

# Achieving Interaction Fluidity in a Wizard-of-Oz Robotic System: A Prototype for Fluid Error-Correction

Carlos Baptista De Lima  
c.v.baptistadelima@swansea.ac.uk  
Swansea University  
Swansea, United Kingdom

Julian Hough  
julian.hough@swansea.ac.uk  
Swansea University  
Swansea, United Kingdom

Frank Förster  
f.foerster@herts.ac.uk  
University of Hertfordshire  
Hatfield, United Kingdom

Patrick Holthaus  
p.holthaus@herts.ac.uk  
University of Hertfordshire  
Hatfield, United Kingdom

Yongjun Zheng  
y.zheng20@herts.ac.uk  
University of Hertfordshire  
Hatfield, United Kingdom

## Abstract

Achieving truly fluid interaction with robots with speech interfaces remains a hard problem, and the experience of current Human-Robot Interaction (HRI) remains laboured and frustrating. Some of the barriers to fluid interaction stem from a lack of a suitable development platform for HRI for improving interaction, even in robotic Wizard-of-Oz (WoZ) modes of operation used for data collection and prototyping. Based on previous systems, we propose the properties of interruptibility and correction (IaC), pollability, latency measurement and optimisation and time-accurate reproducibility of actions from logging data as key criteria for a fluid WoZ system to support fluid error correction. We finish by presenting a Virtual Reality (VR) HRI simulation environment for mobile manipulators which meets these criteria.

## CCS Concepts

• **Human-centered computing** → **HCI theory, concepts and models**; • **Computing methodologies** → *Robotic planning*; • **Computer systems organization** → **External interfaces for robotics**.

## Keywords

Human-Robot Interaction, Fluidity, Virtual Reality Environments, Wizard-of-Oz

## ACM Reference Format:

Carlos Baptista De Lima, Julian Hough, Frank Förster, Patrick Holthaus, and Yongjun Zheng. 2026. Achieving Interaction Fluidity in a Wizard-of-Oz Robotic System: A Prototype for Fluid Error-Correction. In *Proceedings of 2026 ACM/IEEE International Conference on Human-Robot Interaction (HRI'26)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/XXXXXXXX.XXXXXXX>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
*HRI'26, Edinburgh, Scotland, UK*

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 978-1-4503-XXXX-X/2018/06  
<https://doi.org/XXXXXXXX.XXXXXXX>

## 1 Introduction

The challenge of achieving fluid interaction remains prevalent within the field of human-robot interaction (HRI). Despite advances in robot vision, motion, manipulation and automatic speech recognition, state-of-the-art HRI is frequently slow, laboured, and fragile. Improving the feeling of fluidity is a difficult task to achieve, with a variety of challenges from the robot control algorithms to the physical constraints of the robots [1, 8, 9].

While fluidity has not been formally defined, it is felt by humans [9]. We define fluidity and fluid interaction as the ability to seamlessly transition in turn-taking to allow appropriate overlap of turns between agents, including multimodally, and to allow action using prediction. For example, in human-human pick-and-place situations, an instruction like “put the remote on the table” with a repair like “no, the left-hand table” typically has no delay in reacting to the initial instruction, and adaptation to the correction is instant. Fluid interaction for robots consists of more seamless, human-like transitions from processing speech to taking physical actions with no delays, permitting appropriate overlap between the two, and the ability to repair actions in real time as humans do [5].

While there has been some work on incremental processing for increasing the fluidity of HRI [1, 9], to allow for the development of control algorithms by data-driven means, Wizard-of-Oz (WoZ) interactions, where an experimental confederate ‘Wizard’ tele-operates a robot out of view of the user, are required. However, it is not clear to what extent current WoZ systems are capable of achieving and recording data for developing fluid HRI, nor which properties are required to achieve this.

After more precisely motivating our work and setting the contribution in context, we identify some core criteria of Wizard-of-Oz systems needed for interactive fluidity, briefly review existing WoZ-enabled robots and associated HRI experiments, specifically focusing on how they do or do not meet these criteria. We then present a prototype system in a VR environment which meets them.

## 2 Motivation & Related Work

Riek [11] provides a systematic review of 54 WoZ experiments published in the primary HRI publication venues from 2001–2011. They observed that research mainly used WoZ systems for simulating the natural language processing and non-verbal action functionality of systems (72.2% and 48.1% of papers, respectively). They found

**Table 1: An overview of existing HRI WoZ systems and their properties necessary for fluidity. Keys for the different interaction types: C = Controller, M = Mouse, K = Keyboard, P = Pen & Paper, H = Haptic/Touch, G = Gesture, T = Tablet Wizarding, S = Speech X = evidenced, ? = Mentioned category but not clearly discussed, - = Not Mentioned**

System	Robot name	Robot type	Real or Virtual	User interaction		Wizard interface	Interruptibility and correction			
				User interaction	Wizard interaction		Interruptibility and correction	Pollability	Reproducibility	Latency
[3]	Pepper	Social Robot	VR	H	M+K	Dashboard screen GUI	-	-	-	45ms
[7]	UR10	Manipulator Arm	RR	P	K	Command-line interface	X	X	-	-
[12]	Pepper	Social Robot	RR	?	M+K	Dashboard screen GUI	-	-	X	-
[14]	CityBot	Mobile Car	VR	M+K+C	G+S	Dashboard screen GUI	-	-	X	X
[13]	Citybot	Mobile Car	RR	S	M	Dashboard screen GUI	?	-	-	-
[17]	Pepper	Social Robot	RR	S + G	M	Dashboard screen GUI + Live Camera Feed	-	-	-	-
[2]	xArm 7	Manipulator Robot	RR	-	C	Dashboard screen GUI	X	-	-	X
[22]	Luxai QT robot	Social robot	RR	P	T	Dashboard screen GUI	-	-	-	-
[21]	Luxai QT robot	Social robot	RR	P	T	Dashboard screen GUI	-	-	-	-
[6]	Tiago+	Mobile Manipulator	RR	S	M+K	Dashboard screen GUI + Live Camera Feed	-	-	-	-
[20]	Multiple	Social Robots	RR	-	-	-	-	-	-	-
[18]	ResolutionBot	Social Robot	RR	H+S	C+M+K	Dashboard screen GUI + Live Camera Feed	-	-	-	?
[15]	UR10	Manipulator Arm	RR	S+T	M+K	-	-	-	-	-
[19]	Nao Robot	Social Robot	RR	S	?	-	-	-	-	-

that Wizards also performed a significant amount of navigation and mobility tasks (29.6% of papers) while manipulation-based tasks were rarely employed (3.9% of papers). The author hypothesises this is largely due to the difficulty of fully automating speech-based systems as well as maintaining effective tracking. Two themes that occur throughout the reviewed papers are the concern of accurately representing the natural delay that robots have within the Wizarded systems, and accounting for errors from the Wizard, intentional or otherwise. Both themes are important when considering fluidity and the potential for repairing ongoing actions.

One shortcoming of most current WoZ systems is that they tend to employ custom Graphical User Interfaces (GUIs) fit only for their specific problem with a finite set of potential actions [10]. The GUIs themselves can often lead to creating additional latency between a user's command and the execution of the action, as found by Runzheimer et al [14].

## 2.1 Defining Criteria for Fluid Interaction

Based on the authors' direct experiences of designing in-robot interaction control algorithms and an initial literature search, we posit the following criteria for WoZ systems to be capable of fluid interaction with users: *Interruptibility and correction (IaC)*, *Pollability*, *Latency measurement and optimisation*, and *Reproducibility of action timings*, which are explained below.

**Interruptibility and correction:** is the ability of the Wizard to either change and update the current ongoing action of a robot to a new one, or to cancel all ongoing actions. As robots are not guaranteed to deliver perfect processing, the ability to correct the robot partway through is important for building a robust robot that people can use in real-world contexts. Additionally it is important to have a method to cancel all ongoing actions for safety reasons. In fact, safety features in personal care robots are mandatory.<sup>1</sup> By delivering in these aspects, we can assure that we can confidently repair any incorrect actions, including those made by the wizard.

**Pollability:** refers to the definition described by Aylett and Romeo, who define something pollable as needing to know "what state the system is in and what it has done at any point in time" [4]. While their case focuses on speech-based robots, we widen pollability to the progress of completion of any action type. For example,

<sup>1</sup><https://www.iso.org/standard/53820.html>

if a robot like Niryo<sup>2</sup> is used to pick and place items, the system should be pollable such that the Wizard is able to discern how far it has progressed towards completing a pick or place action at any given time, and can therefore make an optimal decision about subsequent actions, including repairing or cancelling.

**Latency Measurement and Optimisation:** is the need for latency to be a central design concern. For effective and fluid interactions, latency must be accounted for and minimised, as even small latencies, when compounded in different parts of the system, can lead to noticeable delays.

**Reproducibility of Action Timings:** is a WoZ system's ability to replicate how users interacted with the robot after an interaction with high timing accuracy. One of the main reasons this is important is data collection for improved future interaction – the data gathered by accurately logging a user's actions can be either analysed, or used to train autonomous functions in the same robot and improve user experience in terms of responsiveness.

## 2.2 Review of Existing Systems

In an update to [11]'s review, we analyse the extent to which state-of-the-art WoZ systems exhibit these 4 criteria, we conducted a literature review of recent work on WoZ systems, which resulted in 14 papers, the full results of which will be published in a follow-up paper.<sup>3</sup> Additional classifications for each system were made for *Wizard and User modes of interaction*, and whether the system was *Virtual or Real*. The results can be found in Table 1 and are described below:

**Virtual or Real:** We found the majority of studies describe WoZ systems for controlling real-world robots. [14] implemented a

<sup>2</sup><https://niryo.com/>

<sup>3</sup>The inclusion criteria which define our scope were:

- The term 'robot' or a reference to the type of robot was referenced in either the title or the abstract.
- Papers that had at least one reported study using a Wizard-of-Oz system.
- Papers were classified as a study and not a report, book chapter or abstract.
- The paper was peer-reviewed and accepted for publication.
- Papers were included if the Wizarded systems involved real robots or virtual simulations of the robot.

Exclusion criteria were:

- Papers that focused on virtual agents without a real-world robotic equivalent instead of robots.
- Papers which were not available through the university services.

simulation of the CityBot and three scenes in Unity<sup>4</sup> with the user wearing the Meta Quest 2 headset.<sup>5</sup> Despite the benefits of a VR simulation, the challenges of achieving fidelity to the real-world robot and environment constitute significant software design and engineering overhead, which are possible reasons for the relative lack of VR HRI systems despite their benefits.

**Wizard and User Modes of Interaction:** There is no uniform method of designing the Wizard and User modes of interaction. For the Wizard, the use of a mouse and keyboard was the most popular mode of control, with most interfaces using a GUI dashboard with various buttons and keyboard shortcuts to execute pre-defined actions. In many cases it was not clear whether these interfaces displayed the robot's visual stream to the Wizard, or another view, such as an omniscient or third-party observation point. In many cases, it was not made clear if any action could be interrupted, cancelled, or even tracked in terms of its progression. The user interaction modes vary from haptic, to tablets to speech, depending on the affordances of the robot.

**Interruptibility and Correction:** If the WoZ system was reported to be able to repair current actions or cancel existing actions, it was considered to have the IaC property. Only two papers clearly achieved this. [7] stated that their system had an "interrupt node" which allowed the Wizard to send a signal causing the robot to withdraw a safe distance to avoid a collision with the participant. It is also noted that their system had an "override command" used to disable any running actions. The authors do not specifically mention whether the interrupt node could also be used to update the existing action.

**Pollability:** Only one paper [7] on a proposed system explicitly dealt with pollability being a core part of the system via their 'robot\_feedback' publisher which was displayed back to the user. The authors specify that the 3 states they defined were 'Nod', 'Observe' and 'Pose' and note that the Wizard could choose and control these states. However, it was unclear how the Wizard was confident in knowing how close the robot was to completing a task, or any granular information on task progress.

**Latency:** Only three studies mention and discuss latency. [14] discusses how their Wizard interface was complicated and difficult to use, increasing the latency between a user's request and the robot commencing the action. In particular, their interface was too cumbersome for just an individual Wizard to manage alone. Their UI contained a series of buttons, sliders and input fields for scene changes, alongside having separate MIDI controllers for audio responses. Despite the Wizard being trained and having to become familiar with the system, the management of all the various controls led to delays between the commands of the participant and the wizard; especially in scenarios where the wizard did not expect an interjection from the participant leading to not so fluid interaction.

**Reproducibility of Action Timings:** Only two examples of a reproducible system for timings of action were found. [14] recorded the views and audio of both the user and the Wizard - data which was then downloadable and viewable. One particular issue they came across with their setup was that they found it impossible to effectively synchronise the various streams due to the streams

coming from different sources with a variety of different settings and corrupted streams, due to their complex setup.

### 3 Virtual Reality HRI Platform Prototype

As discussed above, no existing system we reviewed meets our criteria fully, with limitations ranging from high latency to over-complexity of Wizard interfaces. We have developed a test-bed WoZ system which adequately satisfies the criteria in ways which we explain below. While applicable to a variety of robots, this system uses the mobile manipulator "Fetch" robot<sup>6</sup>, controlled by speech commands by a user in the same space.<sup>7</sup>

The system architecture is shown in Figure 1. The ROS and Gazebo<sup>8</sup>-powered *Robot and Environment Simulator* is housed in a Docker container, which sends ROS messages to the Unity-based *Visualizer*. The movement and planning for our Fetch model is implemented by the ROS-based 'MoveIt'<sup>9</sup> framework. The Visualizer, which is continuously listening for ROS messages to immediately update the relevant objects' position, provides synchronous, separate User and Wizard views of the scene. The user accesses the environment using a VR headset and hand-held controllers, while the Wizard has a separate screen to control the robot using mouse clicks. Once the Wizard has clicked on the environment for Fetch to complete a navigation or pick-and-place action, a ROS message is sent to our Python-based *Action Control and Logging module*. This control module sends the request to initiate the action back to the Robot and Environment Simulator for execution.

To be able to receive ROS messages in Unity, we use a ROS-TCP connector<sup>10</sup> and to receive ROS messages back from Unity we use a ROS-TCP-Connector package in the Docker container<sup>11</sup>. To keep the robot visualisation consistent, we use the Fetch UDRF descriptor file and meshes and import the robot model into the Unity environment using the UDRF-Importer package<sup>12</sup>.

**Wizard Interface:** The Wizard sees the simulated robot camera feed on the Fetch model from the Visualizer and has full continuous audio access to the User's microphone. Responding as fast as possible to user commands, the Wizard mouse-clicks on a target destination for moving and selecting objects and destinations. Once this is registered, a ROS message is sent to our control module. Visualisations signal the Wizard's current target location (red cone), target object (green highlight) and/or object destination (blue spot on surface). There is a simple dashboard with a 'Cancel all actions' button and a message area to signal illegal clicks.

**User Interaction:** The user wears a VR headset displaying the Unity Scenes and a microphone to interact with the (Wizarded) robot via speech. They also have limited movement capabilities through the use of hand-held controllers to navigate the environment.<sup>13</sup> All sensor data is sent to the control and logging module.

<sup>6</sup><https://www.zebra.com/us/en/products/autonomous-mobile-robots.html>

<sup>7</sup>The toolkit source code is available at <https://github.com/fluidity-project/fluidity-hri/> and a demo video of the system's correction behaviour is at <https://fluidity-project.github.io/media/HRIErrors.mp4>

<sup>8</sup><https://gazebo.org/>

<sup>9</sup><https://moveit.ai/>

<sup>10</sup><https://github.com/Unity-Technologies/ROS-TCP-Connector>

<sup>11</sup><https://github.com/Unity-Technologies/ROS-TCP-Endpoint>

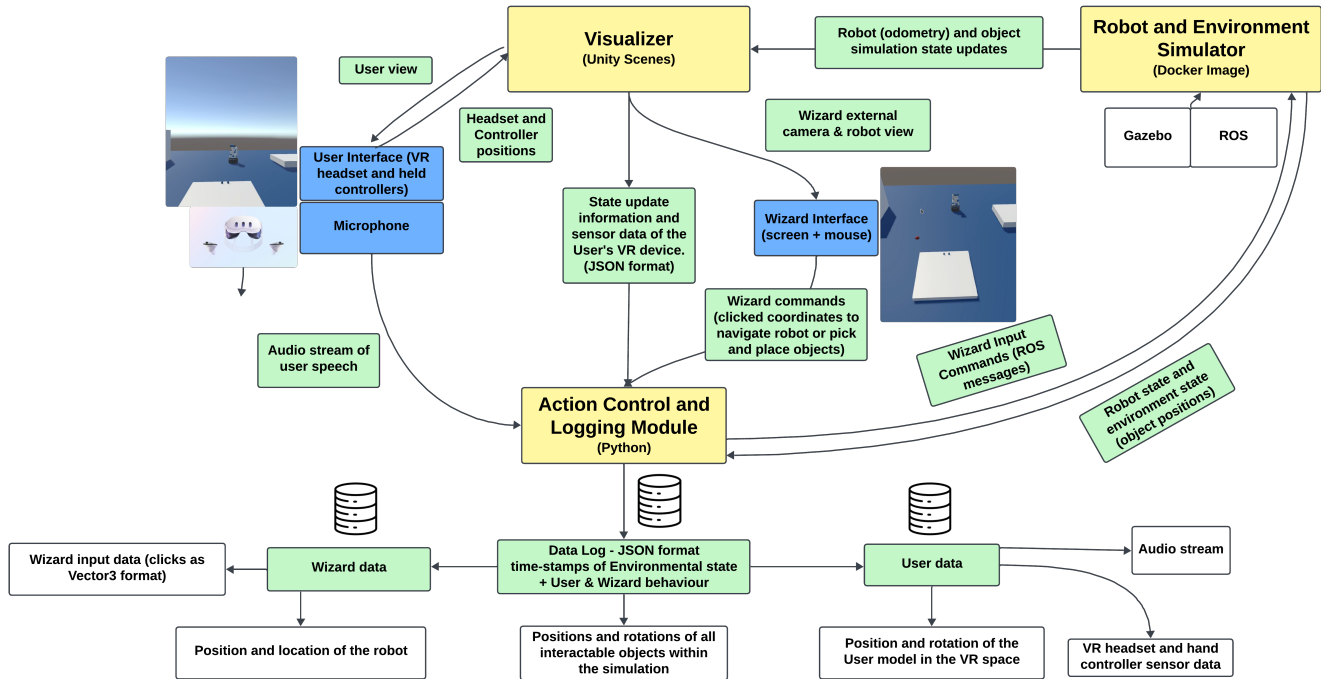
<sup>12</sup><https://github.com/Unity-Technologies/URDF-Importer>

<sup>13</sup>All Unity-compatible hardware can be used. We use the Meta Quest 3 VR headset and Meta Quest Touch Plus - <https://www.meta.com/gb/quest/quest-3/>

<sup>4</sup><https://unity.com/>

<sup>5</sup><https://www.meta.com/gb/quest/products/quest-2/>

**Figure 1: A diagram showing the system architecture of the proposed VR-HRI system. Hardware interfaces are blue boxes, processors are yellow, and data types are green.**



**Logging data:** In addition to the audio feed from the user’s microphone being recorded, all the following data is logged and time-stamped in a JSON format (with timestamps and an ID tag): 1) Wizard actions: time-stamped vectors of mouse clicks. 2) The robot’s state: position of its base and joint state. 3) Interactable objects within the space: positions and rotations. 4) The user’s view: position of the camera in the Unity scene as controlled by the VR headset. 5) User’s controller actions. In playback mode, log files can be read in by the system to simulate the interaction.

### 3.1 Discussion: Meeting of Fluidity Criteria

While the immediate next step in our future work is evaluating the system with real users, here we discuss how our VR WoZ prototypes meet our criteria identified in 2.1.

**Interruptibility and Correction** is met in our system by 1) Goal update functionality, which allows the Wizard to update the current action of the robot by a simple override with a button click on a new location or object; 2) The *cancel* button, which allows us to revoke all current goals to stop the robot from completing any remaining actions and come to a default state as quickly as possible, modulo deceleration.

**Pollability** is met as the Wizard can directly see the progress of the action through their view, and the data log records the state of the robot during any given action.

**Latency Measurement and Optimization** is achieved by using ROS nodes which tether the robot simulation to both User and Wizard views. The system can also attribute its overall latency to its various causes: 1) The delay from the user requesting an action via speech to the Wizard initiating that action. 2) The delay from

the Wizard’s perception of the action request to the action starting. 3) The delay from the action starting to the user perceiving the start of that action. 4) The delay from interrupting or repairing any given action. For minimising 1) and 3) we allow full access to the audio stream in the Wizard’s interface and continuous visual access to the synchronous Unity views. We minimise 2) by experimenting with different automatic recovery behaviours from the Fetch MoveIt package, which are shown to have a large effect on user perceptions of the robot’s abilities [16], here making them less conservative to lower the latency of task execution. For 4), fluid repair is achieved by the Wizard being able to select new target destinations and objects during ongoing motion, similar to [1].

**Reproducibility** is achieved by the accurate time-stamped logging of the data described above. While this data does not exhaustively include all visual information, full reconstruction of the User and Wizard behaviour and the changing states of the environment is possible, as are real-time User, Wizard and third-party views.

## 4 Conclusion and Future Work

Based on requirements for fluid HRI for WoZ-controlled robots, we have presented a prototype VR platform which allows for the functional properties of Interruptibility and correction, pollability, latency measurement and optimisation, and reproducibility of action timings. Our toolkit is generalizable beyond the Fetch robot to any ROS-based simulation and Unity environment, allowing customizable worlds and robots. In future work, we will run a user study exploring how effective the system is in facilitating fluid interactions both in WoZ mode and in autonomous mode experiments with systems trained on the WoZ-collected data.

## Acknowledgements

Baptista de Lima, Hough and Förster's work is funded by the FLUIDITY project (EPSRC grant EP/X009343/1).

## References

- [1] Mitchell Abrams, Thies Oelerich, Christin Hartl-Nesic, Andreas Kugi, and Matthias Scheutz. 2025. Incremental Language Understanding for Online Motion Planning of Robot Manipulators. In *Proceedings of IROS*.
- [2] Alexander Arntz. 2024. Enabling Safe Empirical Studies for Human-Robot Collaboration: Implementation of a Sensor Array Driven Control Interface. In *International Conference on Human-Computer Interaction*. Springer, 42–57.
- [3] Alexander Arntz, André Helgert, Carolin Straßmann, and Sabrina C Eimler. 2024. Enhancing Human-Robot Interaction Research by Using a Virtual Reality Lab Approach. In *2024 IEEE International Conference on Artificial Intelligence and Extended and Virtual Reality (AIxVR)*. IEEE, 340–344.
- [4] Matthew Peter Aylett and Marta Romeo. 2023. You Don't Need to Speak, You Need to Listen: Robot Interaction and Human-Like Turn-Taking. In *Proceedings of the 5th International Conference on Conversational User Interfaces*. 1–5.
- [5] Carlos Valter Baptista De Lima, Julian Hough, Frank Förster, Patrick Holthaus, and Yongjun Zheng. 2024. Improving Fluidity Through Action: A Proposal for a Virtual Reality Platform for Improving Real-World HRI. In *Proceedings of the 12th International Conference on Human-Agent Interaction*. 358–360.
- [6] Judith S Heinisch, Jérôme Kirchhoff, Philip Busch, Janine Wendt, Oskar von Stryk, and Klaus David. 2024. Physiological data for affective computing in HRI with anthropomorphic service robots: the AFFECT-HRI data set. *Scientific Data* 11, 1 (2024), 333.
- [7] David Hinwood, James Ireland, Elizabeth Ann Jochum, and Damith Herath. 2018. A proposed wizard of OZ architecture for a human-robot collaborative drawing task. In *International Conference on Social Robotics*. Springer, 35–44.
- [8] Patrick Holthaus, Trenton Schulz, Gabriella Lakatos, and Rebekka Soma. 2023. Communicative Robot Signals: Presenting a New Typology for Human-Robot Interaction. In *International Conference on Human-Robot Interaction (HRI 2023)*. ACM/IEEE, Stockholm, Sweden, 132–141. doi:10.1145/3568162.3578631
- [9] Julian Hough and David Schlangen. 2016. Investigating fluidity for human-robot interaction with real-time, real-world grounding strategies. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 288–298.
- [10] Nikolas Martelaro. 2016. Wizard-of-oz interfaces as a step towards autonomous hri. In *2016 AAAI spring symposium series*.
- [11] Laurel D. Riek. 2012. Wizard of Oz studies in HRI: a systematic review and new reporting guidelines. *J. Hum.-Robot Interact.* 1, 1 (July 2012), 119–136. doi:10.5898/JHRI.1.1.Riek
- [12] Finn Rietz, Alexander Sutherland, Suna Bensch, Stefan Wermtter, and Thomas Hellström. 2021. WoZ4U: an open-source wizard-of-oz interface for easy, efficient and robust HRI experiments. *Frontiers in Robotics and AI* 8 (2021), 668057.
- [13] María Trinidad Rodríguez-Domínguez, María Isabel Bazago-Dómine, María Jiménez-Palomares, Gerardo Pérez-González, Pedro Núñez, Esperanza Santano-Mogena, and Elisa María Garrido-Ardila. 2024. Interaction Assessment of a Social-Care Robot in Day center Patients with Mild to Moderate Cognitive Impairment: A Pilot Study. *International Journal of Social Robotics* 16, 3 (2024), 513–528.
- [14] Tabea Runzheimer, Stefan Friesen, Sven Milde, Johannes-Hubert Peiffer, and Jan-Torsten Milde. 2024. Exploring VR Wizardry: A Generic Control Tool for Wizard of Oz Experiments. In *International Conference on Human-Computer Interaction*. Springer, 60–73.
- [15] Moritz Schmidt and Claudia Meitinger. 2024. Convenience vs. Reliability? Evaluation of Human-Robot Interaction Preferences in a Production Environment. In *International Conference on Human-Computer Interaction*. Springer, 168–179.
- [16] Trenton Schulz, Rebekka Soma, and Patrick Holthaus. 2021. Movement acts in breakdown situations: How a robot's recovery procedure affects participants' opinions. *Paladyn, Journal of Behavioral Robotics* 12, 1 (2021), 336–355.
- [17] Barbara Sienkiewicz, Gabriela Sejnova, Paul Gajewski, Michal Vavrecka, and Bipin Indurkha. 2023. How language of interaction affects the user perception of a robot. In *International Conference on Social Robotics*. Springer, 308–321.
- [18] Yao-Lin Tsai, Chinmay Wadgaonkar, Bohkyung Chun, and Heather Knight. 2022. How service robots can improve workplace experience: Camaraderie, customization, and humans-in-the-loop. *International Journal of Social Robotics* 14, 7 (2022), 1605–1624.
- [19] Caroline L Van Straten, Jochen Peter, Rinaldo Kühne, and Alex Barco. 2022. The wizard and I: How transparent teleoperation and self-description (do not) affect children's robot perceptions and child-robot relationship formation. *Ai & Society* 37, 1 (2022), 383–399.
- [20] Paulina Zguda, Alicja Wróbel, Paweł Gajewski, and Bipin Indurkha. 2024. "He Can Walk, He Just Doesn't Want To"-On Machine/Human-Likeness of Robots in Polish Children's Perception. In *International Conference on Human-Computer Interaction*. Springer, 221–239.
- [21] Jianling Zou, Soizic Gauthier, Salvatore M Anzalone, David Cohen, and Dominique Archambault. 2022. A wizard of oz interface with qtrobot for facilitating the handwriting learning in children with dysgraphia and its usability evaluation. In *International Conference on Computers Helping People with Special Needs*. Springer, 219–225.
- [22] Jianling Zou, Soizic Gauthier, Hugues Pellerin, Thomas Gargot, Dominique Archambault, Mohamed Chetouani, David Cohen, and Salvatore M Anzalone. 2024. R2C3, a rehabilitation robotic companion for children and caregivers: the collaborative design of a social robot for children with neurodevelopmental disorders. *International Journal of Social Robotics* 16, 3 (2024), 599–617.