

# HI-DWA: Human-Influenced Dynamic Window Approach for Shared Control of a Telepresence Robot

Juho Kalliokoski, Basak Sakcak, Markku Suomalainen, Katherine J. Mimnaugh, Alexis P. Chambers, Timo Ojala, and Steven M. LaValle

**Abstract**—This paper considers the problem of enabling the user to modify the path of a telepresence robot. The robot is capable of autonomously navigating to a goal predefined by the user, but the user might still want to modify the path, for example, to go further away from other people, or to go closer to landmarks she wants to see on the way. We propose Human-Influenced Dynamic Window Approach (HI-DWA), a shared control method aimed for telepresence robots based on Dynamic Window Approach (DWA) that allows the user to influence the control input given to the robot. To verify the proposed method, we performed a user study (N=32) in Virtual Reality (VR) to compare HI-DWA with switching between autonomous navigation and manual control for controlling a simulated telepresence robot moving in a virtual environment. Results showed that users reached their goal faster using HI-DWA controller and found it easier to use. Preference between the two methods was split equally. Qualitative analysis revealed that a major reason for the participants that preferred switching between two modes was the feeling of control. We also analyzed the effect of different input methods, joystick and gesture, on the preference and perceived workload.

## I. INTRODUCTION

Telepresence robots are one of the most prominent tools that can enable *hybrid meetings* such that part of the people are physically present but others who wish to participate can only do so virtually. The use cases range from important personal milestones (birthdays, weddings, graduations) to business meetings and factory tours. The commercial telepresence robots, essentially Skype-on-wheels, are sufficient for video conferencing. However, a recent study showed that in a hybrid meeting, virtually present people did not speak as much and found a shared task more difficult compared to physically present people [1]. These limitations increased the interest in immersive robotic telepresence for which the user embodies a mobile robot in a remote location through a Head-Mounted Display (HMD). Despite being more complex and prone to issues such as VR sickness [2], [3], the immersiveness of an HMD has the potential to overcome the gap between physically and virtually present people.

Even though shared control is a well-known problem in robotics, there is limited research on aspects related to telepresence, either for “regular” or immersive telepresence. Manually controlling the robot through joystick commands

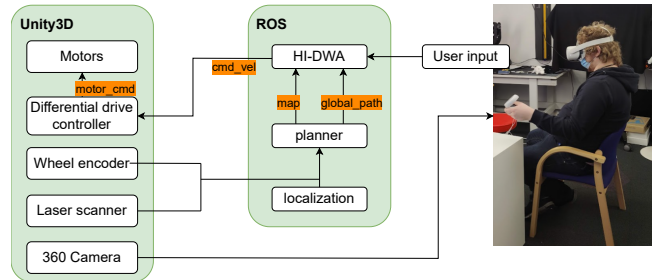


Fig. 1. Navigation framework for a telepresence robot employing the proposed HI-DWA shared control method and its communication with the simulation environment Unity.

is the simplest method to enable the user to affect the robot motion. However, there is evidence that people find it tiring [4] leading to the research towards autonomous or semi-autonomous methods, such as the “waypoint navigation” found in the commercial telepresence robot Double for which the user points at a point within the visible area and the robot navigates towards there autonomously. This type of navigation is also shown to be useful for immersive telepresence [5]. However, as telepresence robots share an environment with people, a difficult problem of human-aware navigation [6] should be addressed. In addition to safely avoiding people in the environment, a telepresence robot should also ensure that the person on board feels comfortable with the robot motion. Since aspects related to comfort and preference vary from one person to another, it seems natural to endow the user with effortless methods to make slight changes to the robot’s autonomous path, to easily address issues such as going too close to people or too far from points of interest not known to the autonomous planner.

In this paper, we address the problem of semi-autonomous navigation of a telepresence robot that allows the user to influence the trajectory executed by the robot with minimal effort. We propose HI-DWA for robot control, that is an adaptation of the popular DWA [7], which searches for the best control input that optimizes a relevant objective by integrating the system dynamics for each admissible control input and scoring the resulting trajectories. The key idea of HI-DWA is to penalize the deviation from the user input. Thus, we retain the collision avoidance property of DWA, while allowing the user to influence the robot’s motion. We performed a user study (N=32) in which the participants wearing HMDs compared the proposed control method with

\*This work was supported by a European Research Council Advanced Grant (ERC AdG, ILLUSIVE: Foundations of Perception Engineering, 101020977), Academy of Finland (projects PERCEPT 322637, CHiMP 342556), and Business Finland (project HUMOR 3656/31/2019).

Authors are with Center of Ubiquitous Computing, Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland. {name.surname}@oulu.fi

switching between manual and autonomous modes (a method often used in commercial telepresence robots, such as Double 3) for a simulated robot in a virtual environment that mimics immersive telepresence. They were encouraged to alter the robot's path. The results showed that even though participants found the proposed method easier, there was no difference in preference. Further analyses to interpret this result indicated that a major reason for seeing no difference in preference is that many users liked having full control over the robot's motion. Also, for the use case of immersive telepresence, we found that adjusting the path using gestures (moving hand) was preferred over joystick.

## II. RELATED WORK

In robotic telepresence a robot in the local environment is connected to and controlled by a user in a remote location [8]. The user is able to move the robot around and interact with other people in the remote environment. The term telepresence was originally coined by Minsky [9] and the most prominent telepresence implementation in research is a mobile robot with a camera, a standard screen, and some kind of controller [8]. Such telepresence robots are used for, for example, education [10], social interaction [8], and telemonitoring on the basis of medical consultation [11].

Shared control refers to systems in which the human and the robot are cooperating to achieve a specific goal. Whereas shared control is often used with manipulators [12] or autonomous cars [13], there is also a growing literature of shared control for autonomous robots. Shared control is often achieved by blending trajectories or policies [14], using the autonomous controller as a safeguard for detecting collisions, or simply switching between autonomous and manual modes [15], [16].

In trajectory blending, commands coming from the autonomous controller are blended with the trajectory indicated by the user [17] [18] [19] [20]. However, apart from some of these ([17] [20]), there is no proper collision detection for the resulting trajectory. In contrast, using the autonomous controller as a safeguard means that normally user has full control of the robot, but if some hazard or obstacle is detected by the robot it will take over the control preventing a collision. This method was already used in lunar rovers [21] and has been used in many mobile robots afterwards [22][23][24]. The proposed method differs from these by properly addressing the possible obstacles like in a safeguard method, but also taking the advantage of autonomous planning so that the user does not have to control the robot actively.

## III. PROPOSED SHARED CONTROL METHOD

We consider a telepresence robot shown in Fig. 2 that is a differential drive robot comprising two driving wheels and four omnidirectional wheels for balance. The robot kinematics is given by the model

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega,\end{aligned}\tag{1}$$

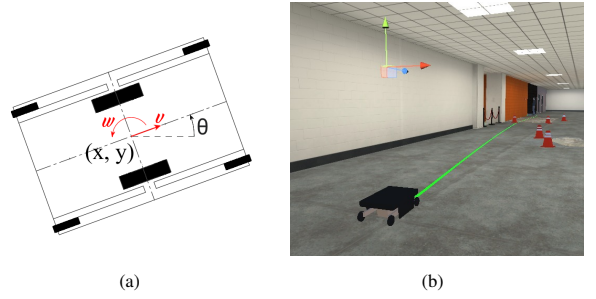


Fig. 2. (a) Diagram of the robot used in the experiment with state and control variables shown. (b) The simulated robot in virtual environment.

in which  $(x, y)$  is the robot position and  $\theta$  is the orientation with respect to a global reference frame, and the control input  $u = (v, \omega)$  corresponds to the linear and angular velocities with respect to the robot-fixed reference frame. The robot configuration is expressed as  $q = (x, y, \theta)$  and it is constrained in the set  $Q \subset \mathbb{R}^2 \times S^1$ . The control input is subject to actuation constraints (acceleration limits) and it takes part in a compact set of admissible controls  $U \subset \mathbb{R}^2$ .

Let  $E \subset \mathbb{R}^2$  be a planar environment in which the robot is moving. There are obstacles which are open sets and subsets of  $E$  that prohibit the robot to have certain configurations due to collisions. The obstacles change dynamically and this information is (locally) available to the robot. Therefore, let  $E_{obs}(t) \subset E$  be the union of all the obstacles known to the robot at time  $t$ . The part of the planar environment that the robot can be in without collisions at time  $t$  is then denoted by  $E_{free}(t) = E \setminus E_{obs}(t)$ . Let  $\mathcal{A} : Q \rightarrow E$  be a mapping from the robot configuration to its footprint in the environment. The free configuration space is defined as  $Q_{free}(t) = \{q \in Q \mid \mathcal{A}(q) \in E_{free}(t)\}$ . Let  $q_g = (x_g, y_g, \theta_g)$  and  $q_0 = (x_0, y_0, \theta_0)$  be the goal and the initial configurations, respectively. Conventionally, the motion planning problem is defined as finding a control-trajectory  $\tilde{u} : [0, T] \rightarrow U$  such that the state-trajectory  $\tilde{q} : [0, T] \rightarrow Q$  computed as forward integrating (1) starting from  $q_0$  satisfies  $\tilde{q}(t) \in Q_{free}(t), \forall t \in [0, T]$  and  $\tilde{q}(T) = q_g$ . Typically, a trajectory that is optimal with respect to a relevant metric is sought and the final time  $T$  is not fixed.

As common in the literature, we consider that the motion planning of the robot is achieved by two interacting modules: the *planner* and the *controller*. The planner is responsible for computing a length-optimal path to the goal (not necessarily feasible with respect to the robot kinematics), and the task of the controller is to compute the control input that tracks the output of the planner. We assume that a planner is in place and address the inclusion of the user input at the controller level. Before describing our problem, we will briefly explain the planner. Since the considered environment is dynamic, the planner is invoked at regular intervals  $t_k, k = 0, \dots, K$  to re-plan using the currently available information. Suppose the estimate of the robot configuration at  $t_k$  is available and denote it by  $\hat{q}_k = (\hat{x}_k, \hat{y}_k, \hat{\theta}_k)$ . Then, given the current environment  $E_{free}(t_k)$  and  $\hat{q}_k$ , the output of the planner is a length-optimal path  $\pi_k : [0, 1] \rightarrow E_{free}(t_k)$  such

that  $\pi_k(0) = (\hat{x}_k, \hat{y}_k)$  and  $\pi_k(1) = (x_g, y_g)$ . It is often referred to as the global path. Note that the computed path is not kinematically feasible<sup>1</sup>. However, it is assumed that the planner considers the robot size so that the path has sufficient clearance. Since the computational load of searching the whole position space is high, it is expected that the planner works at a lower frequency. Hence, it is mainly the task of the controller to avoid obstacles.

We consider a controller inline with methods based on searching the control input space such as DWA [7] and trajectory rollout [25]. These methods search a limited set of admissible controls (velocities) with respect to the actuation constraints such that given the current linear and angular velocities of the robot, only the ones that are achievable within a short length of time are considered. Consequently, assuming that the controls will stay constant, the respective state-trajectories are evaluated with respect to an objective function that captures the obstacle clearance and vicinity to the global path, for example. The control input corresponding to the best trajectory is then passed to the robot. This procedure is repeatedly executed for each  $t$  considering  $Q_{free}(t)$  and  $\pi_k$  such that  $t \in [t_k, t_{k+1})$ .

Due to the computationally infeasible exhaustive search over the set of admissible controls, typically, only a set of sampled controls are passed to the trajectory evaluation step. Let  $U^\Delta(t)$  be the set of sampled admissible controls at time  $t$  and let  $\Delta\tau$  be the time window used to integrate the dynamics. The set of control pairs such that respective trajectories are collision free are denoted as  $U_{free}^\Delta(t)$ . This implies that for each element  $u \in U_{free}^\Delta(t)$  the trajectory  $\tilde{q} : [t, t + \Delta\tau] \rightarrow Q$  computed considering a constant control for  $\Delta\tau$  satisfies  $\tilde{q}(\tau) \in Q_{free}(t), \forall \tau \in [t, t + \Delta\tau]$ . Then, the best control input to pass to the robot at time  $t$  is determined as

$$u_t^* = (v_t^*, \omega_t^*) = \arg \min_{u_t \in U_{free}^\Delta(t)} s_{nv} J_{nv}(u_t) \quad (2)$$

in which  $J_{nav}(u) = (J_1(u), J_2(u), \dots, J_d(u))^T$  is a  $d$ -dimensional vector of objective functions capturing the aspects relevant to path tracking, obstacle avoidance, and goal achievement and  $s_{nav} = (s_1, s_2, \dots, s_d)$  is the respective vector of weights.

We address the problem of designing a controller that not only tracks the path as described in the previous paragraph but also takes into account the input given by the operator. This corresponds to enabling the operator to apply slight modifications to the trajectory executed by the robot without directly controlling the robot motion. Suppose that the operator can indicate a preference for robot motion which is then mapped to a control command  $u_h = (v_h, \omega_h)$ . We propose a DWA-based controller to incorporate the user input in the selection of the best control input pair to pass to the robot. To this end, we augment the minimization problem in (2) with an additional cost function that penalizes the difference between the selected control pair and the operator input.

<sup>1</sup>A planner that computes kinematically feasible paths can also be used. In that case,  $\pi$  maps to  $Q_{free}(t_k)$  and satisfies (1)

This way, the control input that is closer to the input given by the operator is preferred while taking into account other aspects related to navigation. The cost function to evaluate the control input pairs and respective trajectories is described as

$$J(u_t) = \begin{cases} s_{nv} J_{nv}(u_t) + s_{sh} J_{sh}(u_t) & \text{if } \gamma(t) = 1 \\ s_{nv} J_{nv}(u_t) & \text{otherwise,} \end{cases} \quad (3)$$

in which  $(v_h, \omega_h)$  is the pair of control inputs given by the operator at time  $t$  and  $\gamma$  is a function that maps  $t$  to 1 if user input is given and to 0 otherwise. The cost component resulting from the user input is defined as

$$J_{sh}(u_t) = \left( |v_h - v_t|, |\omega_h - \omega_t| \right)^T$$

and  $s_{sh} = [s_v, s_\omega]$  is the vector of respective weights. Consequently, the best control input in  $U_{free}^\Delta(t)$  is determined as

$$u_t^* = (v_t^*, \omega_t^*) = \arg \min_{u_t \in U_{free}^\Delta(t)} J(u_t). \quad (4)$$

We introduce a delay in the transition to autonomous navigation to ensure that once the user input ceases, the resulting motion does not involve sudden rotations to compensate for the potential divergence from the global path due to user input. Let  $t$  be the instance that the operator stops interacting with the robot. Then, a constant pseudo-input is given to the controller such that  $v_h$  is the velocity command given at  $t$  and  $\omega_h = 0$  for all  $\tau \in [t, t + \delta]$ , in which  $\delta$  is the delay.

## IV. EXPERIMENT

### A. Controllers

The experiment is designed to compare three controllers. Two of them implement the shared control approach described in Section III with different input methods, and one implements a switch between the autonomous navigation and direct control of the operator. Here we explicitly describe these methods.

We integrate the controllers within the Nav2 project [26] for Robot Operating System 2 (ROS2). We use the default planner and localization plugins provided by Nav2 and modify the DWB controller which derives from DWA to integrate ours. In particular, we use the default critics (objective functions forming the vector  $J_{nv}$ ) and their default weights. Therefore,

$$J_{nv}(u_t) = s_{pa} \text{PathAlign}(u_t) + s_{pd} \text{PathDist}(u_t) \\ + s_{bo} \text{BaseObstacle} + s_{ga} \text{GoalAlign}(u_t) \\ + s_{gd} \text{GoalDist}(u_t) + s_{rg} \text{RotateToGoal},$$

in which for each trajectory corresponding to the numerical integration of the control  $u_t$  for  $\Delta\tau$ ,  $\text{PathAlign}$  ( $s_{pa} = 32.0$ ) and  $\text{PathDist}$  ( $s_{pd} = 32.0$ ) penalize the trajectory based on the distance from the global path and how well it is aligned to it, respectively. Similarly,  $\text{GoalAlign}$  ( $s_{ga} = 24.0$ ) scores a trajectory based on how well the robot aligns with the goal pose and  $\text{GoalDist}$  ( $s_{gd} = 24.0$ ) scores it based on how close the trajectory gets the robot to the

goal pose. `BaseObstacle` ( $s_{bo} = 0.02$ ) scores a trajectory based on its distance from the obstacles. Finally, it includes also a binary critic named `Oscillation` that prevents backwards-forwards motion by penalizing such trajectories with infinite cost. We refer the reader to Nav2 documentation [27] for more detailed explanations.

**Switching (SW):** Switching control refers to handing the direct control of the robot to the operator when the operator wants to alter the course of the robot motion; similar systems have been proposed in [15] and implemented in the commercial Double 3 telepresence robot. For the remaining times, the robot is moving autonomously. The control input to send to the robot at time  $t$ , that is,  $u_t^*$ , is determined as

$$u_t^* = \begin{cases} u_h = (v_h, \omega_h) & \text{if } \gamma(t) = 1 \\ \arg \min_{u_t \in U_{free}^\Delta(t)} s_{nv} J_{nv}(u_t) & \text{otherwise.} \end{cases} \quad (5)$$

To make the switching explicit, we implemented a button such that once it is pressed the control is transferred to the operator. If after pressing the button no user input is given, the respective user input is  $u_h = (v_h, \omega_h) = (0, 0)$  and the robot stops. The user input is given using the joystick of an Oculus Quest 2 controller. Let  $(p_x, p_y) \in [-1, 1] \times [-1, 1]$  be the joystick coordinate corresponding to the user input such that  $(0, 0)$  is the origin, it is mapped to  $u_h = (v_h, \omega_h)$  as follows:

$$v_h = \begin{cases} 0 & \text{if } |p_y| \leq 0.1 \\ p_y^2 \text{sgn}(p_y) v_{max} & \text{otherwise,} \end{cases} \quad (6)$$

and

$$\omega_h = \begin{cases} 0 & \text{if } |p_x| \leq 0.1 \\ p_x^2 \text{sgn}(p_x) \omega_{max} & \text{otherwise.} \end{cases} \quad (7)$$

The following two methods implement the shared control described in Section III.

**Shared Control-Joystick (SJ):** Similar to Switching Control, also in this case, the operator uses a joystick to provide input to the system. To keep controlling the robot simpler, we allow the user only to affect the rotational speed but not the linear speed. Therefore, for all  $u_h$ ,  $v_h$  is set to  $v_{max}$  to avoid prioritizing controls corresponding to rotate in place and  $\omega_h$  is computed using (7). The control input  $u_t^*$  passed to the robot at time  $t$  is determined using (4). To find a good combination of the relative weights, that is,  $s_{sh} = (s_v, s_\omega)$  for penalizing the costs  $J_{sh}(u_t)$  that take part in  $J(u_t)$  and to determine a sufficient delay  $\delta$ , we ran a pilot test ( $N = 8$ ). The participants tried this control method with four different parameter combinations, resulting in four conditions. For  $(s_v, s_\omega, \delta)$  we tested the following four combinations:  $(200, 400, 2s)$ ,  $(400, 800, 2s)$ ,  $(200, 400, 1s)$ ,  $(400, 800, 1s)$ . When asked which condition they felt was the best, 6 out of 8 participants picked condition 2, one participant picked conditions 3 and 4 as equally good, and 1 participant picked conditions 1 and 2 as equally good. Since condition 2 was selected by the majority of the users we used the parameters used in condition 2 for the main study ( $s_{sh} = (s_v, s_\omega) = (400, 800)$  and  $\delta = 2s$ ).

**Shared Control-Gesture (SG):** This method differs from SJ only in terms of the way the input is received from the operator in form of gesture control and a button to indicate the trajectory. When the operator presses the button, we obtain the current position and orientation of the hand from the tracking system of the Oculus Quest 2 controller. Then this is mapped to coordinates in the  $x - y$  plane. The initial coordination of the hand is set as the origin of this plane and the normalized horizontal movement of the hand is mapped to  $w_h$  similar to (7).

## B. Hypotheses

We pre-registered the following two hypotheses, with the procedure and analyses to be used in the study, in Open Science Foundation (OSF) <https://osf.io/q9ubx>. In a scenario of a semi-autonomous telepresence robot navigating an environment in which users immersed in the robot through a head-mounted display (HMD) can alter the path executed by the robot either by switching to manual guidance or by indicating the direction that they would like to go, we will test the following hypotheses:

- H1:** SJ condition is preferred as indicated by asking directly which condition was preferred.
- H2:** Perceived workload is lower under SJ as indicated by asking directly which condition was easier and NASA Task Load Index (NASA-TLX) questionnaire after each condition.

## C. Study Setup

The hypotheses were tested using a simulated environment in Unity. The virtual environment was loosely based on the University of Oulu campus, so although some participants could recognize parts of the environment, they could not use this knowledge into their advantage. The simulated telepresence robot used in the experiment comprised of a mobile base as described in Section III and a simulated 360° camera attached 1.5 meters above the robot from which the users could see the virtual world using a virtual reality headset. We also ensured that the dynamics governing the robot motion in simulation were sufficiently realistic. The autonomous navigation of the robot was based on the Robot Operating System (ROS) and its Nav2 project [26]. To avoid high computational loads that might hamper the viewing comfort, we used two separate computers; a ROS-based application that ran on a Linux laptop and Unity-based one on a Windows laptop. The connection was established via a ROS TCP Connector[28] (Fig. 1). In the experiment, the robot was navigating towards a fixed goal configuration using the autonomous navigation system together with the controllers described in the previous section. The virtual environment contained regions called *regions to avoid* that consisted of potholes, scaffolding, cardboard boxes, traffic cones, and bumpy areas. These were not visible to the robot's sensors and were not marked on the map so that they could not be avoided like other obstacles. However, they were passable for the robot. Fig. 2b shows an instance from the experiment with traffic cones and potholes.

#### D. Procedure

The participants were presented with three conditions corresponding to the SW, SJ, and SG control methods. The conditions SW and SJ were presented in counterbalanced order such that both videos were seen first and second an equal number of times by the participants, and the SG condition was presented always as the last one. Upon arrival, the participants were greeted by a researcher and signed a form to indicate their consent to participate. Next, the experimenter asked the participants if they were feeling nauseous or had a headache in an effort to pre-screen people feeling sick already before the experiment began. Then, the participants were told the general instructions by the experimenter, and shown how to put on the HMD.

After the experimenter made sure that the participant knew how to put on the HMD properly, they read out the instructions for the first task. The participants were told that they were late for a meeting where they were going to participate using a telepresence robot. The robot is capable of navigating autonomously, that is, computing a path to a goal and tracking while avoiding obstacles. The goal was set as a point in the meeting room and to have more control over the experiment, the users were not allowed to change the goal of the robot. There were potholes and scaffolding along the robot's default path without user deformations, which are not visible to the robot's sensors. Despite not being explicitly told to avoid those regions, the participants were encouraged to do so by saying that they could get slowed down or feel uncomfortable (bumpy tiles) if they did not. They were also told that the path the robot would try to follow was shown as a green line on the floor. Before each task, the participant was asked to practice the controls in a short practice environment, with the robot again following a path with similar scaffolding obstacles as in the real path, lasting approximately 2-3 minutes.

After the practice, the participants were asked if they are ready for the task, and the proper task scenario was started by the experimenter. After the task, the participants were asked to take off the HMD and fill out a questionnaire regarding their experience with that specific control method. The same procedure was repeated two more times with the other control methods. Finally, the participants were given 20€ Amazon vouchers for participating. After each experiment, all the equipment was disinfected as a precaution regarding COVID-19. Precautions were also taken during the experiment by using disposable face covers with the HMD, the experimenter always wearing a mask, and the experimenter keeping a safe distance from the participant except if help was needed.

#### E. Measures

For each task, we measured the path that the robot took. To see where the subjects altered the path of the robot, we placed time stamps on the moments when the subject used their controller. We also measured the head movements of the subjects using the Oculus Quest 2's tracking system to

see where the subjects were looking at any point during the task.

At the beginning of each questionnaire, we asked the participants to fill out a Simulator Sickness Questionnaire (SSQ) [29]. It is a questionnaire often used to measure the sickness that results from using VR, and consists of questions regarding 16 different sickness symptoms that are scored based on severity of experience. Weighted scores are used to calculate the total score for the sickness. Higher scores indicate greater levels of sickness experienced. Participants were also asked to fill out a NASA Task Load Index questionnaire, which is used to measure six dimensions of workload (mental demand, physical demand, temporal demand, performance, effort, and frustration) of the task. Each dimension is compared between the methods individually to find if one of the methods is perceived as more demanding than the others. These questions were followed by a forced-choice question if there was any point where the participant wanted to alter the path but didn't, and 7-point Likert-scale questions about the participant's perceived control over the robot, the ease of altering the path, and their comfort while altering the path.

After the second task, we additionally asked forced-choice questions about the participants' preference between the two methods, and a comparison between the two methods on their control and ease of use. We also asked open-ended questions about the reasons for their choices and if there were any specific situations where they would prefer one of the methods over another.

Finally, after the third task, we asked forced-choice questions about whether the SG method made them feel more as if they were there in the environment or if they felt more in control of the robot than in the previous tasks. We also asked their preference over all three different methods and open-ended questions about reasons for their choices. In the end we asked about participant's VR and gaming experience and their demographics. The questionnaire after the third task, including also all previously asked questionnaires except the forced-choice between first two methods, can be found from the same OSF page as the hypotheses (Section IV-B).

#### F. Participants

Participants were recruited from the University of Oulu campus and community. We aimed to have 32 participants, but due to the exclusion of four people from the study, we ended up running 36 total. Three of the excluded participants quit the experiment due to sickness symptoms, and one was excluded for not following the instructions and moving the robot outside of the intended environment, which caused them to get stuck and thus they were not able to finish their task without a reset. Of the 32 included participants, 19 were men and 13 were women. The responses of the participants to how often they use VR systems were: 21.9% never used any before, 43.8% just a couple of times and at least once, 18.8% once or twice a year, 9.4% once or twice a month, and 3.4% once or twice a week.



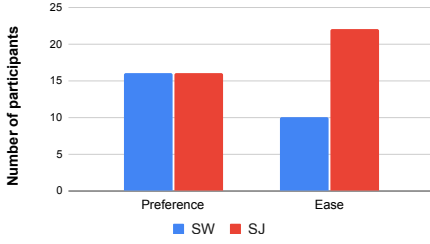


Fig. 3. The distributions of responses to the questions regarding preference, easiness, and comfort.

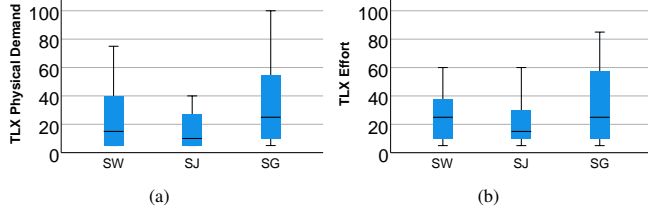


Fig. 4. Comparison of relevant TLX scores (a) physical demand (b) effort.

## V. RESULTS

Two confirmatory hypothesis tests were preregistered. All tests were run in SPSS with significance levels set to 0.05 and with a 95% confidence interval.

### A. Confirmatory analysis

Fig. 3 shows the distributions of the responses given by the participants to the forced-choice questions regarding preference and ease of use, and comfort. When asked “Which control method did you prefer?” 16 out of 32 participants (50%) selected the SJ condition showing no tendency in either direction in preference. When asked “Which control method was easier to use?” 22 out of 32 participants (68.75%) selected the SJ condition. An exact binomial test with exact Clopper-Pearson 95% CI was performed, showing that the condition SJ was found significantly easier compared to the SW condition,  $p = 0.026$  (one-sided) and had a 95% CI of 50.0% to 83.9%.

We compared the differences between the TLX-scores across conditions for each dimension of workload and found a statistically significant difference only in the effort dimension. A Wilcoxon Signed-Ranks test is performed to compare the TLX effort scores for SJ ( $Mdn = 15.0$ ) and SW ( $Mdn = 25.0$ ) (see Fig. 4b for the score distributions). The test indicated that SJ elicited statistically significantly lower effort scores compared to SW,  $Z = -1.687$ ,  $p = .047$ ,  $r = 0.30$  (one-sided).

### B. Exploratory analysis

In addition to the confirmatory analyses, we performed exploratory analyses to interpret the results better.

1) *Quantitative*: The task completion time data was analyzed to see if one method would result in faster task completion. The respective distributions for SJ and SW conditions did not follow a normal distribution as indicated by a Shapiro-Wilk test,  $W = 0.773$ ,  $p < 0.001$  and  $W = 0.852$ ,  $p < 0.001$ , respectively. Therefore, a Wilcoxon Signed-Rank test (two-sided) was performed to compare the differences

between the task completion times for SJ ( $Mean = 172.37s$ ) and SW ( $Mean = 187.17s$ ) conditions. The test indicated that SJ elicited significantly faster task completion compared to SW,  $Z = -4.207$ ,  $p < 0.001$ ,  $r = 0.74$ .

To see if one control method helped to avoid *regions to avoid* more, we analyzed the number of regions that the path executed by the robot intersected with. Comparing the number of regions not avoided for SJ ( $Mean = 0.219$ ) with SW ( $Mean = 0.406$ ) we did not observe any significant difference as indicated by a Wilcoxon Signed-Ranks test (two-sided),  $Z = -1.281$ ,  $p = 0.213$ ,  $r = 0.226$ . Considering the obstacles, in total, we observed three collisions under SW condition. There were no collisions for SJ as it is inherently collision-free.

We tested whether one condition induced more sickness. Comparing the weighted total SSQ scores across the conditions SJ ( $Mean = 44.5294$ ) and SW ( $Mean = 48.3863$ ), we did not observe a significant difference between the scores, as indicated by a Wilcoxon Signed-Ranks test (two-sided),  $Z = -0.809$ ,  $p = 0.427$ ,  $r = 0.14$ . However, we observed a carryover effect on sickness with 15 minutes breaks; total weighted SSQ scores after the task 2 ( $Mean = 54.1131$ ) was significantly higher compared to the task 1 ( $Mean = 38.80$ ), as indicated by a Wilcoxon Signed-Ranks test (two-sided),  $Z = -3.648$ ,  $p < .001$ ,  $r = 0.64$ .

To find out the reasons for participants’ choices for their preference between the SW and SJ methods, we filtered the participants into two groups based on their preference. We then compared the Likert-scale ratings for the easiness and the feeling of control over the robot within those groups. From the participants who preferred the SW method, when comparing their ratings of the control over the robot, we found out that there is, at best, weak evidence that the SW method ( $Mean = 1.9375$ ) had more control over the robot than the SJ method ( $Mean = 2.8750$ ), as seen from a Wilcoxon Signed-Ranks test (two-sided),  $Z = -1.943$ ,  $p = 0.052$ ,  $r = 0.343$ . However, when comparing the easiness ratings within these participants, there was no significant difference between the SW method ( $Mean = 2.0625$ ) and the SJ method ( $Mean = 2.5625$ ) as seen from a Wilcoxon Signed-Ranks test (two-sided),  $Z = -1.144$ ,  $p = 0.253$ ,  $r = 0.202$ . From the people who preferred the SJ method, a Wilcoxon Signed-Ranks test indicated that there was no significant difference between the means of the control over the robot ratings between the SW method ( $Mean = 2.3750$ ) and the SJ method ( $Mean = 2.4375$ ),  $Z = -0.265$ ,  $p = 0.791$ ,  $r = 0.047$ , but there was again, at best, weak evidence that the easiness score was lower with the SJ method ( $Mean = 1.6250$ ) than with the SW method ( $Mean = 2.3125$ ),  $Z = -1.942$ ,  $p = 0.052$ ,  $r = 0.343$ . Similarly, of the participants who preferred the SW method, only 10 out of 16 (62.5%) said that it also felt easier, while all 16 participants who preferred the SJ method also picked it as easier method.

With our exploratory third control method, we wanted to see if participants would prefer using their gestures (hand) to give inputs to the shared controller instead of the joystick.

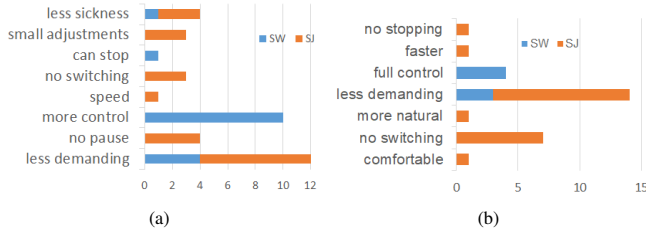


Fig. 5. Frequently found codes for questions (a) “Please explain why you prefer that control method”. (b) “Please explain why that control method felt easiest to use”

When we asked “Which of the three methods did you prefer?”, the SG method was preferred by 15 out of 32 participants (46.9%) while the SW method was preferred by only 8 (25.0%) and the SJ method was preferred by only 9 participants (28.1%). When asked “How easy was it to alter the path of the robot”, participants felt that the SJ method was significantly easier to use ( $Mean = 2.0937$ ) than the SG method ( $Mean = 2.9687$ ), as indicated by a Wilcoxon Signed-Ranks test (two-sided),  $Z = -2.381$ ,  $p = 0.015$ ,  $r = 0.421$ . When comparing the physical demand score between the SJ method ( $Mean = 3.969$ ) and the SG method ( $Mean = 7.031$ ), the SG method had significantly higher score, as indicated by a Wilcoxon Signed-Ranks test (two-sided),  $Z = -2.785$ ,  $p = 0.004$ ,  $r = 0.492$ . Similarly the participants felt that the SG method required more effort ( $Mean = 6.969$ ) than the SJ method ( $Mean = 4.875$ ), as indicated by a Wilcoxon Signed-Ranks test (two-sided),  $Z = -2.034$ ,  $p = 0.041$ ,  $r = 0.360$ . Other TLX scores did not have significant differences. TLX score distributions can be seen in Fig. 4.

2) *Qualitative*: The open-ended data was analyzed using the thematic analysis method with inductive approach [30].

Fig. 5a shows the frequently found codes in the responses to the open-ended question asking why participants preferred the control method, divided by which method was preferred. The most frequently found code was *less demanding*. Eight (50%) participants who preferred SJ and four (25%) participants who preferred SW found these methods less demanding (“It felt that it would be less laborious to just make small adjustments to the path when needed rather than drive manually”). The biggest factor for participants who preferred SW was having *more control* that is found in the responses of 10 out of 16 participants (“With the first method it was always clear who was in control. The second method felt like a constant struggle.”).

The responses to the open-ended questions regarding why the control method felt easiest to use the greatest number of comments (44%) said that it was *less demanding*, followed by *no switching* (22%) and *full control* (13%). See Fig. 5b for the frequently found keywords in the participants’ responses.

## VI. DISCUSSION

An interesting takeaway from the performed study was the discrepancy between ease of use and preference: even though preference between SJ and SW was split exactly equal, participants found SJ statistically significantly easier

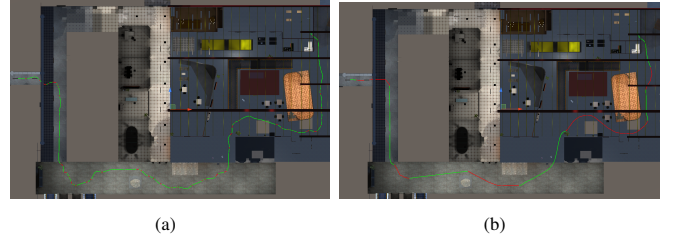


Fig. 6. Example paths taken using (a) SJ and (b) SW. Red parts are where the participant used their controller to give input.

compared to SW. Whereas it was expected that experienced gamers not to have any issues with manual joystick control over such a short time period (approximately 3 minutes), we did not observe any correlation between the gaming background and preference of control method, even though several participants explicitly said that it did (“It did not demand much mind effort to accomplish. and having joystick in the hand is a good experience from past playing video games.”).

One reason why participants preferred SW but found SJ easier was the *feeling of control*. For example, one participant preferred SW because “You can have full control and navigate the path from the location of your choosing”, but found shared control easier because “its easier because you can relax and see where you want to change path. However on like the first, you can feel sleepy or over relaxed with the second” for whom the second condition was SJ. This result could be related to our study setup, for which the goal configuration was constant throughout the experiment; in a real scenario, users would have chosen the goal themselves, either via waypoint navigation (see Section I) or via choosing the destination on a minimap.

Another frequently mentioned notion, related to the feeling of control, was the ability to stop. Whereas taking full manual control of the robot stopped the robot, with shared control, the robot always drove towards the goal. Besides not including waypoint navigation, this was another deliberate choice for the study setup: we were worried about too much freedom of choice for the participants to confound the study with multiple simultaneous locomotion modalities. However, if they had had more control over the robot’s destination and stopping, there is a chance users would have felt more in control even in the SJ condition. Also, with waypoint navigation, the participants could have chosen whether to use shared control or waypoint navigation depending on the size of the obstacle: however, giving them this freedom would have made analysis of the results more difficult. Several participants noticed that the proposed method was especially useful for small corrections “It felt that it would be less laborious to just make small adjustments to the path when needed rather than drive manually.”, which is indeed the intended use case. Fig. 6 presents an example of this case such that SJ is used to make small alterations to the path whereas SW is used along larger portions of the path executed by the robot.

Since the main contribution of this paper is the underlying

shared control mechanism, which can be used with either immersive or regular telepresence robots, the gesture condition, mainly meant for immersive telepresence robots, was left last as exploratory. However, the results were encouraging; whereas using extensive gesture motions for control is not encouraged in VR, due to fatigue, in this case the participants had armrests to rest their arms on, and the control is planned to be needed only on occasions. Also, the joystick of the HMD controller is quite small and not very accurate, which could have been difficult to use for non-gamers. There are also interesting reasons in the qualitative data for preferring the gestures, such as “*It was more comfortable, a lot easier and it felt more real*”. The “felt more real” part could be related to *presence*, which is one of the main reasons for using HMD for telepresence. Thus, future research on whether body-based control increases the feeling of presence would be interesting.

## VII. CONCLUSION

In this paper we presented HI-DWA, a novel shared control method based on DWA aimed especially for telepresence robots such that the users can influence the control input to the robot resulting in trajectories according to their choice. We performed a user study in VR, where the users avoided regions unseen by the robot’s sensors either with the proposed method or switching between autonomous and manual modes. We showed that the participants found shared control easier, even though preference was split equally between the proposed method and switching; the feeling of control was often mentioned as the reason for users who preferred the switching, even though there was no statistically significant difference in a Likert-scale question of feeling of control. In the future study, we will allow the users to select the goal. We expect it to increase the feeling of having control over the robot and thus, affect their preference for different control methods.

## REFERENCES

- [1] B. Stoll, S. Reig, L. He, I. Kaplan, M. F. Jung, and S. R. Fussell, “Wait, can you move the robot? examining telepresence robot use in collaborative teams,” in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 14–22.
- [2] J. J. LaViola Jr, “A discussion of cybersickness in virtual environments,” *ACM Sigchi Bulletin*, vol. 32, no. 1, pp. 47–56, 2000.
- [3] S. M. LaValle, *Virtual reality*. Cambridge University Press, 2021.
- [4] I. Rae and C. Neustaedter, “Robotic telepresence at scale,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 313–324.
- [5] G. Baker, T. Bridgwater, P. Bremner, and M. Giuliani, “Towards an immersive user interface for waypoint navigation of a mobile robot,” in *The Second International Workshop on Virtual, Augmented and Mixed Reality for Human-Robot Interaction*, 2020.
- [6] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, “Human-aware robot navigation: A survey,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.
- [7] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [8] A. Kristoffersson, S. Coradeschi, and A. Loutfi, “A review of mobile robotic telepresence,” *Advances in Human-Computer Interaction*, vol. 2013, 2013.
- [9] M. Minsky, “Telepresence,” *Omni Magazine*, vol. 38, no. 4, p. 217230Murray, 1980.
- [10] J. Botev and F. J. Rodríguez Lera, “Immersive robotic telepresence for remote educational scenarios,” *Sustainability*, vol. 13, no. 9, p. 4717, 2021.
- [11] K. A. R. Carranza, N. J. B. Day, L. M. S. Lin, A. R. Ponce, W. R. O. Reyes, A. C. Abad, and R. G. Baldovino, “Akibot: A telepresence robot for medical teleconsultation,” in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. IEEE, 2018, pp. 1–4.
- [12] M. J. Zeestraten, I. Havoutis, and S. Calinon, “Programming by demonstration for shared control with an application in teleoperation,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1848–1855, 2018.
- [13] M. Marciano, S. Díaz, J. Pérez, and E. Irigoyen, “A review of shared control for automated vehicles: Theory and applications,” *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 6, pp. 475–491, 2020.
- [14] A. D. Dragan and S. S. Srinivasa, “A policy-blending formalism for shared control,” *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.
- [15] M. Chiou, R. Stolkin, G. Bieksaite, N. Hawes, K. L. Shapiro, and T. S. Harrison, “Experimental analysis of a variable autonomy framework for controlling a remotely operating mobile robot,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3581–3588.
- [16] G. Petousakis, M. Chiou, G. Nikolaou, and R. Stolkin, “Human operator cognitive availability aware mixed-initiative control,” in *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*. IEEE, 2020, pp. 1–4.
- [17] J. G. Storms and D. M. Tilbury, “Blending of human and obstacle avoidance control for a high speed mobile robot,” in *2014 American Control Conference*, 2014, pp. 3488–3493.
- [18] P. Pappas, M. Chiou, G.-T. Epsimos, G. Nikolaou, and R. Stolkin, “Vfh+ based shared control for remotely operated mobile robots,” *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Nov 2020. [Online]. Available: <http://dx.doi.org/10.1109/SSRR50563.2020.9292585>
- [19] M. Chiou, M. Talha, and R. Stolkin, “Learning effects in variable autonomy human-robot systems: how much training is enough?” in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 720–727.
- [20] H. Wang and X. P. Liu, “Adaptive shared control for a novel mobile assistive robot,” *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1725–1736, 2014.
- [21] E. Krotkov, R. Simmons, F. Cozman, and S. Koenig, “Safeguarded teleoperation for lunar rovers: From human factors to field trials,” in *IEEE Planetary Rover Technology and Systems Workshop*, vol. 26, 1996, p. 28.
- [22] J. Luo, Z. Lin, Y. Li, and C. Yang, “A teleoperation framework for mobile robots based on shared control,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 377–384, 2020.
- [23] J. Jiang and A. Astolfi, “Shared-control for the kinematic model of a mobile robot,” in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 62–67.
- [24] T. Fong, C. Thorpe, and C. Baur, “A safeguarded teleoperation controller,” in *IEEE International Conference on Advanced Robotics (ICAR)*, no. CONF, 2001.
- [25] B. P. Gerkey and K. Konolige, “Planning and control in unstructured terrain,” in *ICRA workshop on path planning on costmaps*. Citeseer, 2008.
- [26] S. Macenski, F. Martín, R. White, and J. G. Clavero, “The marathon 2: A navigation system,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2718–2725.
- [27] (2020) Navigation 2. [Online]. Available: <https://navigation.ros.org/>
- [28] U. Technologies. (2021) Ros tcp connector. [Online]. Available: <https://github.com/Unity-Technologies/ROS-TCP-Connector>
- [29] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, “Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness,” *The international journal of aviation psychology*, vol. 3, no. 3, pp. 203–220, 1993.
- [30] M. Q. Patton, “Qualitative research,” *Encyclopedia of statistics in behavioral science*, 2005.