2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 27 - October 1, 2021. Prague, Czech Republic

2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-1714-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IROS51168.2021.9636008

Cursor-based Robot Tele-manipulation through 2D-to-SE2 Interfaces

Maria E. Cabrera[1,*], Kavi Dey[2,*], Kavita Krishnaswamy[3], Tapomayukh Bhattacharjee[1], Maya Cakmak[1]

Abstract— **Cursor-based tele-operation interfaces for manipulators can enable widely available and accessible control of robots to make many near term applications possible. However, their efficiency is restricted by the challenge of controlling 6 Degrees-of-Freedom (DoF) with 2D input from the cursor. Existing interfaces make use of different strategies to tackle this challenge, including viewpoint constraints, mode switching, and visual overlays, but it is unclear how these strategies impact the efficiency and accessibility of the interface. In this paper we characterize the design space of cursor-based robot control interfaces and compare alternatives in two user studies. Study 1 (N=216) compares nine alternative interfaces focusing on control of 3 DoFs to understand the differences of the interfaces at the basic level and examine the impact of task parameters on efficiency. Study 2 (N=60) compares a subset of the interfaces integrated into a system that allows full control of a robot manipulator from three orthogonal views. We also present a framework for heuristically evaluating accessibility of these interfaces and discuss the efficiency and accessibility trade-off with recommendations.**

I. Introduction

Robot tele-operation has been a topic of interest in robotics for several decades, yet remains relevant with more and more robots being deployed to perform real world tasks. Tele-operation enables applications that are currently not possible with autonomy and serves as a fall-back or recovery mechanism for failures of autonomy. Existing tele-operation strategies for robot manipulators using widely available input devices, such as the common point and click mouse, are currently too inefficient to be practical. Further, these interfaces pose many accessibility challenges to users with mobility limitations, who could arguably benefit the most from near-term applications of mobile manipulators enabled by tele-operation.

A number of tele-operation interfaces for manipulators based on cursor input have been developed in prior work (see Sec. II). At the core of the inefficiency of these interfaces is the challenge of controlling 6 Degree-of-Freedom (DoF) robot end-effectors with a cursor that provides only 2D input. Most interfaces employ a combination of three strategies: (1) *Viewpoint constraints*: Interfaces involve 3D rendering of the robot that can be viewed from different perspectives, often changeable by the user. The viewpoint constrains the meaning of the cursor input. (2) *Mode switching*: The cursor input can be used to control different DoFs in different modes, enabled by mode-switching mechanisms. (3) *Visual overlays*: Similarly, the cursor input can be used to control different DoFs in different parts of the screen indicated by visual elements overlaid on the screen.

[1] University of Washington, USA {mecu,tapo,mcakmak}@cs.uw.edu
[2] Seattle Academy, USA kaviasher@gmail.com
[3] U. of Maryland, Baltimore County, USA kavi1@umbc.edu
[*] Co-lead authors

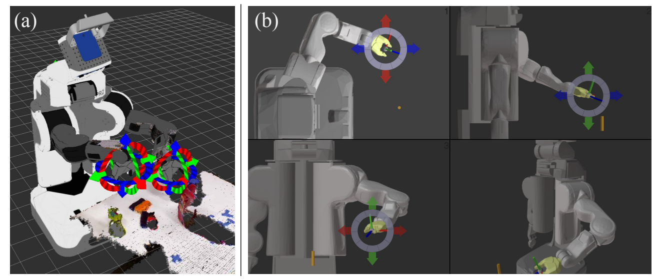

Fig. 1: (a) Typical interfaces that allow 6 DoF (SE3) control of end-effectors and free-form view changes are generally inaccessible. (b) To reduce the number of possible control inputs at any given time we use fixed orthogonal views and focus on 3 DoF (SE2) control in each view.

This set of strategies present a large design space with many variations, but it is unclear how choices in the design space impact the efficiency and accessibility of the resulting interface. In this paper we set out to characterize the interface design space and evaluate the performance of different alternatives in this space. To simplify the analysis, we first focus on the control of 3 DoFs, position and orientation in 2D (*i.e.,* SE2), with a cursor. We construct nine different interfaces inspired by existing tele-operation interfaces for manipulators (Fig. 1). In our first user study (N=216) we evaluate the efficiency of these interfaces in different settings and examine their dependence on task parameters. We then take a subset of the constructed SE2 interfaces and combine them in a 3-view interface to fully control a robot manipulator. We evaluate the performance of these interfaces in object grasping tasks in our second user study (N=60).

Our paper makes several contributions:

1) A characterization of the design space of cursor-based remote tele-operation interfaces and a common framework for representing different interfaces with Finite State Machines.
2) Empirical data from two user studies comparing different interfaces in low-level and high-level tasks.
3) A framework for heuristically evaluating the accessibility of user interfaces.
4) Recommendations informed by study findings for striking the right balance between interface accessibility and efficiency.

II. Related Work

Robot tele-operation has been studied for many centuries across application areas ranging from search&rescue [1] and space exploration [2] to surgery [3]. Prior literature surveys on the topic have characterized the space of solutions and identified challenges that impact their efficiency and usability [4], [5]. Although some challenges like image quality or network delays have been mitigated with advancing communication technologies, many challenges like depth perception or self occlusion remain relevant for today's interfaces.

For tele-operating robot manipulators (also referred to as *tele-manipulation*), one particular type of interface based on *interactive markers* [6] has become particularly popular. These interfaces often involve a 3D rendering of the robot that can be viewed from different viewpoints, augmented with interactive visualizations attached to parts of the robot that can be activated and moved by the user to provide control input to the robot (an example is shown in Fig. 1). These have been used across interfaces developed for assistive robots [7], [8] and featured in several of the DARPA robotics challenge entries [9], [10], [11], [12]. Prior user studies have demonstrated the advantages of different variations of this type of interface [13], [14]. Alternative interfaces using different input devices, such as exoskeletons [15], pedals, joysticks [16], and haptic devices [17] also exist.

While direct tele-operation of robots remains relevant, one recent trend in robot tele-operation is towards semi-autonomous solutions [18], [19]. This includes work on *virtual fixtures* which guide or prevent operator actions to increase efficiency and accuracy [20], [21], [22], [23], work on *assistive tele-operation* which involves recognizing the user's intention from their control input and assisting their actions with autonomy [24], [25], [26], [27], [28], [29], as well as work on assisted mode-switching through predictions of future actions [30], [27], [31].

Accessibility of tele-operation interfaces for users with mobility impairments has also been recognized as an important issue in tele-operation research. As part of the *Robots for Humanity* project [5] researchers developed several tele-operation interfaces for one particular user with severe mobility impairments to perform tasks like fetching items from drawers, trick-or-treating at Halloween, and shaving [7], [32], [33], [34]. Mast et al. developed tele-operation interfaces for informal and professional caregivers to assist older adults [35], [36], [37]. Tsui et al. evaluated interface choices, such as joystick versus touchscreen, for the control of a wheelchair-mounted robot arm [38]. Our previous work explored the design of accessible alternatives to existing interfaces for one of the authors of this paper who has Spinal Muscular Atrophy, requires assistance for all activities of daily living, and is only able to control a cursor via her right-hand index finger [39]—this work led to some of the interfaces studied in this paper. Accessibility of tele-operation has also been studied for the control of wheelchairs [40] and telepresence devices [41].

A large body of work in human-computer interaction investigates accessibility of user interfaces for people with limited fine motor skills—particularly relevant to our work is research on accessible pointing of cursors [42], [43]. Some researchers have investigated ways in which regular interfaces could be adapted, automatically or by end-users, to best match the individual user's input range [44], [45].

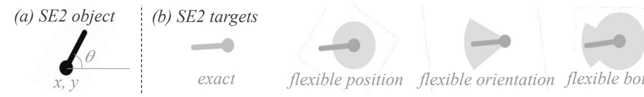

Fig. 2: Visual representation of (a) an abstract SE2 object specified by a position (x, y) and orientation θ; (b) SE2 targets with light-grey regions indicating flexibility in position and orientation.

III. Cursor-based Tele-Manipulation Framework

Typical interfaces that allow 6 DoF (SE3) control of end-effectors and free-form view changes (Fig. 1(a)) are largely inaccessible for users with mobility impairments [39], [34], [32]. To reduce possible control inputs at any given time we use fixed orthogonal views and focus on 3 DoF (SE2) control in each view (Fig. 1(b)).

A. 2D-to-SE2 Interfaces

SE2 objects are 2-dimensional directional objects whose pose is specified by a position (x, y) and an orientation θ. Controlling SE2 objects with a 2D cursor requires structure provided by the interface with mode switching and visual overlays, as described in Sec. I. We refer to the combination of elements that allow using the cursor to control SE2 objects as *2D-to-SE2* interfaces. In this paper we study variations of five different 2D-to-SE2 interfaces inspired by existing robot teleoperation interfaces. Fig. 3 shows the visual elements of each interface and describes their behavior in the form of a Finite State Machine (FSM). The supplementary video shows how the different interfaces work.

Fixed. This interface involves a fixed panel of six buttons, each assigned to the movement of one DoF in one direction. The arrow shape, orientation, and arrangement of the buttons are chosen to reflect the associated movement. This interface can be represented with a 7-state FSM, which starts in a state where the object is not moving. Activating any one of the six buttons shifts the interface to a state in which the SE2 object is moving in a particular dimension and direction. The pose of the SE2 object is updated continuously by a small increment[1], with an empirically adjusted frequency (10Hz), while in these states. The object stops moving when the button is deactivated. Activation-deactivation of a button can be either (i) through a *mouse press* while the cursor is on the button, followed by a *mouse release*; or (ii) two consecutive *mouse clicks*. This presents two variations of the Fixed interface which we refer to as *press/release-based transitions* versus *click-based transitions*.

ArrowRing. This interface involves buttons in the form of arrows and a ring that are attached to and move together with the SE2 object. Horizontal arrows activate movement in x, vertical arrows activate movement in y, and the ring activates rotation θ. While the interface is in a moving state, the pose of the SE2 object is continuously updated based

[1]To allow faster coverage of longer stretches, the distance added to the current pose increases over time up to a maximum value.

on the latest pose of the cursor ($\mathbf{m}$). The mapping from $\mathbf{m}$ to x and y is simply a small offset that depends on where the cursor was when the moving state was activated. The mapping from $\mathbf{m}$ to θ uses the center of the ring (*i.e.,* the current position of the SE2 object) as an anchor point to determine a rotational change based on how the cursor has moved from its initial pose in the moving state. This interface also supports press/release-based and click-based transitions. Transitions between not moving and moving states can be through a mouse press/release or two consecutive clicks, while the cursor is on the arrow or ring regions.

CircleRing. This interface involves buttons in the form of a circle and a ring that are attached to the SE2 object. The circle activates simultaneous translational movement in both dimensions (x, y) and the ring activates rotational movement θ. While in translational movement mode, the pose of the cursor is mapped directly to the pose of the object, with a small offset. The ring works the same way as in the ArrowRing interface. This interface also allows transitions based on press/release and subsequent clicks.

TargetAnchor. This interface involves directly specifying the target pose where the cursor is to be moved independent of its current pose, as opposed to the first three interfaces. An initial press or click anywhere on the screen specifies the position (x, y) of the object, without yet moving it. Instead a *ghost* SE2 object at the specified new position is visualized. In this state, the object is *anchored* to a position but its orientation is not yet specified. The next transition, with the release or another click of the mouse, specifies the orientation based on the anchor point and new cursor pose at the time of the transition. The orientation of the ghost visualization, corresponding to intermediate cursor positions, is updated continuously while in the anchored state to allow the user to find the desired orientation before committing to it. In addition, the ghost visualization turns green if the intermediate pose reaches the target. The object actually moves to the target when both position and orientation have been specified.

TargetRing. The last interface also allows directly specifying the target pose. A click anywhere on the screen, except for the ring attached to the object, directly specifies the position (x, y) of the object and moves it there. A click on the ring directly specifies the target orientation θ of the object and directly rotates it there. This is a stateless interface that requires only one type of mouse event for interface actions, hence does not have a variation that use press/release events.

The five interfaces differ in various ways. While the first three interfaces specify movement of the object from its current pose, the latter two interfaces directly specify where the object should end up, independent of where it currently is. The Fixed interface does not use cursor information other than for which button is being pressed, while the other interfaces map the 2D cursor information to movement or target value of position or orientation. All interfaces but TargetAnchor independently specify position and orientation, while TargetAnchor requires specifying both every time

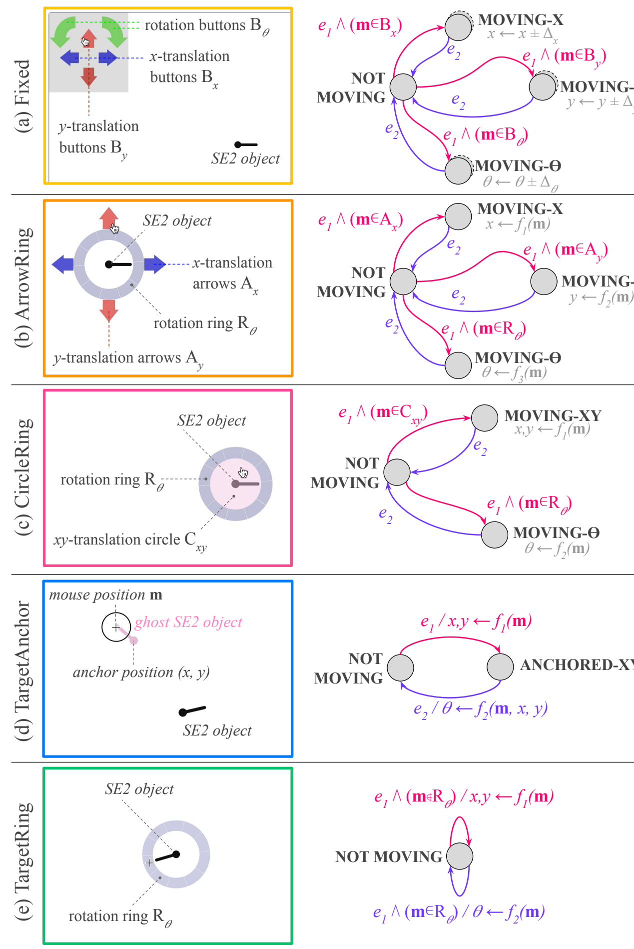

Fig. 3: Five 2D-to-SE2 interfaces studies in this paper. Visual elements of the interface are shown on the left and a Finite State Machine (FSM) representation of the interface is shown on the right. In the first three interfaces (a-c), the mouse event e_1 (mouse press or click) triggers transitions to different states where the pose of the SE2 object is continuously updated. Which state the FSM transitions to is determined by the screen region that the cursor is at the time of the event (*e.g.,* on a button or ring). The second mouse event e_2 (mouse release or another click) transitions the FSM back to the NOT MOVING state. In the latter two interfaces (d,e) the pose of the SE2 object is updated during state transitions based on the cursor.

the object is moved. Additionally, the five interfaces have reducing number of states, with the last one, TargetRing, being stateless (*i.e.,* having only one state).

B. *From SE2 to SE3*

To allow SE3 control of a robot arm, using 2D-to-SE2 interfaces, we combine three, fixed orthogonal views of the robot. Each view allows control of two out of the three position DoFs and one out of the three orientation DoFs of the robot's end-effector, *i.e.,* (x, y, θ_z), (x, z, θ_y), and (y, z, θ_x). Hence, the three views combined give full SE3 control of all DoFs of the end-effector, and any pair of views give full control over the position DoFs[2]. The SE2 object manipulated in each view is a projection of the robot's gripper onto the corresponding orthogonal 2D plane (Fig. 4).

[2]Available at `https://mayacakmak.github.io/se2/se3/demo.html`

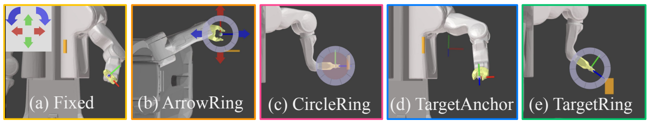

Fig. 4: The five 2D-to-SE2 interfaces applied to the gripper movement of the PR2 robot for full SE3, shown from different orthogonal views.

The center point (x, y) is chosen as the mid point between the robot's fingers, as manipulation tasks often require placing the fingers relative to something in the environment, rather than an arbitrary point on the robot's end-effector.

Only one of three views is active at any given time. The user simply clicks on another view to switch to it. The robot and controls in inactive views are still visible but grayed out. A fourth view shows the robot from an oblique corner view to allow better 3D scene understanding, akin to 3D Computer Aided Design (CAD) tools.

C. Implementation

We implement the different interfaces within a web application that runs on a browser using HTML/CSS/JavaScript. Most interface components are created as Scalable Vector Graphics (SVGs) elements and manipulated through mouse events captured through JS. While some details like color and scale were taken directly from existing robot control interfaces, others were empirically adjusted for usability.

We implement SE3 control for the PR2 robot, which has a 7 DoF manipulator, rendered on a browser using the 3D JavaScript library `three.js`[3]. We use optimization-based inverse kinematics (IK) to find a robot arm joint configuration that satisfies the end-effector pose specified by the user. The optimization manipulates the joint angles of the seven PR2 arm joints to achieve a target position and orientation. We use the Nelder-Mead algorithm [46], with an objective function that combines the current error in position and rotation (between the PR2 hand and the pose specified by the user), and the joint angle difference between the current and previous frame. The frame to frame comparison helps create smooth movement between them. The IK solver is implemented using the JS library `fmin`[4] and runs completely in a browser 60 times a second. When no IK solutions are found for the specified configuration, the robot arm moves as close as possible and it becomes dark gray.

D. Heuristic Assessment of Accessibility

Breaking down each interface to its elements and identifying the different user inputs needed to control an SE2 object, allows us to assess how accessible each interface is based on the extensive literature on user input accessibility. Next we enumerate dimensions in which accessibility of control interfaces can vary and discuss where the different interfaces fall in terms of accessibility.

Continuous/proportional input. Interfaces that do not require precise positioning or movement of the cursor to specify continuous quantities are more accessible as they eliminate potential difficulties and errors [32], [44]. The Fixed interface in this paper does not require any continuous input and is therefore the most accessible in this dimension. The ArrowRing interface uses a smaller subset of the continuous input while in different modes, changing x, y, and θ separately, which can increase tolerance to errors. All our SE3 interfaces use discretization of views, removing the need for continuous input to specify the view point.

Number of discrete actions. Interfaces with smaller numbers of possible discrete actions at any given time are more accessible, as they make it easier to specify the desired action with limited input [32], [30], [47]. Interfaces often reduce the number of possible actions by splitting them into different modes. In our case, the interface with the largest number of states (Fixed) has the highest number of possible transitions from the NOT MOVING state (*i.e.,* six), making it less accessible in comparison. However, this number is still very low, which makes this heuristic insignificant compared to the previous one.

Press/drag input. Some interfaces require continuous input to be provided while maintaining a pressed state of the click device which can be straining and tiring [39]. The press/release-based versions of the ArrowRing, CircleRing, and TargetAnchor interfaces all require dragging and are hence less accessible than their click-based counterparts. Even though the press/release version of the Fixed interface does not require dragging, it still requires holding the button down for longer durations.

Button size and arrangement. Fitt's law indicates that larger buttons that are nearby are easier and faster to reach with cursor movements, which holds true for users with mobility impairments [48], [49], [50], [43]. In the Fixed interface all buttons are arranged within a smaller region of the screen reducing the need for larger movements across the screen and the need to track the object as it moves, but the sizes of the buttons are relatively small. The circle region in CircleRing is easier to get to compared to the four smaller arrows around the ring in ArrowRing, making it more accessible.

Overall the most critical difference among the interfaces is between Fixed, which requires no continuous input, and others. We also expect the click-based versions of the interfaces to be more accessible than press/release-based ones.

IV. Study 1: Cursor-to-SE2 Control Interfaces

Our first study investigates the efficiency and usability of different 2D-to-SE2 interfaces. We aim to understand how quickly users can move a directional object on a plane, whose pose is specified in SE2, to a given target using the different interfaces. We also aim to characterize how the time to reach the target is impacted by the distance and size of the target, akin to Fitt's law [48] that specifies the relationship between the expected time for a cursor to reach a target 2D region.

[3]https://threejs.org/
[4]https://github.com/benfred/fmin

A. Study Design

We perform a between-groups study in which each participant only interacts with one type of interface. We include click-based transition versions of all five interfaces and the press/release-based transition versions of the first four interfaces (Fig. 3), resulting in nine conditions.

The task of moving an SE2 object to a target is varied systematically by changing the distance and size of the target. Both are separately varied along position and orientation dimensions. The starting pose of the object is kept at the center of the screen at 0^o. The target distance is sampled around three increasing values between zero and the distance to the edge of the screen; the actual position is randomized. The orientation of the target is sampled around three values from 0^o to 180^o; the sign of the orientation distance (+/-) is randomized. Size for (x, y) is varied by increasing the radius around the center of the target within which the object would be considered as having reached the target. Values are sampled around two points between the radius of the center point indicator and the radius of the orientation indicator. Size of the target (*i.e.,* how much the angle is allowed to deviate from the exact target direction) is sampled around two points between 0^o and 90^o. This results in 36 total combinations of sizes and distances in position and orientation of the target. Hence each participant completes 36 target reaching tasks, in randomized order. The visualization of example targets are shown in Fig. 2(b).

B. Procedure

Participants first watch an instructional video that explains the task of SE2 manipulation, introduces all elements of the interface, shows how to move and rotate the SE2 object, introduces flexible targets, demonstrates how targets can be reached using the interface, and gives an overview of the rest of the study. Next participants practice using the interface to reach randomly generated targets. They are allowed to move onto the tests after they have successfully reached at least five different targets. Then they start the actual interface tests. Before each test the participant presses a button that says "I'm ready", followed by a 3 second countdown. The trial starts at the end of the countdown, when the new target appears with the SE2 object reset to the center of the screen. It ends when the target is reached. After completing all trials, the participant respond to a questionnaire.

C. Measurements

We measure the efficiency of the interfaces based on task completion time and number of clicks per task trials. We perform statistical comparisons of efficiency metrics using a factorial model with 6 treatments: interface type (Fixed, ArrowRing, CircleRing, TargetAnchor, TargetRing), transition type (press/release, click), and the four task parameters (distance and size in (x, y) and θ). Subjective evaluation of the interfaces is based on the questionnaire, which involves the NASA TLX scale to assess workload and additional statements about perceived usability, learnability, efficiency, error-proneness, and accessibility with 5-point agree-disagree

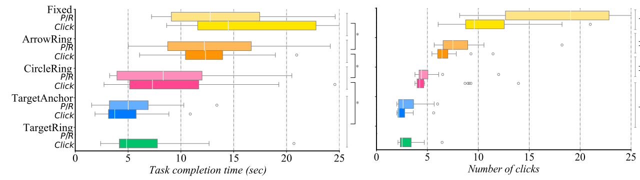

Fig. 5: Study 1: Average task completion time and number of clicks across tasks and participants for two versions (P/R and Click) of the five different 2D-to-SE2 interfaces.

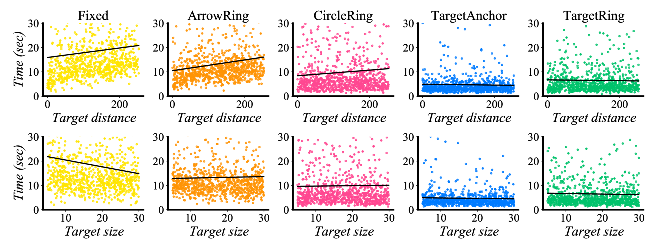

Fig. 6: Study 1: Distribution of task completion time across two task parameters: (top) Euclidean distance and (bottom) size of the target in (x, y) for the click-based transition versions of the five interfaces.

Likert scales. Open-ended questions ask participants to elaborate on their answers to the Likert-scale questions.

D. Findings

Our study was completed by 216 participants (24 in each condition) recruited over Amazon Mechanical Turk. Fig. 5 presents the average task completion times and number of clicks across the nine versions of the 2D-to-SE2 interfaces. Fig. 6 shows how task completion times are distributed across task parameters. We make several observations.

Target specification is more efficient than movement specification. Both TargetAnchor and TargetRing have significantly smaller task completion times compared to other interfaces, averaging around 6 seconds and 3 clicks to complete (Fig. 5). CircleRing was the most efficient of the movement specification interfaces, but still on average 3 seconds slower and needed at least 2 additional clicks to complete tasks. The standard deviations in completion time of the movement-based interfaces (Fixed, ArrowRing, CircleRing) were large, mainly due to outliers in individual tasks that took much longer than usual.

Impact of task difficulty. As expected, movement-based interfaces are least impacted by the increasing task difficulty, with no differences in completion time for easy (near and large) and hard (far and small) targets (Fig. 6). The Fixed interface was most impacted by task parameters, with more pronounced trends of increased task completion time with increased distance or decreased size. The effects of changing the target's distance (in position and orientation) was only significant between the closest and farthest regions for both number of clicks and completion times.

Press/release versus click makes minor difference. The interaction between interface and transition type (press/release versus click) is significant only for the Fixed interface with larger completion time but smaller number of clicks for click-based transitions. There were no other significant differences between the two versions of the interfaces.

No subjective differences. Surprisingly we found no significant differences among the interfaces across the different NASA TLX scales and the Likert-scale questions. We attribute this to the between-groups design of our study and the unfamilarity and abstractness of the task, where participants had no point of comparison for the interface they experienced. Their comments confirmed that they thought even the least efficient interfaces were reasonable for the task at hand, *e.g.,* one participant said "*It was easy enough..I'm not sure how it could be made more intuitive.*" for the Fixed (click) interface which was the least efficient among the nine.

V. Study 2: SE3 Control Interfaces

Study 1 allows us to understand how interface decisions impact efficiency at the basic interaction level, in isolation from the complexities of robot control interfaces. In Study 2 we aim to understand how those low level differences translate to the full control of a robot manipulator in an interface that incorporates the same 2D-to-SE2 strategies. To that end we evaluate the SE3 interface described in Sec. III-B in a user study.

A. Study Design, Procedure, Measurements

The study design and procedure are similar to that of Study 1. Instead of both variations of the five interfaces, we only included the variations that appeared to be more efficient in Study 1—hence we used press/release transitions for the movement-specifying interfaces and click transitions for the target-specifying interfaces. Rather than trying to get the end-effector to a specified target we adopt a more realistic task, where the goal is to move the end-effector to a pose from which an object could be grasped. Hence each task involves an object presented in front of the robot. There are five tasks total; the first two involve a cylindrical object to be grasped from the side like a water bottle, and the latter three involve a rectangular prism object resembling a remote control lying on a flat surface at different orientations about the gravity axis, to be grasped from the top. The five tasks are roughly in order of increasing difficulty, the first two only requiring translations, and the latter three requiring rotations around one or two axes. The order of tasks is therefore kept constant across participants. Measurements are also similar to Study 1, with the addition of number of view switches and time spent in each of the three orthogonal views. Statistical tests exclude the transition type factor.

B. Findings

Study 2 was completed by 60 participants (12 in each condition) recruited from Amazon Mechanical Turk. Fig. 7 shows the average task completion time and number of clicks used in each interface. Additionally, Fig. 8 shows participant responses to NASA TLX and Fig. 9 shows the time spent in different views. We make the following observations.

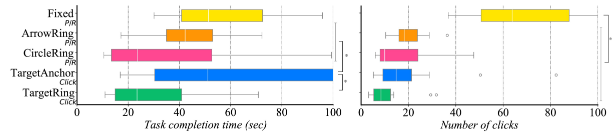

Fig. 7: Study 2: Average task completion time and number of clicks across tasks and participants for the five different SE3 interfaces (error bars denote standard deviation).

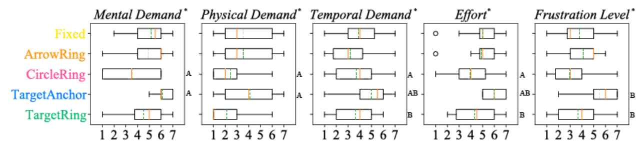

Fig. 8: Study 2: Participant responses to the NASA TLX questionnaire scales. The x-axis represents the 7-point scale, with "Frustration Level" reverse-coded. "*" represents significant differences between groups, letters specify which groups ($p < 0.05$).

Most and least efficient different from Study 1. CircleRing and TargetRing interfaces were the most efficient both in terms of task completion time and number of clicks. Unexpectedly, TargetAnchor, which was the overall most efficient interface in Study 1, was significantly less efficient than all other interfaces in Study 2. This was likely because the ghost end-effector visualization, which included both of the orthogonal axes, was less intuitive than having a single orientation indicator as in Study 1. One participant mentioned that "*the [SE2 object] was very sensitive*" and that it was "*difficult to attain an exact movement*".

The increased task completion time was not necessarily reflected in the number of clicks for TargetAnchor. Similarly, the Fixed interface which needed a significantly larger number of clicks (more than double) compared to all other interfaces, was not necessarily less efficient in terms of completion time.

Efficiency reflected in subjective ratings. Participants rated the TargetAnchor interface as having the highest mental demand, effort, and frustration level (Fig. 8). They also disagreed more that this interface was intuitive, easy to learn, and easy to recover from errors, and agreed more that it was error prone, compared to all other interfaces. Responses between TargetAnchor and CircleRing, and between TargetAnchor and TargetRing were significantly different ($p < 0.05$) in four and three of the five sections in NASA-TLX respectively, found using an HSD Tukey adjustment after One-Way ANOVA using the interfaces as different treatments. The differences are represented with same letters on different groups in Fig. 8.

Use of orthogonal views. Participants spent more of their time (50-60%) in the *top view* for Fixed, ArrowRing, and CircleRing interfaces across all tasks, while TargetAnchor and TargetRing had uniform distributions across views (Fig. 9). The top view might have been preferred because is was in the top-left of the screen and was selected by default. While participants often used only a subset of two views for individual tasks, the pair chosen for the task differed

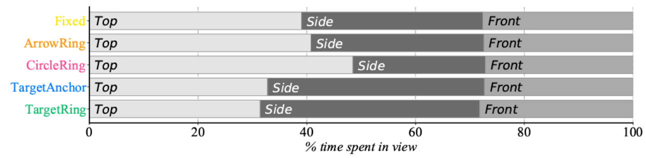

Fig. 9: Study 2: Percentage of time spent in each view (Top, Side, Front) for each interface.

from task to task depending on how the robot arm needed to move to grasp the object. While the number of view switches was similar across the different interfaces, the increasing task difficulty resulted in significantly more view switches and significantly higher task completion times.

VI. Discussion

Efficiency-accessibilty trade-off. Our work characterizes the impact of low-level interface choices on the efficiency of controlling an SE2 object with a 2D cursor and demonstrates how multiple 2D-to-SE2 interfaces can be combined through orthogonal views to control a robot. While Study 1 suggests that target-specifying interfaces are more efficient than movement-specifying ones, TargetAnchor was the least efficient in Study 2 and TargetRing was not significantly different from the movement-specifying interfaces. Study 1 indicates a clear trade-off between efficiency and accessibilty—the Fixed interface, which is most accessible due to not requiring precise pointing with the cursor, was also the least efficient and most impacted by task difficulty, but this difference in efficiency was reduced in Study 2. This makes the Fixed interface a great choice to balance efficiency and accessibility. For additional efficiency when continuous cursor input is not an issue, CircleRing for movement specification and TargetRing for target specification are the favorable options.

Limitations. The key metric in comparing the efficiency of different interfaces in our studies was task completion time—the time it takes to successfully move the visual representation of an SE2 object to a target. In study 2, we assumed that the robot's arm could instantaneously move to the specified target which is not true in practice. This assumption might have made the efficiency difference between target-specifying and movement specifying interfaces larger than it would be in practice since the robot arm will take longer to move to an arbitrary target than to a nearby pose specified with cursor movement.

Both of our studies had a between-participants design to minimize any interface ordering effects, due to the difficulty of unlearning an interface to start using a different one; as well as fatigue due to the large number of conditions and tasks included in the studies. The downside of a between participant study is that the subjective data is inconclusive in terms of which interface people would prefer independent of its efficiency. While there were no significant differences in subjective ratings in Study 1 due to the simplicity of the task, there were differences for almost all questions in Study 2, despite the between-study design.

While we presented a way to heuristically assess the accessibility of an interface, in the future we hope to empirically evaluate the accessibility of the different interfaces with people who have mobility limitations and use assistive devices to access a cursor.

References

[1] R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Trans. on Systems, Man, and Cybernetics (SMC), Part C: Applications and Reviews*, vol. 34, no. 2, pp. 138–153, 2004.

[2] T. Fong, R. Berka, M. Bualat, M. Diftler, M. Micire, D. Mittman, V. SunSpiral, and C. Provencher, "The human exploration telerobotics project," in *Global Space Exploration Conference*, 2012.

[3] M. J. Lum, D. C. Friedman, G. Sankaranarayanan, H. King, K. Fodero, R. Leuschke, B. Hannaford, J. Rosen, and M. N. Sinanan, "The raven: Design and validation of a telesurgery system," *The International Journal of Robotics Research*, vol. 28, no. 9, pp. 1183–1197, 2009.

[4] T. B. Sheridan, "Teleoperation, telerobotics and telepresence: A progress report," *Control Engineering Practice*, vol. 3, no. 2, pp. 205–214, 1995.

[5] J. Y. Chen, E. C. Haas, and M. J. Barnes, "Human performance issues and user interface design for teleoperated robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 6, pp. 1231–1245, 2007.

[6] D. Gossow, A. Leeper, D. Hershberger, and M. Ciocarlie, "Interactive markers: 3-d user interfaces for ros applications [ros topics]," *IEEE Robotics & Automation Magazine*, vol. 4, no. 18, pp. 14–15, 2011.

[7] M. Ciocarlie, K. Hsiao, A. Leeper, and D. Gossow, "Mobile manipulation through an assistive home robot," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 5313–5320.

[8] T. L. Chen, M. Ciocarlie, S. Cousins, P. M. Grice, K. Hawkins, K. Hsiao, C. C. Kemp, C.-H. King, D. A. Lazewatsky, H. Nguyen *et al.*, "Robots for humanity: A case study in assistive mobile manipulation," 2013.

[9] N. Alunni, C. Phillips-Grafftin, H. B. Suay, D. Lofaro, D. Berenson, S. Chernova, R. W. Lindeman, and P. Oh, "Toward a user-guided manipulation framework for high-dof robots with limited communication," in *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[10] A. Stentz, H. Herman, A. Kelly, E. Meyhofer, G. Haynes, D. Stager, B. Zajac, J. Bagnell, J. Brindza, and C. Dellin, "Chimp, the cmu highly intelligent mobile platform," *Journal of Field Robotics*, vol. 32, no. 2, pp. 209–228, 2015.

[11] H. Yanco, A. Norton, W. Ober, D. Shane, A. Skinner, and J. Vice, "Analysis of human-robot interaction at the darpa robotics challenge trials," *Journal of Field Robotics, Special Issue on DARPA Robotics Challenge*, 2015.

[12] J. Mainprice, C. Phillips-Grafflin, H. Suay, N. Alunni, D. Lofaro, D. Berenson, S. Chernova, R. W. Lindeman, and P. Oh, "From autonomy to cooperative traded control of humanoid manipulation tasks with unreliable communication: System design and lessons learned," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2014, pp. 3767–3774.

[13] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2012, pp. 1–8.

[14] D. Kent, C. Saldanha, and S. Chernova, "A Comparison of Remote Robot Teleoperation Interfaces for General Object Manipulation," in *ACM Intl. Conference on Human Robot Interaction (HRI)*, 2017.

[15] D. Buongiorno, D. Chiaradia, S. Marcheschi, M. Solazzi, and A. Frisoli, "Multi-dofs exoskeleton-based bilateral teleoperation with the time-domain passivity approach," *Robotica*, vol. 37, no. 9, p. 1641–1662, 2019.

[16] T. Klamt, D. Rodriguez, L. Baccelliere, X. Chen, D. Chiaradia, T. Cichon, M. Gabardi, P. Guria, K. Holmquist, M. Kamedula *et al.*, "Flexible disaster response of tomorrow: Final presentation and evaluation of the centauro system," *IEEE robotics & automation magazine*, vol. 26, no. 4, pp. 59–72, 2019.

[17] L. Guanyang, G. Xuda, L. Lingzhi, and W. Yan, "Haptic based teleoperation with master-slave motion mapping and haptic rendering for space exploration," *Chinese Journal of Aeronautics*, vol. 32, no. 3, pp. 723–736, 2019.

[18] M. A. Goodrich, D. R. Olsen, J. W. Crandall, and T. J. Palmer, "Experiments in adjustable autonomy," in *Proceedings of IJCAI Workshop on autonomy, delegation and control: interacting with intelligent agents*. Seattle, WA, 2001, pp. 1624–1629.

[19] J. M. Beer and L. Takayama, "Mobile remote presence systems for older adults: acceptance, benefits, and concerns," in *6th International Conference on Human-Robot Interaction*. ACM, 2011, pp. 19–26.

[20] J. Abbott, P. Marayong, and A. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *Robotics research*. Springer, 2007, pp. 49–64.

[21] F. Ryden, H. Chizeck, S. Kosari, H. King, and B. Hannaford, "Using kinect and a haptic interface for implementation of real-time virtual fixtures," in *RSS Workshop on RGB-D*, 2011.

[22] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 1139–1144.

[23] M. Li and A. M. Okamura, "Recognition of operator motions for real-time assistance using virtual fixtures," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on*. IEEE, 2003, pp. 125–131.

[24] A. D. Dragan, S. S. Srinivasa, and K. C. Lee, "Teleoperation with intelligent and customizable interfaces," *Journal of Human-Robot Interaction*, vol. 2, no. 2, pp. 33–57, 2013.

[25] K. Hauser, "Recognition, prediction, and planning for assisted teleoperation of freeform tasks," *Autonomous Robots*, vol. 35, no. 4, pp. 241–254, 2013.

[26] K. Khokar, R. Alqasemi, S. Sarkar, K. Reed, and R. Dubey, "A novel telerobotic method for human-in-the-loop assisted grasping based on intention recognition," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 4762–4769.

[27] H. Admoni and S. Srinivasa, "Predicting user intent through eye gaze for shared autonomy," in *2016 AAAI Fall Symposium Series*, 2016.

[28] M. Gao, J. Oberländer, T. Schamm, and J. M. Zöllner, "Contextual task-aware shared autonomy for assistive mobile robot teleoperation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3311–3318.

[29] 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.

[30] L. V. Herlant, R. M. Holladay, and S. S. Srinivasa, "Assistive teleoperation of robot arms via automatic time-optimal mode switching," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2016, pp. 35–42.

[31] D. E. Gopinath and B. D. Argall, "Active intent disambiguation for shared control robots," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2020.

[32] P. M. Grice and C. C. Kemp, "In-home and remote use of robotic body surrogates by people with profound motor deficits," *PloS one*, vol. 14, no. 3, p. e0212904, 2019.

[33] P. M. Grice, M. D. Killpack, A. Jain, S. Vaish, J. Hawke, and C. C. Kemp, "Whole-arm tactile sensing for beneficial and acceptable contact during robotic assistance," in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2013, pp. 1–8.

[34] P. M. Grice and C. C. Kemp, "Assistive mobile manipulation: Designing for operators with motor impairments," in *RSS 2016 Workshop on Socially and Physically Assistive Robotics for Humanity*, 2016.

[35] M. Mast, M. Burmester, K. Kruger, S. Fatikow, G. Arbeiter, B. Graf, G. Kronreif, L. Pigini, D. Facal, and R. Qiu, "User-centered design of a dynamic-autonomy remote interaction concept for manipulation-capable robots to assist elderly people in the home," *Journal of Human-Robot Interaction*, 2012.

[36] M. Mast, M. Španěl, G. Arbeiter, V. Štancl, Z. Materna, F. Weisshardt, M. Burmester, P. Smrž, and B. Graf, "Teleoperation of Domestic Service Robots: Effects of Global 3D Environment Maps in the User Interface on Operators' Cognitive and Performance Metrics," in *Social Robotics*. Springer, 2013, pp. 392–401.

[37] M. Mast, Z. Materna, M. Španěl, F. Weisshardt, G. Arbeiter, M. Burmester, P. Smrž, and B. Graf, "Semi-autonomous domestic service robots: Evaluation of a user interface for remote manipulation and navigation with focus on effects of stereoscopic display," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 183–202, 2015. [Online]. Available: http://dx.doi.org/10.1007/s12369-014-0266-7

[38] K. Tsui, H. Yanco, D. Kontak, and L. Beliveau, "Development and evaluation of a flexible interface for a wheelchair mounted robotic arm," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, 2008, pp. 105–112.

[39] K. Krishnaswamy, T. Adamson, M. Cakmak, and T. Oates, "Multi-perspective, multimodal, and machine learning for accessible robotic web interface," in *Proceedings of the 2018 Rehabilitation Engineering and Assistive Technology Society of North America Annual Conference*. RESNA, 2018, pp. 1–4.

[40] R. Hazlett-Knudsen, M. A. Smith, and A. Behal, "Knowledge based design of user interface for operating an assistive robot," in *International Conference on Human Centered Design*. Springer, 2011, pp. 304–312.

[41] K. M. Tsui, J. M. Dalphond, D. J. Brooks, M. S. Medvedev, E. McCann, J. Allspaw, D. Kontak, and H. A. Yanco, "Accessible human-robot interaction for telepresence robots: A case study," *Paladyn, Journal of Behavioral Robotics*, vol. 1, no. open-issue, 2015.

[42] J. O. Wobbrock and K. Z. Gajos, "A comparison of area pointing and goal crossing for people with and without motor impairments," in *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, 2007, pp. 3–10.

[43] J. O. Wobbrock, K. Shinohara, and A. Jansen, "The effects of task dimensionality, endpoint deviation, throughput calculation, and experiment design on pointing measures and models," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 1639–1648.

[44] S. Carter, A. Hurst, J. Mankoff, and J. Li, "Dynamically adapting guis to diverse input devices," in *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, 2006, pp. 63–70.

[45] K. Z. Gajos, J. O. Wobbrock, and D. S. Weld, "Automatically generating user interfaces adapted to users' motor and vision capabilities," in *Proceedings of the 20th annual ACM symposium on User interface software and technology*, 2007, pp. 231–240.

[46] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[47] S. Jain and B. Argall, "Robot learning to switch control modes for assistive teleoperation," in *RSS 2016 Workshop on Planning for Human-Robot Interaction: Shared Autonomy and Collaborative Robotics*, 2016.

[48] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement." *Journal of experimental psychology*, vol. 47, no. 6, p. 381, 1954.

[49] A. Sears, "Layout appropriateness: A metric for evaluating user interface widget layout," *IEEE Transactions on Software Engineering*, vol. 19, no. 7, pp. 707–719, 1993.

[50] I. S. MacKenzie and W. Buxton, "Extending fitts' law to two-dimensional tasks," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1992, pp. 219–226.