2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Design of a High-level Teleoperation Interface Resilient to the Effects of Unreliable Robot Autonomy

Samuel S. White[1*], Keion W. Bisland[1], Michael C. Collins[2], and Zhi Li[1]

***Abstract*— High-level control is generally preferred for the control of complex robot platforms and by users inexperienced with robot teleoperation. However, high-level teleoperation interfaces can be less effective if the robot autonomy is not reliable. To address this problem, it is important to understand how the users' preference of teleoperation interface may vary with the reliability of the robot autonomy, and understand what design features ameliorate the frustration and effort caused by unreliable autonomy.**

This paper proposes a graphical user interface for high-level robot control. The framework of the interface enables teleoperators to control a robot at the action level, and incorporates a simple but effective design that enables teleoperators to recover from task failure in a number of ways. We conducted a user study ($N = 25$) to compare the performance and user experience when using the proposed high-level interface to a low-level interface (i.e., gamepad) for robot low-level control, on a representative manipulation task. We also investigated if the high-level teleoperation interface remains effective if the reliability of robot autonomy decreases. Our results show that a high-level interface able to handle the most frequent errors is resilient to the effects of unreliable robot autonomy. Although the total task completion time increased as the robot autonomy becomes unreliable, the users' perception of workload and task performance are not affected. Through the user study, we also reveal the desirable interface features.

I. Introduction

The design of a teleoperation interface defines the roles that both the human and the robot play in completing the tasks. Even for the same task, the optimal choice of teleoperation interface depends on the capabilities of robot autonomy and human teleoperators. Overall, teleoperation interfaces for high-level robot control are preferred if the robot has sufficient autonomy for the task, whereas direct control interfaces are preferred if the teleoperators need more freeform control for unstructured tasks. However, it is unclear if preference for user interfaces can be affected by the reliability of the robot autonomy. In a recent DARPA challenge [1], the effectiveness of a system where a high-level interface was used except in the event of a failure (at which point an operator took over with low-level control) was demonstrated. However, it is also possible that users may still prefer high-level control interfaces if the interfaces provide autonomous failure correction. The robot autonomy for failure recovery actions does not have to be perfect, as long as it can fix the most frequent errors in the robot autonomy.

In this paper, we present a design for a high-level teleoperation interface that includes error handling and supports control over the failure recovery action, in addition to control over manipulation actions. We consider the teleoperation of a cup-stacking task, which requires the control of gross manipulation actions (e.g. picking and moving), and precise manipulation action (e.g. placing and stacking). We conducted a user study ($N = 25$) to compare the proposed interface (which was preferred for the teleoperation of similar tasks in a pilot study) with direct teleoperation using a gamepad interface. We also compared the performance and user experience with the proposed interface across several degrees of reliability of the robot autonomy. We find that:

1) With reliable robot autonomy, users prefer using high-level control through a GUI interface over direct teleoperation using gamepad, because of the better task performance and lower workload;
2) With unreliable robot autonomy, users still prefer the high-level control over direct teleoperation if the interface provides high-level control with interactive failure correction;
3) The decreasing reliability of robot autonomy will affect the task performance in terms of total task completion time, yet the user-perceived workload is not affected due to the error correction function of the interface.

II. Related Work

A. Teleoperation Interfaces for Low-Level and High-Level Control

Teleoperation interfaces generally fall into two categories: The low-level interfaces allow the user to directly dictate the robot's motions and states; The high-level interfaces allow the users to control the robot actions, given the robot autonomy for motion planning and execution. The representative devices for low-level teleoperation include desktop or hand-held interfaces such as joysticks and gamepad [2], motion mapping interfaces of varying accuracy [3]–[6], and virtual reality (VR) interfaces with a headset and hand-held controllers [7]. These interfaces are suitable for the teleoperation of freeform tasks because they intuitively map between human action and robot motion. It is desirable for interfaces like these to have teleoperation assistance for improving the task accuracy and reducing user effort. For instance, Dragan and Srinivasa [8] arbitrated between user

The first three authors contributed equally to this work. Asterisk indicates corresponding author.

[1] Samuel S. White, Keion W. Bisland and Zhi Li are with the Robotics Engineering Program, Worcester Polytechnic Institute, Worcester, MA 01609, USA {sswhite,kbisland,zli11}@wpi.edu

[2] Michael C. Collins is with the Computer Science Department, Worcester Polytechnic Institute, Worcester, MA 01609, USA mcollins@wpi.edu

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

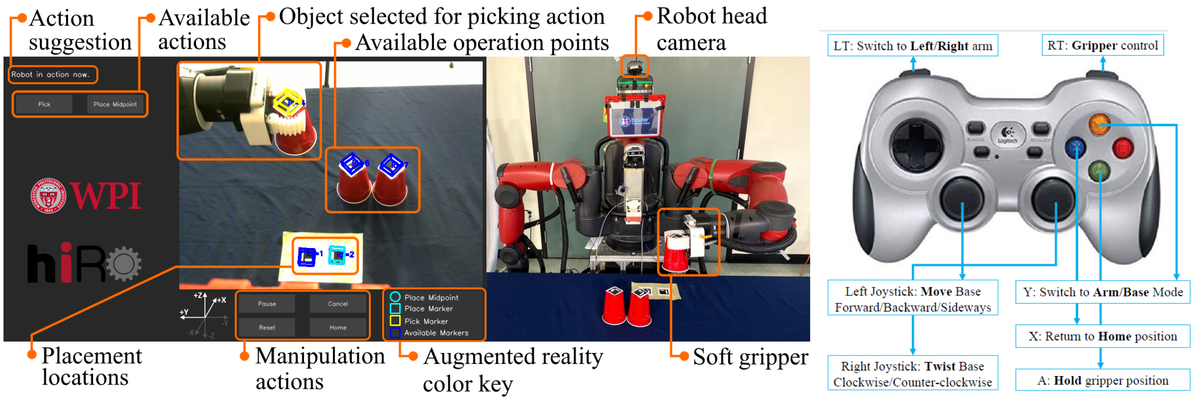

Fig. 1: (Left) The proposed GUI interface for high-level teleoperation; (middle) robot platform; (right) gamepad interface for low-level teleoperation.

input and predicted intent to create seamless teleoperation through shared control based on policy-blending. In addition, virtual or augmented reality and haptic feedback are also integrated into interfaces to improve the perception of remote tasks. For instance, a haptic interface with force feedback was used to merge user input with autonomous objectives, including collision avoidance and smoothness for control of mobile robot trajectories [9]. Mixed reality head-mounted display (HMD) visualizations were used to show intended trajectories of a humanoid robot for manipulation tasks [10]. On the other hand, interfaces including Graphical User Interfaces (GUI), speech [11], gestures [12], and brain-computer interfaces (BCI) [13] are usually preferred for high-level teleoperation. Note that GUI can be used for both low-level control and high-level control. For example, the web-based graphical user interface (GUI) developed by Kent, Saldanha, and Chernova [14] combines free-form control, constrained positioning, and point-and-click control of robot manipulation.

Overall, high-level interfaces are preferred by novice users and are advantageous for more structured tasks where the robot autonomy is sufficient to handle the task complexity. It has been shown that as long as the robot autonomy is reliable, robot teleoperation operators make less errors [15] and the effort of completing tasks is reduced as the robot autonomy increases [16].

However, such teleoperation interfaces may be unsuitable for robot control if the robot autonomy cannot perform the task reliably. Moreover, the users' preference of teleoperation interfaces may also be affected by the usability of the interface, given the reliability of the robot autonomy. Indeed, the development of user interfaces and robot autonomy are coupled, with research suggesting that GUI's are more usable with highly reliable autonomy [17]. An interface design that supports both high-level and low-level teleoperation allows the user to freely switch between them according to the capability and reliability of the robot autonomy with respect to the difficulty of the task [18]. Another strategy to combat this is variable autonomy, where robot autonomy and user control can be balanced as necessary to handle errors and variable task complexity [13], [19]. However, to novice users, it is always more preferred to reduce the amount of direct and tedious control of robot motions. For instance, error handling can be encoded into robot autonomy such that the robot detects errors and asks the operator to handle them [1]. In addition, prior research also shows that the errors in robot autonomy not only negatively affect performance, but also impact the operator's trust [20], perhaps to the point that the operator would prefer an interface that lends more control over the robot. Therefore, it is also important to investigate how the reliability of robot autonomy impacts user preference of teleoperation interfaces [21].

B. Failure handling with Human-Robot Interaction

Honig and Oron-Gilad [22] identified several consequences of how a robot handles a failure, including willingness to use the system again, task performance, trust in the system, and perception of the system. Failure handling for teleoperation systems breaks down into three components:

- Detecting failures
- Communicating the failure to the user
- Failure recovery

To detect failures in robot autonomy, the robot must have a way to identify inconsistencies between the robot's world model and reality [23]. Major catastrophe can be avoided by notifying the operator after a potential error is detected, thus enabling the operator to take control and remedy the situation [1]. Communicating the fault can be done in several ways, ranging from expressive motions [24] to directly asking for a form of assistance [25] to justifying the mistake as a means of repairing lost trust [26]. Analysis of the detected error state can be an important step before communicating the error or attempting to recover, especially with unstructured environments [27] or complex interactions [28].

If there is a time-critical situation with too much information for a human to process quickly, autonomously recovering from the failure rather than returning control to the operator is often the best decision [29]. This enables autonomy to safely operate under difficult conditions that are prone to failures [29]. Appropriate failure recovery is especially crucial to long-term robot autonomy systems that are deployed for months or years at a time [30], [31]. While recovering from failure is important, it is not the last step - the robot still needs to continue its mission. Marion et al.

[32] give an example of an interface that allows the robot to resume attempting a previously failed task after a human correction.

III. Interface Design and Implementation

Fig. 1 shows our proposed Graphical User Interface (GUI). We designed the GUI to be suitable for a wide range of object manipulation tasks performed in the workspace, which involve the actions of picking, moving, and placing objects. The set of manipulation actions include:

(a) **Pick** - Select which object should be picked up
(b) **Place** - Select where to place the object
(c) **Home** - Resets the robot to its neutral position

The camera view has augmented reality to display the default "operation points" (i.e. the available markers) in dark blue, including the locations of the objects and the goal positions for placing the object. The interface also allows users to select the midpoint between any two default operation points. In addition, the interface provides a set of functional actions including:

(a) **Act** - Asks the robot to perform a manipulation action
(b) **Cancel** - Stops the ongoing action
(c) **Reset** - Stops the ongoing action and resets the selection of the operation points
(d) **Try Again** - Reattempts the picking action in the case of a detected failure
(e) **Continue** - Resumes placing action in the case of a detected failure

Given the current task status, the interface will only show the applicable actions and hide the ones that are not relevant. For tasks with predefined steps, the interface will follow a state machine that defines the task logic and display the suggested actions given the current task status.

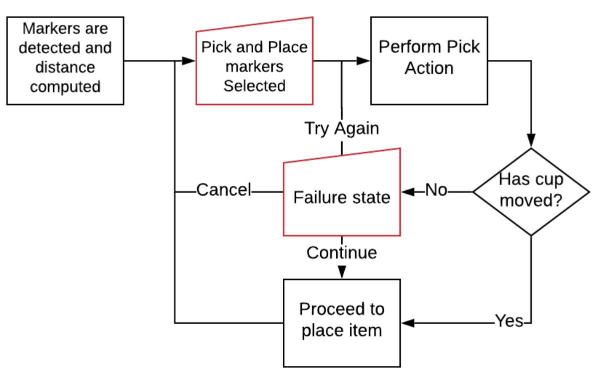

Fig. 2: Control flow for the interactive failure recovery.

Our proposed GUI features interactive high-level control with failure detection and recovery. The failure recovery addresses the most frequently occurring type of error in this task, i.e. error in grasping. Fig. 2 shows the control flow for the interactive failure process. The error detection relies on the computer vision that tracks the ArUco marker on the object being picked up. An error is detected if the object intended for pickup does not have any change in its height while the robot is attempting to manipulate it. According to our pilot study, this error detection approach can capture above 95% of grasping errors, though 30% of the detected errors could be false positives resulting from occlusion of the ArUco markers. If an error is detected, the interface will display a window which shows three action options: "try again", "cancel", and "continue". The users can review the camera view to decide the actions to take. The users can choose to:

1) "Continue" to take the action if they think the detected error is a false positive
2) "Try again" to have the robot attempt to grasp the object once more
3) "Cancel" the grasping action if necessary

Regarding the implementation of the interface, we use several methods to improve the accuracy of computer vision. We use a web camera mounted on the head of the robot to track the position and orientation of the ArUco markers attached to the objects [33]. Through calibration we are able to increase the marker tracking accuracy to be within 2 cm across the entire workspace. We also use a multi-stage filtering algorithm to remove markers which are detected by the computer vision but do not actually exist. This is a common problem for ArUco marker detection algorithms. Specifically, we examine the list of detected markers IDs and check if the marker ID has been recently detected around this location a set number of times. In the case of the sudden large displacement of a marker, this means it will be removed from the list for a short period of time before reappearing once its location is stable.

IV. Experiment

A. Robot platform

Our robot platform, Tele-Robotic Intelligent Nursing Assistance, is a mobile humanoid robot originally developed for nursing assistance tasks [34]. The robot system consists of a humanoid robot torso (Rethink Baxter), a pair of soft robot grippers (UBIROS GentleDuo), and an omni-directional mobile base (HStar). The robots are equipped with many telepresence cameras, yet for this study we only use a web camera (Logitech HD 720P) mounted on the robot head for computer vision and telepresence with augment reality.

The computer vision package takes in the video feed from the camera and outputs a list of existing markers and their locations. The GUI consumes this list and uses it to show the augmented reality display to the user. Between the GUI and the robot's motion planner, there exists a state machine that handles communication and maintains the robot's world model. Within this state machine, the current command from the high-level user interface as well as the detected and selected markers are all considered in determining the appropriate robot motions. Commands are given through a set of ROS [35] topics, each of which contains information about a different part of the command. The state machine runs on a continuous loop, checking the commands and the current positioning of the robot, and then sending an end-effector location and grasp amount to the motion planner. It also sends its state information back to the user interface, in order to communicate failure detection and task completion to the user. The motion planner determines the route and joint positions required to reach the commanded end-effector location, and then causes the robot to perform the required motions.

B. Procedure, Task, and Data Collection

We conducted a user study comprised of two experiments with $N = 25$ participants (16 male and 9 female, age: 22.16 ± 2.50 years old) in order to evaluate the effectiveness of the proposed GUI interface design. Before the experiments, the participants took a pre-study survey to report their education background and experience that may affect their robot teleoperation skills. Most of our participants (96%) stated that they possess engineering and technical background. On a 7-point Likert scale (1 for "No Experience" and 7 for "Very Experienced"), most users reported a moderate amount of experience in robot operation (5 ± 1.5), 3D software (5 ± 2), and video games(4 ± 1.5).

After the pre-study survey, participants were asked to conduct Experiment I, which compares robot teleoperation using a gamepad and the proposed GUI interface at a baseline 0% artificial autonomy error. The gamepad directly controls the robot's motions and states, while the proposed GUI controls the robot's actions for manipulation and error recovery. The task was to stack a set of three cups with markers on them into a pyramid formation at a designated location. For each interface, participants were given 5 minutes to learn and practice the interface. After practicing each interface, the participants were given 5 minutes to perform the cup-stacking task. The order in which the participants used the interfaces was randomized for each participant. The participants then conducted Experiment II, in which teleoperators used the proposed GUI to control the robot to perform the same task as the previous experiment. A key difference in the second experiment was the different percentage chances of positioning error in pickup introduced to the robot autonomy. In Experiment II, we randomized the order of the three levels of reliability of the robot autonomy, (15%, 30% and 45% chance of error in the autonomous picking-up action). For both of the experiments, we collected objective data including the total task completion time and time for each action, as well as a NASA-TLX survey after every teleoperation task. At the end of Experiment I and after every task in Experiment II we used a customized questionnaire to survey the participant's preference of interface. After both experiments were completed, participants were surveyed about the effectiveness of the interface design features including:

1) The error handling
2) The colored markers in the augmented reality
3) The dialog box prompting user actions
4) The minimal number of buttons
5) The ability to give action commands
6) The cancel button
7) The reset button

The participants used a 5-point Likert scale to rate each of the design features, with 1 for "not useful" and 5 for "highly useful". We also surveyed the users to determine to what extent the users agree that the proposed GUI (relative to the gamepad):

1) Is easier to learn
2) Allows them to complete the task faster
3) Is simpler to use
4) Is more intuitive
5) Is more engaging
6) Is more efficient
7) Is more reliable
8) Required less effort
9) Better enables multitasking

The participants used a 5-point Likert scale to rate each of these statements, with 1 for "strongly disagree" and 5 for"strongly agree".

V. Results

A. Task performance

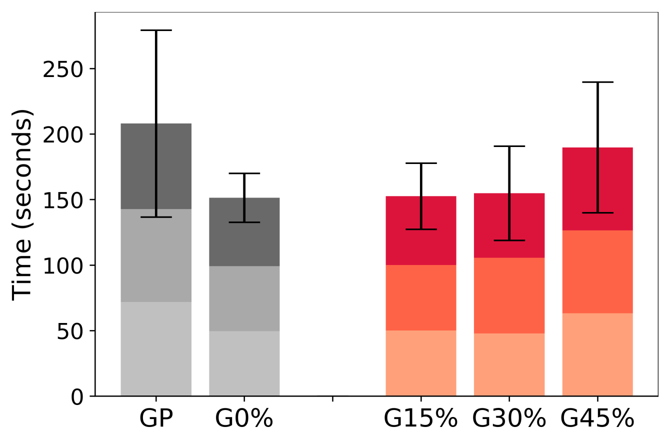

Fig. 3: Comparison of total task completion time: Grey bars for Experiment I and Red bars for Experiment II. The average time for handling the first, second and third cups were indicated using the light to dark colors. Results from NASA-TLX survey for Experiment I. GP: gamepad; GX%: GUI with X percent artificial error.

Fig. 3 shows the task performance comparison in Experiment I (grey bars) and Experiment II (red bars). Our ANOVA analysis results indicate: 1) with fully reliable robot autonomy, the proposed GUI for robot high-level control has significantly shorter total task completion time than the direct low-level control using the gamepad interface ($p < 0.001$); 2) there is a gradual increase in both average and variation of completion time as the injected error percent increases, with a significance of $p < 0.05$.

B. User Experience

We further investigate how the choice of interfaces and reliability of robot autonomy influence the users perception of the task. Shown in Fig. 4, the proposed GUI outperforms

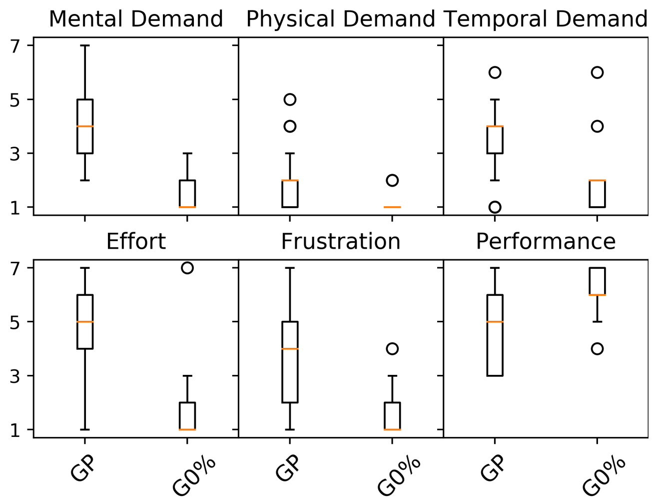

Fig. 4: Results from NASA-TLX survey for Experiment I. GP: gamepad; GX%: GUI with X percent artificial error.

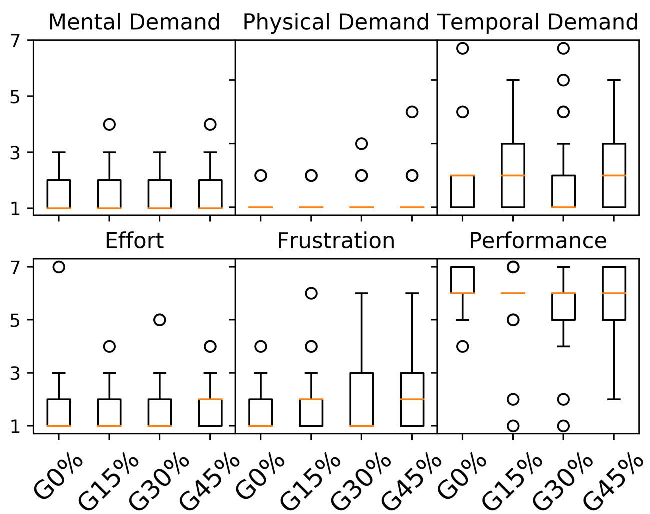

Fig. 5: Results from NASA-TLX survey for Experiment II. GX%: GUI with X percent artificial error.

the gamepad in all the aspects of the NASA-TLX survey ($p < 0.001$). Shown in Fig. 5, as greater percentages of errors were introduced to the robot autonomy, we only found significant difference in frustration between 15% and 45% of errors (Wilcoxon: $F = 9.0$, $p = 0.000197$).

Our customized questionnaire surveys the users' preference of interface design features. Overall, users like all the design features in the proposed interface design (see Section IV-B). The rating for these features is 4 ± 1.47, in the 5 Likert scale, with 1 for "not useful" and 5 for "very useful".

We also found the users dominantly prefer the proposed GUI to perform robot teleoperation on a daily basis even if the robot autonomy is unreliable, as long as the interface allows them to recover the errors. When the percentage of introduced errors is under 30%, 19 out of 25 users prefer to use the proposed GUI. Only when the error rate increases to 45% did one user change their preferred interface from the robot high-level control using the proposed GUI to

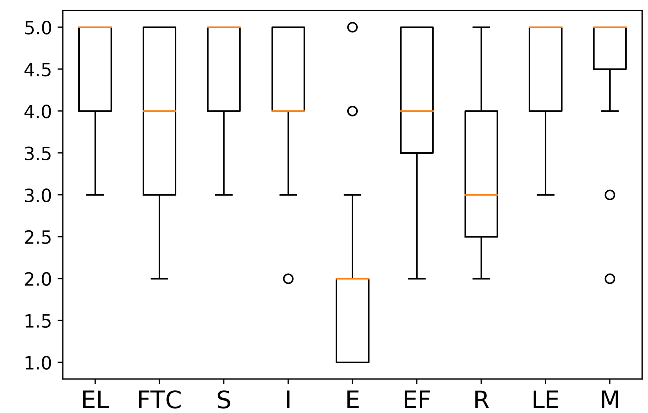

Fig. 6: Users' reasons to prefer GUI relative to the gamepad interface. Users provided their rating using a 5-point Likert scale by 'strongly disagree'(1), 'neutral'(3), and 'strongly agree'(5). The reasons for their preference include: ease of learning (EL), faster to complete task (FTC), simpler to use (S), more intuitive (I), more engaging (E), more efficient (EF), more reliable (R), less effort (LE), better enables multitasking (M).

robot low-level control using the gamepad. Fig. 6 shows the detailed reasons for which the users prefer the proposed GUI over robot low-level control using the gamepad. It is worth noticing that the only thing that may cause the users to not like the proposed GUI interface is the level of engagement in robot teleoperation.

VI. Conclusion

We proposed a graphical user interface design for robot teleoperation that can handle the robot error recovery using high-level control. Our user study shows that high-level control using the proposed GUI outperforms low-level control using a gamepad in both task performance and user experience. Our study also shows that to some extent the high-level error correction makes the proposed GUI robust against unreliable robot autonomy: although the task performance (in terms of total task completion time) was reduced as the robot autonomy becomes less reliable, the users' perception of mental demand, effort, and performance of the task were not significantly affected. One limitation of our user study is that we have evaluated the proposed GUI for one example of a manipulation task. The implementation of the GUI with high-level control for failure recovery is merely a proof-of-concept. Additionally, more research needs to be performed to generalize these findings for a larger range of user backgrounds. Nevertheless, as far as we are aware, our user study is the first to investigate how varying reliability of autonomy affects user preference of a teleoperation interface with error handling.

References

[1] C. G. Atkeson, B. P. W. Babu, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin *et al.*, "No falls, no resets: Reliable humanoid behavior in the DARPA robotics challenge," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 623–630.

[2] M. Stilman, K. Nishiwaki, and S. Kagami, "Humanoid teleoperation for whole body manipulation," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 3175–3180.

[3] Y. Ou, J. Hu, Z. Wang, Y. Fu, X. Wu, and X. Li, “A real-time human imitation system using kinect,” *International Journal of Social Robotics*, vol. 7, no. 5, pp. 587–600, 2015.

[4] C. Stanton, A. Bogdanovych, and E. Ratanasena, “Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning,” in *Proc. Australasian Conference on Robotics and Automation*, 2012.

[5] J. Oh, I. Lee, H. Jeong, and J.-H. Oh, “Real-time humanoid whole-body remote control framework for imitating human motion based on kinematic mapping and motion constraints,” *Advanced Robotics*, vol. 33, no. 6, pp. 293–305, 2019.

[6] Y.-H. Seo, I.-W. Jeong, and H. S. Yang, “Motion capture-based wearable interaction system and its application to a humanoid robot, amio,” *Advanced Robotics*, vol. 21, no. 15, pp. 1725–1741, 2007.

[7] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, “High speed whole body dynamic motion experiment with real time master-slave humanoid robot system,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.

[8] 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.

[9] C. Masone, M. Mohammadi, P. Robuffo Giordano, and A. Franchi, “Shared planning and control for mobile robots with integral haptic feedback,” *The International Journal of Robotics Research*, vol. 37, no. 11, pp. 1395–1420, 2018.

[10] 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, 2019.

[11] B. Wang, Z. Li, and N. Ding, “Speech control of a teleoperated mobile humanoid robot,” in *2011 IEEE International Conference on Automation and Logistics (ICAL)*. IEEE, 2011, pp. 339–344.

[12] I. Ajili, M. Mallem, and J.-Y. Didier, “Gesture recognition for humanoid robot teleoperation,” in *2017 26th IEEE International symposium on robot and human interactive communication (RO-MAN)*. IEEE, 2017, pp. 1115–1120.

[13] K. Muelling, A. Venkatraman, J.-S. Valois, J. E. Downey, J. Weiss, S. Javdani, M. Hebert, A. B. Schwartz, J. L. Collinger, and J. A. Bagnell, “Autonomy infused teleoperation with application to brain computer interface controlled manipulation,” *Autonomous Robots*, vol. 41, no. 6, pp. 1401–1422, 2017.

[14] D. Kent, C. Saldanha, and S. Chernova, “Leveraging depth data in remote robot teleoperation interfaces for general object manipulation,” *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 39–53, 2020.

[15] ——, “A comparison of remote robot teleoperation interfaces for general object manipulation,” in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 2017, pp. 371–379.

[16] D.-J. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang, and A. Behal, “How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 1, pp. 2–14, 2011.

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

[18] I. Havoutis and S. Calinon, “Supervisory teleoperation with online learning and optimal control,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1534–1540.

[19] P. Schillinger, S. Kohlbrecher, and O. von Stryk, “Human-robot collaborative high-level control with application to rescue robotics,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2796–2802.

[20] M. Desai, P. Kaniarasu, M. Medvedev, A. Steinfeld, and H. Yanco, “Impact of robot failures and feedback on real-time trust,” in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2013, pp. 251–258.

[21] M. Desai, M. Medvedev, M. Vázquez, S. McSheehy, S. Gadea-Omelchenko, C. Bruggeman, A. Steinfeld, and H. Yanco, “Effects of changing reliability on trust of robot systems,” in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2012, pp. 73–80.

[22] S. Honig and T. Oron-Gilad, “Understanding and resolving failures in human-robot interaction: Literature review and model development,” *Frontiers in psychology*, vol. 9, p. 861, 2018.

[23] S. Gspandl, S. Podesser, M. Reip, G. Steinbauer, and M. Wolfram, “A dependable perception-decision-execution cycle for autonomous robots,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2992–2998.

[24] M. Kwon, S. H. Huang, and A. D. Dragan, “Expressing robot incapability,” in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 87–95.

[25] R. A. Knepper, S. Tellex, A. Li, N. Roy, and D. Rus, “Recovering from failure by asking for help,” *Autonomous Robots*, vol. 39, no. 3, pp. 347–362, 2015.

[26] F. Correia, C. Guerra, S. Mascarenhas, F. S. Melo, and A. Paiva, “Exploring the impact of fault justification in human-robot trust,” in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 507–513.

[27] D. Crestani, K. Godary-Dejean, and L. Lapierre, “Enhancing fault tolerance of autonomous mobile robots,” *Robotics and Autonomous Systems*, vol. 68, pp. 140–155, 2015.

[28] P. Trung, M. Giuliani, M. Miksch, G. Stollnberger, S. Stadler, N. Mirnig, and M. Tscheligi, “Head and shoulders: automatic error detection in human-robot interaction,” in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, 2017, pp. 181–188.

[29] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, “Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1722–1729.

[30] N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrova, J. Young, J. Wyatt, D. Hebesberger, T. Kortner *et al.*, “The strands project: Long-term autonomy in everyday environments,” *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 146–156, 2017.

[31] L. Kunze, N. Hawes, T. Duckett, M. Hanheide, and T. Krajník, “Artificial intelligence for long-term robot autonomy: A survey,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4023–4030, 2018.

[32] P. Marion, M. Fallon, R. Deits, A. Valenzuela, C. Pérez D'Arpino, G. Izatt, L. Manuelli, M. Antone, H. Dai, T. Koolen *et al.*, “Director: A user interface designed for robot operation with shared autonomy,” *Journal of Field Robotics*, vol. 34, no. 2, pp. 262–280, 2017.

[33] C. Gamagedara, “aruco-markers,” https://github.com/fdcl-gwu/aruco-markers, 2019.

[34] Z. Li, P. Moran, Q. Dong, R. J. Shaw, and K. Hauser, “Development of a tele-nursing mobile manipulator for remote care-giving in quarantine areas,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3581–3586.

[35] Stanford Artificial Intelligence Laboratory et al., “Robotic operating system.” [Online]. Available: https://www.ros.org