

TELESIM: A Modular and Plug-and-Play Framework for Robotic Arm Teleoperation using a Digital Twin

Florent P Audonnet¹, Jonathan Grizou¹, Andrew Hamilton¹ and Gerardo Aragon-Camarasa¹

Abstract—Teleoperating robotic arms can be a challenging task for non-experts, particularly when using complex control devices or interfaces. To address the limitations and challenges of existing teleoperation frameworks, such as cognitive strain, control complexity, robot compatibility, and user evaluation, we propose TELESIM, a modular and plug-and-play framework that enables direct teleoperation of any robotic arm using a digital twin as the interface between users and the robotic system. Due to TELESIM’s modular design, it is possible to control the digital twin using any device that outputs a 3D pose, such as a virtual reality controller or a finger-mapping hardware controller. To evaluate the efficacy and user-friendliness of TELESIM, we conducted a user study with 37 participants. The study involved a simple pick-and-place task, which was performed using two different robots equipped with two different control modalities. Our experimental results show that most users were able to succeed by building at least a tower of 3 cubes in 10 minutes, with only 5 minutes of training beforehand, regardless of the control modality or robot used, demonstrating the usability and user-friendliness of TELESIM.

I. INTRODUCTION

Robot teleoperation is difficult for non-experts [1], [2]. Recently, the ANA Avatar XPRIZE Challenge [3] set a series of challenging tasks to test the limits of teleoperation. The best systems that completed the challenges were rewarded with a prize pool of \$10 million. At its core, the challenge involves the direct teleoperation of a robot with minimal latency and the capacity to experience the environment from the robot’s perspective. However, direct teleoperation still places a heavy physical and mental strain on the user, as Pettinger *et al.* [4] reported that a user performing a pick and place task was faster and had fewer errors while reporting the task was more accessible when shared autonomy systems were enabled. Hence, researchers from HCI, medicine, robotics, and others have explored different means of control for teleoperation to address these limitations. While most research efforts focus on a physical control device such as a Virtual Reality Controller [5], [6], a Joystick [7], [8], or phone [9], others decided to use cameras to track the whole body [10], [11], or just the gaze [12]. There has yet to be an overall consensus on the most appropriate type of control for direct teleoperation with specific applications requiring specific implementations.

In this paper, we develop a modular and plug-and-play direct teleoperation framework called TELESIM that non-

This research has been supported by EPSRC DTA No. 2605103 and EPSRC Grant No. EP/S019472/1

¹ School of Computing Science, University of Glasgow, G12 8QQ, Scotland, United Kingdom
f.audonnet.1@research.gla.ac.uk;
gerardo.aragoncamarasa@glasgow.ac.uk

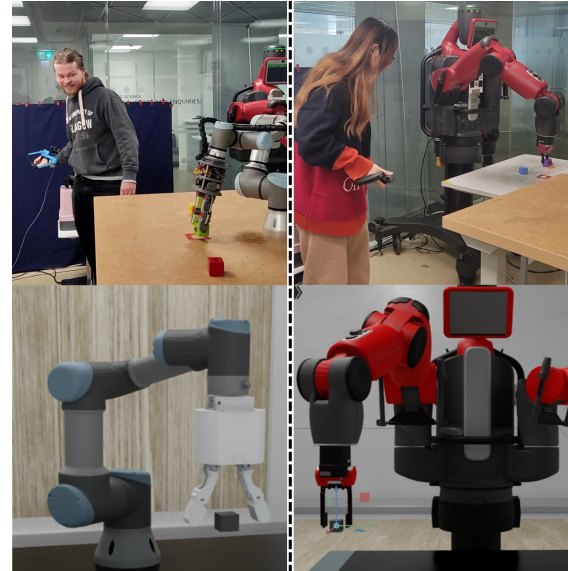


Fig. 1: Our modular and plug-and-play TELESIM framework is being used to control a UR3 Robot (top-left) and a Baxter Robot (top-right) and its digital twin (bottom-right).

experts can use without specialised training using off-the-shelf Virtual Reality (VR) technologies. Specifically, our objective is to allow for the direct teleoperation of any robotic arm using a digital twin as the interface between the user and the robotic system. According to Semeraro (2021), a digital twin can be defined in various ways. However, in the context of this paper, a digital twin is defined as a virtual representation or model that engages with a physical system throughout its life cycle. We then demonstrate TELESIM’s user-friendliness using a user study and the users’ success rate at completing the task using two different types of control and grasping systems. Specifically, we use a virtual reality controller and a finger mapping hardware controller mounted on two robotic manipulators using different grasping systems. We compare their performance to study whether additional degrees of freedom in the control scheme enhance performance while performing a simple task. Our contributions are:

- A modular and plug-and-play framework for teleoperation for any robotic arm using a digital twin.
- An experimental validation for testing the framework’s performance through a simple non-expert task.
- A rigorous evaluation involving 37 participants demonstrating the user-friendliness of TELESIM.

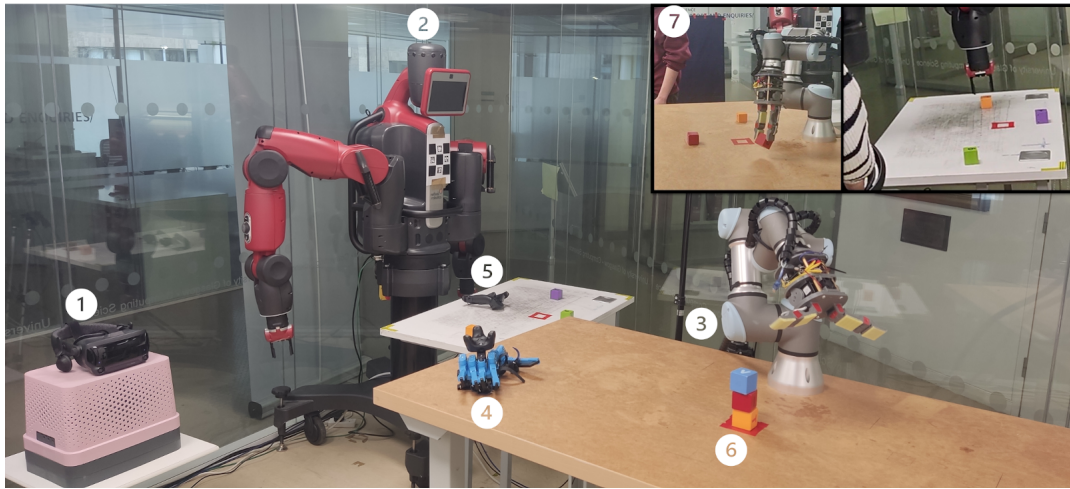


Fig. 2: Overview of the experimental setup. The Steam Index VR Headset [13] is marked as (1) on the far left, which acts as the world’s origin. The Baxter robot on the left (2) is controlled by the Steam Index controller (5). In front of it, the UR3 is on the right (3), with the Yale OpenHand T42 gripper [14], controlled by the Senseglove and HTC Vive tracker (4) on the left side of the brown table. Additionally, in the upper right corner (7), a view of the starting setup of the task, which consists of 3 cubes in a triangle pattern (described in Section IV), while on the brown table, the cubes are arranged in the goal configuration (6).

II. BACKGROUND

Direct teleoperation is considered a stepping stone for shared autonomy [15]. This is because direct teleoperation causes significant cognitive strain on the user [4], and the user may not be capable of millimetre-scale adjustment to the position of the robot end effector. While in medicine, the user’s movement is scaled down to allow for more precision [16], [6], it may not be suitable for all types of manipulation tasks as some require significant arm movements to move an object from one place to another.

Hence, researchers have explored different control methods to reduce the cognitive strain while giving the highest precision. For instance, low degree-of-freedom control methods such as a keyboard [17], a joystick [18], [7], a touchscreen [19], or a gamepad [20] have brought an improved level of control [17] to address the user’s mental strain. However, with the advent of VR technologies, researchers have investigated whether these technologies are appropriate for direct teleoperation. For example, they have proposed using a VR controller such as [21], [5] or a phone [22]. While others have investigated the use of motion mapping of the user’s body [10], [23] or only gaze control [12]. However, for the latter, the added mobility generates a higher cognitive load [4], and mapping motions to robot movements is challenging due to differences in kinematics chains between robot arms and users [24].

Recently, Gottardi *et al.* [25] have investigated combining multiple control systems, such as a VR controller and a tracking band on the upper arm, to track the user’s movements. Rakita *et al.* [6] also compared various control methods: a stylus, a touchscreen, and a VR controller. These were then integrated into a custom inverse kinematics solver that adjusted the tolerance level when matching the end-effector pose to that of the user. The authors showed that users

preferred the VR controller as they were more successful at completing pick-and-place tasks, such as picking up bottles or plates. To mitigate the limitation of direct teleoperation, researchers have focused on how much shared autonomy improved the success of a given task. For this, research works have aimed at comparing direct teleoperation with respect to an assisted version to analyse the impact of shared autonomy on task success. For example, Chen *et al.* [10] created a system in which the operator, using a joystick, manipulated the robot’s end effector to an object, and then the robot could either grasp the object autonomously or assist the user in fine-tuning the robot position for a more optimal grasp. Later, [11], [4], [25] built on [10] where the user teleoperated the robots directly to a planned position but allowed the robot to perform the grasp automatically or, in the case of [4], turn a valve handle. Lin *et al.* and Gottardi *et al.* conducted a user survey and confirmed that users preferred the shared autonomy approach, as it reduced complexity and mental strain. Furthermore, [25] observed results similar to [8], who hypothesised that users preferred to give up control if it meant increasing the task completion rate. However, Javdani *et al.* [8], [26] have falsified this hypothesis using a system similar to [11], [4], [25]. The authors concluded that users preferred to lose control if it meant an increase in a task’s success rate only for a more complex task, while, for simple tasks, users still preferred to have more control. These works have focused on one robotic system and conducted their experimental survey on a small user base (between 8 and 23 participants). Furthermore, they focused on different autonomy levels and not on different control methods. Finally, they used either MoveIt! or their planning interface for controlling the robot.

Although this paper focuses on TELESIM as a framework, our evaluation also addresses four main limitations of

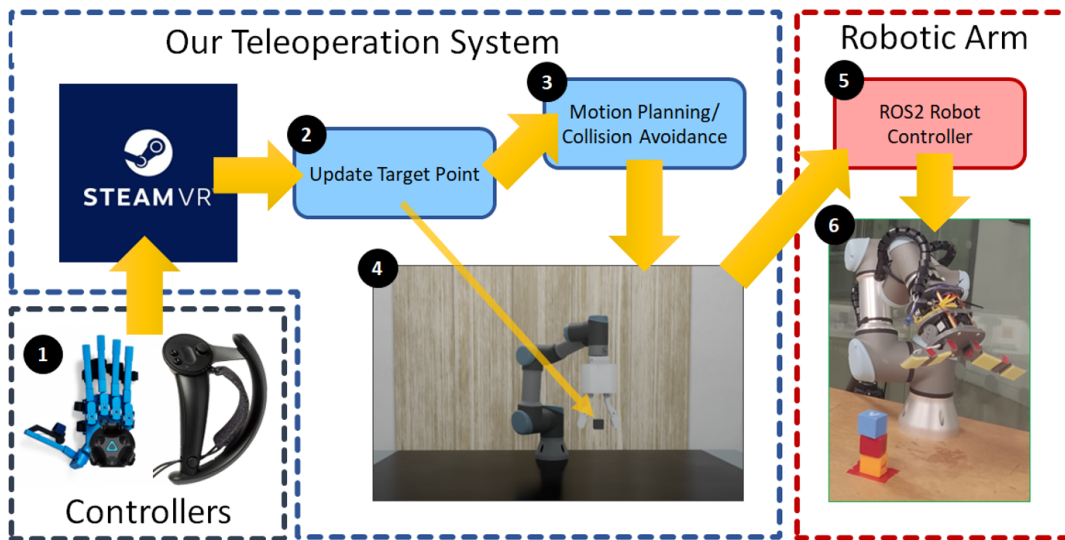


Fig. 3: Overview of TELESIM, our direct teleoperation framework. The controllers (in the black dotted line) (1) can be any system that outputs a 3D pose, while our framework is depicted in the blue dotted line. It accepts the pose given by (1) to update the 3D pose of a cube in the digital twin. The robot then calculates a path to this cube in real time while avoiding collision with the world (4). Finally, TELESIM can be plugged into any robotic system (6) via a ROS2 robot controller (5), as shown in the red dotted line.

previous work: (1) researchers have only used one robot per study, (2) with either MoveIt! or their own implementation of path planning, (3) using either one or more tracking methods and (4) testing their system with user studies with a small user base, which does not represent a statistically significant sample. It also addresses a gap discussed by Rea & Seo [1], which states that there needs to be more non-expert evaluation of robotic teleoperation for general tasks such as picking and placing common objects. Therefore, to advance the state-of-the-art in robotic teleoperation, we investigate the performance of different control modalities for direct teleoperation using a VR controller and finger mapping as different ways of interacting with the gripper while relying on VR tracking. Similarly, we evaluate our framework on 2 different robots, a Rethink Robotics Baxter and a Universal Robotic 3, which demonstrates the modularity of our system, while MoveIt! and other systems needed to be fine-tuned to work [27]. We ask 37 non-expert participants to carry out a simple pick-and-place task. Additionally, by using two different control modalities, our goal is to bridge the gap between VR control methods [21], [5] and complete body mapping [10], [11] by investigating the performance of hand and finger tracking through a SenseGlove.

III. DIRECT TELEOPERATION FRAMEWORK

TELESIM (Fig. 3) consists of 3 components. The controller (1) can be any control system that outputs a 3D pose. In this paper, the controller is either a VR controller or a SenseGlove with an HTC Vive Tracker. Our framework (2) calculates a motion plan to the pose given by (1) and sends it through ROS2 (Robot Operating System) [28] to any robotic arm that supports ROS control (3). In this paper, we use either a Baxter robot or a UR3 robot. The following

subsections detail the design and implementation of each component.

A. Hardware Setup

The hardware used in this paper consists of a SteamVR [29] tracking a Steam Index controller and HTC Vive Tracker, as shown in Figs. 2 points 5 and 4, respectively. The Baxter robot (point 2 in Fig. 2) is controlled using the Steam Index controller, and the gripper can be closed by pressing the main trigger on the Steam Index controller. The UR robot (point 3 in Fig. 2) is controlled using a Senseglove development kit, which allows for mapping of all individual finger movements, with an HTC Vive Tracker mounted on top of the middle finger of a T42 gripper from the Yale OpenHand project [14]. These control methods are represented as point 1 in Fig. 3

The T42 gripper is an underactuated 2-finger gripper that we modified to have a 1 degree of freedom per finger by fixing the 2nd joint of each finger, counting from the base. The interface with the Senseglove is thus restricted to movements of the user's thumb and index finger, and these are mapped to a single degree of freedom to result in the opening and closing of the gripper. We adopted this approach to compare the Steam Index controller and the Senseglove fairly. However, it still allowed us to test the increase in complexity of adding an extra degree of freedom for the user to control. We also developed a custom controller for the gripper using two separate Arduino UNOs, as they only have one serial port per board. Joint positions are converted from degrees to motor commands and then sent to the first Arduino through a serial interface. Then, commands are relayed to the second Arduino through I2C and transmitted to the two MX-28 Dynamixel [30]. The second Arduino is also responsible for reading the motors' current position and

load and transmitting it back to the computer. Additionally, we restricted the amount of force the user can transmit to the gripper, as without it, it leads to breakage of the controlling string or exceeding the amount of resistance allowed by the motor. We decided not to implement haptic feedback as it would give the users an advantage of sensing whether an object is grasped. Thus, this will result in an unfair comparison between the VR controller and the SenseGlove. Therefore, we leave haptic feedback for future work.

The VR headset (1 in Fig. 2) acts as the origin of both robots, giving the user an easy reference point for teleoperation. The SteamVR outputs the controller position in a 3D space with respect to the headset. The origin is thus transformed into the user's resting hand position when initiated. This method of tracking the position is preferred by multiple researchers [4], [31], as well as many of the participants in the ANA Avatar XPRIZE Challenge [5], [32]. The Senseglove is also used by the winning team [32], but to control a Schunk robotic hand that replicates a human hand.

B. Digital Twin

The position in the 3D space from the SteamVR is transmitted through ROS2 to a full digital twin created in NVIDIA Isaac Sim [33]. This flow of information can be seen in Fig. 3 point 2. Isaac Sim, a recent ray-tracing simulation software, is used to calculate the robot motion plan using RMPFlow [34], which is a motion generation based on Riemannian Motion Policies [35] (Fig. 3 point 3). Isaac Sim was chosen as it is the most realistic simulation software compatible with ROS2 from our simulation benchmark [36].

Isaac Sim takes in a URDF (Unified Robotics Description Format) of a robot for visualisation along with a robot description file, describing the joints that can be actuated by the motion planner and the robot's collision as spheres, as Isaac Sim uses them for collision checking. This file can be created with an included extension (Lula Robot Description Editor [37]). The ability to add new robots is a significant part of what makes TELESIM modular, along with ROS2.

We decided to use ROS2 as it is the most used framework for controlling various robots, making our framework plug-and-play. Views of the UR3 and the Baxter robot from Isaac Sim are shown in 1. The grey square in between the gripper (Fig. 3 point 4) is the point that is controlled by the teleoperation system and indicates where Isaac Sim must find a path. The fact that Isaac Sim acts as a complete digital twin allows the robot to avoid collision with the world around it and damage itself. Additionally, both robots have systems that allow them to work alongside humans. Our system is capable, with minimal configuration, of handling the restrictions placed by their need to be safe around humans. Finally, Isaac Sim transmits the position of each joint to the real robot through ROS2 and passes this information to the robotic system, as shown in Fig. 3 points 5 and 6.

C. Robot Control

ROS control [38] provides a wrapper that facilitates the interface of different robots. However, each robot needs to

be adapted to work with specific hardware. For this paper, we implemented the Universal Robot ROS2 Control package to transfer the joint states from Isaac Sim to the robots (Fig. 3 point 5). Specifically, for the UR3, the size of the gripper and the safety regulations of the laboratory reduced the available workspace for the robot. We handled these limitations by adding safety planes and limitations on the range of motion of the real robot. For Baxter, we used a ROS1 to ROS2 bidirectional bridge, as Baxter only works with ROS1 internally. The pipeline is thus converting Isaac Sim's joint states into Baxter messages in ROS2 and then they are sent through the bridge as ROS1 messages to the robot. The robot outputs its current state, which is converted to ROS2 through the same bridge.

Our framework introduces a slight amount of lag (100-150 ms) between the user movement and the robot movement, partly due to the path planning step, the time taken by the actuator to move the robot arm to the desired position and mostly due to safety restrictions that caps the maximum speed of our robots. The faster the user moves from one position to another, the higher the delay as the arm tries to catch up. The user easily accounts for this by making small and slow movements, as confirmed by Schwarz and Behnke [39]. Although we acknowledge lag is present in our system, Schwarz and Behnke [39] found that minor delays do not impact performance. None of our users mentioned lag as an issue in our experiments (see Section V). We recorded the amount of lag by following the same principle used by Rakita *et al.*[6], which recorded the scene using a 100Hz camera starting with both the robot and human at a complete stop. We recorded the difference between the starting frame of the movement by the operator and the starting frame of movement for the robot. This approach has some subjectivity due to identifying which frame the movement started.

IV. EXPERIMENTS

A. Methodology

As described in Section III-A, the user controls a robot by teleoperating it using either a VR controller or the SenseGlove. For this experiment, we consider a simple task where the user has to pick up 3 cubes on a table. The user must then bring them individually to the centre of the table to complete a tower. Once the tower is completed, the cubes are returned to their original position, and the user is asked to repeat the task as many times as possible within 10 minutes.

The task definition consists of 3 cubes positioned at each vertex of an isosceles triangle and at similar distances from the robots' base on each robot's table (Fig. 2 point 7). We have placed markers of the vertices of this isosceles triangle on each robot's operating table for repeatability of the task between attempts and among users. The front cube at the top of the triangle is positioned such that it is at the maximum reach limit of each robot's overhead rotation. This means that the end-effector's z-axis is perpendicular to the table, which makes it difficult for the user to pick up the cube from overhead. Thus, the user has to add some rotation in the x- or y-axis of the gripper to pick the cube successfully.

The right cube is the furthest away from the user in both robots and adds a degree of difficulty due to the user's viewpoint, but still within reach. Finally, the left cube is placed such that the user has to move their body. That is, for Baxter, the location where the user needs to pick up the left cube is approximately at the waist of the user, while for the UR3, the location is on the other side of (farthest from) the headset. The cube positions were chosen to let users be spatially aware of their position and its relationship to the robot. Finally, the location where the users need to stack the three cubes is easily accessible by the robot, and this position is marked by a red square in red tape, 2 cm larger than the cube. This position can be seen in Fig. 2 point 6, with the cubes stacked as required. For a better view of the setup, we refer the reader to the video attached¹. To teleoperate using the Baxter robot, the users need to stand with their back to the VR headset, while for teleoperating using the UR3, users need to stand with the headset on their right and the Baxter robot behind them. This difference in position is due to the space constraint of the room in which we ran our experiments. The operating room can be seen in Fig. 2, with the headset on the left of the picture shown as point 1.

B. User Survey

In our experiments, we asked 37 participants (29 male and 8 female) from various backgrounds aged 19 to 51 (mean: 25.32, 1 standard deviation: 6.26) to teleoperate both robots and stack 3 cubes without a monetary reward. We did not intend for our experiment to be influenced by the participants' gender, but we have included this information for completeness. Furthermore, our study had more participants than previous research, e.g. Javdani *et al.* [26], which only had 23 participants. Lastly, we observed that we had reached the saturation point, where the inclusion of each new participant did not significantly alter the mean and standard deviation. Participants reported having, on a 5-point Likert scale (going from "No Experience" with a score of 1 to "Experienced" with a score of 5), a 2.97 mean experience with Virtual Reality with a standard deviation of 1.2. They also reported having a mean experience of 2.76 with a robot with a standard deviation of 1.24. These questions are all relative to the user's understanding.

Each participant completed the short questionnaire described above at the beginning of the experiment. After being asked to position their back to the VR headset, an explanation was given on how to control the robot, emphasising that all of their hand movements and rotation will be mapped one-to-one to the robot. They were instructed to try to grasp a cube from both sides. They had 5 minutes to get used to the control without a specific task objective. Most of the participants picked up and placed a cube during this time.

After 5 minutes, the participants performed the task of stacking the 3 cubes in the given location without any restriction on the cubes' pose and order. Users were asked to stack cubes as many times as possible in 10 minutes. Once a

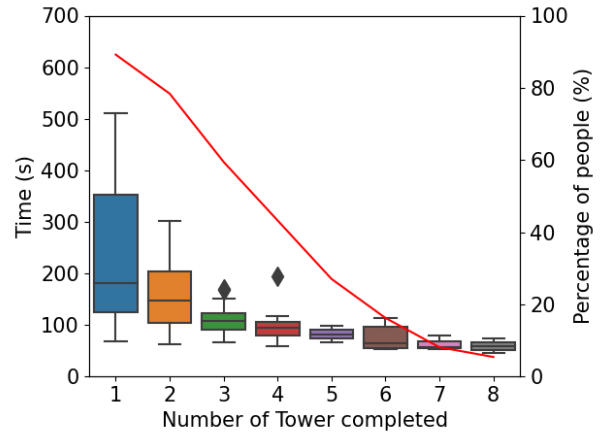


Fig. 4: Average Time Taken and Population percentage for each tower completed for the Baxter robot. The right axis corresponds to the red line, while the left corresponds to the box plot.

tower has been completed, we pause time and reset the cubes to their initial configuration. Users' actions were recorded, such as the time taken for each tower and for individual actions for each pick, place, and drop (i.e. failures).

After 10 minutes, users were allowed to take a break while answering the Single-Ease Question (SEQ) [40]. SEQ is a 7-point Likert-scale question asking, "How difficult was the task?". It was chosen instead of other metrics such as the System Usability Scale (SUS) [41] as Hodrien and Fernando [40] have argued that it is a good end-of-task metric. Then, they were asked to repeat the same experiment but with the UR3 robot. While this could lead to some learning bias, the fact that every participant performed the experiment this way, we expect the bias to be inconsequential. Additionally, the results for the UR3 described in Section V do not show any improvement compared to Baxter. This leads to the experiments taking a total time of $2 \times (5 + 10)$ min, so 30 minutes plus approximately 5 minutes of questionnaires.

V. EVALUATION

Fig. 4 shows that 85% of the participants can build at least one tower in 10 minutes using Baxter and the VR controller. However, there is a steady decline for each of the following towers, with only 5% of the users able to build 8 towers. This is in direct comparison to the UR3 as shown in Fig. 5, with slightly less than 50% of the population failing to build one tower and 5% managing to build 4, half as many towers for Baxter.

The box plots in Fig. 4 and 5 show the average and variance of the time taken by users to complete the towers. In particular, the first tower for both robots took most of the task duration because some participants could not build one tower. This time completion trend shows TELESIM's user-friendliness as 60% of users for Baxter managed with minimal training to complete a full tower, which means 3 different pick-and-place operations in around 2 minutes.

¹<https://cvas-ug.github.io/telesim>

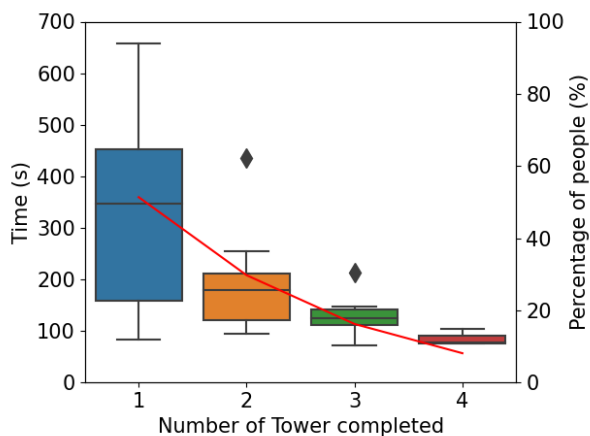


Fig. 5: Average Time Taken and Population percentage for each tower completed for the UR robot. The right axis corresponds to the red line, while the left corresponds to the box plot.

TABLE I: Additional Statistics Collected

(a) Baxter

	Min	Mean \pm Std	Max
Placing Rate	25.00%	77.42% \pm 15.54%	100.00
Dropping Rate	3.70%	23.83% \pm 14.08%	66.67
Collapse Rate	5.56%	18.44% \pm 11.66%	57.14
Still in Place Rate	24.31%	75.21% \pm 15.20%	95.92

(b) UR

	Min	Mean \pm Std	Max
Placing Rate	12.50%	46.29% \pm 17.97%	86.67
Dropping Rate	13.33%	53.37% \pm 18.63%	87.50
Collapse Rate	4.76%	22.25% \pm 12.46%	50.00
Still in Place Rate	14.88%	46.93% \pm 16.75%	84.44

Similarly, this can also be observed for the UR3 as 25% of the users managed to build a tower in 4 minutes.

Table I shows the additional statistics collected during the experiment, such as the percentage of times the user dropped a cube that caused the tower to collapse. Table Ia indicates that for the Baxter robot, 75% of the picking actions resulted in a correctly placed cube that did not collapse due to incorrect placement or the user inadvertently moved the robot in the tower’s path. Similarly, in Table Ib, 46% of all the picking actions resulted in a correct place. The difference in the number of towers built, shown in Fig. 5, can be explained by a greater difficulty in placing the cube. Specifically, our results indicate that there is a significant difference in the difficulty of placing a cube ($P < 0.01$), as well as the amount of time a cube is dropped ($P < 0.01$). These similarities are logical as the placing and dropping rates are complementary since these are the only two outcomes after picking up a cube. The differences can be explained by the different control methods, especially the dropping rate, which is most likely linked to the weakness of the gripper, such as the

limitation of the grip strength of the closed finger, to prevent the cable from breaking, and the limited range of motions since we observed that some users let the cube fall while the gripper was closed. However, successful users moved slowly to prevent unnecessary movement, thus reducing the risk of dropping. It is difficult to relate these results with the literature as previous research has not performed similar experiments. The collapse rate in both Table Ia and Table Ib is similar, and the statistical analysis shows that the difference is not significant ($P > 0.05$) and indicates that the type of robot does not influence the difficulty in operating the robot around obstacles.

Results of the Single Ease Question (SEQ) asked at the end of each task, in which a higher score means that TELESIM is easy to use, indicate that participants who performed better gave a higher SEQ score. Specifically, Baxter obtained a mean of 3.32 with a standard deviation of 1.27, while UR3 obtained a mean of 2.19 with a standard deviation of 1.14. Additionally, the mean difference between Baxter and UR corresponds with fewer towers built for the UR robot.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have investigated the performance of TELESIM by conducting a medium-scale user survey with 37 participants who were asked to build towers of 3 cubes by teleoperating robots. We tested TELESIM’s modularity on two different robots with two different control modalities. Our experimental results show that TELESIM is modular, plug-and-play and user-friendly, as not only were we able to deploy it on 2 robots with different modalities, but most users were able to succeed by building at least once a tower of 3 cubes, with only 5 minutes of training, regardless of the control modality, robot used and hardware limitations. We thus bridged the gap pointed out by Rea & Seo [1], where they state that there is a lack of non-expert evaluation of robotic teleoperation for general tasks such as picking and placing common objects. TELESIM is available on GitHub², allowing developers to perform teleoperation on their robots with minimal setup time.

Our underlying motivation for choosing direct teleoperation in this paper is to establish a baseline for further research on shared autonomy, which could combine human intuition and a high-level overview of a task while giving freedom to the robot to perform, for example, accurate picking and placing objects. Additionally, we plan to remove the constraint of having the VR headset behind the user and allow them to wear the headset to operate either in VR in the digital twin view of Isaac Sim or Augmented Reality by allowing the user to move around the robot and have different viewpoints while manipulating, thus enhancing the precision of the teleoperation. Finally, we want to explore the relationship between previous experience with VR and robotic technology and success at completing the tasks to identify the level of competence required to teleoperate robots successfully.

²<https://cvas-ug.github.io/telesim>

REFERENCES

- [1] D. J. Rea and S. H. Seo, "Still Not Solved: A Call for Renewed Focus on User-Centered Teleoperation Interfaces," *Frontiers in Robotics and AI*, vol. 9, 2022. [Online]. Available: Link
- [2] T. Muto and J.-H. Lee, "A touch based Multi-Modal interface for teleoperation to improve usability for the non-expert," in *2012 IEEE/SICE International Symposium on System Integration (SII)*, Dec. 2012, pp. 589–594.
- [3] "Ana avatar xpriza challenge," 2023. [Online]. Available: Link
- [4] A. Pettinger, C. Elliott, Z. Fan, and M. Pryor, "Reducing the Teleoperator's Cognitive Burden for Complex Contact Tasks Using Affordance Primitives," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 6, 2020.
- [5] S. Dafarra, K. Darvish, R. Grieco, G. Milani, U. Pattacini, L. Rapetti, G. Romualdi, M. Salvi, A. Scalzo, I. Sorrentino, D. Tomè, S. Traversaro, E. Valli, P. M. Viceconte, G. Metta, M. Maggiali, and D. Pucci, "iCub3 Avatar System," arXiv, Tech. Rep. arXiv:2203.06972, Mar. 2022, arXiv:2203.06972 [cs] type: article. [Online]. Available: Link
- [6] D. Rakita, B. Mutlu, and M. Gleicher, "A Motion Retargeting Method for Effective Mimicry-based Teleoperation of Robot Arms," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. Vienna Austria: ACM, Mar. 2017, pp. 361–370. [Online]. Available: Link
- [7] S. Scherzinger, A. Roennau, and R. Dillmann, "Learning Human-Inspired Force Strategies for Robotic Assembly," arXiv, Tech. Rep. arXiv:2303.12440, Mar. 2023, arXiv:2303.12440 [cs] type: article. [Online]. Available: Link
- [8] S. Javdani, S. Srinivasa, and A. Bagnell, "Shared Autonomy via Hindsight Optimization," in *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, Jul. 2015. [Online]. Available: Link
- [9] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese, "Human-in-the-Loop Imitation Learning using Remote Teleoperation," arXiv:2012.06733 [cs], Dec. 2020, arXiv:2012.06733. [Online]. Available: Link
- [10] N. Chen, K. P. Tee, C.-M. Chew, and R. Yan, "Intuitive interaction for robotic grasping," in *Proceedings of the Workshop at SIGGRAPH Asia*, ser. WASA '12. New York, NY, USA: Association for Computing Machinery, Nov. 2012, pp. 105–111. [Online]. Available: Link
- [11] T.-C. Lin, A. Unni Krishnan, and Z. Li, "Shared Autonomous Interface for Reducing Physical Effort in Robot Teleoperation via Human Motion Mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 9157–9163, iSSN: 2577-087X.
- [12] H. Admon and S. Srinivasa, "Predicting User Intent Through Eye Gaze for Shared Autonom," *2016 AAI Fall Symposium Series: Shared Autonomy in Research and Practice*, 2016.
- [13] "Valve Index." [Online]. Available: Link
- [14] "Yale OpenHand Project - Model T42," 2023. [Online]. Available: Link
- [15] D. Zhang, R. Tron, and R. P. Khurshid, "Haptic Feedback Improves Human-Robot Agreement and User Satisfaction in Shared-Autonomy Teleoperation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 3306–3312, iSSN: 2577-087X.
- [16] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers, "Robotic Surgery: A Current Perspective," *Annals of Surgery*, vol. 239, no. 1, pp. 14–21, Jan. 2004. [Online]. Available: Link
- [17] K. D. Katal, C. Y. Brown, S. A. Hechtman, M. P. Para, T. G. McGee, K. C. Wolfe, R. J. Murphy, M. D. Kutzer, E. W. Tunstel, M. P. McLoughlin, and M. S. Johannes, "Approaches to robotic teleoperation in a disaster scenario: From supervised autonomy to direct control," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Chicago, IL, USA: IEEE, Sep. 2014, pp. 1874–1881. [Online]. Available: Link
- [18] R. M. Aronson, T. Santini, T. C. Kübler, E. Kasneci, S. Srinivasa, and H. Admoni, "Eye-Hand Behavior in Human-Robot Shared Manipulation," in *2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar. 2018, pp. 4–13, iSSN: 2167-2148.
- [19] Y. P. Toh, S. Huang, J. Lin, M. Bajzek, G. Zeglin, and N. S. Pollard, "Dexterous telemanipulation with a multi-touch interface," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. Osaka, Japan: IEEE, Nov. 2012, pp. 270–277. [Online]. Available: Link
- [20] M. Micire, M. Desai, J. L. Drury, E. McCann, A. Norton, K. M. Tsui, and H. A. Yanco, "Design and validation of two-handed multi-touch tabletop controllers for robot teleoperation," in *Proceedings of the 16th international conference on Intelligent user interfaces*. Palo Alto CA USA: ACM, Feb. 2011, pp. 145–154. [Online]. Available: Link
- [21] L. Wang, Q. Li, J. Lam, Z. Wang, and Z. Zhang, "Intent inference in shared-control teleoperation system in consideration of user behavior," *Complex & Intelligent Systems*, Oct. 2021. [Online]. Available: Link
- [22] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei, "ROBOTURK: A Crowdsourcing Platform for Robotic Skill Learning through Imitation," in *Proceedings of The 2nd Conference on Robot Learning*. PMLR, Oct. 2018, pp. 879–893, iSSN: 2640-3498. [Online]. Available: Link
- [23] E. Rosen, D. Whitney, E. Phillips, G. Chien, J. Tompkin, G. Konidaris, and S. Tellex, "Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays," *The International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1513–1526, Oct. 2019, publisher: SAGE Publications Ltd STM. [Online]. Available: Link
- [24] F. Kennel-Maushart, R. Poranne, and S. Coros, "Manipulability optimization for multi-arm teleoperation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 3956–3962, iSSN: 2577-087X.
- [25] A. Gottardi, S. Tortora, E. Tosello, and E. Menegatti, "Shared Control in Robot Teleoperation With Improved Potential Fields," *IEEE Transactions on Human-Machine Systems*, pp. 1–13, 2022, conference Name: IEEE Transactions on Human-Machine Systems.
- [26] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 717–742, Jun. 2018, publisher: SAGE Publications Ltd STM. [Online]. Available: Link
- [27] T. Schüppstuhl, K. Tracht, and A. Raatz, Eds., *Annals of Scientific Society for Assembly, Handling and Industrial Robotics 2021*. Cham: Springer International Publishing, 2022. [Online]. Available: Link
- [28] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, May 2022, publisher: American Association for the Advancement of Science. [Online]. Available: Link
- [29] "SteamVR on Steam." [Online]. Available: Link
- [30] "ROBOTIS e-Manual." [Online]. Available: Link
- [31] D. Whitney, E. Rosen, D. Ullman, E. Phillips, and S. Tellex, "ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 1–9, iSSN: 2153-0866.
- [32] M. Schwarz, C. Lenz, R. Memmesheimer, B. Pätzold, A. Rochow, M. Schreiber, and S. Behnke, "Robust Immersive Telepresence and Mobile Telemanipulation: NimbRo wins ANA Avatar XPRIZE Finals," arXiv, Tech. Rep. arXiv:2303.03297, Mar. 2023, arXiv:2303.03297 [cs] type: article. [Online]. Available: Link
- [33] "Isaac Sim," Dec. 2019. [Online]. Available: Link
- [34] C.-A. Cheng, M. Mukadam, J. Issac, S. Birchfield, D. Fox, B. Boots, and N. Ratliff, "RMPflow: A Computational Graph for Automatic Motion Policy Generation," in *Algorithmic Foundations of Robotics XIII*, ser. Springer Proceedings in Advanced Robotics, M. Morales, L. Tapia, G. Sánchez-Ante, and S. Hutchinson, Eds. Cham: Springer International Publishing, 2020, pp. 441–457.
- [35] N. D. Ratliff, J. Issac, S. Birchfield, and D. Fox, "Riemannian Motion Policies," arXiv, Tech. Rep. arXiv:1801.02854, Jul. 2018, arXiv:1801.02854 [cs] type: article. [Online]. Available: Link
- [36] F. P. Audonnet, A. Hamilton, and G. Aragon-Camarasa, "A Systematic Comparison of Simulation Software for Robotic Arm Manipulation using ROS2," in *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, Nov. 2022, pp. 755–762, iSSN: 2642-3901.
- [37] "Lula Robot Description Editor — Omniverse IsaacSim latest documentation." [Online]. Available: Link
- [38] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. R. Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, and E. F. Perdomo, "ros.control: A generic and simple

- control framework for ROS,” *Journal of Open Source Software*, vol. 2, no. 20, p. 456, Dec. 2017. [Online]. Available: Link
- [39] M. Schwarz and S. Behnke, “Low-Latency Immersive 6D Televisualization with Spherical Rendering,” in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, Jul. 2021, pp. 320–325, iSSN: 2164-0580.
- [40] A. Hodrien and T. Fernando, “A review of post-study and post-task subjective questionnaires to guide assessment of system usability,” *Journal of Usability Studies*, vol. 16, no. 3, pp. 203–232, May 2021.
- [41] J. Brooke, “SUS – a quick and dirty usability scale,” in *Usability Evaluation in Industry*, Jan. 1996, pp. 189–194.