Intentional User Adaptation to Shared Control Assistance

Reuben M. Aronson
reuben.aronson@tufts.edu
Tufts University
Medford, Massachusetts, USA

Elaine Schaertl Short
elaine.short@tufts.edu
Tufts University
Medford, Massachusetts, USA

ABSTRACT

Shared control approaches to robot assistance, which predict a user's goal based on their control input and provide autonomous assistance towards the predicted goal, typically assume that user behavior remains the same despite the presence of the assistance and rely on this assumption to infer user goals. However, people operating assisted systems continuously observe the robot behaving differently from their expectations, which may lead them to adapt their control behavior to better achieve their desired outcomes. In this paper, we show that users both change their control behavior when assistance is added and describe these changes as responses to the new system dynamics. In a computer-based bubble popping study, participants report changing their strategies with different levels of assistance, and analysis of their actual control input validates this change. In an in-the-wild robot study, participants teleoperated a robot to pick up a cup despite the presence of "assistance" that drives the system away from the true goals of the task. Participants can overcome the "assistance" and reach the goal, which requires them to correct for the novel system dynamics. These results motivate further research in user-centered design and evaluation of assistive systems that treat the user as intentional.

CCS CONCEPTS

• **Human-centered computing**; • **Computer systems organization → Robotic control**;

KEYWORDS

shared control, shared autonomy, assistive robotics, teleoperation

ACM Reference Format:
Reuben M. Aronson and Elaine Schaertl Short. 2024. Intentional User Adaptation to Shared Control Assistance. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24), March 11–14, 2024, Boulder, CO, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3610977.3634953

1 INTRODUCTION

Shared control systems enable users to more easily and effectively teleoperate robots to perform tasks. These systems generally assume knowledge of autonomous strategies to achieve pre-specified tasks that users may want to perform. The assistance interposes itself between the user input and the underlying system and adjusts the user's control signal towards the autonomous strategy.

This work is licensed under a Creative Commons Attribution-NoDerivs International 4.0 License.

HRI '24, March 11–14, 2024, Boulder, CO, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0322-5/24/03.
https://doi.org/10.1145/3610977.3634953

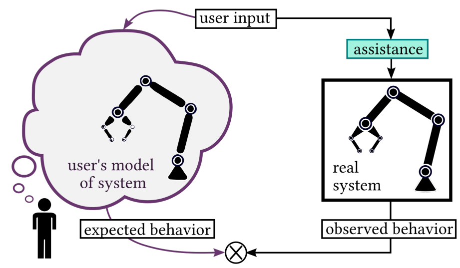

Figure 1: Shared control systems work by capturing user input and adjusting it to be more effective. In this framing, users observe system behavior that conflicts with their expectations from their internal model. The stationarity assumption requires that users ignore this conflict.

To provide assistance for multiple different tasks, shared control systems typically infer which of the known tasks the user is trying to perform from user behavior. For this step, it is common to assume that user input will remain *stationary* regardless of the presence of the assistance: that is, that users will not change how they provide input when the assistance is added. Under this assumption, goal inference is performed by comparing the user's observed control input with task strategies for the underlying system. Furthermore, the assumption bypasses the circularity that arises from trying to adapt the assistance behavior to a user who is, in turn, adapting to the changing assistance. Shared control systems increase success rate and decrease required input in a variety of tasks [5, 20].

However, this assumption of stationarity is in conflict with users' actual experience of the system. As people interact with the robot, they continuously observe its actual behavior and how it differs from their internal models of how the system would behave (Fig. 1). In fact, the psychological theory of motor learning [18, 26] proposes that it is by observing this feedback that people learn to control systems in general. We theorize that as users' experience with the system increases, they intentionally adapt their behavior to compensate for the effect of the assistance system. According to this theory, improvements in task metrics are not only due to the assistance itself, but are also driven by users taking advantage of the changes in system dynamics that result from adding assistance.

In this paper, we show that user input behavior changes when goal-directed assistance is added. Furthermore, users describe themselves as modeling the new system behavior and strategizing about how to adapt their inputs. In support, we present two user studies in which participants controlled a robot in the presence of shared control-based assistance. First, we describe a computer-based study in which participants controlled a ball to pop bubbles. We show that participants explicitly identify changes in the assistance behavior between conditions and strategize about how to adapt their control strategy to maximize their scores. Second, we present a public-space user study in which participants controlled a robot to pick up a cup despite the presence of "assistance" that directed them towards incorrect goals. Participants still succeeded on the task despite the changes in dynamics introduced by the assistance. These results demonstrate that users consistently and intentionally change their behavior in the presence of assistance.

The idea that users adapt to assistance introduces a new, user-centered paradigm for designing assistance systems. In this paradigm, the assistance does not simply act as a proxy of the user's intent: it instead becomes part of the system that the user controls. Then, the effect of the assistance is to add an additional dynamic system with its own state (e.g., probability distribution over possible goals) and dynamics (e.g., Bayesian updates based on the user's input). Considering an active user introduces new design criteria, such as valuing transparency in goal prediction or treating the control process as collaborative, that do not apply when designing assistance to autonomously execute a user's intent. Showing that users do, in fact, react to assistance systems justifies this user-centered perspective on assistive teleoperation.

2 BACKGROUND

2.1 Related work

Shared control is a paradigm for human-robot interaction in which both the user and an assistive algorithm simultaneously provide control input to a robot. This approach has improved success rates over direct user control in a wide variety of tasks; a general overview is given in Cimolino and Graham [5]. These systems are further surveyed in Losey et al. [20], which divides shared control assistance into an *intent detection* phase, during which the system infers the user's likely intended goal or task from their input behavior or other passive signals, and an *arbitration* phase, during which the user's input signal is combined with automatically generated assistance. Goal inference is often performed using Bayesian inference [13] by treating user inputs as observations conditioned on the user's intended goal and the underlying system dynamics; typical user models include Boltzmann-rational behavior [3] or maximum entropy inverse reinforcement learning [30].

Numerous works have validated the effectiveness of this approach for robot teleoperation, especially in assistive applications. The method helps with tasks such as driving a wheelchair [4, 6, 11, 12] or controlling a robot arm [2, 7, 10, 14, 15, 17, 28], especially in the presence of low-dimensional or noisy input devices [8, 22]. These methods vary in how they model tasks, detect user intent, and arbitrate between user input and assistance behavior, but the intent detection and arbitration pipeline is consistent.

While research in teleoperation has generally focused on algorithmic developments, Rea and Seo [27] calls for a focus on the user experience in teleoperation, and several works consider this perspective for shared control systems. Nikolaidis et al. [24] and Parekh et al. [25] model the user's willingness to adapt to the robot's behavior so that the assistance system can converge on a collaborative strategy. Other strategies [9, 16] modify the robot behavior so that the user's control inputs are likely to be informative about their goals earlier in the task. Jun Jeon et al. [17] adds an entropy objective to ensure that their goal inference algorithm does not trap users into a single task. Zurek et al. [31] models how well user behavior aligns with existing tasks and treats low alignment as a signal to add new task models to the system. Our work showing that users adapt to the presence of assistance contributes to this developing focus on user-centered analysis in shared control.

2.2 Overview of the assistance algorithm

For all assistance behavior used in the studies, we adapt the shared autonomy algorithm given in Javdani et al. [14], which we summarize here. This assistance behavior consists of three stages: goal inference, assistance generation, and arbitration.

2.2.1 Goal inference. We assume that the robot control behavior is modeled as a Markov decision process (MDP) consisting of states S, actions A, and transition T. We assume that the user's goal is a single element of a pre-specified set of goals G, and each goal g defines a reward function $r_g(x, a)$ which is optimized by action value function $Q_g(s, a)$ and policy $\pi_g(s)$. The goal of this first step is to find a probability distribution $p(g \in G)$ that represents the chance, based on observed evidence, that the user's true goal is g.

To determine this goal probability, we treat the user's input action $u_t \subset A$ as an observation of their goal provided in state $s_t \in S$ (which we drop for ease of notation). This formulation enables us to use Bayesian inference to determine the goal probability over time [13]. From a known prior $p(g)$, the goal probability distribution is updated at each time step as $p(g|u_0, \cdots, u_t) \propto p(u_t|g)p(g|u_0, \cdots, u_{t-1})$. For the observation probability $p(u_t|g)$, we assume that user follows Boltzmann rationality [3] $p(u|g) \propto \exp \beta Q_g(u)$, where $Q_g(u)$ is given above and β sets the sensitivity of the goal inference. For all tasks, the MDP represents x-y translation in a plane. States and goal locations are given as points $s = (s_x, s_y)$ and $g = (g_x, g_y)$. Actions A are bounded velocity vectors in the plane, $\{(a_x, a_y) \in \mathbb{R}^2 : |(a_x, a_y)| \leq 1\}$, and transitions are defined by vector translation: $T(s, a) = s + a$.

2.2.2 Assistance generation. Given a goal distribution $p(g)$ at state s_t, the optimal assistance command was calculated as $a_t = E_{p(g)} a^*_g$, i.e., the expected value of the optimal actions a^*_g for each goal g over the goal distribution. a_t is then normalized to unit length. This calculation follows a policy blending approach [7], which is sufficient for these simple tasks.

2.2.3 Arbitration. Once the assistance command a_t has been generated, the final robot action a_{appl} is generated by an arbitration step: $a_{\text{appl}} = \gamma u_t + (1 - \gamma) a_t$ [23]. The blending factor γ indicates the relative contribution of the user command and the assistance command to the resulting motion; $\gamma = 1$ is direct control, and $\gamma = 0$ is *indirect* control in which the robot uses the input signal only for goal updates and has full control over the output behavior.

3 BUBBLE POPPING STUDY

The *intentional* user model theorizes that participants change their behavior in the presence of assistance by adapting to the new dynamics of the system. In this first study, we investigate this theory in a simple, 2-D bubble popping task, performed in two different assistance conditions and with direct control. Participants report

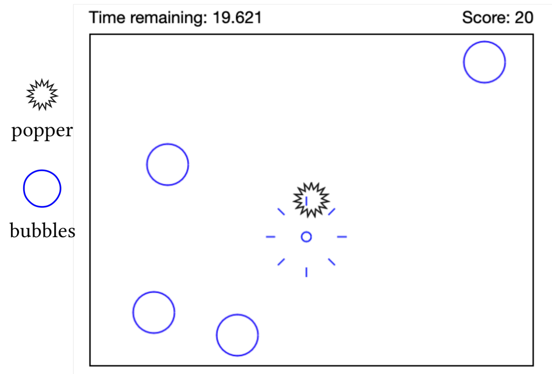

Figure 2: Overview of bubble popping game. Participants moved the popper (spiked ball) to pop the bubbles (blue circles) as they moved around the screen. The game also showed the remaining time (top left) and current score (top right, with each bubble worth 10 points).

that they notice the change in system dynamics change and strategize about how to adapt their control behavior, which we validate by finding quantitative differences in input between conditions.

3.1 Interaction

Participants played a game in which they moved a spiked ball (the "popper") to pop bubbles that appeared from the edge of the screen and moved around randomly (see Fig. 2). Participants commanded linear and diagonal directions of input using the arrow keys on the keyboard. The popper was designed to be very sensitive to input, moving at 1000 pixels per seconds in a 640×480 window. This rapid speed made perfect control difficult without assistance, thus emulating the difficulty of direct control of a more complex, robotic system and motivating a preference for assisted control. At most six bubbles appeared at a time. Each emerged near an edge of the window with a random initial velocity biased towards the center of the window and moved with a randomly generated acceleration. The study was approved by the Tufta Institutional Review Board.

First, participants gave consent, read instructions, and played one practice game at a slower speed (400 pixels per second) and a shorter duration (30 seconds). Participants repeated the practice session as many times as they wanted. Next, participants provided demographic information and then played three 60-second sessions of the game at full speed, once per condition. After each session, participants filled out a questionnaire, and answered a final questionnaire after all sessions were complete.

For each full session, participants controlled the popper in a different assistance condition: *direct control*, *assisted*, and *guided autonomy*. Conditions were fully counterbalanced across participants in a within-participants study design. To increase consistency, bubbles within each condition followed the same sequence of trajectories, though the bubbles' times of appearance and popping depended on user behavior. To prevent participants from noticing the repeated trajectories, a $\pm 90°$ rotational transformation was applied to the trajectories in each condition, similar to Leyzberg et al. [19]. Transformations were balanced across conditions.

3.2 Conditions

To measure how people's inputs varied with the level of assistance, we applied three different levels of assistance in the three conditions. The assistance calculation is described in Sec. 2.2, with details below. For each condition, the assistance algorithm set the blending parameter γ between the user input and the assistance input.

Direct control. Participants' inputs were provided unaltered to the popper, corresponding to $\gamma = 1$. Due to the input limitations, velocities were limited to horizontal, vertical, and diagonal vectors.

Assistance. The final applied velocity was given by the mean of the assistance and the user's direct input, corresponding to $\gamma = 0.5$.

Guided autonomy. The assistance command was used directly ($\gamma = 0$); the user's input only affected the behavior of the bubbles through changing the system's inferred goals.

3.3 Assistance generation

In this task, we set $r_g(s,a) = -1$ if $|g_t - s| \leq D$ and 0 otherwise, with D a distance parameter and g_t representing the location of bubble g at time t. This sparse reward function drives the popper in the direction of the bubble at a constant speed and is analytically solvable in this simple environment. Therefore, $Q_g(s,a) = a^T (g_t \hat{-} s)$, with $(g_t \hat{-} s)$ the unit vector in the direction of g_t from s, normalized to 0 if $|g_t - s| < D$. The corresponding policy is to assign $\pi_g(s) = (g_t \hat{-} s)$. The distance threshold $D = 25$px was chosen to match the radius of the popper.

This task setting incorporated bubbles (goals) that appeared and disappeared, which requires an adaptation of the goal inference step to accommodate. We treat an appearance or disappearance event as occurring between user inputs, so we can treat these events separately from the observation updates. Given a current goal set G_t, when a new goal g' appeared, it was assigned uniform probability $p(g') = \frac{1}{|G_t|+1}$. Then, the probability mass was removed from the remaining goals as $p'(g) = \frac{|G_t|}{|G_t|+1} p(g)$. To remove a goal $g^{\times}$, the probability $p(g^{\times})$ was redistributed among the other goals by normalization, so $p'(g) = \frac{p(g)}{\sum_{g' \in G \setminus \{g^{\times}\}} p(g')}$.

3.4 Hypotheses

H1. The presence of assistance will increase number of bubbles popped and decrease user input. This finding shows that the assistance in this domain has similar benefits as are generally found in studies of shared control assistance.

H2. Users' reported strategies will vary between levels of assistance. H2 shows that the users must be aware of the differences in the system caused by the changes in assistance level.

In addition, we use the qualitative results to conduct an exploratory analysis of the quantitative data. This analysis serves to validate that people's reported strategies actually led to measurable differences in their input behavior.

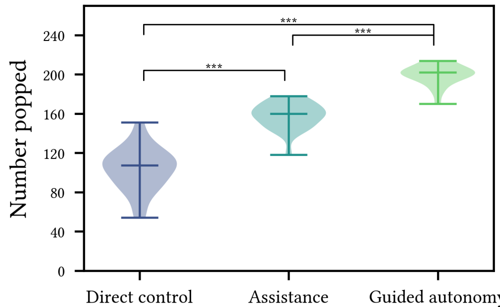

Figure 3: Number of bubbles popped per condition.

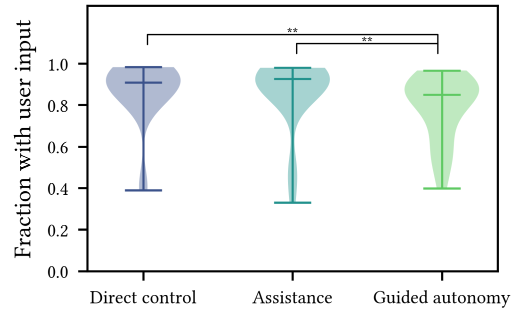

Figure 4: Fraction of the trial in which user input was nonzero per condition.

3.5 Results

3.5.1 Participants. Participants were solicited via word of mouth and hallway recruitment. Participants used their own laptops or a experimenter-provided laptop to navigate to a study website hosted on a research server to provide consent and perform the interaction. The entire process took about 15 minutes, and each participant received a $5 Amazon gift card.

The study had a total of **N**=24 participants. Of these, 19 were male, 5 were female, and 0 other. 21 of the participants were aged 18-24, and 3 aged 25-34. There were 21 students and 3 other professionals. Of the students, majors or areas of study included 9 computer science, 6 mechanical engineering, 4 math, and 2 other related technical fields. 2 participants reported playing video games daily; 7 reported weekly; 5 monthly; 7 rarely, and 3 never.

3.5.2 Score and Effort Varied By Condition. First, we analyze the effect of the assistance on study performance. Typically (e.g in [14]), it is found that increased levels of assistance lead to better success rates with less user effort. To evaluate the usefulness of our assistance behavior and to show that this task is representative of assisted teleoperation, we perform the same analyses here.

We find that increased levels of assistance clearly led to increased numbers of bubbles popped within the time period. The data is shown in Fig. 3. Using a Tukey HSD test, the overall differences were found to be significant ($F(2, 69) = 218.6, p < 10^{-29}$), and all pairwise post-hoc found significance at $p < 10^{-12}$. Similarly, we find that the *guided autonomy* condition requires less user input than either of the other conditions with less autonomy, shown in Fig. 4. Fraction with user input is determined by dividing the total duration of time stamps during which a control key was pressed by the total trial time; since the strict upper bound of 1 introduces a nonlinearity, nonparameteric statistics were used. A Friedman test shows a significant difference between conditions ($\chi^2(2) = 13.0, p < 0.002$); post-hoc comparisons with a paired Wilcoxon U-test and Holm-Bonferroni correction show significance between *direct control* and *guided autonomy* ($U = 49.0, p_{\text{corr}} < 0.007$) and *assistance* and *guided autonomy* ($U = 35.0, p_{\text{corr}} < 0.002$).

Together, these results validate that the assistance improves task performance and decreases required user input, in line with prior results on shared control and supporting H1. Reproducing these performance results suggests that user behavior found here may generalize to other assisted tasks.

3.5.3 Self-Reported Strategies Varied By Condition. Next, we consider whether participants used different control strategies in different conditions. If the control strategies differ, participants must be reacting to the assistance behavior that they perceive and adapting the behavior appropriately.

To understand if people explicitly reason about their strategies for each condition, we asked users three open-response questions after each trial:

(1) How much do you feel the robot's ability to complete the task depended on your input?
(2) What strategies did you use when controlling this system?
(3) Did you change your strategy for using this system compared previous one(s)? If so, how? (Skipped for the first trial.)

To evaluate these responses, we used open coding to identify 7 themes. One rater coded the data and built a codebook; a second rater re-coded the data using the codebook. Any disagreements were resolved through discussion. Number of responses coded with each code and subdivided by condition are reported in Tab. 1. Quotes are labeled with participant ID and condition.

Direct control. Strategies in the direct control condition generally consisted of ad-hoc, *specific strategies*. These strategies were independent of the robot's behavior, for example, "going along diagonals" (P8-DC), "lining up multiple bubbles" (P14-DC), or aim for the bubbles "along the edges" (P13-DC). These ad-hoc strategies were described as effective, though the features they relied upon — such as finding "patterns in where the bubbles appeared" (P12-A) — may not actually exist in the underlying system. Similarly, for the direct control condition, participants most strongly emphasized the importance of their *active control* behavior in accomplishing the task. They reported that "the task was entirely dependent upon [their] input" (P9-DC) or emphasized that the system "depended on [their] input more than" in other conditions (P4-A). Participants considered themselves the primary agent in controlling the task in the direct control condition, and came up with various different strategies to approach the problem of control.

Assistance. In the assistance condition, strategies were somewhat more consistent than in the direct control case, and the focused on providing high-level control to the robot. Many participants reported *targeting clusters* of bubbles together, described as "go[ing] in the general direction of a group of bubbles" (P6-A) or "huddl[ing]

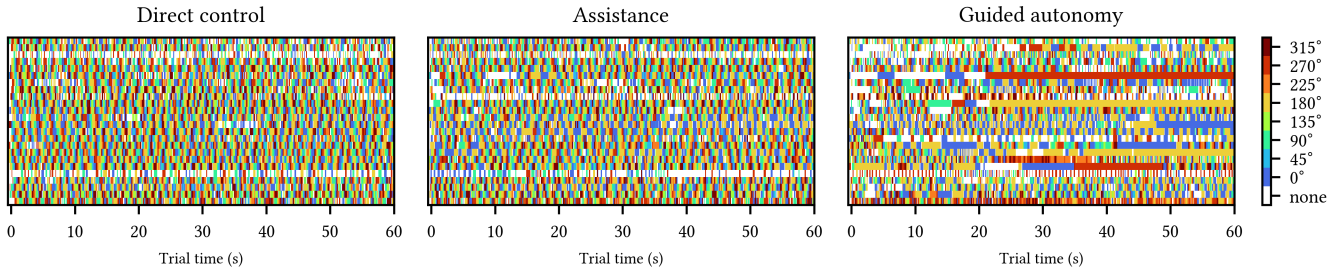

Figure 5: Visualization of all control input values provided by participants over the course of each game. Color indicates the direction of control input; the keyboard-based input scheme ensured that all inputs were axis-aligned or diagonal.

Code	Direct control	Assistance	Guided auton.
specific strategy	**16**	7	6
active control	**15**	10	4
target clusters	7	**11**	0
general direction	3	**10**	5
follow robot	2	7	**18**
random control	1	0	**10**
minimal input	0	2	**8**

Table 1: Counts of participant responses that were coded with each code, separated by condition, out of 24 total participants.

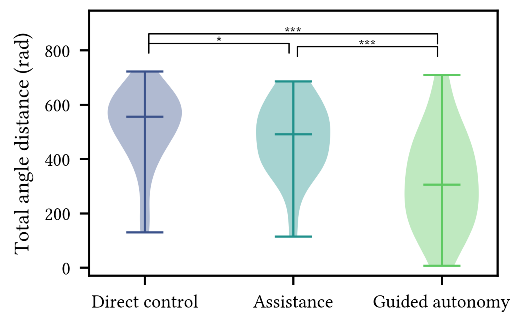

Figure 6: Total angular distance traveled by the control input during each game. Participants who changed their control input direction rapidly covered larger angular distances, whereas participants who provided minimal input or who rarely changed input direction had smaller total distances. Starting or stopping control incur a distance of $\pi/4$, the same as transitioning between adjacent control directions.

towards clusters" (P1-A). This group focused their control effort in directing the *general direction* of the popper. Participants noted that "if [they] aimed at a general direction, the robot did the fine tuning" (P7-A), and put "less emphasis on the more precise movement and just moving it in the general direction" (P17-A). Participants noticed that the assistance was particularly effective at performing fast, localized motion so they "stopped caring about accuracy since the system took care of it for me" (P11-A). This change made the game "easier and more relaxing" (P23-A). Input in the assisted control condition was both more consistent between participants than in the direct control condition (using fewer ad-hoc strategies) and tended to show an emergent hierarchical control behavior, with the participant "aiming at groups of bubbles and letting it do its thing on its own once [they] got close enough" (P20-A). This emergence of a distinct strategy indicates that users are responding to the presence of assistance and adjusting their behavior to match.

Guided autonomy. The guided autonomy condition was primarily distinguished by participants emphasizing their lack of control over the popper and describing their struggles to find strategies that improve system performance. Participants described the system behavior as *following the popper*, claiming that "[i]t would completely disregard" their input (P13-GA) and wondering if "the robot might just be doing it all by itself" (P4-GA). Some participants thought that providing any *random input* helped, saying that "[i]t just found the circles on its own as long as I was pressing buttons" (P11-GA) and that they "didn't really even think of which key I was even pressing" (P2-GA). Participants explicitly rejected strategies developed for other conditions: "Initially, I tried to use my old strategy but then I realized that I was not in control" (P17-GA);"I just tried to aim like last time but I devolved to mashing the arrow keys and it got everything" (P5-GA). Other participants provided *minimal input* to the system, saying they "basically did almost nothing except maybe press a button every once in a while" (P8-GA) or "allowed the robot to play the game until it was too far away from the balls to make a decision and then [they] sent it in the right direction" (P9-GA). Participants noticed that the system was less responsive to their input and developed new, often arbitrary, strategies in the hope that their inputs would improve the system performance.

Overall, participants report that they can identify differences in the system behavior in different assistance conditions and that they change their strategies to match, supporting **H2**.

3.5.4 Input Behavior Matched Reported Strategies. In their responses, participants claim to notice changes in system behavior and develop new strategies in response. Here, we show that these changes in strategy corresponded to measurable changes in input behavior.

We visualize the input behavior of all participants in all conditions in Fig. 5. General condition-dependent trends appear in the visualization. Most notable, the guided autonomy condition is characterized by substantial periods of constant input behavior, matching the *random control* strategies asserted by participants. The difference in input behavior between direct control and assistance is more subtle, but direct control seems to show more rapid changes in control direction and fewer periods of no control applied.

We can further characterize these differences by comparing how much participants changed their control input direction within each trial. From the qualitative results above, participants in the direct control condition believed the score depended strongly on their own input behavior and developed strategies to maximize their score; we might, therefore, expect to observe many rapid changes in input direction to precisely target bubbles. In the assistance condition, participants reported using high-level, general control behavior and leaving the fine motion to the assistance, which might lead to less overall rate of change in the direction of input. The guided autonomy condition led to participants controlling randomly by selecting a single key to hold down or providing minimal input, which should lead to the least total change in input direction.

Indeed, we find in Fig. 6 that the total distance traveled in angle of input control direction varied significantly by condition. Using a repeated measures ANOVA, we find a significant main effect of condition ($F(2, 46) = 19.44, p < 10^{-6}$), with post-hoc significance found for all pairs of conditions using a paired t-test with Holm-Bonferroni correction: direct control and assistance ($t(23) = 3.12, p_{\text{corr}} < 0.02$), direct control and guided autonomy ($t(23) = 3.51, p_{\text{corr}} < 0.001$), and assistance and guided autonomy ($t(23) = 4.50, p_{\text{corr}} < 0.0002$). This exploratory analysis suggests that the reported strategies were not merely a rationalization of identical behavior. Instead, the different strategies reported by participants led to measurably different control input behavior, further validating that participants indeed responded differently to different assistance behavior.

4 "ASSISTED" GRASPING STUDY

To understand people's adaptation to shared control and validate that our results hold in interactions with a physical robot, we conducted a public-space study with a robot arm in a pickup task with shared control. Unlike in the previous study, however, the "assistance" was tuned to be effectively adversarial: the assistive algorithm did not know about the the user's actual goal. If people were not aware of the "assistance" and operated the robot as if in direct control, it would be impossible to complete the task: the "assistance" would drive the robot further from the goals, and users would not be able to compensate for the changes. However, we find that users succeed in the task despite the presence of the adversarial dynamics, which is only possible because users do perform this compensatory adjustment. Furthermore, users explicitly strategize about how to adapt their control behavior to these new dynamics (as in the previous study) and show an increase in performance with more experience with the robot, which further demonstrates the presence of this reactive adjustment.

4.1 Task

Data collection for the study was performed in a public space over a single day. The robot was set up on a table in a public walkway and passersby were invited to participate. Participants controlled a Kinova Gen3 Lite robot arm to pick up one of two candy-filled cups on a table. The robot started above the cups with its gripper pointed downward. Participants controlled the x-y motion of the robot along with a *conflicting* "assistance" system. When participants pressed a button on the controller, the robot transitioned to the second stage and began moving steadily downward towards the cups in addition to maintaining the previous x-y control. Once the robot reached a fixed height, the gripper closed. If the robot was lined up successfully during the downward motion, it grasped one of the cups and the trial was considered a success; if neither cup was grasped, it was a failure. The robot performed an autonomous pouring motion, then the robot and cups were reset.

4.2 Procedure

Participants first provided consent and completed a demographic survey on an experimenter-provided laptop, then waited for the robot to become available. Next, they were introduced to the robot, the task, and its controls. They performed eight trials in two sets of four each (without a practice period), then completed the survey by answering two qualitative questions and providing an email address for compensation. The entire interaction took approximately 10 minutes, excluding wait time for the robot. Participants were compensated with \$3.50 Amazon gift cards. As an additional incentive, the cups were filled with candies of the participant's selection. When the cup was successfully grasped, the robot poured out the candy and the participant kept it. The study protocol was approved by the university's institutional review board.

Participants performed two sets of four trials each. The first four trials occurred at the **baseline** difficulty level, described below. After those trials were complete, the difficulty level was changed based on the participant's number of successes during the first set. If the participant succeeded on 0 or 1 trial, the second set of trials was performed in the **easier** difficulty level; if they succeeded on exactly 2 trials, the assistance was maintained at the **baseline** difficulty level; and if the participant succeeded on 3 or 4 of the trials, they next experienced the **harder** difficulty level. Adjusting the level of difficulty of the task helped to compensate for the large variance in participants' initial levels of skill at controlling the robot. Tuning the task difficulty level to more closely match the participant's initial skill made the study success rate more sensitive to how participants adapted with more experience.

In addition to these modifications to assistance parameters, the "goal" locations varied between participants and set of trials. In all trials, the assistance used three "goal" configurations positioned in a triangle around the workspace, with the cups in the interior of the triangle. There were four goal configurations, corresponding to the direction that the triangle pointed: **up**, **down**, **left**, **right**. The goal configuration for the first set of trials was block-randomized between participants. For the second set of trials, the goal configuration was inverted relative to the first set. An overhead view of the interaction appears in Fig. 7.

4.3 "Assistance" system

The x-y motion of the robot was determined using the participant's raw input data via the shared autonomy assistance described in Sec. 2.2. However, unlike in the previous study, the "assistance" system directed the users to "goals" composed of arbitrary points in space and not located above the cups. In this scenario, the "assistance" signal draws the robot *away* from the cups, so the user must overcome the dynamics of the "assistance" to achieve their goal.

Figure 7: Overhead view of robot in the claw game interaction. A user controlled a Kinova Gen3 Lite robot to pick up one of two cups, despite "assistance" that drove them *away* from the cup locations.

To compute the assistance, we set the reward function of each goal to $r_g(x, a) = -\alpha$ if $|g - x| > D$ or $-\frac{\alpha}{D}|g - x|$ otherwise. This piecewise linear reward function matches the one used in Javdani et al. [14] and helps align the robot more precisely with the goal location when it is nearby. The scaling parameter α controls the overall robot speed. We set $D = 5$cm experimentally.

4.4 Difficulty levels

Participants could experience three different task difficulty levels, set by adjusting the arbitration parameter γ and the reward scaling parameter α.

Baseline. In this level, the arbitration parameter γ was set to 0.5, which causes the same behavior as the *assistance* condition in Study 1. The robot speed remained nominal at $\alpha = 0.5$.

Easier. To make the control easier, the arbitration parameter was changed to $\gamma = 0.8$, so that the user input contributed more to the final motion than the assistance did. In addition, the overall robot speed was halved, so $\alpha = 0.25$.

Harder. For participants who achieved success in the first set of trials, the arbitration parameter was lowered to $\gamma = 0.2$, so that the user input has less direct impact on the robot motion. In addition, the overall speed was scaled up by a factor of 1.5, making $\alpha = 0.75$.

4.5 Hypotheses

H1. Users will succeed on the task despite the presence of adversarial "assistance". If users tried to control the robot directly and did not adapt to the unexpected dynamics, the "assistance" would drive the robot away from the goal and make the task very difficult. The fact that users can succeed means they must be adapting to the assistance dynamics.

H2. Users will report explicitly reasoning about how the system works and strategizing about how to control it.

H3. User success rates will increase as they perform more trials at the same difficulty level. An increase in success rate with experience can only occur if users are changing their policies over time, which is inconsistent with the assumption of stationary control behavior.

4.6 Results

4.6.1 Participants. Over the course of 6.5 hours, 35 participants were recruited to interact with the robot. 5 interactions were discarded due to data quality or study performance issues, leading to **N=30** overall participants. Participant demographics largely reflected the university setting of the study. 13 were male, 15 female, and 2 unspecified. 21 were aged 18-24, 7 aged 25-34, 1 aged 35-44, and 1 65 or older. Almost all participants (26) were students; others included 1 professor and 3 staff members. Computer science was the primary area of study (16), and the remaining participants studied technical fields except for 2 art students. 1 reported playing video games daily; 5, weekly; 5, monthly; 10, rarely; and 9, never.

4.6.2 Overall Performance. Success rates on the task demonstrate that it was possible for users to control the robot and achieve their goals. Over each participant's first sets of trials, which all occurred in the *baseline* difficulty level, participants successfully grasped a cup in 42/120 (35%) of trials, and successful trials took an average of 29.9 seconds. After the first set, participants performed a second set in the *easier*, *baseline* (again), or *harder* difficulty level depending on their performance in the first set. 16/30 (53.3%) of participants succeeded on fewer than two trials, so their second set was performed in the *easier* difficulty. They succeeded in 61/64 (95.3%) of trials, and successful trials took an average of 10.6 seconds. 9/30 participants (30%) succeeded twice in the first set and continued in *baseline* difficulty, during which they succeeded in 16/36 trials (44.3%) and took an average of 14.9 seconds on successful trials. The remaining 5/30 participants (16.7%) succeeded on three or more trials in the first set, so they transitioned to the *harder* difficulty level. Over their second sets of trials, they succeeded in 10/20 (50%), and successful trials took 26.4 seconds on average. The highest performing participant overall succeeded in 7/8 trials. 23/30 participants (77.7%) succeeded at least once in the first set of trials, and all participants succeeded in at least one trial among both of their sets.

The success rates in the task show that users can perform this robot control task despite the conflicting assistance. While success rate varied widely among participants, even the lowest-performing participants succeeded in at least one trial. This find supports **H1** and provides evidence that participants must be using control strategies other than what would be appropriate for direct control.

4.6.3 Participants Use Concrete Strategies. Next, we consider how participants explain their own strategies to accomplish the task. After participants finished both sets of trials, they answered two open-response questions taken from the the first study:

(1) How much do you feel the robot's ability to complete the task depended on your input?
(2) What strategies did you use when controlling this system?

Reviewing user responses suggests that several participants tried to stabilize the robot above the cup before initiating downward motion so that they needed to provide minimal control as the cup descended. For example, P29 reports that "I sort of just hover the claw above the cup and then initiate the drop and make slight adjustment[s] along the drop." This strategy takes advantage of stable points in the combined dynamics of the robot and the assistance, explicitly described by P9 ("[I] tried to stabilise [the robot] in a position") and P33 ("I was usually able to get [the robot] positioned well enough that, even when the robot shifted as it descended, I didn't need to touch the analog stick."). These stable points were not an explicit design objective of the assistance; instead, they emerged from the coupled dynamics between the assistance and the underlying system behavior. Users were sensitive enough to the behavior of the system to identify these stable points intuitively.

In addition, some participants report providing exploratory inputs to the robot to understand its dynamics before controlling it: "I first tried moving the arm around with the joystick to try to understand how it responded to input" (P20). These exploratory actions are inconsistent with goal-directed models of user input, and misinterpreting motion as goal-directed can derail certain assistance systems completely [2]. Participant descriptions of their own strategies not only support **H2**, but they motivate further examination of the user experience in controlling an assisted system.

4.6.4 Success Increases With Experience. Finally, we consider if user performance on tasks increased with experience. To control for participant skill and adjusted difficulty levels, we count the number of successful trials in the *early* part of each participant's sets, i.e., trials 1, 2, 5, and 6. We compare this count to their number of total successes in the *later* half of each set, comprising trials 3, 4, 7, and 8. We find that 4 participants succeed one fewer time; 13 participants succeeded the same number of times; and the remaining 13 succeeded an additional time. A paired t-test shows a significant difference between the number of successes ($t(29) = 2.34, p < 0.03$), with a mean increase in the number of successes of 0.3 (95% CI: $[0.038, 0.56]$). These results provides evidence that success rates increased with experience, supporting **H3**.

5 DISCUSSION AND CONCLUSION

In this paper, we show that users change their control input in the presence of automated assistance. When the assistance is useful, participants describe how they learn the new system behavior and strategize about how to best take advantage of it. When the assistance conflicts with user objectives, users can override the "assistance" to accomplish the task. These findings motivate future investigations into how users respond to the presence of assistance.

Furthermore, these findings suggest a new, user-centered approach to analyzing assistance systems. Rather than optimizing over nominal user input behavior, future work can explicitly consider the system from the user's perspective. To the user, the assistance acts as a separate dynamical system coupled to the underlying system, not unlike how virtual fixtures [21, 29] introduce stabilizing "forces" around particular locations or motions. Unlike virtual fixtures, though, the assistance can take advantage of known task models and task switching dynamics. User-centered assistance acts by changing the system dynamics to ones that a user can better understand and control to accomplish their tasks.

Several design implications result from this user-centered perspective on assistance. For example, complex assistance systems face a tradeoff between the increased task modeling power they provide and the decreased ability of users to understand their dynamics and control the entire system optimally. It is important for the goal prediction system not only to be capable, but to be interpretable, so that the entire system is transparent to the user [1]. Further, the requirement of a perfect, complete task model may be lessened: the system can rely on the user's expertise for some parts of the task, thus reframing the assistance as collaboration. This perspective also allows researchers to investigate the coupled system via system analysis techniques, e.g., measuring how quickly the system reacts to a change in the user's goal or if the assistance introduces new stable states, as were found by users in the second experiment. Demonstrating that users can take an active role in controlling assistive systems is foundational to this user-centered perspective on assistance design.

One limitation of this work is that the tasks that users performed are particularly simple, so this ability of users to adapt may not transfer to more complex tasks. Simple tasks were used here so that participants could learn how the systems behaved and change their input strategy within short interactions, so these results cannot prove that users adapt to assistance in all tasks. However, user adaptation is compatible with the increases in performance found with automated assistance, and cooperative adaptation may even make the assistance *more* effective than if the user policy remained stationary. Future work can examine how users respond to more complex assistance systems in more sophisticated tasks.

Another limitation that pertains especially to the second study is that the level of difficulty of the task, and the amount of experience participants had with the task before operating the robot, was not tightly controlled. While conducting the study in a public space substantially improved recruitment, participants often watched other people controlling the robot before they began, which may affect how they responded to the assistance. In addition, the ease of the *easier* difficulty level meant that participants had a near-perfect success rate, so improvement with experience was difficult to measure. This study is best understood as demonstrating that participants *can* override conflicting assistance to accomplish their goals rather than precisely characterizing this adaptation.

In this paper, we show that users respond to the presence of goal-directed assistance by learning how it works and adapting their control inputs to achieve their goals. Users adjust their performance to collaborate more effectively with assistance that shares their goals and to compensate for "assistance" that makes their goals harder to achieve. This variation in user behavior does not need to be rejected as noise that degrades the performance of an optimal assistive system. Rather, treating users as intentional agents opens up new ways of developing and evaluating goal-directed assistance systems: the future of assistance is collaborative.

6 ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under grant no. IIS-2132887. Thanks to Hang Yu for assisting with study execution and Katherine Allen for assisting with data coding.

REFERENCES

[1] Victoria Alonso and Paloma De La Puente. 2018. System transparency in shared autonomy: A mini review. *Frontiers in Neurorobotics* 12, November (11 2018), 358702. https://doi.org/10.3389/FNBOT.2018.00083/BIBTEX

[2] Reuben M Aronson and Henny Admoni. 2022. Gaze Complements Control Input for Goal Prediction During Assisted Teleoperation. In *Robotics: Science and Systems*. https://doi.org/10.15607/rss.2022.xviii.025

[3] Chris L Baker, Joshua B Tenenbaum, and Rebecca R Saxe. 2007. Goal Inference as Inverse Planning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*. 29.

[4] Tom Carlson and Yiannis Demiris. 2008. Collaborative control in human wheelchair interaction reduces the need for dexterity in precise manoeuvres. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 59–66.

[5] Gabriele Cimolino and T. C.Nicholas Graham. 2022. Two Heads Are Better Than One: A Dimension Space for Unifying Human and Artificial Intelligence in Shared Control. In *Conference on Human Factors in Computing Systems - Proceedings*. Association for Computing Machinery. https://doi.org/10.1145/3491102.3517610

[6] Eric Demeester, Alexander Hüntemann, Dirk Vanhooydonck, Gerolf Vanacker, Alexandra Degeest, Hendrik Van Brussel, and Marnix Nuttin. 2006. Bayesian estimation of wheelchair driver intents: Modeling intents as geometric paths tracked by the driver. In *IEEE International Conference on Intelligent Robots and Systems*. 5775–5780. https://doi.org/10.1109/IROS.2006.282386

[7] Anca D Dragan and Siddhartha S Srinivasa. 2013. A policy-blending formalism for shared control. In *International Journal of Robotics Research*, Vol. 32. SAGE PublicationsSage UK: London, England, 790–805. https://doi.org/10.1177/0278364913490324

[8] Deepak Gopinath, Mahdieh Nejati Javaremi, and Brenna Argall. 2021. Customized Handling of Unintended Interface Operation In Assistive Robots. In *Proceedings - IEEE International Conference on Robotics and Automation*, Vol. 2021-May. 10406–10412. https://doi.org/10.1109/ICRA48506.2021.9561096

[9] Deepak E Gopinath and Brenna D Argall. 2017. Mode switch assistance to maximize human intent disambiguation. In *Robotics: Science and Systems*, Vol. 13. https://doi.org/10.15607/rss.2017.xiii.046

[10] Alberto Gottardi, Stefano Tortora, Elisa Tosello, and Emanuele Menegatti. 2022. Shared Control in Robot Teleoperation With Improved Potential Fields. *IEEE Transactions on Human-Machine Systems* 52, 3 (6 2022), 410–422. https://doi.org/10.1109/THMS.2022.3155716

[11] Alexander Hüntemann, Eric Demeester, Marnix Nuttin, and Hendrik Van Brussel. 2008. Online user modeling with Gaussian Processes for Bayesian plan recognition during power-wheelchair steering. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. 285–292. https://doi.org/10.1109/IROS.2008.4651040

[12] Alexander Hüntemann, Eric Demeester, Gerolf Vanacker, Dirk Vanhooydonck, Johan Philips, Hendrik Van Brussel, and Marnix Nuttin. 2007. Bayesian plan recognition and shared control under uncertainty: Assisting wheelchair drivers by tracking fine motion paths. In *IEEE International Conference on Intelligent Robots and Systems*. 3360–3366. https://doi.org/10.1109/IROS.2007.4399524

[13] Siddarth Jain and Brenna Argall. 2019. Probabilistic Human Intent Recognition for Shared Autonomy in Assistive Robotics. *ACM Transactions on Human-Robot Interaction* 9, 1 (12 2019), 1–23. https://doi.org/10.1145/3359614

[14] Shervin Javdani, Siddhartha Srinivasa, and Andrew Bagnell. 2015. Shared Autonomy via Hindsight Optimization. In *Robotics: Science and Systems XI*, Vol. 11. Robotics: Science and Systems Foundation. https://doi.org/10.15607/RSS.2015.XI.032

[15] Ananth Jonnavittula and Dylan P Losey. 2021. Learning to Share Autonomy Across Repeated Interaction. In *IEEE International Conference on Intelligent Robots and Systems*. 1851–1858. https://doi.org/10.1109/IROS51168.2021.9636748

[16] Ananth Jonnavittula and Dylan P. Losey. 2022. Communicating Robot Conventions through Shared Autonomy. In *Proceedings - IEEE International Conference on Robotics and Automation*. 7423–7429. https://doi.org/10.1109/ICRA46639.2022.9811674

[17] Hong Jun Jeon, Dylan Losey, and Dorsa Sadigh. 2020. Shared Autonomy with Learned Latent Actions. In *Robotics: Science and Systems*. Robotics: Science and Systems Foundation. https://doi.org/10.15607/rss.2020.xvi.011

[18] John W. Krakauer, Alkis M. Hadjiosif, Jing Xu, Aaron L. Wong, and Adrian M. Haith. 2019. Motor Learning. *Comprehensive Physiology* 9, 2 (4 2019), 613–663. https://doi.org/10.1002/CPHY.C170043

[19] Daniel Leyzberg, Samuel Spaulding, Mariya Toneva, and Brian Scassellati. 2012. The Physical Presence of a Robot Tutor Increases Cognitive Learning Gains. In *Building Bridges Across Cognitive Sciences Around the World - Proceedings of the 34th Annual Meeting of the Cognitive Science Society, CogSci 2012*. 1882–1887.

[20] Dylan P. Losey, Craig G. McDonald, Edoardo Battaglia, and Marcia K. O'Malley. 2018. A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction. https://doi.org/10.1115/1.4039145

[21] Panadda Marayong, Ming Li, Allison M. Okamura, and Gregory D. Hager. 2003. Spatial motion constraints: Theory and demonstrations for robot guidance using virtual fixtures. In *Proceedings - IEEE International Conference on Robotics and Automation*, Vol. 2. 1954–1959. https://doi.org/10.1109/robot.2003.1241880

[22] Katharina Muelling, Arun Venkatraman, Jean-Sebastien Sebastien Valois, John E Downey, Jeffrey Weiss, Shervin Javdani, Martial Hebert, Andrew B Schwartz, Jennifer L Collinger, and J. Andrew Bagnell. 2017. Autonomy infused teleoperation with application to brain computer interface controlled manipulation. *Autonomous Robots* 41, 6 (8 2017), 1401–1422. https://doi.org/10.1007/s10514-017-9622-4

[23] Benjamin A. Newman, Reuben M. Aronson, Siddartha S. Srinivasa, Kris Kitani, and Henny Admoni. 2022. HARMONIC: A Multimodal Dataset of Assistive Human-Robot Collaboration. *The International Journal of Robotics Research* 41, 1 (7 2022), 3–11. https://doi.org/10.1177/02783649211050677

[24] Stefanos Nikolaidis, David Hsu, and Siddhartha Srinivasa. 2017. Human-robot mutual adaptation in collaborative tasks: Models and experiments. *International Journal of Robotics Research* 36, 5-7 (6 2017), 618–634. https://doi.org/10.1177/0278364917690593

[25] Sagar Parekh, Soheil Habibian, and Dylan P Losey. 2022. RILI: Robustly Influencing Latent Intent. In *IEEE International Conference on Intelligent Robots and Systems*, Vol. 2022-Octob. 2135–2142. https://doi.org/10.1109/IROS47612.2022.9981601

[26] Camilla Pierella, Maura Casadio, Ferdinando A. Mussa-Ivaldi, and Sara A. Solla. 2019. The dynamics of motor learning through the formation of internal models. *PLoS computational biology* 15, 12 (2019). https://doi.org/10.1371/JOURNAL.PCBI.1007118

[27] Daniel J Rea and Stela H Seo. 2022. Still Not Solved: A Call for Renewed Focus on User-Centered Teleoperation Interfaces. *Frontiers in Robotics and AI* 9 (2022). https://doi.org/10.3389/frobt.2022.704225

[28] Siddharth Reddy, Anca D Dragan, and Sergey Levine. 2018. Shared Autonomy via Deep Reinforcement Learning. In *Robotics: Science and Systems*. https://doi.org/10.15607/rss.2018.xiv.005

[29] Louis B. Rosenberg. 1993. Virtual fixtures: perceptual tools for telerobotic manipulation. In *1993 IEEE Annual Virtual Reality International Symposium*. Publ by IEEE, 76–82. https://doi.org/10.1109/vrais.1993.380795

[30] Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum Entropy Inverse Reinforcement Learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3 (AAAI'08)*. AAAI Press, 1433–1438.

[31] Matthew Zurek, Andreea Bobu, Daniel S Brown, and Anca D Dragan. 2021. Situational Confidence Assistance for Lifelong Shared Autonomy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2783–2789. https://doi.org/10.1109/icra48506.2021.9561839