RESEARCH ARTICLE

Open Access



Intention-reflected predictive display for operability improvement of time-delayed teleoperation system

Yaonan Zhu^{1*}, Keisuke Fusano¹, Tadayoshi Aoyama¹ and Yasuhisa Hasegawa¹

Abstract

Robotic teleoperation is highly valued for its ability to remotely execute tasks that demand sophisticated human decision-making or that are intended to be carried out by human operators from a distance. However, when using the internet as a communication framework for teleoperation, high latency, and fluctuations make accurate positioning and time-dependent tasks difficult. To mitigate the negative effects of time delay, this paper proposes a teleoperation system that uses cross reality (XR) as a predictive display of the outcome of operators' intended actions and develops a time-delay aware shared control to fulfill the intention. The system targets a liquid pouring task, wherein a white ring that indicates the intended height of the liquid surface is overlayed onto the beaker in a delayed camera image to close the visual feedback loop on the leader side. Simultaneously, the shared control automatically completes the pouring action to track the intended liquid height. The performance of the proposed system is validated based on liquid pouring experiments performed by human subjects. When compared with direct control, the absolute error rate decreased significantly for a constant round-trip time delay of 0.8 s and 1.2 s, similarly for a time-varying delay of 0.4 s and 0.8 s. Moreover, when the time-varying delay was 0.8 s, operators achieved significantly higher accuracy while maintaining comparable operation time. These results indicate that our proposed system improves operability even in the presence of time-varying delays in communication networks.

Keywords Teleoperation, Shared control, Cross reality, Predictive display

Introduction

Robotic teleoperation systems are currently being used in various fields such as medical care [1, 2], disaster sites [3, 4], space operation [5], and assistive robots [6, 7]. Robotic teleoperation offers two main advantages over fully automated robots. The first is the ability to reflect human intentions in tasks that require human decisionmaking during the process. For example, when pouring a drink into a cup, the person pouring the drink does not necessarily determine the exact amount beforehand

¹ Department of Micro-Nano Mechanical Science and Engineering,

but decides during the pouring process that the amount is sufficient and stops pouring to complete the task. This example shows that humans determine the amount they want to pour through visual feedback and might not have a clear answer before performing the task. Similarly, tasks that reflect intentions during the process, such as deciding which drink to take out of the refrigerator after opening it, or positioning objects while tidying a room, require human involvement to physically move objects rather than full automation. The second advantage is that teleoperation provides high adaptability and flexibility through human supervision. In full automation, the environment that surrounds the robots must be precisely perceived in advance. Despite the significant advancements made in automation in recent years, technology has not yet reached the point where robots can function



© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

^{*}Correspondence:

Yaonan Zhu

zhu@robo.mein.nagoya-u.ac.jp

Nagoya University, 1 Furo-cho, Chikusa-ku, Nagoya 464–8603, Japan

with complete autonomy and effectively manage unforeseen situations or unpredictable events [8].

In recent years, cross reality (XR) technologies have gained attention as visual enhancement systems for teleoperation interfaces. XR is a collective term for virtual object and virtual space presentation technologies, including virtual reality (VR), augmented reality (AR), and mixed reality (MR), and is being applied in various fields, such as gaming, medical training, and travel experiences [9]. The benefits of using XR for teleoperation include the ability to present three-dimensional (3D) images and provide an immersive experience, which enhances the transparency and operability of teleoperation. Sun et al. proposed an MR system that displays the virtual scene with a robot model, which follows the realtime joint states, and a point cloud of the task space [10]. In the MR system, the operators are isolated from their local environment, which enhances their concentration to focus on the task space and thereby improves operability and transparency. In [11], the visual feedback of the teleoperation system is enhanced using a VR-based wide-view head-mounted display (HMD). An MR-based immersive 3D visualization is proposed for operator situation awareness [12].

Although the use of XR has improved operability, teleoperation continues to be plagued by time delays. Communicational time delays have been reported to severely affect operability [13]. For teleoperation to be successful, stable transmission of data, such as visual feedback information and the control command, is required. However, in teleoperation scenarios relying on the public Internet or wireless communication, long distances can cause significant delays or jitters in communication.

Previous research has shown that operating robots in time-delayed environments is challenging, and many solutions have been proposed. Song et al. reported that task operation time and the number of task failures increased exponentially with time delay for remote surgery [14]. Orosco et.al. focused on remote surgery with communication delays and demonstrated a reduction in task operation time by applying negative scaling to the operator's input [15]. Storms et al. improved operability under communication delays using shared control to perform obstacle avoidance tasks automatically [16]. Huang et al. proposed a single-leader-multi-follower teleoperation system with automatic leader selection to solve the obstacle avoidance problem under time-varying delays, nonlinearities, and uncertainties [17]. Edwin et al. have developed a skill-based shared autonomy framework that improves follower intelligence was proposed to overcome instabilities in communication networks [18]. Laghi et al. proposed a shared control strategy for bimanual operation assistance, in which the system employs versatile VR controllers for inputs, and assists operations by autonomously coordinating bimanual actions to grasp objects of varying sizes [19]. Additionally, instead of assisting the operation in all degrees of freedom (DOF), Bowman et al. proposed a framework to assist the operation individually for each DOF based on an authority allocation policy which assesses both the levels of disagreement in control allocation and the user's acceptance of the assistance [20].

Although introducing some degree of autonomy on the follower side can assist operators with task completion, it may result in unintended movements contrary to the operator's intention. In such cases, operators may feel that the shared control system is interfering with their operations, reducing intuitiveness and operability [21]. In addition, time delays during immersive teleoperations can induce simulation sickness due to differences in sensation between the operator and the remote robot [22]. Therefore, a system that mitigates the negative effects of time delays, adequately reflects operators' intentions, and provides assistance to fulfill the intentions is highly desirable.

In addition to shared control, the predictive display has been proposed to address communication delays [23, 24]. Predictive display predicts the delayed movement of a robot and displays a real-time phantom robot to the operators in response to their input commands; the real robot's motion then follows the presented phantom image. Martha et al. verified the impact of predictive display on operability improvement for lunar vehicle teleoperation [25]. Burkert et al. proposed a method to alleviate time delay effects in teleoperation systems using a local scene model to predict the trajectory of a remote robot arm and generated photorealistic images as a predictive display to close the feedback loop locally at the operator's side [26]. Instead of an apriori local model, Jagersand et al. proposed a predictive display using realtime model capture and tracking for alignment tasks, in which the display can render textured graphics for improved visual feedback under time delay [27]. The predictive display has been integrated with AR technology to enhance the performance of teleoperated pick and place tasks [28] and teleoperated robotic surgeries [29]. The predictive display has also been applied to remote vehicle operations [30-32]. As the literature suggests, the predictive display has shown improvement in operability under time delay. However, in most of these applications, the predictive display was aimed at addressing constant time delays and focused on prediction without automation-based control assistance. Developing a predictive display that targets time-varying delays and complements human commands through automation is urgently required when using the Internet as the communication framework for teleoperation.

This paper presents a teleoperation system that can fulfill operator intention in a remote environment, even with time-varying delays. The contributions of this study are summarized as follows:

- 1 An XR-based predictive display is proposed to present the outcome of the action performed by the operator, which is regarded as the operator's intention. This predictive display is targeted for timedependent tasks, such as the liquid pouring task under time-varying delays. The operator's intended level of the poured liquid is obtained through a mathematical model. The intended level is then visually represented as a white ring, overlaid on the delayed camera stream; this provides real-time visual feedback for the operator's pouring action.
- 2 A shared control algorithm is proposed to fulfill the intended action given by the operator in the remote environment. The controller tracks the liquid surface height in real-time through point cloud segmentation-based liquid surface detection. By referring to the real-time liquid height, a proportional derivative (PD) controller tracks the operator's intended liquid height and automatically completes the action to pour the desired volume of liquid despite the occurrence of time-varying delays.
- 3 Human subject experiments are performed to evaluate system performance. Teleoperated liquid pouring experiments are performed under two types of communication delays, including constant and time-varying delays with different round-trip values.

These experiments suggest that the proposed system improves the operability of the teleoperation system under time-varying delays in terms of task accuracy and task operation time.

Teleoperation system

In this study, an immersive teleoperation system is proposed. Figure 1 shows the architecture of the proposed teleoperation system. The system obtains the position and posture of the head and right hand through VRbased head and hand tracking (Oculus Quest 2, Meta). The obtained values are transmitted to a remote PC through a 3D development platform (Unity). A homogeneous transformation matrix is then used to convert the Cartesian position and posture of the head and right hand from VR coordinate system to the robot coordinates. Then, the inverse kinematics of the robot are solved to convert the Cartesian target to joint space target angles using TRAC-IK library [33] and sent to each robot. In this study, the head pose is kept constant under experimental conditions and only the hand pose is synchronized with the follower robot.

The video stream from the remote environment is obtained using a stereo camera (ZED Mini, Stereo Labs) mounted on the wrist link of the robot manipulator (Universal Robot, UR3e), and used as the visual feedback. The video stream is first sent to the follower side PC (Fig. 1: PC2) and then to the leader side PC (Fig. 1: PC1) for graphical processing in Unity, and presented to the operator in stereo vision. The presented image resolution on



Fig. 1 Proposed teleoperation system using a shared control and a VR device. The Oculus head-mounted display (HMD) is used for visual feedback, and an Oculus controller is used as a leader device to track the operator's hand movements. The 6-degree-of-freedom robot arm (Universal Robot, UR5e) is used as a follower robot. The Stereo Camera (Stereo Labs, ZED Mini) is used for stereo visual feedback. The round-trip delay of this system is 150 ms

the HMD is 720×2560 pixels with a frame rate of 60 Hz. The intrinsic time delay of this teleoperation system is 30 ms for the transmission of position and posture commands, 80 ms between the follower side PC receiving the command and the follower robot reaching the target position, and 50 ms for the communication delay of the camera image. Additionally, the transmission frequency of command values from PC1 to PC2 is approximately 30 Hz.

Predictive display and shared control

This section describes (1) the proposed XR-based predictive display of the predicted liquid height from the operator's action to provide real-time visual feedback despite the time delay, and (2) a shared control algorithm that automatically complements the pouring action to fill the liquid to the intended height. In the proposed system, the predictive display and the shared control algorithm incorporate to fulfill the user intention. The predictive display utilizes a liquid model to estimate the user intended liquid height and overlays the white ring on the beaker as an indication. Concurrently, the shared control algorithm auto-aligns the pouring spout of the bottle with the beaker and executes the pouring action to track the intended liquid height in real-time. The system is designed to target a liquid pouring task and involves manipulating a robot hand that grasps a plastic bottle and pours its contents into a beaker. In this task, the position of the pouring spout is automatically aligned to match the opening position of the beaker, allowing the operator to focus on controlling a single degree of freedom that rotates around the pouring spout of the bottle.

System flowchart

Figure 2 shows the system flowchart of intention extraction and automatic execution. First, the operator calculates the intended liquid surface height h using the posture θ of the controller and a liquid model. The calculated liquid surface height is overlaid on the beaker shown in the remote follower environment video received after a time delay. As the predictive display, the liquid surface height calculated using the liquid model is presented to the operator without delay. The operator observes the real-time change in the overlaid liquid height and decides whether to continue pouring or stop and finish the task. As this decision-making process is similar to that performed by humans when pouring liquid into a cup, it can be said that the volume of liquid calculated from the liquid model is the intended volume by the operator.

As the feedback loop is closed on the leader side, the operation result does not depend on the remote environment with time delay. Therefore, even if there is a



Fig. 2 Flowchart of the teleoperation system with the model-based predictive display, and shared control to reproduce the intended action. θ is the pose of the controller, and *h* is the intended liquid height calculated by the liquid model

communication delay, such as a delayed video stream from a remote environment, the operator can still complete the task with the intended liquid height without being affected by the delay. In addition, despite the communication delays, the intended height of the liquid during the pouring process can still be transmitted to the follower robot. The shared control will automatically pour the liquid until it reaches the intended liquid surface height. By continuously visualizing the progress of the task with the predictive display, the time delay is overcome. This makes the time-delayed teleoperated pouring task almost identical to the task without time delay, and through the auto-completion capability of the proposed system, the intention of the operator can be successfully realized in the remote environment.

The system behavior for each control step n under time-varying delay is shown in Fig. 3. First, using the liquid model the desired liquid height h_1 can be calculated from the controller's posture θ_1 . As the predictive display is visually presenting h_1 to the operator as a ring without delay, the operator can confirm the outcome (liquid height) of their action (θ_1) and manipulate the liquid in a real-time fashion. Next, h_1 is transmitted to the remote environment with time delay T_d . Then, according to the received liquid height on the follower side, a PD control is performed to complete the pouring action and track



Fig. 3 The system behavior for each control step n under time-varying delays. θ_n is the pose of the controller, and h_n is the intended liquid surface height calculated by a liquid model

the intended reference height h_1 . By repeating this process, the operator can reproduce the desired height of the liquid h_n in the remote environment.

Liquid model-based predictive display for intention reflection

In this system, a liquid model is proposed to convert the controller pose θ to the operator's intended liquid height *h*. Equation (1) shows the formula used to calculate the height of the liquid surface from the controller input angle:

$$h(t) = \int_0^t K\left[\theta(\tau) - \theta_0\left\{h(\tau)\right\}\right] d\tau$$
(1)

where h(t) represents the intended liquid height at the current time step t, K is a constant gain value, $\theta(\tau)$ is the angle of the controller input at the step τ , and $\theta_0(h(\tau))$ represents the angle of the bottle from which the liquid starts flowing given $h(\tau)$ is already poured into the beaker. Equation (1) shows that, similar to the situation when pouring liquid from a bottle, the more tilted the bottle is compared to the angle at which pouring can be started, the greater will be the flow rate of the liquid. Note that, the starting angle $\theta_0(h)$ should be obtained from the real-world data by actually pouring the liquid with the given bottle.

The method to derive $\theta_0(h)$ is described as follows. $\theta_0(h)$ is obtained by tilting a bottle grasped by the robot hand to a certain angle and stopping for 10 s for pouring, during which the height of the liquid surface in the beaker and the given angle of the bottle are recorded. Then, a polynomial regression is applied to obtain the relationship between $\theta_0(h)$ and h. The angle of the bottle was investigated in 1-degree increments in the range of 80° to 87°, and the linear approximation yielded the following equation:

$$\theta_0(h) = 108.07 \cdot h + 78.592 \tag{2}$$

The method to determine the gain *K* is described as follows. The value of the integral term increases with *K*, resulting in a faster response of $h(\tau)$. First, in order to make the model closely simulate the characteristics of the real liquid, the time from the start of pouring until the liquid reaches a height of 5 cm is measured, with the bottle tilt angle fixed at 86°. Then, the same pouring condition is implemented inside the Unity physical simulator. The gain *K* is adjusted inside the simulator recursively to match the time required for pouring in the real environment. This procedure yielded a *K* value of 0.0774.

The operator's intended liquid height h(t) is converted from the operator's input action $\theta(\tau)$ through the combination of Eqs. (1) and (2). As Fig. 4 shows, based on the estimated liquid height, an XR-based predictive display is developed by overlaying a white ring that indicates the height of the liquid surface to the beaker in the delayed camera image. This predictive display is then streamed to the HMD to provide real-time visual feedback. As the intended liquid height is predicted through a mathematical model, the predictive display does not contain any communication time delay and closes the visual feedback



Fig. 4 The predictive display and the pouring process. The predictive display overlays the white ring on the beaker indicating the intended liquid height. The visual feedback loop is closed on the leader side without time delay, then the shared control automatically completes the pouring action until the intended liquid height is achieved. The sub-figure **a** shows the approaching and automatic position alignment, **b** the operator starts pouring action, **c** the white ring goes up to indicate the liquid height, **d** the operator adjusts the white ring to the targeted height, **e** the white ring reaches the targeted height, the operator stops the pouring while the shared control completing the pouring action, **f** liquid is being poured automatically, **g** pouring is finished and the liquid reached the intended height, **h** recover to the initial position



Fig. 5 Block diagram of the PD control for liquid pouring. θ_{ope} is the pose of the controller, \hat{h} is the intended liquid surface height calculated by the model, *e* represents the error, and θ indicates the angular velocity of rotation around the pouring spout of the bottle. Additionally, h_{real} represents the actual height of the poured liquid

loop on the operator side. By relying on this real-time visual feedback, the operator can manipulate the liquid height as intended, and then the shared control automatically completes the pouring action on the remote side. The pouring process is shown in the sub-figures (a–h) of Fig. 4.

Shared control for automatic liquid-filling

Inspired by the research [34] which developed an automatic liquid-filling algorithm using a humanoid robot, a shared control algorithm for automatic liquid-filling was developed. Figure 5 shows the block diagram of the automatic liquid-filling algorithm. The reference value is the intended liquid height h(t), which is obtained by using the aforementioned mathematical model to convert the controller's input posture. The deviation between the target value h(t) and the actual liquid height h_{real} is used as input to the PD controller that outputs the rotational velocity of the bottle rotating around its spout. Consequently, the PD controller controls the volume of the liquid to be poured and adjusts the liquid height in the remote follower environment.



Fig. 6 Liquid height detection method using point cloud. PS indicates plane segmentation: PS detects the plane with the largest possible area from point cloud data and segments the data into plane point clouds and other point clouds. This allows PS to detect the plane of the table (h_1) and the plane of the liquid surface (h_2). The difference between their heights is the liquid height

To perform PD control, the liquid height has to be obtained in real-time. Figure 6 shows an overview of the liquid height detection algorithm. An RGBD camera is mounted horizontally above the workspace of the follower robot to capture real-time point cloud data of the

workspace. The Point Cloud Library (PCL) [35] is used to perform plane detection, which is a PCL algorithm that recognizes the largest plane in the point cloud data and segments it into the largest plane and other point clouds. Using this plane detection algorithm, the point cloud data is first segmented into the table plane where the beaker is placed and other point clouds to obtain the distance (h_1) between the plane and the camera. Then, using the plane segmentation algorithm again on the other point cloud data which is not the plane previously detected, the liquid surface plane can be acquired. The point cloud data recognized as a plane is considered to be the liquid height, and thus, the distance (h_2) from the RGBD camera to the liquid surface is obtained. The liquid height h can then be obtained by subtracting h_2 from h_1 . The automatic liquid filling achieved a filling accuracy of ± 1.5 mm (± 3.4 ml) using this PD controller.

Experiments and discussion

To evaluate the various time delay conditions, including time-varying delay, the experiment was conducted in two stages to minimize the influence of other factors. The task performed in Experiments 1 and 2 was to pour 150 ml of liquid from a bottle into a beaker. The evaluation criteria were the absolute error rate and the task operation time. The absolute error rate is calculated using the following equation:

$$\epsilon = \frac{|M - T|}{T} \times 100 \tag{3}$$

where ϵ is the absolute error rate (%), *M* is the weight of the liquid poured, and *T* is the weight of the target volume of liquid.

The subjects were five individuals who do not regularly use VR. Experiments 1 and 2 are performed under constant and varying time delays, respectively. Experiment 2 was conducted approximately two weeks after Experiment 1. Before starting each experiment, to make the operators fully understand the pouring process, they were made to pour liquid three times under the conditions of direct control (DC) and shared control (SC), without any added communication delay. In this context, direct control refers to operators controlling the single degree of freedom which rotates around the pouring spout, without the aid of liquid height indication and automated completion of the pouring action. In the experiments, the order of the experimental conditions is randomly selected to mitigate the effect of habitation. All study participants gave their informed consent and the study was approved by the Ethics Committee of Nagoya University (No. 23-8). The operators were instructed to pour the liquid quickly and accurately.

Experiment 1 (constant time delay) Experimental setup and procedure

In Experiment 1, evaluations were performed under conditions with different constant time delays. In the experiment, the round-trip communication time delays of 0.4, 0.8, and 1.2 s were prepared. In each condition, the round-trip delay was evenly split into command transmission and visual feedback delays. For each time delay, the two conditions of direct control and shared control were given, and for each condition, 3 trials were performed, resulting in 18 trials for each of the 5 human subjects.

In addition, we obtained the time delay threshold T_c , above which shared control has a better operability than direct control. Absolute error rates and task operation times under each constant time delay condition were compared to determine T_c . The obtained T_c was then used to determine the delay conditions for Experiment 2. It is generally considered that operability decreases under variable communication delay because it is more difficult to predict the robot's trajectory. By design, the operability of our proposed system outperforms direct control given the time-varying delays are set to values larger than the threshold T_c . Therefore, Experiment 2, which will be discussed in Section Experiment 2 (Time-varying delay), is conducted under the conditions of time-varying delays below the obtained T_c .

Results and discussion

The experimental results for the absolute error rate and task operation time are shown in Figs. 7 and 8, respectively. In this experiment, a nonparametric Mann–Whitney test was used to compare the performance of direct control and shared control under each constant time



Fig. 7 Experimental results of absolute error rate in constant time delay conditions. A significant difference was observed between direct control and shared control in terms of constant time delay at 0.8 s and 1.2 s



Fig. 8 Experimental results of operation time in constant time delay conditions. Compared with the shared control, direct control has a greater variation in operation time for all constant time-delay conditions

delay condition. Based on the experimental results (see Fig. 8), no significant differences in task operation time were observed. However, the absolute error rate (see Fig. 7) decreased significantly with shared control for a constant time delay of 0.8 s (DC vs. SC, p < 0.01, average of 10.11 % and 2.21 %), and 1.2 s (DC vs. SC, p < 0.05, average of 13.76 % and 4.57 %). This indicates an operability improvement with shared control when the constant time delay is greater than 0.8 s; therefore, T_c was set as 0.8 s and used to determine the time delay condition for Experiment 2.

Figure 7 shows that the operational error with direct control increased with the communication delay time, indicating that the intended operation becomes difficult when there is a constant time delay in the teleoperation system. This result is consistent with the findings of the previous studies [14]. In contrast, with shared control, time delay has minimal effect on the absolute error rate, indicating that the proposed system can assist in successfully performing the intended tasks even with constant time delays in the communication network.

The interquartile range of the box plots in Fig. 8 indicates that compared with the shared control, direct control has a greater variation in operation times for all constant time-delay conditions. We believe that the variation in operation time is due to differences in the operators' pouring strategies. Table 1 summarizes the average task operation times for each subject. Subject C completed the pouring task in nearly 20 s, whereas subject D took nearly 13 s. The difference in task operation time among the subjects could be due to the difference in their task strategies. When performing teleoperations under time delays, humans often adopt a "move and wait" strategy, which involves stopping until the

CTD [s]	Subject A	Subject B	Subject C	Subject D	Subject E
0.4	17.41	16.98	24.11	12.25	15.97
0.8	16.23	14.73	20.89	14.52	17.75
1.2	14.48	17.24	21.71	14.15	18.39

robot response is confirmed and then performing the next action [36]. Before the start of this experiment, subjects were instructed to pour as quickly and accurately as possible. The subjects were divided into those who did not stop the task in the middle and poured as quickly as possible, and those who stopped the task while adopting the "move and wait" strategy. We believe that this led to a large variation in operation time.

From Fig. 8, when the time delay was 1.2 s, the task was completed in a shorter median time compared with other time delay conditions. Due to the large time delay, the operators may have given an overly large initial input for the robotic arm control, resulting in a faster flow rate at the start of the pouring task. From Fig. 7, the time delay of 1.2 s corresponded to a higher absolute error rate; this implies that the decrease in the task operation time was not due to improved operability, but rather due to over-pouring caused by the deterioration in operability. Unlike pick-and-place or peg-in-hole tasks that can be completed through trial and error adjustments, pouring is a time-dependent and irreversible task; once overshoot occurs there is no way of returning to the original state, which we believe led to these results. To summarize, shared control was able to perform the pouring task with similar median times for all delay conditions, and stable operations were achieved regardless of constant time delays in the communication network.

Experiment 2 (Time-varying delay) Experimental setup and procedure

In Experiment 2, system performance was evaluated under the conditions of time-varying delays with means that are less than the threshold T_c identified in Experiment 1. The round-trip communication time delays were generated with average time delays of 0.2 s, 0.4 s, and 0.8 s. The jitters were set as Gaussian distributions with standard deviations of 10% of the mean time delays. The time delay for command and video communication is half of the round-trip time delay. For each time delay condition, the pouring task was conducted thrice under the conditions of direct control and shared control, resulting in 18 trials for each of the 5 human subjects.



Fig. 9 Experimental results of absolute error rate in time-varying delay conditions. A significant difference was observed between direct control and shared control in the conditions of 0.4 s and 0.8 s



Fig. 10 Experimental results of operation time in time-varying delay conditions. A significant difference was observed between direct control and shared control in terms of time-varying delay at 0.2 s and 0.4 s

Results and discussion

The experimental results for the absolute error rate and operation time are shown in Figs. 9 and 10, respectively. As in Experiment 1, a nonparametric Mann–Whitney test was used to compare the performance of direct control and shared control under the same time delay condition.

From Fig. 9, the absolute error rate decreased significantly at time-varying delays of 0.4 s (DC vs SC, p < 0.05, average of 3.46 % and 2.31 %) and 0.8 s (DC vs SC, p < 0.05, average of 6.32 % and 2.91 %). As a significant difference in performance was observed starting from 0.8 s in Experiment 1, the results from Experiment 2 indicate that shared control has a wider applicability range under time-varying delays. This suggests

that operational difficulty increases due to the variation in time delay, providing further evidence of the proposed system's capability of operational assistance.

From Fig. 10, the operation time with shared control is significantly longer at time-varying delays of 0.2 s (DC vs. SC, p < 0.05, average 11.44 s and 14.04 s), and 0.4 s (DC vs. SC, p < 0.01, average 11.13 and 14.13 s). This suggests that under shared control, tasks take longer to complete due to the automation process within the operation time. However, there was no significant difference in the operation time with a time delay of 0.8 s, indicating that shared control assists the operator to accomplish the task with higher accuracy and comparable operation time.

The effectiveness of shared control can be assessed by considering the absolute error rate and operation time. Under a time-varying delay of 0.4 s, the absolute error rate significantly decreased with shared control, while the operation time significantly increased. This implies that when the time-varying delay is 0.4 s, the operability is improved by means of task accuracy. The result further demonstrates that for tasks requiring high precision, such as pouring liquid in scientific experiments, the proposed shared control method can be effective. Moreover, when the time-varying delay was 0.8 s, shared control successfully assisted operators to complete the pouring task with significantly improved accuracy, while the operation time remained comparable to that with direct control. The results of Experiment 2 suggest that, based on the level of communication delay and the required task accuracy, it is desirable to switch between direct control and shared control for improved operability.

Conclusions and future works

In this paper, we proposed an intention-reflected predictive display to improve the operability of teleoperation systems under constant and time-varying delays. The predictive display uses cross reality (XR) to visually present the outcome of operators' intended actions, and further integrates with a delay-aware shared control algorithm to automatically complete the intended action in the presence of constant or time-varying communication delays. The proposed system was evaluated with human subject experiments with a liquid pouring task. The absolute error rate was shown to decrease significantly for a constant round-trip time delay of 0.8 s and above. Similarly, the absolute error rate decreased significantly for a time-varying delay of 0.4 s or more. When compared with direct control under a time-varying delay of 0.8 s, the proposed shared control system successfully assisted the operators to perform pouring tasks with significantly higher precision and comparable task operation time.

Currently, the real-time liquid height detection employed in our shared control framework is achieved

through point cloud-based plane segmentation, given the controlled lighting conditions, and a simple image background. To extend the applicability of our research to more practical scenarios, future work should explore more complex image backgrounds and lighting conditions. In such cases, image background segmentation can be performed to acquire the region of interest [37], and denoising algorithms can be utilized to enhance the quality of point clouds [38]. The liquid model used in this study is designed for the specific bottle and cup, aiming to estimate the user's intended liquid height. In the future, a more robust liquid model that integrates object recognition and deep learning can be explored to estimate user intention across a variety of bottle and cup shapes. Given the time-dependent nature of the pouring task, reverting to the original state once pouring has started can be challenging. A failing prevention function that halts pouring based on estimated cup capacity, or XR-based indications that display potential liquid spills can be implemented in the future. Moreover, the research outcomes could be significantly enhanced by incorporating sophisticated skill-blending methods [39, 40]. This would broaden the potential applications of our system to complex tasks that require the reflection of human intention, such as in medical applications [41, 42] and cooking applications [43].

Acknowledgements

Not applicable.

Author contributions

YZ and YH initiated this research. KF, YZ, and YH designed and performed the experiments. KF and YZ performed the data analysis, interpreted the experimental results, and wrote the manuscript with the review of TA. All authors read and approved the final manuscript

Funding

This work was supported in part by JST Trilateral AI Research Grant Number JPMJCR20G8, Japan; and in part by JSPS KAKENHI Grant Number JP22K14222.

Availability of data and materials

Not applicable.

Declarations

Ethics approval and consent to participate

All study participants provided informed consent, and the study was approved by the Ethics Committee of Nagoya University (No. 23–8).

Competing interests

The authors declare that they have no competing interests.

Received: 25 April 2023 Accepted: 9 July 2023 Published online: 15 July 2023

References

1. Selvaggio M, Moccia R, Ficuciello F, Siciliano B, et al. (2019) Haptic-guided shared control for needle grasping optimization in minimally invasive

robotic surgery. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3617–3623. IEEE

- Miao Y, Jiang Y, Peng L, Hossain MS, Muhammad G (2018) Telesurgery robot based on 5g tactile internet. Mobile Net Appl 23(6):1645–1654
- Abi-Farraj F, Pedemonte N, Giordano PR (2016) A visual-based shared control architecture for remote telemanipulation. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4266–4273. IEEE
- Abi-Farraj F, Pacchierotti C, Arenz O, Neumann G, Giordano PR (2020) A haptic shared-control architecture for guided multi-target robotic grasping. IEEE Trans Haptics 13(2):270–285. https://doi.org/10.1109/TOH.2019. 2913643
- Balachandran R, Mishra H, Cappelli M, Weber B, Secchi C, Ott C, Albu-Schaeffer A (2020) Adaptive authority allocation in shared control of robots using bayesian filters. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 11298–11304. IEEE
- Quere G, Hagengruber A, Iskandar M, Bustamante S, Leidner D, Stulp F, Vogel J (2020) Shared control templates for assistive robotics. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 1956–1962. IEEE
- Karamcheti S, Zhai AJ, Losey DP, Sadigh D (2021) Learning visually guided latent actions for assistive teleoperation. In: 3rd Conference on Learning for Dynamics and Control, PMLR 144:1230–1241
- Selvaggio M, Cognetti M, Nikolaidis S, Ivaldi S, Siciliano B (2021) Autonomy in physical human-robot interaction: a brief survey. IEEE Rob Automat Lett 6(4):7989–7996. https://doi.org/10.1109/LRA.2021.3100603
- Ziker C, Truman B, Dodds H (2021) Cross reality (xr): Challenges and opportunities across the spectrum. In: Innovative learning environments in STEM higher education: Opportunities, challenges, and looking forward, Springer International Publishing, Cham, pp. 55–77.
- Sun D, Kiselev A, Liao Q, Stoyanov T, Loutfi A (2020) A new mixed-realitybased teleoperation system for telepresence and maneuverability enhancement. IEEE Trans Human-Mach Syst 50(1):55–67
- Fernando CL, Furukawa M, Kurogi T, Kamuro S, Minamizawa K, Tachi S, et al. (2012) Design of telesar v for transferring bodily consciousness in telexistence. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5112–5118. IEEE
- Schwarz M, Rodehutskors T, Droeschel D, Beul M, Schreiber M, Araslanov N, Ivanov I, Lenz C, Razlaw J, Schüller S et al (2017) Nimbro rescue: solving disaster-response tasks with the mobile manipulation robot momaro. J Field Robot 34(2):400–425
- Zhu Y, Aoyama T, Hasegawa Y (2020) Enhancing the transparency by onomatopoeia for passivity-based time-delayed teleoperation. IEEE Robot Autom Lett 5(2):2981–2986
- Xu S, Perez M, Yang K, Perrenot C, Felblinger J, Hubert J (2014) Determination of the latency effects on surgical performance and the acceptable latency levels in telesurgery using the dv-trainer[®] simulator. Surgical endoscopy 28:2569–2576
- Orosco RK, Lurie B, Matsuzaki T, Funk EK, Divi V, Holsinger FC, Hong S, Richter F, Das N, Yip M (2021) Compensatory motion scaling for timedelayed robotic surgery. Surg Endoscopy 35(6):2613–2618
- Storms J, Chen K, Tilbury D (2017) A shared control method for obstacle avoidance with mobile robots and its interaction with communication delay. Int J Robot Res 36(5–7):820–839
- Huang F, Chen X, Chen Z, Pan Y-J (2022) A novel smms teleoperation control framework for multiple mobile agents with obstacles avoidance by leader selection. IEEE Trans Syst Man Cybernetics Syst 53(3):1517
- Babaians E, Yang D, Karimi M, Xu X, Ayvasik S, Steinbach E (2022) Skill-cpd: Real-time skill refinement for shared autonomy in manipulator teleoperation. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6189–6196. IEEE
- Laghi M, Raiano L, Amadio F, Rollo F, Zunino A, Ajoudani A (2022) A target-guided telemanipulation architecture for assisted grasping. IEEE Robot Automat Lett 7(4):8759–8766
- Bowman M, Zhang X (2023) Dimension-specific shared autonomy for handling disagreement in telemanipulation. IEEE Robot Automat Lett 8(3):1415–1422
- Morita T, Zhu Y, Aoyama T, Takeuchi M, Yamamoto K, Hasegawa Y (2022) Auditory feedback for enhanced sense of agency in shared control. Sensors 22(24):9779

- Suomela J. (2001) Tele-presence aided teleoperation of semi-autonomous work vehicles. Licenciate thesis, Helsinki University of Technology, Espoo, Finland. http://urn.fi/URN:NBN:fi:aalto-202104145457
- Bejczy AK, Kim WS, Venema SC (1990) The phantom robot: predictive displays for teleoperation with time delay. In: Proceedings., IEEE International Conference on Robotics and Automation, pp. 546–5511. https:// doi.org/10.1109/ROBOT.1990.126037
- Moniruzzaman M, Rassau A, Chai D, Islam SMS (2022) Teleoperation methods and enhancement techniques for mobile robots: a comprehensive survey. Robot Autonomous Syst 150:103973
- Mathan S, Hyndman A, Fischer K, Blatz J, Brams D (1996) Efficacy of a predictive display, steering device, and vehicle body representation in the operation of a lunar vehicle. In: Conference Companion on Human Factors in Computing Systems, pp. 71–72
- 26. Burkert T, Leupold J, Passig G (2004) A photorealistic predictive display. Pres Teleoperat Virt Environ 13(1):22–43
- Jagersand M, Rachmielowski A, Lovi D, Birkbeck N, Hernandez-Herdocia A, Shademan A, Cobzas D, Yerex K (2010) Predictive display from computer vision models. In: The 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space I-SAIRAS, pp. 673–680
- Xiong Y, Li S, Xie M (2006) Predictive display and interaction of telerobots based on augmented reality. Robotica 24(4):447–453
- Richter F, Zhang Y, Zhi Y, Orosco RK, Yip MC (2019) Augmented reality predictive displays to help mitigate the effects of delayed telesurgery. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 444–450. IEEE
- Davis J, Smyth C, McDowell K (2010) The effects of time lag on driving performance and a possible mitigation. IEEE Trans Robot 26(3):590–593
- Brudnak MJ (2016) Predictive displays for high latency teleoperation. In: Proc. NDIA Ground Veh. Syst. Eng. Technol. Symp., pp. 1–16
- Dybvik H, Løland M, Gerstenberg A, Slåttsveen KB, Steinert M (2021) A low-cost predictive display for teleoperation: investigating effects on human performance and workload. Int J Human-Comp Stud 145:102536
- Beeson P, Ames B (2015) Trac-ik: An open-source library for improved solving of generic inverse kinematics. In: 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pp. 928–935. IEEE
- Do C, Burgard W (2019) Accurate pouring with an autonomous robot using an rgb-d camera. In: Intelligent Autonomous Systems 15: Proceedings of the 15th International Conference IAS-15, pp. 210–221. Springer
- Rusu RB, Cousins S (2011) 3d is here: Point cloud library (pcl). In: 2011 IEEE International Conference on Robotics and Automation, pp. 1–4. IEEE
- Ferrell WR (1965) Remote manipulation with transmission delay. IEEE Trans Human Factors Electr 1:24–32
- Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, Whitehead S, Berg AC, Lo W-Y et al (2023) Segment anything. arXiv. https://doi. org/10.48550/arXiv.2304.02643
- Zhou L, Sun G, Li Y, Li W, Su Z (2022) Point cloud denoising review: from classical to deep learning-based approaches. Graph Models 121:101140
- Mower CE, Moura J, Vijayakumar S (2021) Skill-based shared control. In: Robotics: Science and Systems 2021. The Robotics: Science and Systems Foundation. https://doi.org/10.15607/RSS.2021.XVII.028
- Hansel K, Urain J, Peters J, Chalvatzaki G (2022) Hierarchical policy blending as inference for reactive robot control. In: 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 10181–10188. IEEE
- Shi C, Luo X, Qi P, Li T, Song S, Najdovski Z, Fukuda T, Ren H (2016) Shape sensing techniques for continuum robots in minimally invasive surgery: a survey. IEEE Trans Biomed Eng 64(8):1665–1678
- Nguyen CC, Wong S, Thai MT, Hoang TT, Phan PT, Davies J, Wu L, Tsai D, Phan H-P, Lovell NH et al (2023) Advanced user interfaces for teleoperated surgical robotic systems. Adv Sensor Res. https://doi.org/10.1002/ adsr.202200036
- Si W, Guan Y, Wang N (2022) Adaptive compliant skill learning for contactrich manipulation with human in the loop. IEEE Robot Automat Lett 7(3):5834–5841

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- ► High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at > springeropen.com