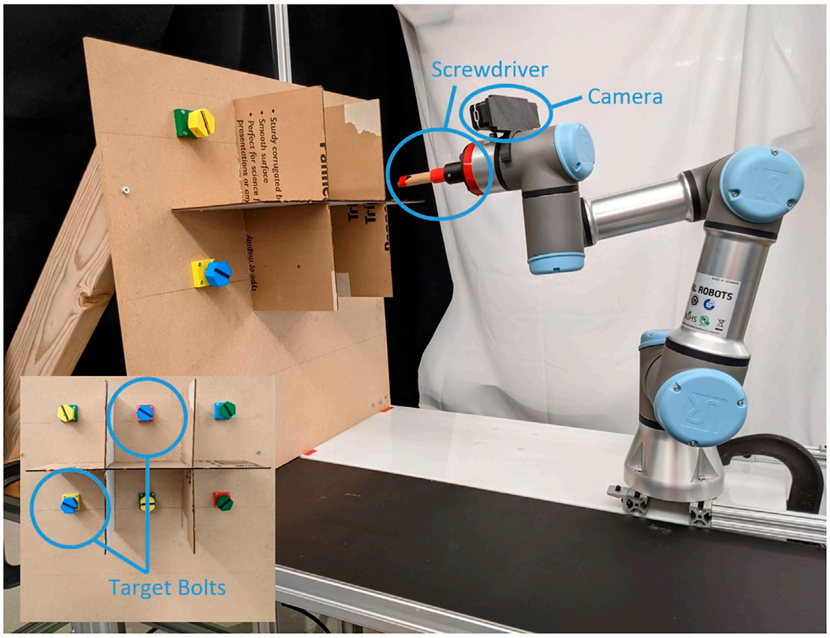

**FIGURE 1 |** Illustration of the UR3e robotic arm in its initial configuration for the screwdriver experiment. In the bottom left is a front-facing view of the blue target bolts that users were requested to tighten in the teleoperation experiment. The robot's end effector has been extended to include a wooden screwdriver, as well as a camera for the precision necessary to complete this task.

user's intentions, *e.g.*, their intended joint configuration, from a set of task-specific goals. The robot will then compute commands that best align with this estimation of intent, and subsequently arbitrate between these autonomous commands and any user inputs to finalize on an optimal outcome for the task objective.

Before proceeding with the implementation of these core mechanisms, we briefly outline in **Section 2.1** the chosen manipulator model. **Sections 2.2** and **2.3** then describe the intention estimation and arbitration steps, respectively, with the latter employing joint space polytopes for assistive teleoperation. Of the virtual fixture methods previously applied to arbitration (O'Malley et al., 2005; Selvaggio et al., 2018; Zeestraten et al., 2018), we believe this manuscript is the first to depict a use-case for motion polytopes.

## 2.1 Manipulator Model

We consider a manipulator with $n$ degrees-of-freedom operating in 6-dimensional space. The end effector pose is represented by a vector $\mathbf{x}_n \in \mathbb{R}^6$, consisting of the position $\mathbf{x}_n^p \in \mathbb{R}^3$ and unit quaternion orientation $\mathbf{x}_n^q \in \mathbb{R}^3$, all of which can be derived from the forward kinematic chain $fk$:

$$\mathbf{x}_n = fk(\mathbf{q}), \tag{1}$$

where $\mathbf{q} = [q_1, \ldots, q_n]$ are the joint configuration variables. The twist at the end effector frame $\boldsymbol{\nu}_n$ can then be obtained from the differential kinematic model:

$$\boldsymbol{\nu}_n = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}_n \dot{\mathbf{q}}, \tag{2}$$

with $\mathbf{v}$ and $\boldsymbol{\omega}$ denoting translational and angular velocities, respectively. $\mathbf{J}_n$ is the 6 × $n$ Jacobian matrix defined at the end effector frame $n$, and $\dot{\mathbf{q}} = [\dot{q}_1, \ldots, \dot{q}_n]^T$ is the joint velocity vector.

## 2.2 Human Intention Estimation

Estimating a human's intention is a multidisciplinary subject that plays an integral part in shared control, as it can guide the robot's decision-making on how to best assist (Demiris, 2007; Jain and Argall, 2019). Throughout this paper, we adopt a loose terminology for *intention information* (Losey et al., 2018), where goals, targets and intentions will be used interchangeably. Furthermore, we assume a discrete goal space $\mathcal{G}$ exists for the specific task and that it is known to both the human and robot at run time. For our target domain of assisted teleoperation in screwing bolts, the goal space spans all bolt locations with each possessing an ideal joint state based on pre-recorded expert trajectories. In other words, we store a dictionary of mappings from goal labels to idealistic joint configurations. Hence, the overall aim of our intention estimation process is to infer a user's goal of interest $\hat{g} \in \mathcal{G}$ and yield its corresponding joint configuration $\hat{\mathbf{q}}$.

Following an extensive body of literature that uses Hidden Markov Models (HMMs) for human intention estimation (Tanwani and Calinon, 2017; Jain and Argall, 2019; Brooks and Szafir, 2020), we also develop an HMM to infer target bolts from manipulator data. First, we consider a sequence of observations $\mathbf{o}_{1:t} = (\mathbf{o}_1, \ldots, \mathbf{o}_t)$ as the Euclidean distances from the current end effector's position $\mathbf{x}_t^p$ to all other goals:

$$\mathbf{o}_t = \| \mathbf{x}_t^p - \mathbf{g}_i^p \|, \forall g_i \in \mathcal{G}, \tag{3}$$

where the frame $n$ subscript is removed for notational simplicity and $\mathbf{g}_i^p$ are the three position variables of the end effector at the $i^{\text{th}}$ bolt. Proximity to goal is a traditional measure of intent in shared control (Dragan and Srinivasa, 2013; Javdani et al., 2015; Jain and Argall, 2019) and serves as the single observation source for our simple task domain.

In the same vein as prior work, we then let the hidden state of the HMM represent the user's current goal $g_t$ and treat intention estimation as a Bayesian filtering problem where the aim is to derive the posterior probability $P(g_t | \mathbf{o}_{1:t})$ (Jain and Argall, 2019). Many shared control frameworks apply such Bayesian reasoning to maintain a *belief* across all task goals, as the robot can then reason over the entire goal space $\mathcal{G}$ to select actions that sustain assistance even under uncertainty (Javdani et al., 2015). Confidence levels surrounding this predictive distribution can also be exploited to dictate how robot-user control is arbitrated.

The posterior probability for a particular goal at time $t$, also known as the belief state $b_t(g_t)$, is computed using Bayes rule:

$$\begin{aligned} b_t(g_t) &= P(g_t \mid \mathbf{o}_{1:\,t}) \\ &= \frac{P(\mathbf{o}_t \mid g_t) P(g_t \mid \mathbf{o}_{1:t-1})}{P(\mathbf{o}_t \mid \mathbf{o}_{1:t-1})} \propto P(\mathbf{o}_t \mid g_t) P(g_t \mid \mathbf{o}_{1:\,t-1}), \end{aligned} \tag{4}$$

which is simplified in HMMs by the assumption that observations $\mathbf{o}_{1:t}$ exhibit the Markov property of conditional independence across timesteps. Partitioning $P(g_t | \mathbf{o}_{1:t-1})$ leads to a recursive update of the belief state:

$$b_t(g_t) = P(g_t \mid \mathbf{o}_{1:t}) \propto P(\mathbf{o}_t \mid g_t) \sum_{g_{t-1} \in \mathcal{G}} P(g_{t-1} \mid \mathbf{o}_{1:t-1}) P(g_t \mid g_{t-1}) \tag{5}$$

$$\propto P(\mathbf{o}_t \mid g_t) \sum_{g_{t-1} \in \mathcal{G}} b_{t-1}(g_{t-1}) P(g_t \mid g_{t-1}). \tag{6}$$

where $P(\mathbf{o}_t \mid g_t)$ and $P(g_t \mid g_{t-1})$ are the emission and transition probabilities, respectively. As a result, the filtering process boils down to a repeated assignment of posterior probabilities for each $g \in \mathcal{G}$ given incoming observations. To complete the HMM description, we configure the starting probability $P(g_0)$ to be uniform.

Lastly, the inferred goal $\hat{g}_t$ is decoded using *maximum a posteriori* estimation as in Jain and Argall (2019):

$$\hat{g}_t = \arg\max_{g_t \in \mathcal{G}} P(g_t \mid \mathbf{o}_t). \tag{7}$$

A lookup of $\hat{g}_t$ in the aforementioned dictionary of goal states to joint configurations will produce the associated $\hat{\mathbf{q}}_t$.

## 2.3 Polytopes for Arbitration

For arbitration in the shared control, we propose to utilize polytopes as a means of generating virtual fixtures during teleoperation. While virtual fixtures often relate to the surrounding environment, optimum performance requires the robot's intrinsic constraints, such as joint limits, to also be considered. In robotics, the most commonly used performance measure is the manipulability ellipsoid first presented in Yoshikawa (1984). Much work has focused on embedding supplementary information within this ellipsoid such that it considers the effects of joint position constraints (Tsai, 1986), joints velocity limits (Lee, 1997), or obstacles in the environment (Vahrenkamp et al., 2012; Vahrenkamp and Asfour, 2015). In contrast to the manipulability ellipsoid, polytopes give an exact representation of velocity limits (Kokkinis and Paden, 1989) and are less susceptible to error (Krut and Pierrot, 2004). Indeed, as polytopes are geometric objects, they can be combined through union or intersection to represent the capacities of composite serial or parallel chains (Long and Padir, 2020). Finally, since polytopes are represented as a system of linear inequalities, supplementary constraints can be easily added, for example friction cones (Caron et al., 2017), workspace danger zones (Long and Padır, 2018) or a zero-moment point (Rasheed et al., 2018).

We build on the work first proposed in Long et al. (2019), where a *constrained motion polytope* is constructed and used for teleoperation by generating a series of virtual guides. These guides consider the distance-to-collision with environmental obstacles, joint position limits and joint velocity limits. The polytopes present the operator with a Cartesian representation of both the workspace and configuration space constraints during teleoperation in a cluttered environment.

A polytope, $\mathcal{P}$ can be represented in two ways: as the convex hull of its vertex set, known as the $\mathcal{V}$-representation and denoted as $\mathcal{P}^V$, or as volume bounded by a finite number of half-spaces, known as the $\mathcal{H}$-representation and denoted as $\mathcal{P}^H$, written respectively as:

$$\mathcal{P}^V = \left\{ \mathbf{x} : \mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{y}_i \middle| \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i = 1 \right\}, \quad \mathcal{P}^H = \mathbf{A}\mathbf{x} \leq \mathbf{b}, \tag{8}$$

where $\mathbf{y}_i$ denotes the $i$th element of the vertex set and $\mathbf{x}$ is any point inside $\mathcal{P}$, $\mathbf{A}$ contains the half-spaces' normals, and $\mathbf{b}$ is the shifted distance from the origin along the normal. Converting between the $\mathcal{V}$ and $\mathcal{H}$ representations can be carried out in several ways, for example using the double description method (Fukuda and Prodon, 1996). The manipulability of a serial manipulator can be obtained by first constructing the joint space polytope in H-representation:

$$\mathcal{Q}^H = \begin{bmatrix} \mathbb{I}_n \\ -\mathbb{I}_n \end{bmatrix} \dot{\mathbf{q}} \leq \begin{bmatrix} \dot{\mathbf{q}}_{max} \\ -\dot{\mathbf{q}}_{min} \end{bmatrix}, \tag{9}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix and $\dot{\mathbf{q}}_{max}$ and $\dot{\mathbf{q}}_{min}$ are robot's maximum and minimum joint velocities. Using the double description method, an equivalent polytope defined by its vertices is written as:

$$\mathcal{Q}^V = \{ \dot{\mathbf{q}}_1^v, \dot{\mathbf{q}}_2^v, \ldots, \dot{\mathbf{q}}_{2^n}^v \}, \tag{10}$$

where $\dot{\mathbf{q}}_i^v$ denotes the $i$th vertex of $\mathcal{Q}$. A manipulability polytope, denoted as $\mathcal{P}$, representing the Cartesian-space velocities can then be obtained by transforming the vertices of (**Eq. 10**) to Cartesian space using (**Eq. 2**). $\mathcal{P}$'s vertex set representation is given as:

$$\mathcal{P}^V = \{ \boldsymbol{v}_1^v \quad \ldots \boldsymbol{v}_{2^n}^v \} = \{ \mathbf{J}_n \dot{\mathbf{q}}_1^v \ldots \mathbf{J}_n \dot{\mathbf{q}}_{2^n}^v \}. \tag{11}$$

The convexity of a polytope is preserved under affine transformation, thus a bounded volume of $\mathcal{P}$ can be easily obtained which represents the system's manipulability and serves as an exact indicator of robot performance.

To obtain the *constrained motion polytope* for a manipulator, a joint space polytope is first constructed using position deviations instead of instantaneous velocities.

In order to reformulate (**Eq. 11**) for allowable motions, the robot's motions are discretized into timesteps. At timestep $k$, let the robot's pose be denoted as $\mathbf{x}_k$ and the instantaneous joint velocity as $\dot{\mathbf{q}}_k$. Over a period of $\delta t$ seconds, assume the end effector travels a distance of $\delta \mathbf{x}_k$, then the resulting end-effector pose at instant $k + 1$ can be obtained by linearizing (**Eq. 1**) using (**Eq. 2**), namely:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta \mathbf{x}_k \approx fk(\mathbf{q}_k + \delta \mathbf{q}_k), \tag{12}$$

$$\delta \mathbf{x}_k = \mathbf{J}_n \delta \mathbf{q}_k \tag{13}$$

where $\delta \mathbf{q}_k$ denotes the displacement of the joint variables over the timestep and is defined as:

$$\delta \mathbf{q}_k = \dot{\mathbf{q}}_k \ \delta t. \tag{14}$$

For a point $\mathcal{I}$ on the robot's kinematic chain whose position with respect to the world frame is denoted by the vector $\mathbf{r}_i$, the displacement due to $\delta \mathbf{q}_k$ is written as:

$$\delta \mathbf{x}_k^i = \mathbf{J}_i \delta \mathbf{q}_k, \tag{15}$$

where $\mathbf{J}_i \in \mathbb{R}^{3\times n}$ denotes the kinematic Jacobian matrix that relates the velocities of the preceding joints in kinematic chain to the translational velocity at point $\mathcal{I}$. Using the above linearization, limits of translational motion for any point on the manipulator body can be defined based on the location of environmental obstacles. The translational motion of point $\mathcal{I}$ *towards* an environmental obstacle $\mathcal{O}$, denoted as $\delta \mathbf{x}_{o,k}^i$, is defined as:

$$\delta \mathbf{x}_{o,k}^i = \hat{\mathbf{r}}_{io}^T \ \mathbf{J}_i \ \delta \mathbf{q}_k, \tag{16}$$

where $\mathbf{r}_{io} = \mathbf{r}_i - \mathbf{r}_o$ is the relative position between $\mathcal{O}$ and $\mathcal{I}$, $\|\mathbf{r}_{io}\|$ is the norm of this vector, *i.e.,* the distance, while $\hat{\mathbf{r}}_{io}$ denotes the corresponding normalized unit vector. Hence, to prevent a potential collision between $\mathcal{I}$ and $\mathcal{O}$, an allowable motion constraint defining the maximum displacement of $\mathcal{I}$ can be expressed as follows:

$$\hat{\mathbf{r}}_{io}^T \ \mathbf{J}_i \ \delta \mathbf{q}_k \le \|\mathbf{r}_{io}\|, \tag{17}$$

This can be repeated for a set of $l$ points discretized along the kinematic chain, leading to the following set of linear inequalities:

$$\begin{bmatrix} \hat{\mathbf{r}}_{1o}^T \ \mathbf{J}_1 \\ \vdots \\ \hat{\mathbf{r}}_{lo}^T \ \mathbf{J}_l \end{bmatrix} \delta \mathbf{q}_k \le \begin{bmatrix} \|\mathbf{r}_{1o}\| \\ \vdots \\ \|\mathbf{r}_{lo}\| \end{bmatrix} \tag{18}$$

In our implementation, we select at every instant the point along each link nearest to an environmental obstacle, *i.e.,* $l = n$. Hence (**Eq. 18**) represents the set of instantaneous collision-free joint deviations for each link on the robot's kinematic chain, which is then repeated for all obstacles in the environment.

Aside from obstacles, a robot's motion is restricted by positional limits of the joints. To integrate these into the polytope, the following linear inequalities are included:

$$\begin{bmatrix} \mathbb{I}_n \\ -\mathbb{I}_n \end{bmatrix} \delta \mathbf{q}_k \le \begin{bmatrix} \mathbf{q}_{max} - \mathbf{q}_k \\ \mathbf{q}_k - \mathbf{q}_{min} \end{bmatrix}, \tag{19}$$

where $\mathbf{q}_{max}$ and $\mathbf{q}_{min}$ are vectors of upper and lower positional joint limits, respectively.

Finally, it should be noted that the linearization error increases as the joint displacement increases, thus a maximum limit is imposed to ensure a satisfactory approximation of link motion:

$$\begin{bmatrix} \mathbb{I}_n \\ -\mathbb{I}_n \end{bmatrix} \delta \mathbf{q}_k \le \begin{bmatrix} \delta \mathbf{q}_{max} \\ \delta \mathbf{q}_{max} \end{bmatrix}, \tag{20}$$

where $\delta \mathbf{q}_{max}$ is the vector of linearization limits for each joint of the kinematic chain. Increasing the values of $\delta \mathbf{q}_{max}$ expands the free space polytope at a cost of reduced fidelity. While not necessary, for simplicity, we let $\delta q_1 = \delta q_2 = \ldots = \delta q_n = \epsilon_{lin}$. The linearization limits can, for various reasons, be altered at run time. For instance, limits could be increased to enlarge the solution space in **(Eq. 22)**. Alternatively, the limits could be shrunk progressively in order to guide the user towards a defined configuration as shown in **(Eq. 24)**.

By stacking (**Eqs. 18–20**), an $\mathcal{H}$-representation of a joint space polytope that approximates, at any instant, the maximum range of joint displacements with respect to the system's constraints is obtained in the form:

$$\mathbf{A}_k \delta \mathbf{q}_k \le \mathbf{b}_k. \tag{21}$$

The polytope can then be transformed to a $\mathcal{V}$-representation using the double description method (Fukuda and Prodon, 1996), after which the Cartesian representation is obtained using the differential kinematic model (**Eq. 2**). For a full derivation of the above procedure, the reader is invited to refer to Long et al. (2019).

The derived *constrained motion polytope* bounds a volume within which the robot can move without violating any constraints. In the shared control application, it is also desirable to guide the user towards a goal configuration, hence we propose to use the polytope as a supplementary guide by modifying the extents based on proximity to a goal. In order to do so, the linearization limit, $\epsilon_{lin}$, is modified as a function of distance from the current end effector pose to the goal pose, as given in **(Eq. 3)**. Consequently, the free space within which the robot can move is reduced. An example of this effect is shown in **Figure 2**.

During the teleoperation, the user's input generates a desired twist $\boldsymbol{v}_d$ for the robot end effector, which is then converted into a joint velocity command $\delta \mathbf{q}_i$ and sent to the robot controller by solving the following optimization problem using SNOPT (Gill et al., 2005):

$$\begin{aligned} &\underset{\delta \mathbf{q_i}}{\text{minimize}} \ \|\boldsymbol{v}_d - \mathbf{J}_i \delta \mathbf{q}_i\| \\ &\text{subject to } \mathbf{A}_i \delta \mathbf{q}_i \le \mathbf{b}_i. \end{aligned} \tag{22}$$

$\mathbf{q}_i$, $\mathbf{J}_i$, $\mathbf{A}_i$ and $\mathbf{b}_i$ denote the robot's current joint configuration, kinematic Jacobian matrix and constrained motion polytope hyperplanes.

**Equation 22** minimizes the error between the user input and the robot's motion, however, the linear inequalities may reject feasible

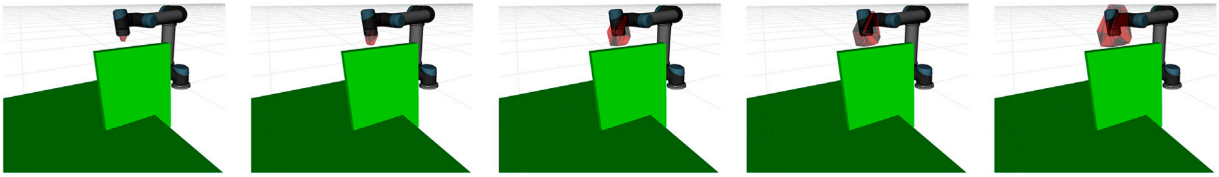

**FIGURE 2 |** Visualization of the shrinking constrained motion polytope, with origin at robot tool frame. From left to right, the linearization limit is selected as $\epsilon_{lin} = 0.038$, $\epsilon_{lin} = 0.060$, $\epsilon_{lin} = 0.1$, $\epsilon_{lin} = 0.117$, $\epsilon_{lin} = 0.138$. The decreasing linearization value shrinks the volume of free space within which the operator can move the end effector, thus guiding the manipulator towards a pre-defined goal configuration.

solutions that lie outside the linearization limits. To overcome this, a set of joint configurations are randomly sampled in the neighborhood of $\mathbf{q}_i$. Let $\mathbf{q}_j$ denote one of the random joint configurations, (**Eq. 22**) is then reformulated as follows:

$$\begin{aligned} &\underset{\delta\mathbf{q}_j}{\text{minimize}} \left\| \boldsymbol{v}_d - \mathbf{J}_i\left(\delta\mathbf{q}_j + \left(\mathbf{q}_j - \mathbf{q}_i\right)\right) \right\| \\ &\text{subject to } \mathbf{A}_j \delta\mathbf{q}_j \leq \mathbf{b}_j. \end{aligned} \tag{23}$$

where the controller input is re-defined as $(\delta\mathbf{q}_j + (\mathbf{q}_j - \mathbf{q}_i))$. This is repeated for each sampled joint state, such that the controller input that best realizes the desired user twist is obtained. The sampling method enables the user to move the robot freely between polytopes while respecting intrinsic and extrinsic constraints. Additionally, the samples enlarge the solution space for each user input albeit at the cost of computation time. Therefore, the constrained motion polytopes can be seen as an arbitration of user inputs to ensure that the resulting robot motions are collision free and respect joint position limits.

During task execution, if the intention estimation is sufficiently confident in the inferred goal from **(Eq. 7)**, *i.e.,* the posterior probability is greater than some pre-defined confidence threshold, then the system begins to guide the user towards the goal configuration by progressively limiting user motion:

$$\epsilon_{lin} = \begin{cases} \epsilon_{max}, & \text{if } \epsilon_{lin} > \epsilon_{max}. \\ \epsilon_{min}, & \text{if } \epsilon_{lin} < \epsilon_{min}. \\ w_1 \cdot min\left(\mathbf{o}_t\right), & \text{otherwise.} \end{cases} \tag{24}$$

where $\epsilon_{\max}$ and $\epsilon_{\min}$ are respectively the maximum linearization value beyond which the error associated with motion linearization is unacceptable and the minimum value beyond which the end effector is immovable, while $w_1$ is a weight that scales the shrinking of the polytope.

Thus, the hyperplane constraints defined in **(Eq. 21)** gradually become more restrictive as the user intention becomes more apparent. This in turn means that the solutions obtained from **(Eq. 23)** are more heavily arbitrated for the precise motion required to complete the screwdriver task.

# 3 VIRTUAL REALITY SYSTEM

The last component of our framework integral to remote teleoperation is the interface, for which we utilize the medium of VR. In remote robot teleoperation, VR has become more widely used in recent years and has been shown to exhibit higher preference and usability over traditional interfaces (Wonsick and Padir, 2020). VR also provides an opportunity to create more immersive and intuitive interfaces due to its capability of interacting and visualizing in 3D. To develop our proposed VR interface, we first introduce a system architecture in **Section 3.1** that can teleoperate the robot manipulator model of **Section 2.1** in the target screwdriver setting. Next, we present in **Section 3.2** a variety of VR prototypes for teleoperation, as well as a pilot user study to gauge their efficacy. Finally, **Section 3.3** outlines the VR interface derived from this initial study.

## 3.1 Architecture

The overall architecture for the VR system is demonstrated in **Figure 3A**. In terms of hardware, the VR system is composed of an HTC Vive headset and a single controller. As for software, the VR application is developed using the Unity engine on the Windows operating system, while the robot is ROS (Quigley et al., 2009) compatible and runs on the Linux operating system. Therefore, to facilitate communication between VR and the robot, we employ an open-source software library called ROS#[1]. Rather than using position control, where waypoints are provided to navigate the arm, or trajectory control, where a generated path dictates the robot's motion (Hetrick et al., 2020), we use a velocity controller to teleoperate the robot.

To teleoperate the arm through VR, we translate relative velocities originating from the Vive controller into command velocities for the robot's end effector. Operators are able to activate velocity control by holding down a button assigned for teleoperation, as depicted in **Figure 3B**, and then move the controller in full 6 degrees-of-freedom. Teleoperation is automatically stopped whenever the button is released. Additionally, to assist in the screwing task, operators are provided with a means of rotating only the screwdriver, similar to a power drill. **Figure 3B** shows the buttons designated to rotate the screwdriver in either a clockwise or counterclockwise direction at a fixed velocity. Similar to the velocity control, rotations are only activated while the button is being held down and stopped upon release.

[1] https://github.com/siemens/ros-sharp

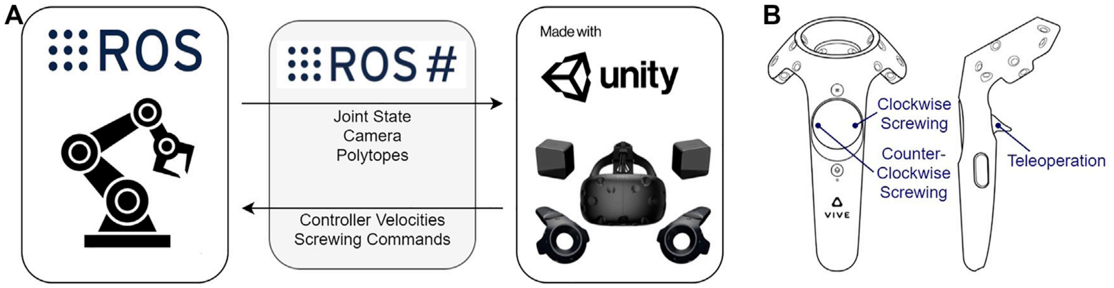

**FIGURE 3 |** Diagram of the HTC Vive virtual reality system and handheld controller interface. **(A)** The entire virtual reality architecture is built along a ROS# communication bridge that transfers data between the robot (*e.g.,* joint state information) and Unity application (*e.g.,* command velocities). **(B)** Handheld controller with button functions to engage in teleoperation and screwing.

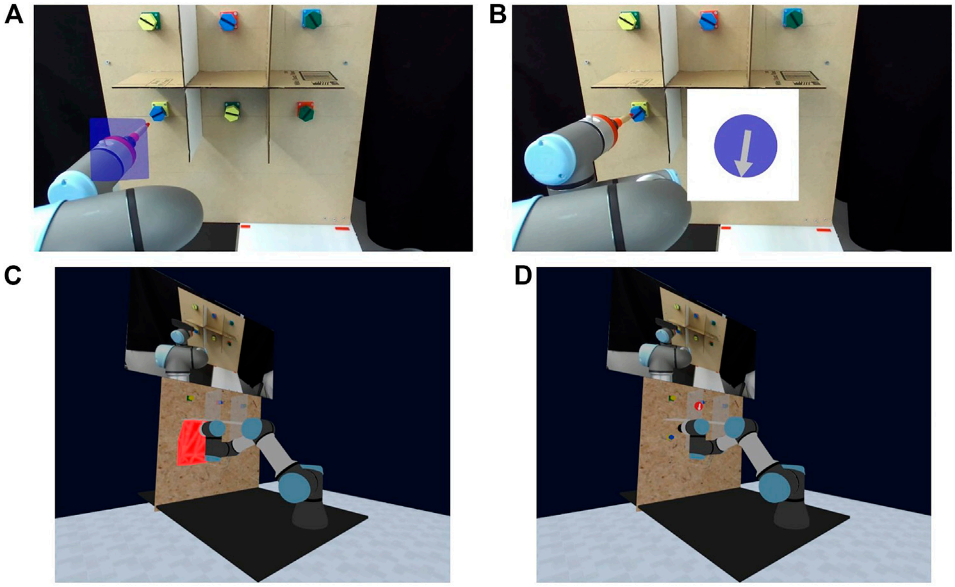

**FIGURE 4 |** Four prototypes of visualizations for the virtual reality interface. **(A)** Camera + Polytope **(B)** Camera + Compass **(C)** Model + Polytope **(D)** Model + Compass.

## 3.2 User-Centered Design

Recent work involving shared control and mixed reality headsets, like the HTC Vive or Microsoft HoloLens, has stipulated that an interface must follow careful design considerations to avoid hindering task performance (Zolotas et al., 2018). Whilst prior studies have illustrated the flaws or benefits of different graphical cues when signaling robot intent during human-robot interaction (Walker et al., 2018; Zolotas and Demiris, 2019), each individual application will warrant a novel perspective. Consequently, we advocate following a user-centered design where human subjects helped resolve the final VR interface for our shared control system, which is presented in **Section 3.3**. This section describes four VR interface prototypes developed for the target setting of teleoperating a screwdriver with a robotic arm (portrayed in **Figure 4**), as well as the results from a pilot user study to evaluate these prototypes. These prototypes differ in two aspects: the environmental view and the choice of visual guide on the internal shared control.

Regarding the environment perspective, participants were shown interfaces that displayed either a pure camera view, **Figures 4A,B**, or a constructed model view of the relative elements of the robot's environment, coupled with a smaller version of the former camera view, **Figures 4C,D**. For all four interface prototypes, a stereo camera was used to capture a 2.5D view of the environment. This camera was mounted behind the robot to provide a complete visual of the scene and robot itself.

As for visual guides, participants were shown either the direct polytope generated from **Section 2.3**, **Figures 4A,C**, or a circular disk with an arrow, *i.e.,* a *compass*, which incorporates information on the internal shared control in a reduced form, **Figures 4B,D**. In the compass visualization, there are three varying elements based on the internal shared control: the size

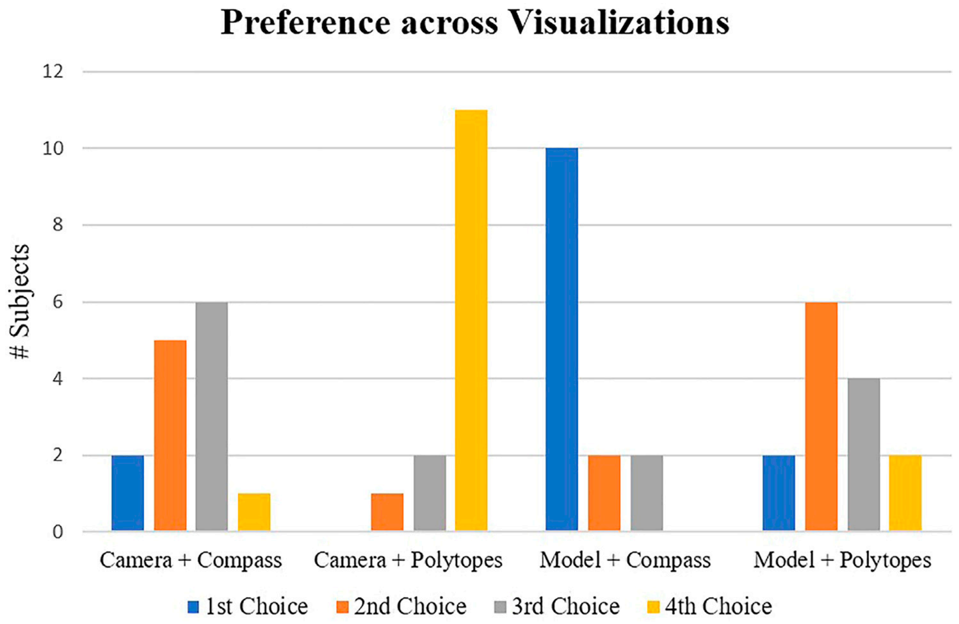

**FIGURE 5 |** User preference ratings across the four visualization prototypes from the pilot study. There is a clear majority of "1st Choice" and "4th Choice" selections for the "Model + Compass" and "Camera + Polytopes" visualizations, respectively.

of the circular disk, the direction of the arrow, and the color coding. The disk behaves and varies like the linearization limit from **(Eq. 24)** by only appearing once a goal estimate exceeds the aforementioned confidence threshold and scaling in size according to the end effector's distance from the inferred goal. Meanwhile, the compass arrow is updated to always point to the center of the operator's desired goal location based on the end effector's current pose. Furthermore, both the polytope and compass visualizations are color-coded to indicate safety from obstacles. Using the ratio of reduction in space between the constrained and allowable motion polytope volumes, we color the compass red for constrained space and blue for wide allowable space. In other words, a smaller ratio highlights close proximity to obstacles with red to warn for danger, while blue signifies a larger ratio and a safer space for operation.

To assess the utility of these interfaces with respect to the screwdriver teleoperation task, we gathered 14 engineering students (1 female, 13 male) aged 19–29 (median: 25) to rate their preferences after viewing all the designs and provide general feedback of their impressions. It is worth noting that the entire subject pool reported familiarity with robotics, coinciding with the target user population of most teleoperation applications. Each participant was also presented a random sequence of the visualizations to offset any effects from ordering.

Participant ratings from this survey are illustrated in **Figure 5**, which demonstrate that the most and least preferred options are predominantly the "Model + Compass" and "Camera + Polytopes" configurations, respectively. There were a few remarks made by volunteers that are worth mentioning here. To begin with, several participants stated that the direct polytope visualization was not immediately intuitive and "hard to understand" or "confusing" as a visual guide. However, many highlighted that the concept of the polytope and how it encapsulates the robot's internal state and environment could be useful if presented in alternative ways. Some participants also commented on how more time spent getting accustomed to their properties, or more "coaching", may have improved their ratings. Another recurring statement was about having a zoomed in

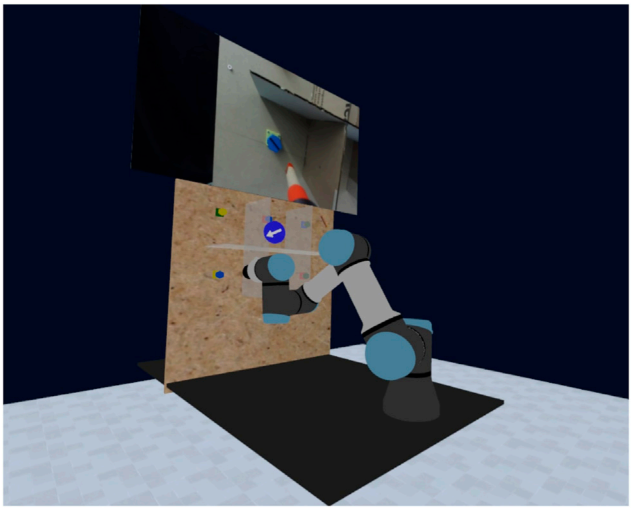

**FIGURE 6 |** Final virtual reality interface bears similarity with the most popular "Model + Compass" prototype, except for two adjustments. First, the image displayed above the model now portrays the view captured from a mounted end effector camera. Second, the compass scaling in size has been adjusted according to user recommendations.

perspective of the bolts, as the stereo camera did not provide enough fidelity or depth perception for the task. Several participants selected the "Model" view as their preferred visualization because it helped them acquire perspective of the task and allowed them additional viewpoints. Other feedback points were to change the scaling of the compass size or increase the transparency of the polytope shading.

### 3.3 Final Interface

With the results from our user-centered design survey, we opted to use the model-based interface with the compass guide as our final interface. Based on the verbal feedback received during the study, we implemented two modifications to this interface. First, we moved the camera's mount point from above-and-behind the robot, which gave a third-person point of view, to a first-person point of view where the camera was directly mounted to the end effector of the robot arm. This modification provided a closer look at the screwdriver and bolts. Second, we adjusted the minimum and maximum scale of the compass to help emphasize the distance to goal. The resulting interface is displayed in **Figure 6**. Despite tailoring this interface to the screwdriver teleoperation task, it could easily generalize to other applications that require both gross and fine manipulation.

## 4 EXPERIMENT

To evaluate our proposed VR system for shared control using motion polytopes, we conducted an experiment where subjects teleoperated a robotic arm to screw bolts tight. The chosen platform for this task was a UR3e collaborative robotic arm with a 3D printed extension at its end effector to hold a wooden screwdriver (refer to **Figure 1**). All other hardware related to the VR setup is described in **Section 3.1**. The shared control processes of intention estimation and arbitration via polytope constraints are implemented atop of the ROS (Quigley et al., 2009).

### 4.1 Experimental Setup and Protocol

A total of 14 volunteers (3 female, 11 male) aged 20–31 (median: 25) were recruited for the experiment. Unlike the pilot study to determine a VR interface, not all subjects had a robotics background for this experiment, thereby reflecting a broader community of potential operators for teleoperation activities. Indeed, three reported no robotics familiarity and there was an even split in those knowledgeable about VR. The purpose of this experiment was to assess whether different control modes and visualizations of the internal robot decision-making would impact an operator's performance and overall experience when guiding a screwdriver. Three modes of control were examined: direct teleoperation (Mode "A"), shared control using joint space polytopes as described in **Section 2** (Mode "B"), and shared control combined with the compass visualization from **Section 3.3** ("B + Viz").

The experimental setup and protocol are as follows. Subjects were initially provided with an information sheet outlining the task specifications before signing a consent form. To reduce the fatigue and workload incurred on participants, the task was simplified to only require a half-turn tightening of the two blue bolts on the wooden board (see bottom left of **Figure 1**). The ordering of different control modes was also randomized to counterbalance the effects of trial order. Prior to each trial, an optional training period with the handheld controller was offered to subjects, where they could operate the robotic arm without wearing the HTC Vive headset. Likewise, the VR interface and visualizations (*e.g.,* the compass) were explained to prevent any confusion on their purpose mid-trial. Moreover, subjects were instructed to perform the task as quickly and safely as possible, which coincides with our metrics for evaluation in the next section.

### 4.2 Evaluation Metrics

An array of quantitative and qualitative metrics is considered for this teleoperation experiment. We adopt two traditional quantitative metrics in the evaluation of shared control against direct teleoperation: time-to-completion and number of collisions. Both of these metrics are recorded manually by the experimenters, with collisions separated into two categories: major and minor. Minor collisions refer to instances where the robotic arm comes into any contact with obstacles, while major events are whenever the UR3e triggers an emergency stop, *e.g.,* due to excess force supplied at the end effector. In the case of emergency stops, a subject's times would be discarded, and they would then proceed onto the next control mode. Our last quantitative measure is the mean temporal distance from environmental object constraints along the task, which can be regarded as an indicator of safety.

Nevertheless, these quantitative metrics do not account for human factors, such as user preference or cognitive workload. As a result, we opt to use two prevalent questionnaires within the field of human-robot interaction: the NASA-TLX (Hart and Staveland, 1988) and the System Usability Scale (SUS) created by Brooke (1996). Both rating scales have been applied in similar mixed reality studies (Brooks and Szafir, 2020; Rosen et al., 2020), with the NASA-TLX denoting perceived workload and the SUS relating to usability. Participants completed these questionnaires immediately after every trial. In addition to the NASA-TLX and SUS, users were also asked after their last experiment trial to provide general feedback via a survey. This feedback included answering three questions on a 5-point Likert scale to assert how *distracting*, *effective* and *predictable* the VR visualizations were at elucidating the shared control. At the end of the experiment, participants had to indicate their preferred mode of control.

### 4.3 Quantitative Results

The quantitative results on total time taken are illustrated in the left-hand plot of **Figure 7**. A one-way repeated ANOVA signals a statistical effect on the basis of timing ($F$ (2, 26) = 5.333, $p$ = 0.011), yet a post-hoc analysis with Tukey's HSD test finds no significance between direct teleoperation and shared control assistance ($p$ > 0.05 for multiple comparisons). Though the general trend suggests quicker times when completing the task

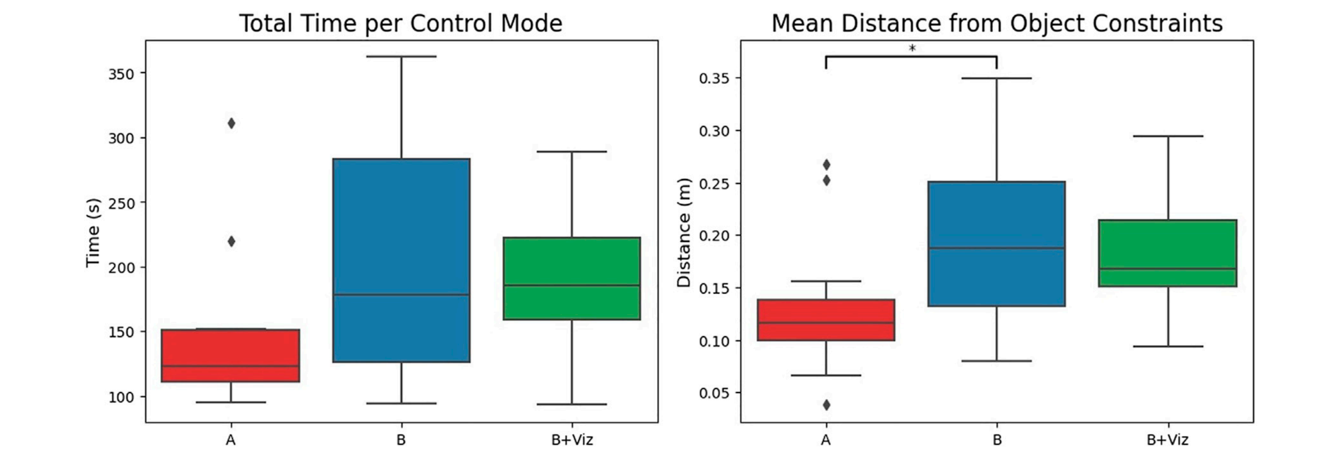

**FIGURE 7 |** Total time taken and average distance from environmental obstacles when completing the screwdriver task using three different control modes. Direct teleoperation garners quicker times than the shared control modes ("B" and "B + Viz") at the expense of safety, as signified by the closer proximity of trajectories to object constraints.

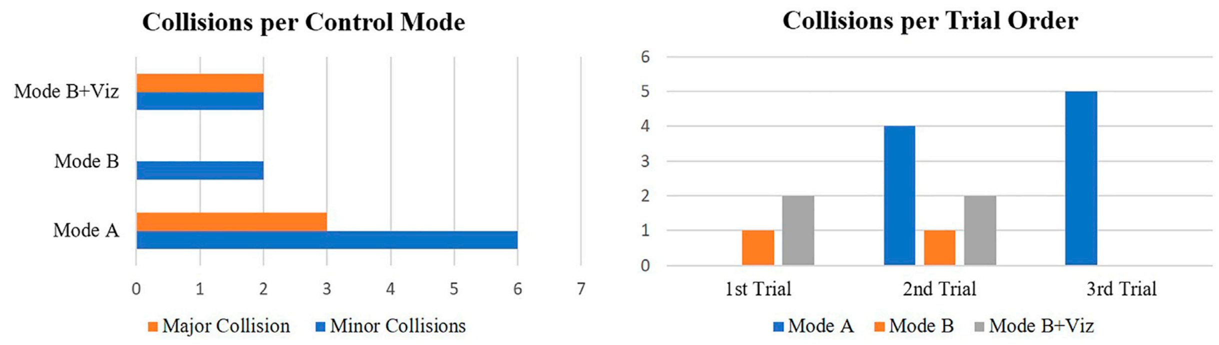

**FIGURE 8 |** Number of collisions for the three control modes illustrated as a function of collision type and trial order. Direct teleoperation had a much higher collision frequency over shared control modes ("B" and "B + Viz"), despite only occupying a single run per subject. Any potential improvements derived from users learning the task across multiple trials also had no positive impact on the collision count for mode "A".

with mode "A" (148.1 ± 64.3s) than modes "B" (205.4 ± 95.4s) and "B + Viz" (192.5 ± 57.3s). This outcome might have been anticipated given that the shared controller is designed to constrain end effector motion in order to ensure safety.

To observe differences in safety, we refer to the average distance from obstacle constraints shown on the right of **Figure 7**. Running a one-way repeated ANOVA yields a statistical difference for this metric ($F$ (2, 26) = 5.116, $p$ = 0.013). Post-hoc analysis with Tukey's HSD corrections indicates that joint state trajectories under direct teleoperation are at significantly closer proximity than those for shared control ($p$ = 0.029), but not the visualization extension ($p$ = 0.109). The lack of significance for mode "B + Viz" is possibly linked to the compass aid interfering with a user's environmental perception. Regardless, it is clear that direct teleoperation possessed higher risk of contact with objects in the environment.

The collision count in **Figure 8** further reinforces this statement by demonstrating a higher frequency of both minor and major events under direct teleoperation. Given the nature of the task and how forceful contact was necessary to screw in the bolts, even "B" modes accrued some collisions. This is primarily because the bolts themselves could not be modelled as obstacle constraints for the motion polytopes. Another noteworthy observation about collisions is related to the effect of trial order (right-hand side of **Figure 8**). For shared control modes, collisions only occurred during the earlier trials where subjects were often still becoming accustomed to the task (emergency stops for "B + Viz" only happened on *first* trials). Whereas for direct teleoperation, collisions cluster around the latter trials with most of them accumulated on the *last* trial.

## 4.4 Survey Results

The survey responses to the SUS and NASA-TLX are summarized in **Figure 9**. A repeated ANOVA test reveals no main significant effects across the groups ($F$ (2, 26) = 2.804, $p$ = 0.079 for SUS; $F$ (2, 39) = 0.101, $p$ = 0.904 for NASA-TLX). Hence, there are no

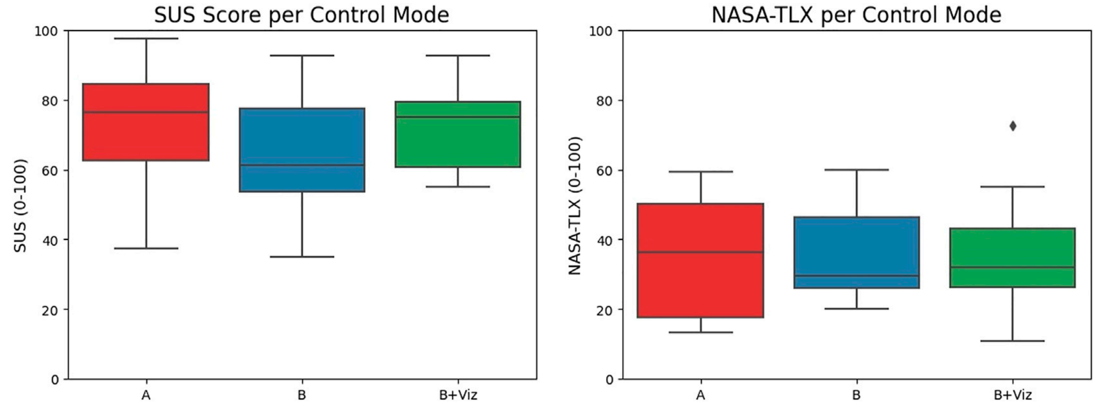

**FIGURE 9 |** SUS score and NASA-TLX for the screwdriver task using the three different control modes. While the SUS score for shared control without visual feedback (mode "B") are the lowest rating of usability, there are no significant differences between the three in perceived workload.

**TABLE 1 |** General ratings on the visualizations from 1–5 (1 = strongly agree, 5 = strongly disagree).

| Question | Mean Rating ± Std. Dev |
|---|---|
| I felt *distracted* by the visualizations | 3.79 ± 1.42 |
| I found the visualizations *effective* for the task | 2.21 ± 1.31 |
| I could *predict* collisions from the visualizations | 3.00 ± 1.57 |

**TABLE 2 |** Preference scores for the three control modes.

| Mode A | Mode B | Mode B + Viz |
|---|---|---|
| 7 | 3 | 4 |

strong conclusions to be drawn from these qualitative results, except that the SUS score for mode "B" (62.3 ± 16.9) is marginally less than that of "A" (72.3 ± 16.9) and "B + Viz" (72.5 ± 12.3). These trends in scores suggest greater usability when either employing the reactive nature of direct teleoperation or the visual guidance of the compass in shared control. Also note that both mode "A" and "B + Viz" exceed an average usability score of 68 (Sauro, 2011), which is a positive outcome for this otherwise challenging screwdriver teleoperation task. As for the NASA-TLX, the perceived workload exerted by participants bears significant similarities across the control modes.

**Table 1** contains the general perceptions of users on the VR interface. Subjects tended to rate the visualizations as *effective* and unlikely to cause *distraction*, but the lack of *predictability* for collisions hints at an ongoing model mismatch. More specifically, the color-coding of the compass was ineffective at warning participants of any imminent danger due to nearby obstacles. The preferred modes shown in **Table 2** also delineate that there is an even split between direct teleoperation and shared control, implying that improvements are warranted before superior preference for mode "B + Viz" is attainable.

# 5 DISCUSSION

There are a few key insights drawn from this screwdriver teleoperation experiment. First and foremost, we observe that the rationale behind our polytope-based shared control was not made abundantly transparent to users and thus persists the problem of model misalignment. This is evident from the user's ratings in **Table 1**, whereby the color-coded compass did not help operators anticipate collisions. A further testament to this point on model mismatch is that the "less reactive" nature of mode "B" was not immediately obvious to various subjects. For our target screwdriver setting, this "slow" behavior was particularly notable, as the shrinking polytope volumes had the adverse effect of preventing users from easily departing a bolt once it had been screwed tight. Various subjects commented on this effect with statements like "it was difficult to pull back" or "it slowed me down".

Constrained motion is a byproduct of our shared control assistance, however it also has the beneficial effect of increasing safety. For instance, the arbitration procedure described in **Section 2.3** helped stabilize operator control when zoning in on bolts and thus reduced the number of collisions with the enclosing borders of the environment. Likewise, the mean distance maintained from obstacle constraints along shared control trials was significantly better than direct teleoperation, accounting for improved safety. While collisions persisted across all control modes, there were none recorded for users with shared control on their final trial. This pattern hints that subjects had either formed a dependency on the safe motion adjustments of the arbitration, or that any potential improvements procured from learning the task could not translate into performance gains with direct teleoperation.

Another finding from this study revolves around user preference. Even after improving the success rate of participants in shared control modes by reducing the number of failed attempts due to major collisions, user preference was still split across the modes. Although speed is not necessarily correlated with user preference and there are other factors at play, such as performance and transparency (Dragan and Srinivasa, 2013), it can still influence people's opinions in toy experiment settings like ours. An example of this is when one subject opted for direct teleoperation as their favorite control setting after successfully completing the task with modes "B" and "B + Viz", despite triggering an emergency stop on mode "A". We stipulate that by simulating incentives to complete the task *successfully*, *e.g.*, offering money, user preferences may have swayed towards favoring shared control.

Lastly, it is worth mentioning that our setup to remote teleoperate a robot wielding a screwdriver is only preliminary and possesses certain limitations. For example, a prominent drawback of our study are its low subject numbers, meaning the results are unlikely to hold sufficient statistical relevance. Moreover, the screwdriver and bolts are from a children's wooden tool set, which simplify the task but reduce it down to a toy setting. As mentioned earlier, the bolts are not modelled as constraints for the polytopes due to the force exertion necessary for screwing, and so the task is not fully accommodated by the shared control methodology. Regardless of these diminutions, the overall experiment strives to mimic real-world applications of remote teleoperation and we hope that it inspires further studies to adopt similar setups.

## 6 FUTURE WORK

A critical finding of our study is that the compass aid derived from the pilot survey was unable to successfully improve the predictability of collisions during teleoperation. In spite of its clear and non-distracting purpose, the shared control remained somewhat elusive to users. Based on this finding and user suggestions, we suspect that an important avenue for future research on model misalignment in shared control is to explore longer term human-robot interaction.

Recent teleoperation frameworks have addressed model mismatch through immersive interface design (Zolotas and Demiris, 2019; Brooks and Szafir, 2020), however the literature is sparse in studies that investigate the role of lifelong assistance (Demiris, 2009). In our pilot survey, numerous participants were appealed by the idea of directly visualizing polytopes, yet struggled to build an intuition for them in the short span of a quick trial. Another encouraging result of our teleoperation experiment was the lack of collisions in later trials under shared control, which prompts an examination into the learning curves of different users. We thus hypothesize that a study into more complex virtual guides for shared control, like polytopes as geometrical objects, require longer experiment scales, *e.g.,* trials spread out across multiple days or interactions. While no haptic interface was incorporated into our proposed framework, future architectures should also endeavor to issue multimodal feedback (Losey et al., 2018), especially for physical tasks like screwing bolts.

An additional course for inquiry will be to improve the means of generating polytopes during arbitration. At present, the constrained motion polytopes only account for robot and environmental constraints, but they could also be tailored to individual user preferences. A simple use-case of this notion could be to dynamically adjust the linearization limit from within the VR interface, thereby allowing the operator a personalized experience. Furthermore, it should be noted that the polytopes are generated at run time in response to robot configuration and environmental changes. At present, an environment model is loaded during the initialization and then added to a maintained collision world. This environment comprises a set of objects defined by collision models, either described by solid primitives or meshes, and pose information. Objects can be added or removed at run time, but this process is cumbersome and impractical. Ongoing work is focusing on integrating *octomaps* (Hornung et al., 2013), 3D occupancy grids based on real-time sensor data, to enable the generation of motion polytopes from dynamic point cloud data.

Finally, our algorithm does not currently respect real-time constraints. Cycle time is minimized by reducing the mesh density of collision objects, but in future work we believe real-time constraints can be satisfied by the following measures. First the control frequency can be further increased by isolating the polytopes from the control loop. In doing so, the polytopes can be generated or updated periodically, independent of user input commands, allowing teleoperation within a feasible space at a much faster cadence. Second, we used SNOPT (Gill et al., 2005) to solve **(Eq. 22)**, but real-time performance could be obtained by using an anytime optimization solver (*e.g.,* an interior point method Wachter, 2002) that returns a feasible solution at any termination point.

## 7 CONCLUSION

In this paper, we introduced a shared control method for teleoperation using constrained motion polytopes and developed a corresponding VR interface. The novelty over previous virtual fixture methods for arbitration in shared control was the introduction of polytopes through the notion of "shrinking" linearization limits. To accommodate remote teleoperation, a model-based VR interface was also presented to guarantee effective telepresence. A pilot survey was first conducted to inform the design of this VR interface and help avoid misunderstandings about the underlying shared control. The resulting system was then evaluated in a human-robot interaction experiment involving a UR3e arm extended with a screwdriver, so as to discern its performance and promise for remote manipulation.

Our experimental results reinforce the usability of our proposed VR system for the screwdriver manipulation task. In particular, we observed that while slower than direct teleoperation, the shared control led to increased safety. Moreover, the shrinking volume of the polytopes prevented erratic motion at close proximity to the bolts, which in turn reduced the likelihood of failing the task. Despite the challenging nature of this screwdriver task, most participants successfully completed every trial using our VR interface, citing the experience as fun and enjoyable.

## DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author.

## ETHICS STATEMENT

The studies involving human participants were reviewed and approved by Northeastern University Institutional Review Board. The patients/participants provided their written informed consent to participate in this study.

## AUTHOR CONTRIBUTIONS

PL authored the "Polytopes for Arbitration" section and created **Figure 2**. MW authored the "Virtual Reality System—Architecture" section and generated **Figures 1**, **3**, **4**, and **6**. MZ authored and compiled the remainder of the manuscript. TP supervised the work.

## FUNDING

This material is based upon work supported by the National Science Foundation under Award No. 1928654, 1935337, and 2031326.

## ACKNOWLEDGMENTS

The authors would like to thank James Winn and Hillel Hochsztein for their assistance in designing the experimental setup.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2021.730433/full#supplementary-material

## REFERENCES

Abbink, D. A., Carlson, T., Mulder, M., de Winter, J. C. F., Aminravan, F., Gibo, T. L., et al. (2018). A Topology of Shared Control Systems-Finding Common Ground in Diversity. *IEEE Trans. Human-mach. Syst.* 48, 509–525. doi:10.1109/THMS.2018.2791570

Bettini, A., Marayong, P., Lang, S., Okamura, A. M., and Hager, G. D. (2004). Vision-assisted Control for Manipulation Using Virtual Fixtures. *IEEE Trans. Robot.* 20, 953–966. doi:10.1109/tro.2004.829483

Bowyer, S. A., Davies, B. L., and Rodriguez y Baena, F. (2014). Active Constraints/virtual Fixtures: A Survey. *IEEE Trans. Robot.* 30, 138–157. doi:10.1109/tro.2013.2283410

Brooke, J. (1996). Sus: a "Quick and Dirty" Usability Scale. *Usability Eval. industry* 189. doi:10.1201/9781498710411-35

Brooks, C., and Szafir, D. (2020).Visualization of Intended Assistance for Acceptance of Shared Control. In IEEE/RSJ International Conference on Intelligent Robots and Systems. Las Vegas, United States: Virtual. doi:10.1109/iros45743.2020.9340964

Caron, S., Pham, Q.-C., and Nakamura, Y. (2017). Zmp Support Areas for Multicontact Mobility under Frictional Constraints. *IEEE Trans. Robot.* 33, 67–80. doi:10.1109/tro.2016.2623338

Demiris, Y. (2009). Knowing when to Assist: Developmental Issues in Lifelong Assistive Robotics. In Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 3357–3360. doi:10.1109/IEMBS.2009.5333182

Demiris, Y. (2007). Prediction of Intent in Robotics and Multi-Agent Systems. *Cogn. Process.* 8, 151–158. doi:10.1007/s10339-007-0168-9

Dragan, A. D., and Srinivasa, S. S. (2013). A Policy-Blending Formalism for Shared Control. *Int. J. Robotics Res.* 32, 790–805. doi:10.1177/0278364913490324

Fukuda, K., and Prodon, A. (1996). Double Description Method Revisited. In *Combinatorics and Computer Science*. Springer, 91–111. doi:10.1007/3-540-61576-8_77

Gill, P. E., Murray, W., and Saunders, M. A. (2005). Snopt: An Sqp Algorithm for Large-Scale Constrained Optimization. *SIAM Rev.* 47, 99–131. doi:10.1137/s0036144504446096

Hart, S. G., and Staveland, L. E. (1988). "Development of Nasa-Tlx (Task Load index): Results of Empirical and Theoretical Research,". *Human Mental Workload*. Editors P. A. Hancock and N. Meshkati (North-Holland: vol. 52 of Advances in Psychology), 139–183. doi:10.1016/S0166-4115(08)62386-9

Hetrick, R., Amerson, N., Kim, B., Rosen, E., Visser, E. J., and Phillips, E. (2020). Comparing Virtual Reality Interfaces for the Teleoperation of Robots. In Systems and Information Engineering Design Symposium. 1–7. doi:10.1109/sieds49339.2020.9106630

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). Octomap: An Efficient Probabilistic 3d Mapping Framework Based on Octrees. *Auton. Robot* 34, 189–206. doi:10.1007/s10514-012-9321-0

Jain, S., and Argall, B. (2020). Probabilistic Human Intent Recognition for Shared Autonomy in Assistive Robotics. *J. Hum.-Robot Interact.* 9, 1–23. doi:10.1145/3359614

Javdani, S., Srinivasa, S. S., and Bagnell, J. A. (2015). Shared Autonomy via Hindsight Optimization. *Robotics Sci. Syst. : Online Proc.* doi:10.15607/RSS.2015.XI.032

Kokkinis, T., and Paden, B. (1989). Kinetostatic Performance Limits of Cooperating Robot Manipulators Using Force-Velocity Polytopes. *ASME Winter Annu. Meet.*, 151–155.

Krut, S., Company, O., and Pierrot, F. (2004). Velocity Performance Indices for Parallel Mechanisms with Actuation Redundancy. *Robotica* 22, 129–139. doi:10.1017/s0263574703005411

Kucukyilmaz, A., and Demiris, Y. (2018). Learning Shared Control by Demonstration for Personalized Wheelchair Assistance. *IEEE Trans. Haptics* 11, 431–442. doi:10.1109/TOH.2018.2804911

Lee, J. (1997). A Study on the Manipulability Measures for Robot Manipulators. *IEEE/RSJ Int. Conf. Intell. Robots Syst.* 3, 1458–1465.

Long, P., Keleştemur, T., Önol, A. Ö., and Padir, T. (2019). Optimization-Based Human-In-The-Loop Manipulation Using Joint Space Polytopes. In International Conference on Robotics and Automation. 204–210doi:10.1109/icra.2019.8794071

Long, P., and Padır, T. (2018). Evaluating Robot Manipulability in Constrained Environments by Velocity Polytope Reduction. In IEEE-RAS International Conference on Humanoid Robots. 497–502doi:10.1109/humanoids.2018.8624962

Long, P., and Padır, T. (2020). Constrained Manipulability for Humanoid Robots Using Velocity Polytopes. *Int. J. Hum. Robot.* 17, 1950037–1950041. doi:10.1142/s0219843619500373

Losey, D. P., McDonald, C. G., Battaglia, E., and O'Malley, M. K. (2018). A Review of Intent Detection, Arbitration, and Communication Aspects of Shared Control for Physical Human-Robot Interaction. *Appl. Mech. Rev.* 70. doi:10.1115/1.4039145

Losey, D. P., Srinivasan, K., Mandlekar, A., Garg, A., and Sadigh, D. (2020). Controlling Assistive Robots with Learned Latent Actions. *IEEE Int. Conf. Robotics Automation*, 378–384. doi:10.1109/icra40945.2020.9197197

Mortimer, M., Horan, B., and Seyedmahmoudian, M. (2017). Building a Relationship between Robot Characteristics and Teleoperation User Interfaces. *Sensors* 17, 587. doi:10.3390/s17030587

O'Malley, M. K., Gupta, A., Gen, M., and Li, Y. (2005). Shared Control in Haptic Systems for Performance Enhancement and Training. *J. Dynamic Syst. Meas. Control.* 128, 75–85. doi:10.1115/1.2168160

Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., et al. (2009). ROS: an Open-Source Robot Operating System. In ICRA Workshop on Open Source Software.

Quintero, C. P., Dehghan, M., Ramirez, O., Ang, M. H., and Jagersand, M. (2017). Flexible Virtual Fixture Interface for Path Specification in Tele-Manipulation. In IEEE International Conference on Robotics and Automation. 5363–5368. doi:10.1109/ICRA.2017.7989631

Rasheed, T., Long, P., Marquez-Gamez, D., and Caro, S. (2018).Tension Distribution Algorithm for Planar mobile cable-driven Parallel Robots. In Cable-Driven Parallel Robots. Springer, 268–279. doi:10.1007/978-3-319-61431-1_23

Rosen, E., Whitney, D., Fishman, M., Ullman, D., and Tellex, S. (2020). Mixed Reality as a Bidirectional Communication Interface for Human-Robot Interaction. *IEEE/RSJ Int. Conf. Intell. Robots Syst.* doi:10.1109/iros45743.2020.9340822

Rosenberg, L. B. (1992). in *The Use of Virtual Fixtures as Perceptual Overlays to Enhance Operator Performance in Remote Environments* (Stanford, CA: Stanford Univ Ca Center for Design Research).*Tech. Rep.*

Sauro, J. (2011). *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. California, United States: Scotts Valley. Measuring Usability LLC.

Selvaggio, M., Abi-Farraj, F., Pacchierotti, C., Giordano, P. R., and Siciliano, B. (2018). Haptic-based Shared-Control Methods for a Dual-Arm System. *IEEE Robot. Autom. Lett.* 3, 4249–4256. doi:10.1109/LRA.2018.2864353

Sibirtseva, E., Kontogiorgos, D., Nykvist, O., Karaoguz, H., Leite, I., Gustafson, J., et al. (2018). A Comparison of Visualisation Methods for Disambiguating Verbal Requests in Human-Robot Interaction. In IEEE International Symposium on Robot and Human Interactive Communication, 43–50. doi:10.1109/roman.2018.8525554

Tanwani, A. K., and Calinon, S. (2017). A Generative Model for Intention Recognition and Manipulation Assistance in Teleoperation. *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 43–50. doi:10.1109/IROS.2017.8202136

Tsai, M.-J. (1986). *Workspace Geometric Characterization and Manipulability of Industrial Robots*. Ohio: Ph.D. thesis, The Ohio State University.

Vahrenkamp, N., Asfour, T., Metta, G., Sandini, G., and Dillmann, R. (2012). Manipulability Analysis. In IEEE-RAS International Conference on Humanoid Robots. 568–573doi:10.1109/humanoids.2012.6651576

Vahrenkamp, N., and Asfour, T. (2015). Representing the Robot's Workspace through Constrained Manipulability Analysis. *Auton. Robot* 38, 17–30. doi:10.1007/s10514-014-9394-z

Van de Merwe, D. B., Van Maanen, L., Ter Haar, F. B., Van Dijk, R. J. E., Hoeba, N., and Van der Stap, N. (2019). Human-robot Interaction during Virtual Reality Mediated Teleoperation: How Environment Information Affects Spatial Task Performance and Operator Situation Awareness. In International Conference on Human-Computer Interaction. 163–177. doi:10.1007/978-3-030-21565-1_11

Wachter, A. (2002). *An interior point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. Ph.D. thesisCarnegie Mellon University.

Walker, M., Hedayati, H., Lee, J., and Szafir, D. (2018). Communicating Robot Motion Intent with Augmented Reality. In ACM/IEEE International Conference on Human-Robot Interaction. (New York, NY, United States: HRI'18), 316–324. doi:10.1145/3171221.3171253

Wonsick, M., Kelestemur, T., Alt, S., and Padir, T. (2021a). Telemanipulation via Virtual Reality Interfaces with Enhanced Environment Models. In IEEE/RSJ International Conference on Intelligent Robots and Systems.

Wonsick, M., Long, P., Onol, A., Wang, M., and Padir, T. (2021b). A Holistic Approach to Human-Supervised Humanoid Robot Operations in Extreme Environments. *Front. Robotics AI* 8, 148. doi:10.3389/frobt.2021.550644

Wonsick, M., and Padir, T. (2020). A Systematic Review of Virtual Reality Interfaces for Controlling and Interacting with Robots. *Appl. Sci.* 10, 9051. doi:10.3390/app10249051

Xi, B., Wang, S., Ye, X., Cai, Y., Lu, T., and Wang, R. (2019). A Robotic Shared Control Teleoperation Method Based on Learning from Demonstrations. *Int. J. Adv. Robotic Syst.* 16, 172988141985742. doi:10.1177/1729881419857428

Yamamoto, T., Abolhassani, N., Jung, S., Okamura, A. M., and Judkins, T. N. (2012). Augmented Reality and Haptic Interfaces for Robot-Assisted Surgery. *Int. J. Med. Robotics Comput. Assist. Surg.* 8, 45–56. doi:10.1002/rcs.421

Yoshikawa, T. (1984). Analysis and Control of Robot Manipulators with Redundancy. In International Symposium on Robotics Research, 735–747.

Zeestraten, M. J. A., Havoutis, I., and Calinon, S. (2018). Programming by Demonstration for Shared Control with an Application in Teleoperation. *IEEE Robot. Autom. Lett.* 3, 1848–1855. doi:10.1109/LRA.2018.2805105

Zolotas, M., and Demiris, Y. (2019). *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3020–3026. doi:10.1109/IROS40897.2019.8968117Towards Explainable Shared Control Using Augmented Reality

Zolotas, M., Elsdon, J., and Demiris, Y. (2018). Head-Mounted Augmented Reality for Explainable Robotic Wheelchair Assistance. In International Conference on Intelligent Robots and Systems, 1823–1829. doi:10.1109/IROS.2018.8594002

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.