NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

RODRIGO PEREIRA CARVALHO

BSc in Electrical and Computer Engineering

# HUMAN ROBOT INTERACTION

## MULTIMODAL CUES FOR NONVERBAL COLLABORATIVE MANIPULATION FRAMEWORK

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
March, 2024

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING

# HUMAN ROBOT INTERACTION

## MULTIMODAL CUES FOR NONVERBAL COLLABORATIVE MANIPULATION FRAMEWORK

**RODRIGO PEREIRA CARVALHO**

BSc in Electrical and Computer Engineering

**Adviser:** José António Barata de Oliveira
*Full Professor, NOVA University of Lisbon*

**Co-adviser:** Francisco Marques
*Research Engineer, UNINOVA-CTS*

**Examination Committee**

**Chair:** Fernando José Almeida Vieira do Coito
*Associate Professor, FCT-NOVA*

**Rapporteur:** João Almeida das Rosas
*Assistant Professor, FCT-NOVA*

**Co-adviser:** Francisco Marques
*Research Engineer, UNINOVA-CTS*

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
March, 2024

**Human Robot Interaction**
**Multimodal Cues for Nonverbal Collaborative Manipulation Framework**



This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.10.16) .

# Acknowledgements

I would like to thank my adviser, Professor José António Barata de Oliveira, who followed my academic path in the last 5 years through the teaching of several classes, sharing his knowledge and passion for the Robotics field, and guiding and supporting the interests of his students.

I would also like to thank my co-adviser, Francisco Marques, for allowing me to partake in the development of this work and for the support given during its implementation and the writing of this document.

I would like to acknowledge all of the professors in the Department of Electrical and Computer Engineering at the Nova School of Science and Technology for their contributions to this degree.

A special thank you to the group of friends I've made during this adventure, the "Motumbas," with whom I celebrated the highs and powered through the lows. They made this journey unforgettable, and I wouldn't want to experience it with anybody else.

At last, I would like to show my appreciation to my family, whose unconditional support and encouragement during my life, especially during this academic period, led to the person that I am today and allowed me to achieve this milestone.

# Abstract

Humans and robots have traditionally been employed for distinct tasks in separate environments in a manufacturing setting. This is due to factors such as their different skill sets and safety concerns. In the event of a task that requires both a human and a robot to participate, this setup fails. In order to achieve a safe and coordinated environment in this scenario, the two must be able to communicate in an effective manner.

This work proposes an approach for implementing a set of nonverbal cues using tools available on an open-source manipulation framework to be used by a robotic manipulator in collaborative sequential and assembly tasks, as well as a potential approach to avoiding collisions between humans and robots.

The cues were tested in a collaborative task in which a robotic manipulator informs a human counterpart about subtasks that the human would be required to perform. Their conveying performance was evaluated through a questionnaire that was answered by 21 participants from various professional backgrounds. The results showed that the implemented cues were capable of conveying their intent with good quality on average, demonstrating that, while more fine tuning is required, using these tools to implement effective communication is feasible.

**Keywords:** Human-robot interaction, Human-robot communication, Nonverbal, Gestures, Collaborative task, Collaborative robot, Manipulation, Safety

# Resumo

Humanos e robôs têm sido tradicionalmente designados para tarefas distintas em espaços separados em ambientes de manufatura. Isto deve-se a fatores como as suas aptidões distintas e por motivos de segurança. No caso de uma tarefa que exija a participação de tanto um humano como de um robô, esta configuração não funciona. De modo a concretizar um ambiente seguro e coordenado num cenário destes, ambos têm de ser capazes de comunicar de forma eficaz.

Este trabalho propõe uma abordagem de implementação de um conjunto de gestos não verbais utilizando ferramentas disponíveis num *framework* de manipulação *open-source* para serem usadas por um manipulador robótico em tarefas sequenciais e de montagem colaborativas, bem como uma possível abordagem para evitar colisões entre humanos e robôs.

Os gestos não verbais foram testados numa tarefa colaborativa em que um manipulador robótico informa um colega humano sobre as subtarefas que este necessita de realizar. A qualidade de perceção dos gestos não verbais foi avaliada através de um questionário que foi respondido por 21 participantes de vários percursos profissionais. Os resultados mostram que os gestos não verbais implementados foram capazes de comunicar sua intenção com boa qualidade em média, demonstrando que, embora seja necessário melhorar certos aspetos, usar estas ferramentas para implementar métodos de comunicação eficazes é viável.

**Palavras-chave:** Interação homem-robô, Comunicação homem-robô, Não verbal, Gestos, Tarefa colaborativa, Robô colaborativo, Manipulação, Segurança

# Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| **2D** | Two-Dimensional |
| **3D** | Three-Dimensional |
| **AI** | Artificial Intelligence |
| **API** | Application Programming Interface |
| **AR** | Augmented Reality |
| **ARHMD** | Augmented Reality Head-Mounted Display |
| **CHOMP** | Covariant Hamiltonian Optimization for Motion Planning |
| **EOL** | End of Life |
| **HRI** | Human-Robot Interaction |
| **MR** | Mixed Reality |
| **OMPL** | Open Motion Planning Library |
| **PTP** | Point-to-Point |
| **RGB** | Red, Green and Blue |
| **ROS** | Robot Operating System |
| **ROS4HRI** | ROS for Human-Robot Interaction |
| **RPC** | Remote Procedure Call |
| **STOMP** | Stochastic Trajectory Optimization for Motion Planning |
| **VR** | Virtual Reality |

**XR** Extended Reality

# 1 Introduction

## 1.1 Scope

From domestic robots that are capable of executing mundane household tasks to service robots that are deployed in hotels, hospitals and other healthcare facilities [1, 2], the use of robots is becoming more prevalent in today's society. This surge in Human-Robot Interactions (HRIs) has brought with it the need for robots to be able to interact in a safe and effective manner with human beings [3]. In the manufacturing industry, robots have generally been employed to replace human operators in dull, physically demanding and repetitive operations like welding and painting since they can generally complete these tasks more quickly and with higher accuracy [4]. Humans, on the other hand, remain significantly superior in scenarios requiring high adaptability or a high cognitive problem solving ability. Because of their method of operation, which often involves executing activities at high speeds with frequently heavy objects or equipment, these robots have been kept separated from unauthorized personnel, surrounded by a cage, and only able to be in its proximity when the power has been switched off. It quickly becomes clear that this approach is inadequate when faced with a task that calls for both human and robotic abilities. The possibility of interaction and collaboration between humans and robots can improve the efficiency or even allow that such tasks may become feasible to execute.

To enable such a feature, the ability for both the human and the robot to communicate with one another is a must. Humans are distinguished by their ability to communicate. Human communication abilities have evolved over time and now comprise from simple gestures, facial expressions and sound cues to a range of spoken languages capable of conveying our complex thoughts in minute detail, and are now an important component of any human interaction.

## 1.2 Problem

The type of collaborative tasks known as sequential or assembly tasks are very prevalent, particularly in the manufacturing industry. A series of steps must be carried out in a

specific order in order to complete the task, and depending on the task's design, both the human and the robot are responsible for different roles. The main issue with sequential or assembly tasks is that they are frequently complex and entail extensive coordination between humans and robots. The robot must be able to understand human instructions and respond appropriately, while humans must be able to accurately interpret the robot's responses and requests. Implementing such capabilities would allow the human-robot team to achieve a greater performance while also providing a means to ensure a safe and coordinated environment. Much research has been conducted in topics such as speech and gesture recognition [5], and action prediction [6] in order to provide the robot with the ability to "understand" the human's instructions and intentions. However, an interaction is a bi-directional event, meaning that it is critical to enable the direction of robot-to-human communication as much as the the direction of human-to-robot communication.

The research question that this dissertation aims to answer is what are the obstacles and potential solutions for implementing effective nonverbal communication between human and robot collaborators in complex sequential or assembly tasks?

## 1.3 Proposed Solution

To address this question, this dissertation seeks to setup a system that enables robot-to-human communication, thereby reducing some of the barriers associated with collaborative assembly tasks. Concretely, the proposed solution focuses on the development of a set of nonverbal cues designed to be used by robotic manipulators. Nonverbal communication is an important component of human interaction, and giving robotic systems similar communication capabilities makes interactions feel more natural and seamless. The goal is to develop these cues in a manner that allows for a simple implementation within manipulation frameworks. The cues aim to facilitate the conveying of actions that enhance collaboration between robots and humans in various sequential and assembly task scenarios. The conveying quality of these cues is intended to be acceptable to the average person, and it will be evaluated using feedback from a questionnaire about their use in a manipulation collaborative task example. The successful development of the proposed method will allow robotic systems to communicate with people in a manufacturing environment more effectively, boosting the overall usability and acceptance of robotic systems in numerous fields.

## 1.4 Document Structure

This document presents the following chapter structure following the introduction:

1. **State of The Art:** This chapter presents the research conducted in the field of human-robot interaction, focusing on the topic of the communication approachs within collaborative tasks.

2. **Proposed Approach and Implementation:** This chapter introduces the proposed system and provides a brief description of its components. It also outlines the resources used for system implementation and delves into the implementation details of some components, with a primary focus on the nonverbal cues.

3. **Test Design:** This chapter describes the design process for the task used in the experimental tests, including breakdowns of the layout and desired workflow.

4. **Experimental Results:** This chapter describes the method used to collect data on the perception of participants about the cues used in the test task, as well as an analysis of the results to determine their effectiveness.

5. **Conclusions and Future Work:** This chapter provides a summary of the document, including the conclusions drawn from the state of the art research as well as the conclusions derived from testing the implemented approach. To close the chapter, some recommendations for future work are presented, including suggestions for improvements to the approach and identification of aspects that were not explored.

# 2

# State of The Art

This chapter presents some of the research done in the area of communication in human-robot interactions, focusing primarily on the robot-to-human communication component used in collaborative tasks.

## 2.1 Communication Techniques and Approaches

In a human-robot interaction, the ability to share and understand information in a clear and succinct manner can help build trust, promote safety and assist collaboration. Similar to human-human interactions, a HRI might involve explicit and/or implicit communication [5]. An explicit communication is performed when the message's meaning is stated in a direct, understandable manner with verbal and written communication being two examples. An implicit communication occurs when the message's meaning must be inferred by the recipient. An example of that is nonverbal communication such as gestures, facial expressions, and posture. When both explicit and implicit communication are used, its called multi-modal communication, which usually happens in an attempt to ensure the recipient's understanding of the message.

Early implementations of HRIs such as [7, 8] relied on these communication methods. The main issues identified from those implementations were: only being able to receive and deliver a few well-defined commands; no mixed initiative, i.e., only the human could initiate an interaction; an inability to communicate about the environment or situation the robot was currently in (dynamically); and the robot's inability to recognize or produce nonverbal cues [9].

Many studies have been conducted since then in order to resolve these issues and enable a more natural and intuitive HRI. Some approaches favor an explicit approach in which verbal communication is the primary mode of communication, whereas others emphasize conveying all information through nonverbal cues. Because all humans communicate using modalities such as voice or body language, communicating with a robot that lacks these abilities would be difficult [10]. However, recent studies, have shown that they can leverage the robot's side with the implementation of Extended Reality (XR) communication

technologies.

## 2.2 Verbal Communication

Verbal communication can be defined as a method of conveying information via the use of words, spoken or written and by nature, humans rely on it in the majority of their interactions. Languages offer a wide variety of vocabulary and grammatical structures that enable speakers to express themselves in complex and subtle ways, making them one of the most versatile forms of communication. This versatility gives people the ability to adjust their sentences to the audience, which opens up the possibility of communicating complicated ideas, feelings, and experiences with a level of clarity and precision that would be challenging through nonverbal communication alone.

In human-human interactions, some research suggests that verbal communication of intent and expectations can significantly improve teamwork between humans when comparing with other communication modalities [11, 12]. Therefore, in a human-robot interaction, using language as a communication interface may bring the same benefits as it allows the robot to express itself more freely and requires minimal user training. However, verbal communication requires a significant amount of time and effort for the sender to produce a coherent message and for the receiver to listen and understand it, particularly in situations where context may be lacking [13]. Also, in order to produce speech, the robot needs to be equip with a text-to-speech or a dialogue management system, which, due to the computational load, often needs to be implemented in separate [14]. As a result, the decision to use verbal communication must be carefully considered, especially in a HRI scenario.

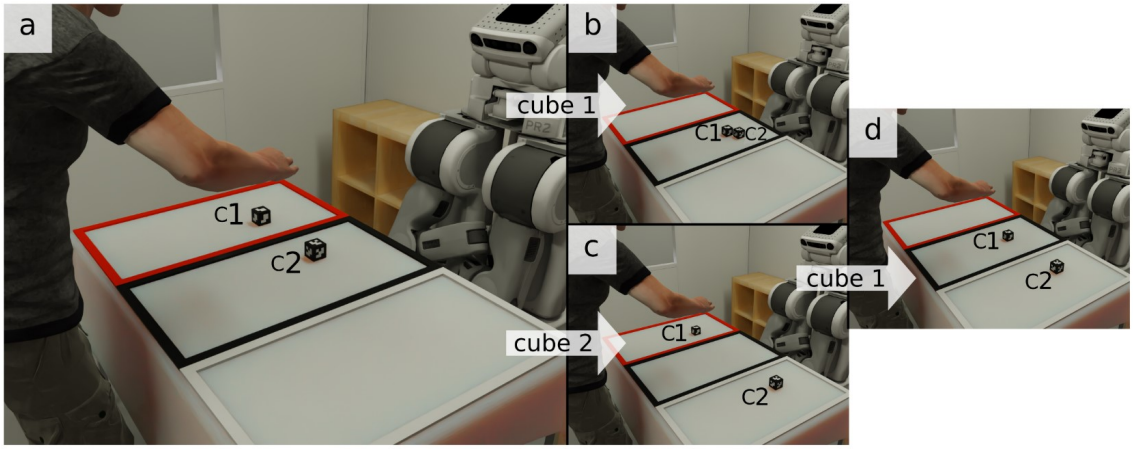



Figure 2.1: Task scenario used to study verbal communication on human-robot collaboration [15].

The research of verbal communication in HRIs focuses mostly on the topic of social robotics [16, 17], with fewer cases concentrating on its implications on cooperation and/or collaboration scenarios. One example is a study done by Buisan, Sarthou, and Alami [15]

in which a robot and a human operator had to position same color cubes in their respective area. The robot could distinguish the cubes but couldn't manipulate them and therefore had to indicate the actions to perform to the human operator. In this situation the robot is able to analyse its environment and convey an instruction in the minimum number of steps and avoid dead-locks in the communication. In this case, a deadlock would be if two cubes ended up in the same area, as depicted in the b) component of Figure 2.1, in which the possible solutions to distinguish both would be through nonverbal cues such as pointing or with a much more complex and detailed verbal instruction. During their study, they concluded that, in situations of low complexity, verbal communication had the lowest "cost" and was effective at conveying the required actions. However, in more complex scenarios, nonverbal cues such as pointing presented a lower "cost" at conveying the same information.

Nikolaidis et al. [18] conducted a comparison of verbal and nonverbal communication using the collaborative task of carrying a table. There were two types of verbal communication: *verbal commands*, in which the robot states how to carry out the action, and *state-conveying actions*, in which the robot expresses an opinion on why to carry out the task in that manner. They demonstrated that verbal communication was the most successful form of communication, as 100% of participants were able to adapt to the robot, as opposed to the 60% achieved with nonverbal communication. However, as the authors point out, verbal communication can become particularly challenging to implement optimally, since this is usually done by translating information encoded in the form of a cost function. Furthermore, the robot must convey information in a well-justified manner that allows the recipient to understand it without overwhelming them with unnecessary information. They also found that when the robot simply stated the action to be performed (*verbal command*), many participants questioned the robot's decision-making. However, when the robot followed the command with an explanation of why it should be done that way, the majority of the participants followed its plan.

Younes et al. [19] implemented a multi-layer AI-generated instruction verbalizer to be used in the context of assembly tasks. For testing, human participants were given instructions on how to perform a sequential assembly task using LEGO™ bricks. For each action, the participants would receive one of four styles of descriptions: (1) the most popular description from data collection; (2) the least popular description from data collection; (3) an AI-generated description without mention of structures in the environment; and (4) an AI-generated description with mention of structures in the environment. They found that, overall, the most popular descriptions from data collection resulted in the fastest completion time of the task, most often associated with the correct placement of the bricks. Both styles of AI-generated descriptions were on par with the least popular descriptions from data collection in terms of completion time. However, on average, the AI-generated descriptions resulted in more correct placements of the bricks when compared with the least popular descriptions. The authors also found that including pointing cues while verbalizing the instructions improved comprehension and helped

reduce the complexity of the verbal explanation.

Singh et al. [20] studied the implementation of verbal explanations performed by a team of collaborative robots, which are meant for interpretation by human bystanders. The implementation divides the robots' plan into partitions, allowing verbalization to support Grice's maxim of quantity [21]. Each partition is described using predefined template words and phrases. Their approach was tested and compared to two others: (1) a so-called "single" approach, in which one of the robots describes the next action and then performs it, and (2) a "random" approach, in which a robot selects one to four actions, waits for them to be performed, and then verbalizes an explanation. Ultimately, they concluded that, while the "random" approach ranked highest in terms of personal preference, their approach outperformed the others in terms of measured recalling accuracy. This demonstrated that, while a more calculated approach to describing the actions resulted in a better understanding of the task, the unpredictable variation in the length of the descriptions made the robots' explanations appear more natural.

As previously stated, trust is an important factor in HRIs, and it is highly dependent on communication methods. Ciocirlan, Agrigoroaie, and Tapus [22] compared trust levels in a HRI when the robot used one of three communication types: (1) no written or verbal communication, (2) non-related written and verbal communication, and (3) related written and verbal communication. They discovered that when the robot used the third technique, the human counterpart's trust level increased more, with signs that the trust level declined less when the robot failed the task.

## 2.3 Nonverbal Communication

Nonverbal communication covers all modes of communication that do not involve the use of words. Despite the fact that verbal communication may appear to be the major source of information in communication, Mehrabian [23] demonstrated that nonverbal communication is the dominant one. It was found that, in a face-to-face conversation, 93% of communication is nonverbal, consisting of 55% body language and 38% tone of voice. Given that, interacting with a robot that lack the ability to express itself through nonverbal cues could feel unnatural. Moreover, not all scenarios lend themselves to verbal communication. In a factory setting, for example, noise levels may interfere with the human's ability to hear the robot or render the robot's speech recognition useless [24].

### 2.3.1 Gestures

One of the aspects studied in nonverbal communication it's the use of gestures in interactions. Gestures are a type of nonverbal communication in which meaning is conveyed by motions of the body, hands, arms, or face. Nehaniv et al. [25] classified the types of human gestures as presented in Table 2.1.

Table 2.1: Classification of Gestures (based on [25]).

| Type | Defining Characteristics |
|---|---|
| "Irrelevant" and Manipulative Gestures | Influence on non-animate environment; Manipulation of objects, side effects of motor behavior, body motion |
| Side Effect of Expressive Behavior | Expressive Marking; Associated to communication or affective states |
| Symbolic Gestures | Conventionalized Signal in Communicative Interaction: communicative of semantic content |
| Interactional Gestures | Used to initiate, maintain, regulate, synchronize, organize or terminate various types of interaction |
| Referential/Pointing Gestures | Pointing of all kinds with all kinds of effectors (including eyes): referential, topicalizing, attention-directing |

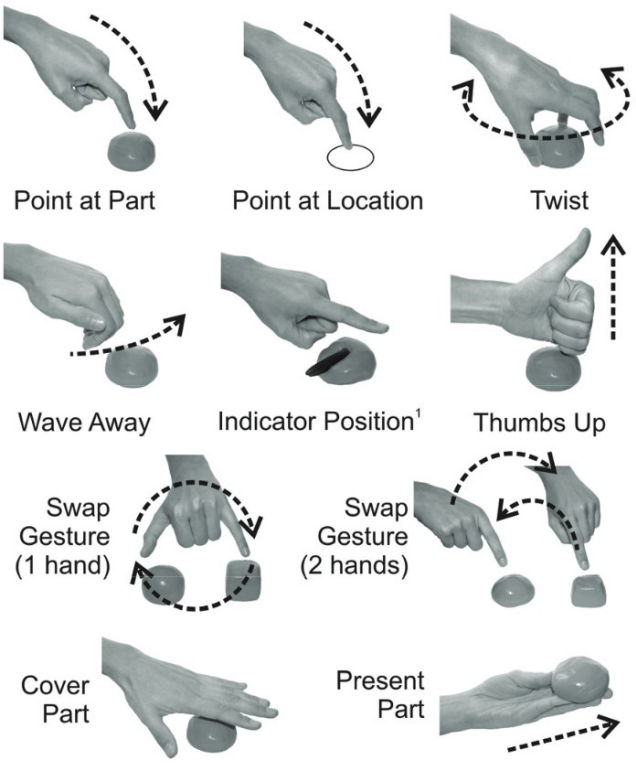



Figure 2.2: Lexicon of gestures derived from a human-human collaborative task [26].

Gleeson et al. [26] conducted a study on the use of communicative gestures by a robotic arm for industrial assembly tasks. The lexicon of gestures implemented, depicted in Figure 2.2, was derived from observing humans perform similar tasks to those used for testing. The gestures can be categorized as part acquisition, part manipulation, or part operations. They compared the intuitive level of the gestures when performed by the robotic arm or

by a human. The results were positive and revealed that the levels of intuitiveness of the gestures were statistically indistinguishable between those performed by the robot and humans. Some caveats of this study are that the gestures were implemented by manually moving the robotic arm through the desired motions and paths, which limits the execution of the same gestures in a slightly different environment configuration. In addition, the human participants who ranked the gestures were already aware of the task to be performed, which may have helped in determining the gesture's intentions.

Similarly, Sheikholeslami, Moon, and Croft [27] performed a study that implemented nonverbal gestures on a three-fingered robotic manipulator. In an equivalent manner, the gestures used were identified by first observing human gestural communication during human-human collaborative tasks and fitted one of the following categories: (1) directional, (2) orientational, (3) manipulation, and (4) feedback. The same gestures performed by humans were recorded and rated for perceived quality. The robotic arm was then manually moved through the desired motions in order to implement the nonverbal gestures. The study found that while the majority of the robot's gestures were reasonably identifiable (recognition rate greater than 60%), human gestures more often outrated the robot's. According to the authors, one possible explanation is that the morphology of the end-effector made it difficult to implement gestures that relied heavily on hand poses, such as the "thumbs up" pose.

The majority of HRI studies found on the topic of gestures focus on the predictability and legibility of arm movements with Dragan, Lee, and Srinivasa [28] being an example of that. They did a study comparing two motion types of a robotic arm: predictable and legible. By predictable motion, it's meant a movement natural to humans or stereotypical of humans that would be expected knowing its intent. On the other hand, a legible motion it's a movement which it's done to help the inference of said intention (Figure 2.3).

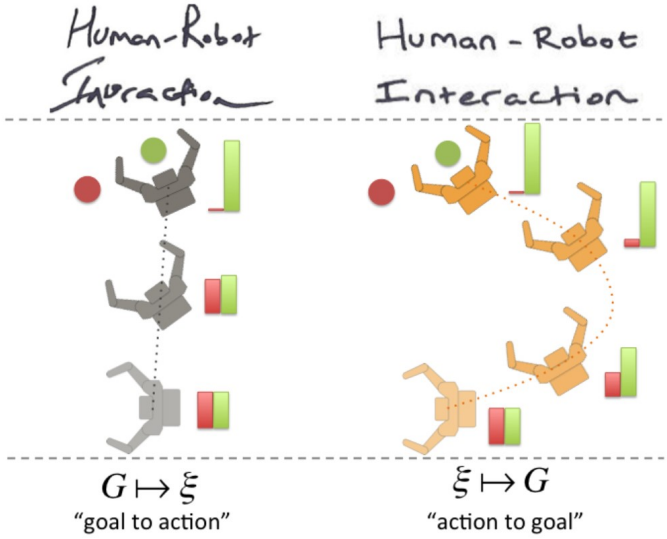



Figure 2.3: Predictable Motion (Left) vs. Legible Motion (Right) [28].

The two types of motion were performed by three different characters: a point robot, a human, and a bi-manual robot, and were compared in two phases: one in which the participant is given the goal and must evaluate the motion based on their expectation of the movement, and another in which the participants had to guess the goal of the motion, recording the both the time it took as well as the prediction. The predictable motion for both the point robot and the human was the one that aligned with what the participants expected. The results for the bi-manual robot were mixed, but favored legible motion. This happened because participants based their expectations on the robot's characteristics, believing that the robot, for example, was not flexible enough to perform a straight movement. When guessing the goal of a motion, the legible motion was the one which performed the best, achieving a higher percentage of correct guesses. This study concluded that the expected motion towards a goal is based on the perception of the character's movement capabilities, and that the type of motion used can change preference depending on the observer's knowledge state.

Holladay, Dragan, and Srinivasa [29] then ran an experiment to compare cost optimization and legible pointing techniques. A ray model approach was used to implement the cost optimization technique, which aligned the end-effector with the goal object with the least amount of travel from the starting point. The legible technique was implemented in the same way, but with the added benefit of allowing the robot to move more freely by accounting for the likelihood of an observer selecting other objects in the environment. When they compared the two, they discovered that the legible pointing consistently produced the best results. Because the clarity of the goal changes with the observer's viewpoint when interpreting a pointing action, the exaggerated pose generated by the legible pointing allowed for a better understanding of the goal.

Following the experiment of Dragan, Lee, and Srinivasa [28], Stulp et al. [30] used a model-free reinforcement learning approach to achieve a more legible motion of a robotic arm through human feedback. They were able to demonstrate that this approach allowed the robot to autonomously improve the legibility of its motions without the implementation of a model based on humans legibility concept.

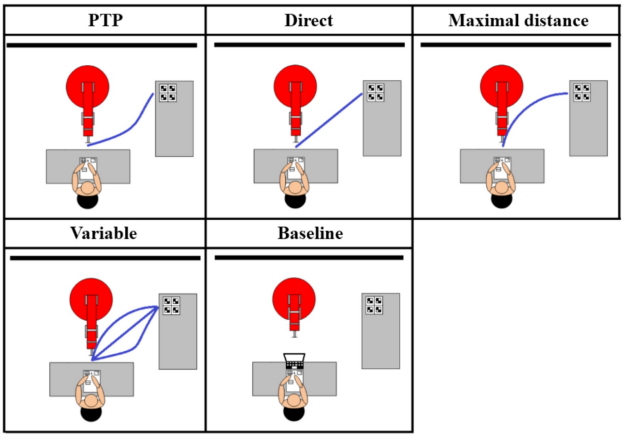



Figure 2.4: Types of motion strategies performed by a robotic arm [31].

Bortot, Born, and Bengler [31] compared the impact of four types of robot motion strategies (Figure 2.4) on human well-being: Point-to-Point (PTP), direct (straight), maximal distance, and variable (randomly selecting one of the previous). They concluded that the movement that resulted in the most well-being was the direct one, which was also voted the most predictable by the human participants.

Huber et al. [32] conducted an experiment comparing a hand-over task performed by a human-human pair and a human-robot pair. Two movement profiles were tested for the robot: *minimum jerk* velocity in spatial coordinates and *convectional trapezoidal* velocity in joint space. Between the two profiles, it was discovered that the *minimum jerk* performed better, being rated as more human-like and safe by the human participants. When the results of the human-human pair and the human-robot pair with that *minimum jerk* profile were compared, it was discovered that the hand-over task took one second longer on average with the human-robot pair, with the release time of the object by the robot being at fault, as the reaction and manipulation times were identical to the human-human pair. This experiment also demonstrated that a maximum velocity of the robot of 1m/s felt subjectively safe for the human counterpart.

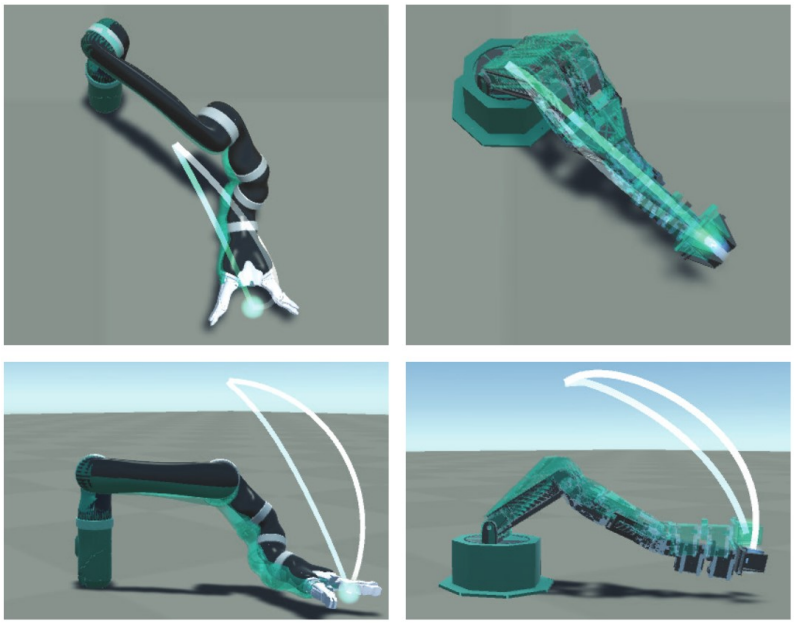


Figure 2.5: Perception of motion in different viewpoints [33].

Bodden et al. [33] also compared three types of motions to assess predictability and intent: minimum path length in joint-angle space (simple), minimum end-effector path length (straight), and a "legible" motion type based on the one by Dragan, Lee, and Srinivasa [28]. All three motions were implemented in three different robotic arms (Mico, Reactor, and UR5) in a virtual environment with the same fixed camera position. Participants in the experiment were instructed to predict the goal position of the motion as soon as they had a guess. They came to the conclusion that the straight motion performed best across all three robot arms. However, each motion type performed very differently

in each robot arm within each measurement. According to the author, this could have happened for three reasons: (1) Score measurements were calculated in relation to the total time of the motion. Because each motion took a different amount of time to complete, it is easier to achieve a higher score without having the best absolute time; (2) by only having one viewpoint, some movements in certain robots appear identical despite being different, as depicted in Figure 2.5, implying that having multiple viewpoints is important to avoid this issue; and (3) because all three robots had different kinematic designs, some movement types are difficult to achieve or become overall similar to another motion type.

### 2.3.2 Gaze

The direction of a person's gaze, or the direction of their eyes, is an essential nonverbal communication modality that can express a variety of emotions and attitudes. Eye contact is seen as an essential part of interpersonal communication because it conveys a sense of connection and engagement. The topic of using gaze as a communication modality in a HRI has primarily been studied from a human-to-robot perspective as a way to give command or infer the human's awareness of its surroundings [34–38], with little research in the opposite direction. Nevertheless, much potential can be seen in using a robot's gaze as a communication tool.

Kshirsagar et al. [39] conducted a study comparing the effects of various robot gaze behaviors in the reaching phase of a handover human-robot collaborative task. Based on human-human handover tasks, the robot would either look only at the human's hand, only at the human's face, or transition from the human's face to the human's hand. Their experiments revealed that the transition gaze was preferred by the human counterpart, while looking at just the hand was the least preferred, despite being the most common behavior seen in human-human handovers.

Faibish et al. [40] conducted a follow-up study in which they investigated the effect of gaze on the reaching, transferring, and retreating stages of a human-robot collaborative task. The robot would either look at the person's hand during the reaching and transferring phases and only at their face during the retreating phase (Hand-Face gaze), look at their face during the reaching phase, switch to the hand during the transferring phase, and look back at the face at the retreating phase (Face-Hand-Face gaze), or only look at the hand during the three phases. These decisions were also based on human-human handover tasks. Similarly, the preferred behavior during the interaction was the Face-Hand-Face gaze, with the least preferred being the Hand gaze. Although participants supported the Face-Hand-Face gaze as more human-like, it is not the most common behavior used in a human-human handover task according to their study, in which the Hand-Face gaze is favored. Unlike humans, the robot can keep track of the object during the handover without sacrificing the social aspect of the task.

In these previous approaches a screen was used to represent eyes and emulate the gaze of the robot. However Terzioğlu, Mutlu, and Şahin [41] showed that it is still possible

to attain good gaze perception by using a robotic arm and its gripper to mimic gaze, eliminating the need for an external device (Figure 2.6).

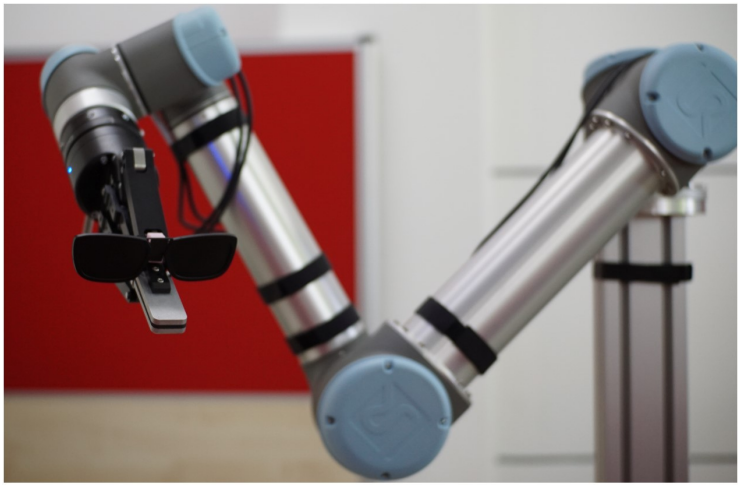


Figure 2.6: Use of a robotic arm as a gaze interface [41].

## 2.4 Extended Reality

The use of extended reality techniques in HRIs has recently been investigated as a supplement to communication methods such as those discussed previously, mostly due to the recent developments in areas such as Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR) [42]. Virtual Reality consists on creating a fully digital and immersive environment where users can interact with it through the use of a headset and other devices. Augmented Reality consists on projecting digital information and content onto the real word. This can be experienced typically through a handheld device with a camera, such as a smartphone or a headset. Mixed Reality involves blending elements of virtual and physical environments to create a new experience that is both real and virtual, allowing users to interact with digital objects as if they were real.

Bolano, Roennau, and Dillmann [43] implemented a system that used both visual and audio feedback to assist the operator and enable a more safe and predictable workspace. The robot had the ability to sound alerts when detected a possible collision. A screen was prompted in front of the operator that showed the motion planning of the robot, such as path and goal position, as well as the robot's perception of the environment. Bolano et al. [44] later added the possibility of projecting some of the information displayed on the screens into the workspace, reducing attention divergence to those external devices (Figure 2.7). They concluded that by providing information to the operator using these methods, they were able to feel more safe and comfortable, and acceptance of the robot.

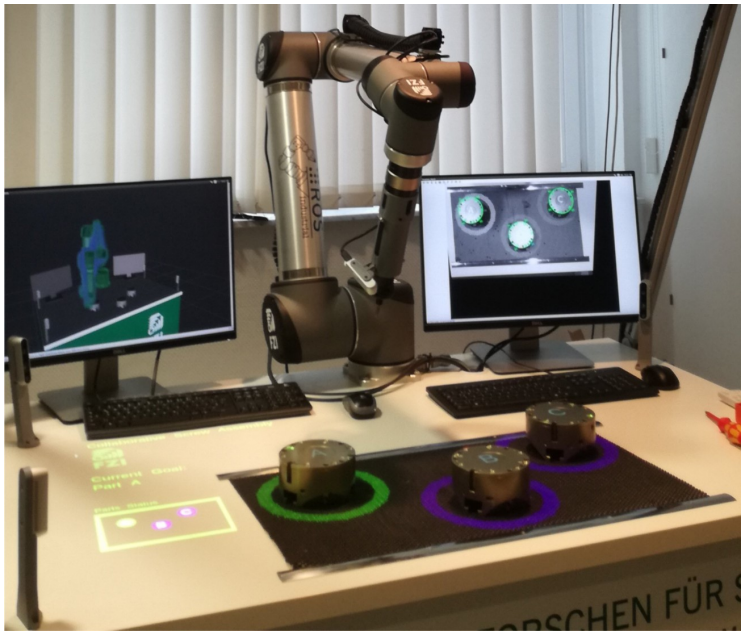


Figure 2.7: Extended reality setup using displays and projectors [44].

Ganesan et al. [45] implemented a mixed reality approach with vision capabilities, able to identify the poses and positions of items in the workspace, as well as a projection mapping technology for both teaching the human operator by showing instructions directly on top of the working object and informing and warning about the robot's intent and safety precautions. They experimented with three modes of communication: (1) printed, in which the task instructions were given to the human operator in written form and pasted to an adjacent wall; (2) mobile display, in which the task instructions were given in a tablet device containing written text, images, and videos; and (3) projection, in which the instructions were projected around the workspace in a just-in-time augmented reality manner. The results showed that the projection approach resulted in fewer error and less time spent on the task execution, and a higher sense of sastisfaction while performing the task.

Tsamis et al. [46] utilized an augmented reality approach for communication with a mobile robot arm. Through an Augmented Reality Head-Mounted Display (ARHMD), the user can communicate with the robot with pointing gestures, and the robot can share its intended motion using spheres along the path trajectory and by demonstrating a preview of the robot's movement using a 3D model of the robot arm. The user is also informed about the robot's working perimeter, with a warning presented when the human operator enters it. They compared their system to a baseline in which the user was provided the identical features from the ARHMD via interaction with a tablet device. They concluded that their approach made task completion feel more intuitive, safer, and faster (averaging less idle time) than the baseline approach. However, the configuration was found to have poor ergonomics due to its weight and narrow field of view.

Walker et al. [47] created a framework composed of a HoloLens ARHMD to communicate drone motion intent in a HRI. In this implementation, features such as arrows to indicate next positions and timers or circles to indicate how long the drone would be in a specific position were used. They came to the conclusion that this implementation increased overall work efficiency because the human counterpart could better predict the robot's movement and thus plan their actions accordingly, reducing the amount of time spent unproductively. However, the ARHMD used in this experiment was also deemed uncomfortable and had a narrow field of view by the participants, making it a poor choice for long HRI scenarios, similarly to [46]. They concluded that by using the projection mode, the human operator boosted task completion efficiency while also providing the human operator with a stronger sense of satisfaction while working.

## 2.5 Summary

Verbal communication seems to be more appropriate and produces better results in scenarios classified as low complexity, and in most scenarios, it can transmit a message to the receiver that correctly conveys its intent without training that person. On the other hand, this mode of communication is easily susceptible to external factors such as ambient noise (common in factory settings) and language barriers. Furthermore, the sentence generation aspect of this communication modality, such as message length and intonation, can influence factors such as message recall accuracy and level of trust. Moreover, apart from the robot, this method also needs additional hardware to be implemented.

While nonverbal communication is not affected by external factors such as ambient noise, making it an ideal modality for use in factory settings, it does appear to have more aspects to account that affect its conveying abilities. First is the robot's morphology, which can be influenced by factors such as the type of end-effector (two-fingered gripper, vacuum, etc.), degrees of freedom and joint types, and level of anthropomorphism, all of which can influence what type of cues can be used. The second factor concerns the receiver's perspective, as some cues, such as referential/pointing cues, can be difficult to depict correctly from certain angles. The third aspect is the robot's movement execution, as the path and trajectory of those movements, as well as the velocity with which they are executed, can have an impact on the human counterpart's well-being in terms of stress, trust, and safety concerns. Compared to verbal communication studies, nonverbal communication studies haven't focused much on the use and analysis of symbolic gestures on practical application, except for the ones performed by Gleeson et al. [26] and Sheikholeslami, Moon, and Croft [27].

Extended reality approaches enable the use of verbal and nonverbal communication in ways that humans cannot replicate on their own. Because of the variety of information available to the human operator, they promote a safer and more informed environment. However, using screens, projectors, and/or ARHMD makes this approach significantly more expensive. In the case of ARHMD, apart from the reports of poor ergonomics and

narrow field of view, the availability of information is limited to those wearing the headset, making the cost of this approach proportional to the number of human operators.

## 2.6 Supporting Concepts

This dissertation made use of open-source tools, libraries, and frameworks built to aid the implementation of robot applications. This section briefly describes each of them.

### 2.6.1 Robot Operating System (ROS)

ROS [48] is an open-source robotics middleware and meta-operating system that has been in use for over ten years and has a large and active community of developers who contribute to its continuous improvement and expansion. It provides a variety of services, such as hardware abstraction, low-level device control, the implementation of commonly used functionalities, the exchange of messages between processes, and package administration.

A ROS-based system is comprised of a number of processes known as **nodes** that are linked in a graph-style peer-to-peer network. These links could be either **topics** or **services**. Topics allow asynchronous publication and subscription of standardized and custom messages containing raw hardware sensory data or processed data from another node. Services allow for a synchronous RPC-style communication.

Furthermore, within the ROS ecosystem, there are available several of powerful tools that allow for a easier management and visualization of all the data that is being shared between all the different nodes. Two examples used are RViz (ROS Visualization) and rosbag. RViz plays a critical role in this dissertation as it facilitates the visualization and debugging of robotic systems in a 3D environment. It enables features such as visualizing robot models, current states and planned trajectories, displaying sensor information, and representing the environment's state. This makes it an invaluable tool for comprehending how all components of the system interact. Rosbag provides the ability to record, play, and manage all content published on topics related to a given simulation session. A bag file contains all of the messages from the previously selected topics, each with its own global timestamp, allowing for later analysis of the data using Python, for example, or visualization using tools like RViz.

Throughout the years, ROS have received several major updates that are divided and identified by its distributions. The most recent and significant one was the release of the ROS2 [49], which was totally revamped to meet the demands of modern robotic issues. These updates in some cases could present compatibility issues with packages developed for older distros. Due to this, the distribution chosen for the implementation of the proposed approach was ROS Noetic Ninjemys, released on May 23, 2020 and with an End of Life (EOL) date of May 20, 2025.

### 2.6.2 MoveIt

MoveIt [50] is an open-source, robot-agnostic motion planning framework for ROS. It's been used with over 150 different robots in several hobbyist projects as well as in professional applications at companies such as NASA, Google, Microsoft, Fetch Robotics, Franka Emika, PAL Robotics, Kinova, and Samsung (Figure 2.8).


(a) Fetch Robotics's Fetch

(b) PAL Robotics' ARI

(c) Kinova's Jaco2

(d) NASA's Robonaut

(e) Franka Emika's Panda

Figure 2.8: Example of robots using MoveIt [51].

This framework offers, through the use of its C++ API, a number of high and low-level capabilities that aid in the development of robotic arm task applications, such as motion planning, manipulation, inverse kinematic solvers, 3D perception, collision checking, and integration with hardware controllers. These capabilities are implemented behind a high level abstract layer where it's possible to plan and execute, in both simulation and hardware, collision-free trajectories in cluttered environments given simple information, such as the final position of the end effector or the target joint values. MoveIt also allows for the modeling of the environment in which the robot finds itself via collision objects, which can be represented by primitives and meshes provided by the user, as well as using depth sensors and point clouds. This enables the previously mentioned collision-free motion planning as well as the ability to accurately manipulate objects.

### 2.6.3 ROS for Human-Robot Interaction (ROS4HRI)

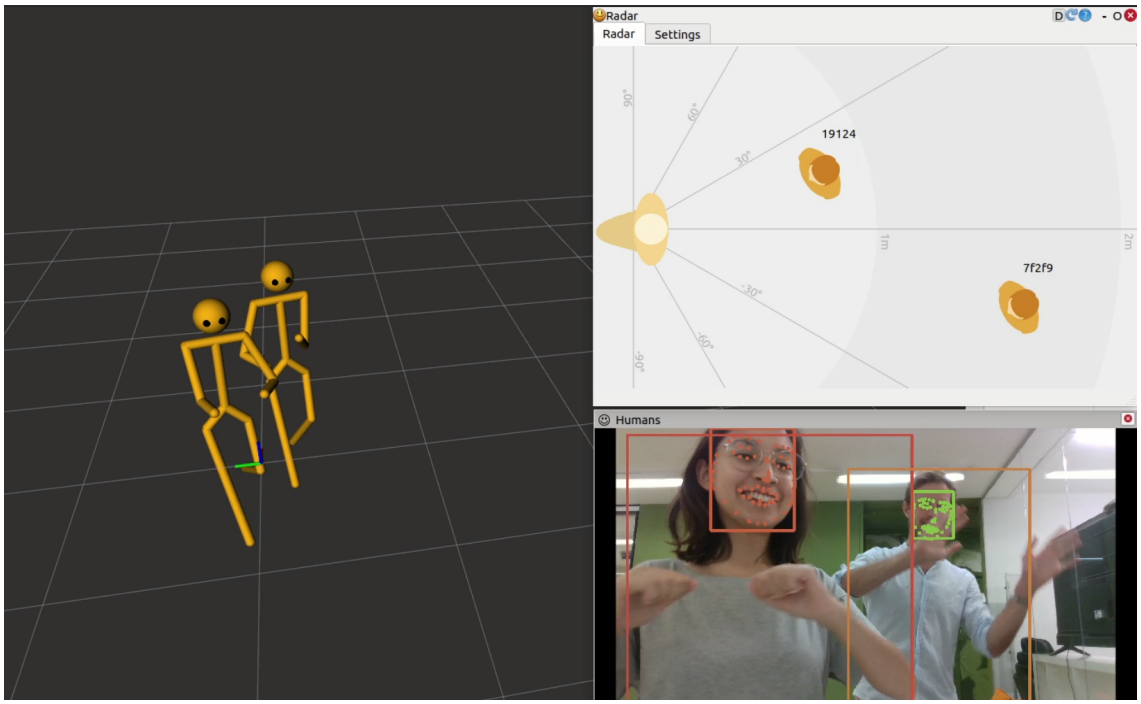



Figure 2.9: Visualisation of outputs generated by components of the ROS4HRI framework [52]. In this are depicted (left) two 3D skeletons on an RViz environment representing both their position in space as well as the body joints positions, (top right) a radar interface representing the position and identification of the humans in the environment relative to the robot, and (bottom right) a camera feed containing face and body bounding boxes, and face landmarks of both humans.

ROS for Human-Robot Interaction (ROS4HRI) [52] is a framework that encompasses all ROS packages, tools, and conventions that support the development of interactive robots within the ROS ecosystem. In this framework are provided ready to use capabilities such as 2D and 3D skeleton tracking, speech recognition, and probabilistic fusion of faces, bodies, and voices, which allow for a full detection and identification of one or more individuals participating in the interaction (Figure 2.9).

# 3

# Proposed Approach and Implementation

This chapter starts with section 3.1 by presenting a brief conclusion based on the state of the art researched. Within the same section, a diagram illustrating the desired and proposed system workflow is presented, accompanied by a brief description of each constituent block and the software resources utilized to implement the system. In section 3.2, it's presented the main hardware resource of the implementation, which is a collaborative robotic manipulator, as well as some factors that led to its choice. Subsequently, the set of nonverbal cues chosen is presented in section 3.3, along with an explanation of the implementation logic for each cue. Section 3.4 provides an implementation approach to provide a more safe environment when working with or in the same environment as a collaborative robot. Section 3.5 depicts which manipulation tool was chosen among the available options and provides a description of each. Section 3.6 concludes the chapter by explaining why the motion planner used in the implementation was chosen.

## 3.1 System Workflow

Although the subject of human-robot interaction is not novel, this field of robotics is still in its early stages of development, and, as mentioned in the previous chapter, the primary focus of many studies has been on developing the direction of human-to-robot communication. On the other hand, the studies conducted in the direction of robot-to-human communication have shown a greater development and practical application of verbal and extended reality communication modalities and techniques when compared to nonverbal. This leaves a gap in our understanding of the latter's potential practical benefits. In light of the aforementioned and as stated in the first chapter, the purpose of this dissertation is to present the implementation of a set of nonverbal cues, and a methodology for creating them, that can be used in collaborative manipulation tasks and test the effectiveness of those cues in this type of environment. To complement the fact that comparatively to the other modalities, the implementation of nonverbal communication

on a robotic manipulator can in most cases be done without additional hardware, this approach will only use open-source resources.

To achieve this goal, the integration of nonverbal cues into manipulation tasks is proposed by following a system that comprises the characteristics and workflow of the one depicted in Figure 3.1.

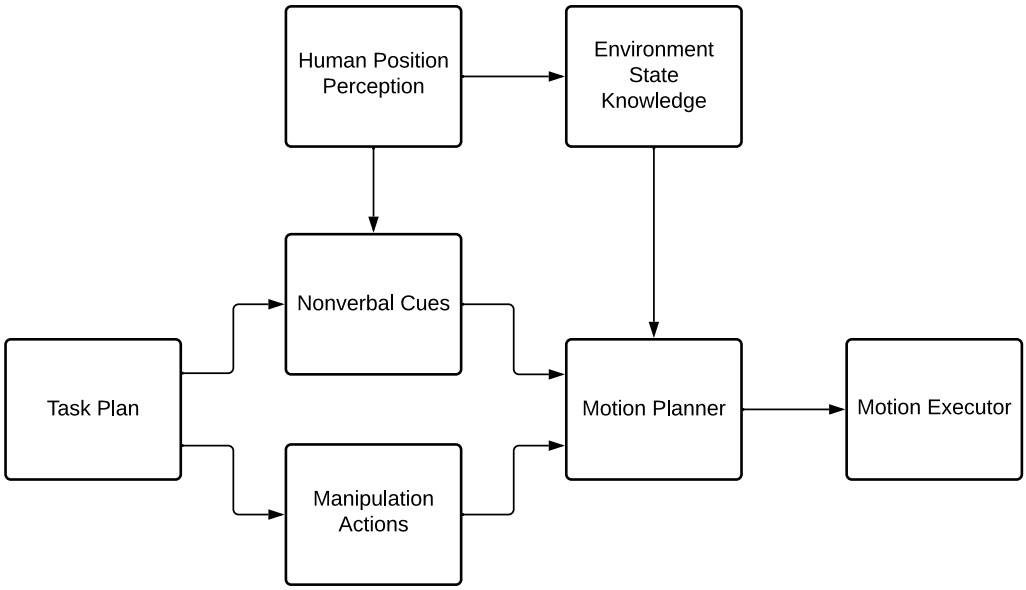



Figure 3.1: Diagram representation of the system workflow.

Following is a general description of each block:

- **Task Plan:** The "Task Plan" component of the system serves the purpose of providing the logical sequence in which sub-tasks will occur. In this case, these sub-tasks can be either nonverbal cues or manipulation actions. The task plan can be a static list of actions manually setup in order to compose a task, or it can employ more advanced approaches, such as using a planning system that dynamically generates the list of actions and their sequence based on the specific task and current environment.

- **Manipulation Actions:** The "Manipulation Actions" component describes the procedures that define manipulative gestures such as grasping and releasing, rotating and tilting, and picking and placing objects or tools, thus enabling the physical interaction between the robotic manipulator and the environment.

- **Nonverbal Cues:** The "Nonverbal Cues" component is the main focus of the dissertation. This component defines the steps that describe nonverbal cues useful in the context of manipulation tasks. There are several classifications for gestures and cues that could be implemented, as shown in Table 2.1, which depend on how they are used and their objective. Because the goal of these cues is to facilitate and enable the execution of manipulation tasks in certain scenarios, referential/pointing and symbolic gestures are the ones considered to be described in this component.

- **Human Position Perception:** The "Human Position Perception" component is responsible for detecting and providing the general position of the human in the working environment. Understanding the position of the human interacting with the robotic manipulator is crucial in this type of application. Not only does it enable the capacity for the robotic manipulator to avoid collisions during communication or manipulation movements, contributing to a safer and more trusting work environment, but it is also important to note that some nonverbal cues may require this positional information to be executed.

- **Environment State Knowledge:** The "Environment State Knowledge" component is responsible for maintaining an updated representation of the environment state. This could include aspects such as the current pose of the robotic manipulator, the pose and geometry of both manipulation objects, those being objects that the robotic manipulator can interact with, and collision objects, which are objects that it must avoid, such as walls, tabletops or people.

- **Motion Planner:** The "Motion Planner" component is responsible for planning collision-free trajectories and their respective time parametrizations that allow the robotic arm to perform the instruction provided by the "Nonverbal Cues" and "Manipulation Actions" modules.

- **Motion Executor:** The "Motion Executor" component of the system is responsible for transmitting the information generated by the "Motion Planner" component to the robotic manipulator's hardware, thus enabling the execution of the intended sub-task actions by the real robot.

The proposed modules of the system were implemented using open source tools, libraries and frameworks developed around and for the Robot Operating System (ROS), briefly described on section 2.6. In this case, given the capabilities of the MoveIt framework, it inherently incorporates the necessary functionalities of the "Environment State Knowledge", "Motion Planner", and "Motion Executor" components of the system. When it comes to the ROS4HRI framework, it enables the representation and exchange of information about the humans interacting with the robot, providing the tools necessary for the implementation of the "Human Position Perception" component.

## 3.2 Robot Model

The robot employed for implementing this approach was ABB's Dual-arm YuMi®- IRB 14000 collaborative robot, depicted in Figure 3.2. This robot features two 7 degree of freedom robotic arms (Figure 3.3), each equipped with a multi-functional end effector that can include servo grippers, each with two fingers, a vacuum system, both meant for part handling and assembly, as well as an integrated infrared camera. The choice

stemmed from the fact that it is a robot that has already been configured to work with MoveIt, freeing up time for the actual implementation of the movements, and has the necessary tools implemented that allow the integration between ABB robot controllers and ROS-based systems through the use of ***abb_robot_driver*** libraries [53]. The latter point, in conjunction with the MoveIt framework, contributes to the "Motion Executor" block by allowing the robot to execute MoveIt's motion instructions. One additional consideration in selecting this robot was its human-like stance, as evidenced by the shape of the body and the position of both arms, which may help the readability and perception of the cues.

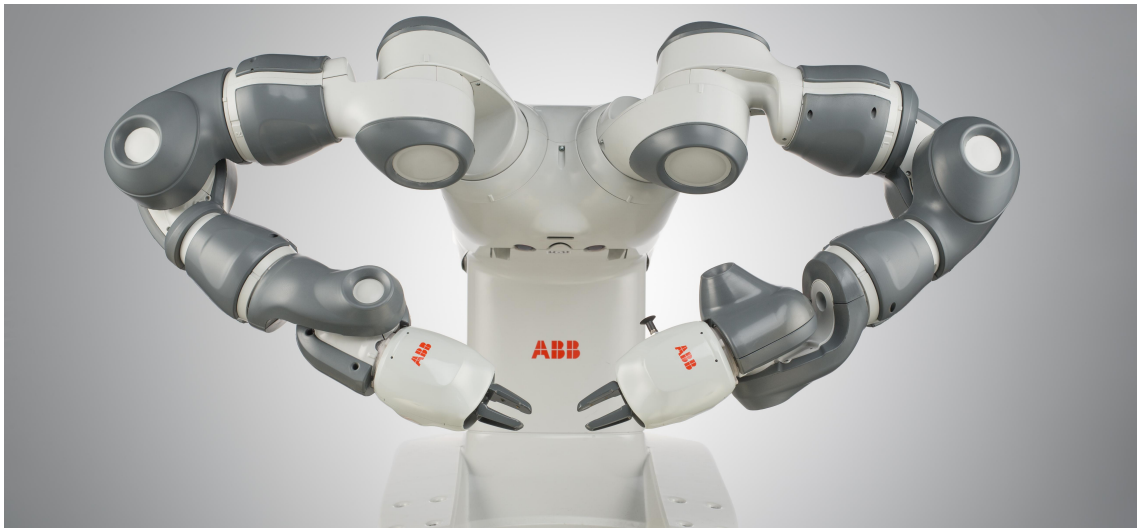



Figure 3.2: ABB's Dual-arm YuMi® - IRB 14000 collaborative robot [54].

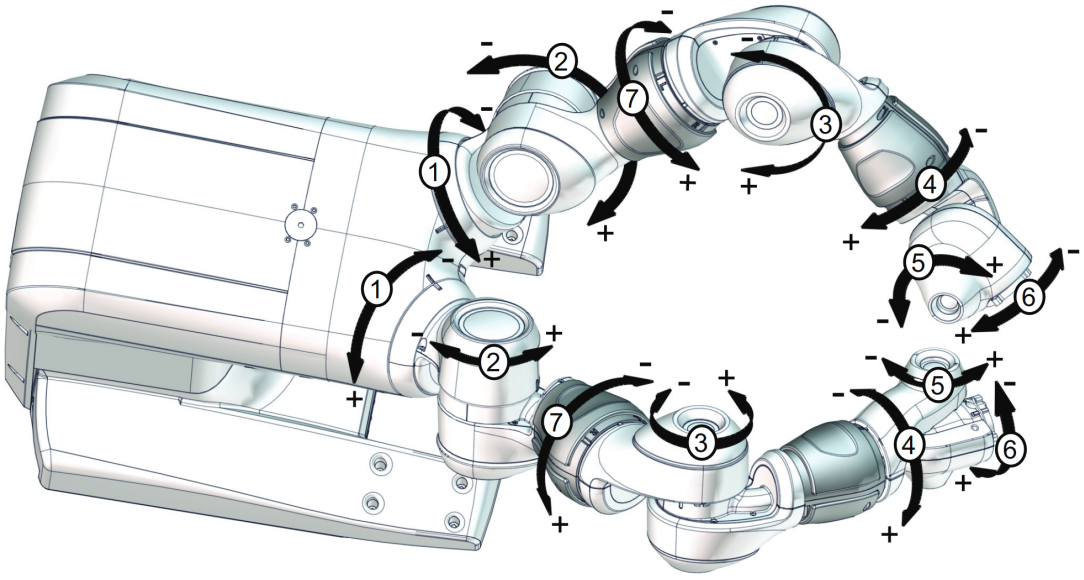



Figure 3.3: Representation of the direction of actuation for each axis of ABB's Dual-arm YuMi® - IRB 14000 collaborative robot [54].

## 3.3 Nonverbal Cues

Before starting the implementation of nonverbal cues, there is a need to first ponder on at least two aspects. The first aspect involves determining the most suitable types of cues for the specific application at hand. As referred to in section 3.1, given that the cues are intended to assist in the execution of a manipulation task, the types of movements more suitable to implement are classified as referential/pointing and symbolic.

The second factor to consider when implementing nonverbal cues is the robot's characteristics, particularly its morphology, which refers to the physical structure of the robot. Aspects such as the number of degrees of freedom, the type of joints that connect each link, and the type of end effector used by the robot may limit the number of cues that can be implemented or enable more complex approaches to implementing them.

With those aspects in consideration, some nonverbal cues can be selected. To figure out how the robot could convey the chosen cues, the simplest process would be to consider how a human would convey that cue and identify the several discrete logical steps that make up the motion. Following some possible adaptations due to limitations in the robot's morphology, those steps can be used to implement the cues on the robot's side. For that, a motion planning framework can be used, given that it provides several tools that facilitate the implementation of diverse applications with robotic manipulators, as described in section 2.6.2. The process of selecting, designing, and implementing the cues is depicted in Figure 3.4.

Taking into account the selected type of movements, the characteristics of the robot model detailed in section 3.2, the intended task that the robot will perform, which will be explained in section 4, and the allotted time for this study's development, the cues listed in Table 3.1 were selected.

Table 3.1: Selection of cues implemented.

| **Referential/Pointing** | **Symbolic** |
|---|---|
| Point To Location | Screw |
| Point To Object | Unscrew |
| Point To Object Side | Pick Object |
| Point To Human | Rotate Object |

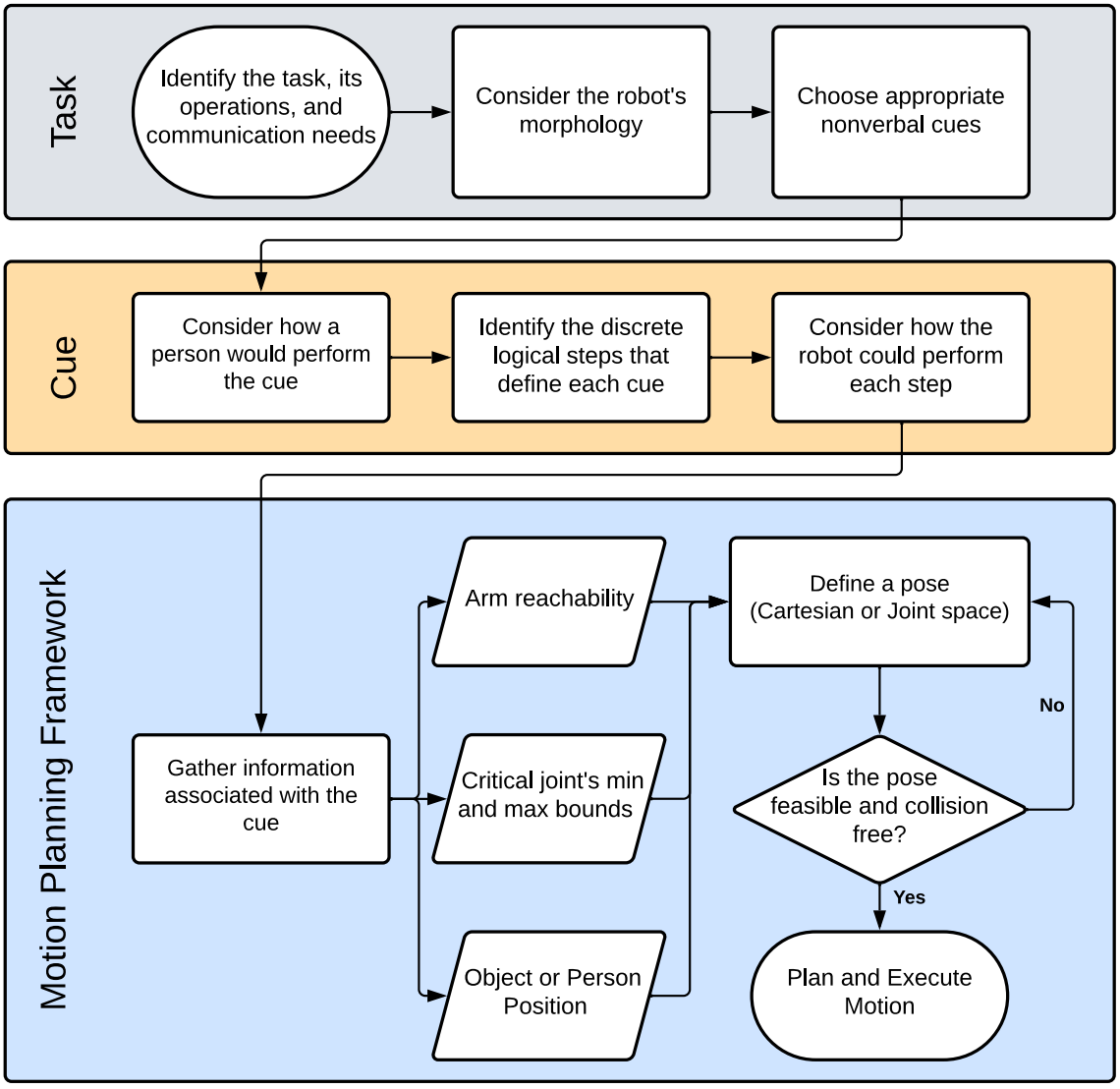



Figure 3.4: Flowchart of the cues' selection, design, and implemention process.

### 3.3.1 Referential/Pointing

Pointing or referential cues are a type of cue essential to implement given that they provide the means to direct the attention of the receiver to a specific location, object, or other target, making them an excellent context tool in communication scenarios.

Considering that the MoveIt framework can handle the motion planning aspect required for pointing movements, the implementation of this type of gesture involves identifying a suitable target pose for the robotic arm to effectively convey the cue. With the only difference being the reference target, which can be a person, an object, or a location, all the selected referential gestures follow the same logic. As a result, with the exception of a specific case described at the end of this section, all pointing gestures were implemented in accordance with the proposed flow:

1. Create a vector from arm position to reference target.

2. Define desired distance to reference target.
3. Define the desired target position for the end effector inline with the created vector and according with the desired distance.
4. Define the desired target orientation for the end effector equal to the vector orientation.
5. Compute possible joint values.
6. Plan the motion to achieve joint values.
7. Execute the planned motion.

The process begins by defining a vector (Figure 3.5), as given by equation 3.1, such that the vector extends from the Cartesian position of the arm's first joint ($P_{FJ}$), which is the shoulder in this case, to the position of the reference target's center ($P_{RT}$). This vector offers both a distance value that can be utilized as a component to define the target's pose position and an orientation that points toward the reference target.

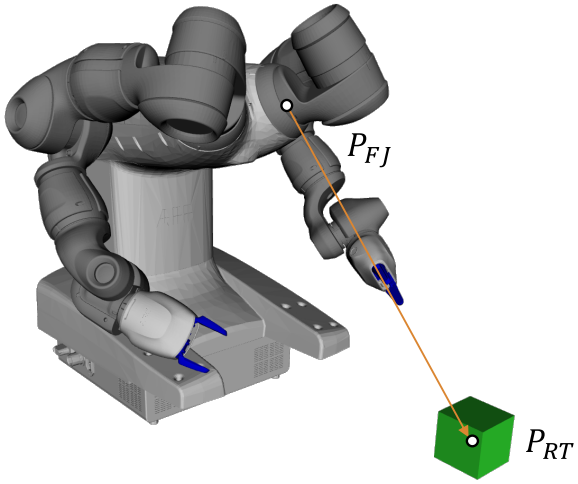



Figure 3.5: Example of the vector that connects the robot's first joint position ($P_{FJ}$) to the center of the reference target ($P_{RT}$).

$$\vec{v}_{P_{FJ} \to P_{RT}} = (P_{RT_x} - P_{FJ_x}, P_{RT_y} - P_{FJ_y}, P_{RT_z} - P_{FJ_z}) \tag{3.1}$$

The target's pose position is determined along the line established by the previously defined vector. Its specific location along this line is influenced by two factors. The first factor has to do with reachability. Since the arm has a max range of operation the target's pose position must be defined within that range, otherwise it will never be an achievable pose. The second factor has to do with target collision. Considering that the vector extends to the center of the reference target, it is crucial to prevent any overlap between the target's

pose position and the geometry of the reference target, since that would result in a collision with the arm. To accomplish this, the position accounts for a safety radius value. This value is chosen so that a hypothetical sphere with that radius could entirely encapsulate the target's geometry plus the length of the end effector. When given the option, the distance to the reference target is chosen in such a way that the target's pose position is as close to it as possible to minimize doubt about the intended target. Following that logic, the distance to the reference target is given by equation 3.2.

$$\text{Distance to Reference Target} = \begin{cases} \text{Max Range} & \text{if } (\text{distance} - \text{radius} \geq \text{Max Range}) \\ \text{distance} - \text{radius} & \text{otherwise} \end{cases} \tag{3.2}$$

The previously defined vector is then scaled to match the chosen distance in magnitude. Subsequently, the target's pose position is determined by then translating the position of the arm's first joint using the scaled vector, as shown in equation 3.3, resulting in a positioning similar to the one depicted in Figure 3.6.

$$P_{TP} = P_{FJ} + \vec{v}_{P_{FJ} \rightarrow P_{RT\,scaled}} \tag{3.3}$$

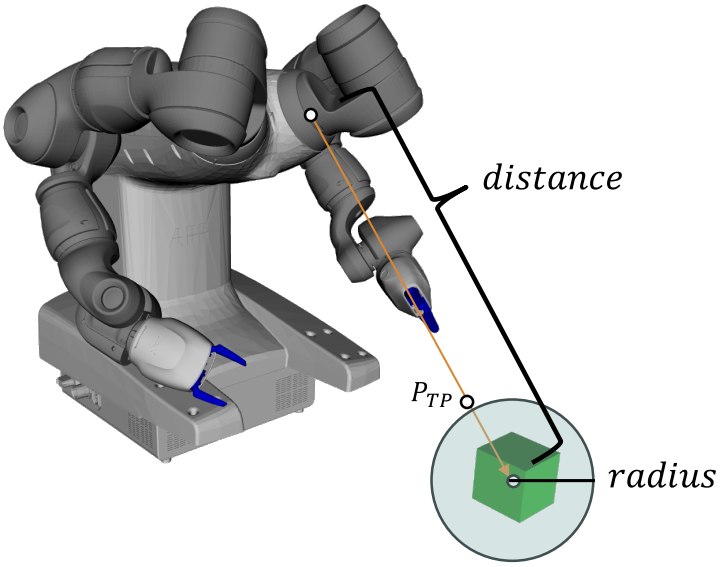



Figure 3.6: Example of the target's pose position ($P_{TP}$) along the vector considering the distance to the target and the safety radius.

The orientation of the target pose is set so that the z-axis of the end effector points in the direction of the reference target. To achieve this, the orientation of the defined vector can be used by applying to a unit quaternion, meaning a quaternion in which the values of x, y, and z are zero and w is one, a rotation around the z-axis (yaw), followed by a rotation

around the y-axis (pitch). The angles of rotation for the yaw and pitch are derived from the equations 3.4 and 3.5, respectively, considering the information depicted in Figure 3.7.

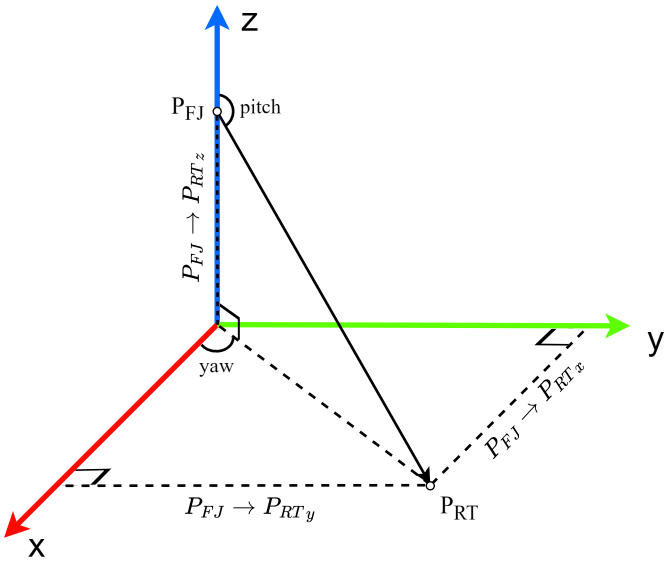



Figure 3.7: Graphical representation of the information used to derive the angle values for yaw and pitch of the target's pose orientation.

$$\text{yaw} = \arctan\left(\frac{\|\vec{v}_{P_{FJ}\rightarrow P_{RT_y}}\|}{\|\vec{v}_{P_{FJ}\rightarrow P_{RT_x}}\|}\right) \tag{3.4}$$

$$\text{pitch} = \frac{\pi}{2} - \arctan\left(\frac{\|\vec{v}_{P_{FJ}\rightarrow P_{RT_z}}\|}{\sqrt{\|\vec{v}_{P_{FJ}\rightarrow P_{RT_x}}\|^2 + \|\vec{v}_{P_{FJ}\rightarrow P_{RT_y}}\|^2}}\right) \tag{3.5}$$

After defining the target pose (Figure 3.8), MoveIt's inverse kinematic services are employed to generate the joint values for the arm to reach that pose. Nevertheless, the attainability of the pose is not guaranteed, as it might lead to self-collision or collision with the environment, which can be verified through MoveIt's framework.

In instances where such collisions occur, an approach was employed to expand the range of potential poses, thereby reducing the likelihood of failing to find a suitable pose. This approach involves generating a set of points that make up a section of a sphere's surface centered on the reference target, surrounding the initial target pose point. Initially, a first layer of points is created, considering a distance to the reference target equal to that of the original target pose. Subsequently, additional layers of points are generated, each considering a greater distance from the target, resulting in a set of points in a structure similar to the one depicted in Figure 3.9. These points are then sorted based on their deviation from the original target pose and subjected to testing using the same inverse kinematics method employed previously.

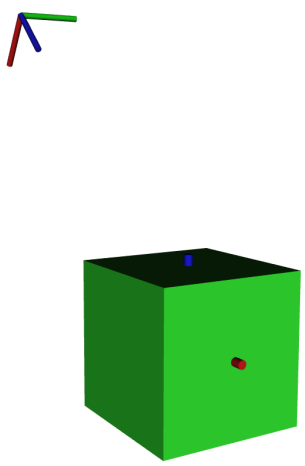


Figure 3.8: Example of an initial target pose calculated during the execution of the Point to Object cue on a green cube. The target pose is represented by a coordinate axis marker in which the x-axis, y-axis, and z-axis are red, green and blue, respectively. The z-axis corresponds to the direction and orientation of the robotic manipulator's end-effector.

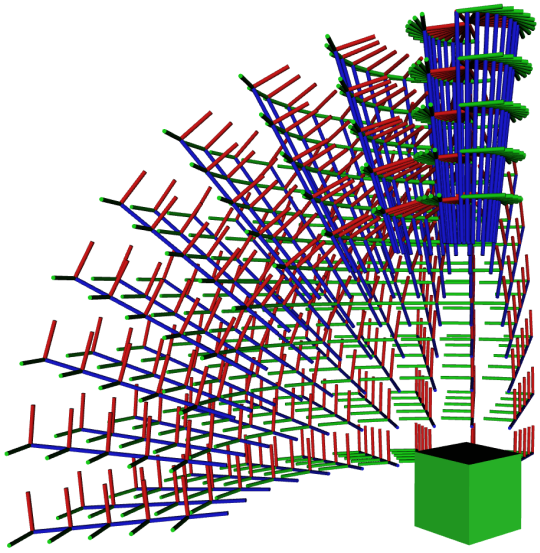


Figure 3.9: Example of five sphere sections made up of possible additional target poses to use during the Point to Object cue execution on a green cube. The target pose is represented by a coordinate axis marker in which the x-axis, y-axis, and z-axis are red, green and blue, respectively. The z-axis corresponds to the direction and orientation of the robotic manipulator's end-effector.

If a valid pose is found, whether it's the original pose or one generated from a subsequent layer, the motion of the arm is planned and executed using the MoveIt framework. However, if none of these poses prove to be valid, conveying or achieving the cue is deemed impossible, and the motion planning is aborted.

This flow is the same for all the selected referential cues, except for the "Point To Object Side" cue, where steps 1 through 4 are carried out differently. In this cue, the objective is

to effectively point to one of the sides of an object. Using the object's reference frame, it was determined that the cue can target one of six sides of the object (Figure 3.10): front (x positive), back (x negative), right (y positive), left (y negative), top (z positive), and bottom (z negative).

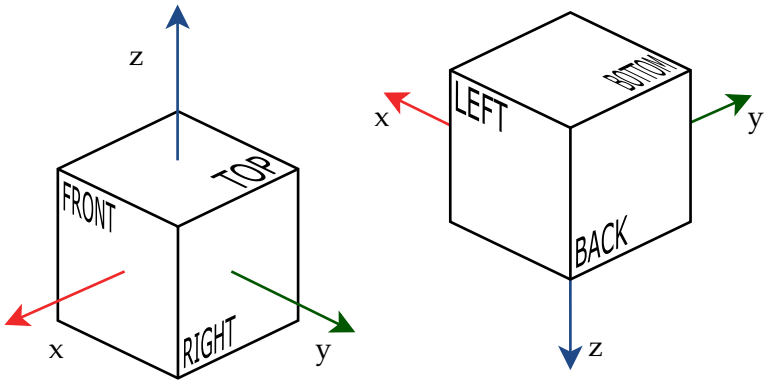



Figure 3.10: Diagram of the possible targetable object's sides.

Instead of defining the target pose through a vector identical to the one described previously, the approach involves translating the position of the object in the direction of the intended side by a distance that complies to the previously mentioned safety radius. Following this translation, an orientation is defined to ensure that the end effector points towards the object's side, thereby completely defining the target pose. Subsequently, the same process as previously described is employed to find a valid pose.

### 3.3.2 Symbolic

Within the context of collaborative manipulation tasks, the implementation of symbolic cues such as the ones selected appears crucial. Despite the impressive capabilities of contemporary robots in terms of manipulation, they still possess inherent limitations. The robotic manipulator's ability to manipulate an object can become impractical or impossible due to factors such as the object's intrinsic characteristics, including its weight or size, as well as reachability constraints.

When comparing the cues of each type, all referential gestures follow a similar pattern, with the only difference being the reference target to which they are being applied, which can be a person, an object, or a location. Symbolic gestures, on the other hand, encompass all gestures used to convey an action and require individual implementation flows, which will be presented in the following sections.

#### 3.3.2.1 Pick Object

The effectiveness of this cue heavily relies on the type of end effector the robotic manipulator employs. While a robotic manipulator can execute various manipulation actions using a vacuum gripper, conveying those actions to a human using it would pose significant

challenges. Given the way humans execute and perceive the picking and grasping action, implementing this cue on a robotic manipulator equipped with at least a two-finger gripper seems to be the most practical approach.

In this specific implementation scenario, the robotic manipulator is equipped with a gripper that features two linearly actuated fingers. The gripper is capable of achieving any aperture between zero and 50 mm (25 mm per finger) within its precision capabilities. However, for this particular implementation, only two states were considered: open (Figure 3.11a) and closed (Figure 3.11b). The cue consists of alternating between the two states, simulating a pinching motion, until the gripper completes two closing actions. Therefore, the cue's sequence can be described as (open) → close → open → close, with the initial movement being to open when the gripper's initial state is closed. When using the MoveIt framework to plan the motions, there is the option to either manually specify the joint value targets of zero and 25 mm for each motion or define an open and closed state for the gripper beforehand, which is usually done for poses that are frequently used.

(a) Opened (b) Closed

Figure 3.11: Gripper states in Pick Object cue.

#### 3.3.2.2 Rotate Object

The primary reason for the selection of this cue is to provide the robotic manipulator with the capability to communicate the desired orientation of an object to a human operator. Given the initial pose of an object, the robot can indicate a sequence of rotation actions that lead to the desired orientation. This could be useful in situations where the robot has a preferred object's picking side or when the human has incorrectly placed the object, for example.

In this implementation, the cue is intended to convey only one simple rotation action. This approach still allows complex rotation actions to be conveyed by combining multiple discrete rotation cues, while also providing the freedom to execute other actions in between them.

It is assumed that an object has three rotational degrees of freedom. Therefore, as depicted in Figure 3.12, it can be rotated in six distinct ways, which include rotations around the x-axis (roll), y-axis (pitch), and z-axis (yaw) in both the positive and negative directions, following the right-hand rule.

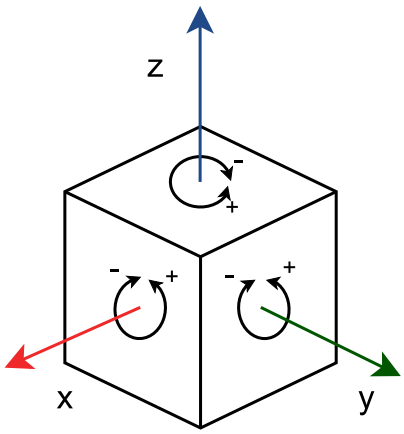



Figure 3.12: Diagram of the rotation axis and direction of an object.

The initial step in the cue's execution is to align the end effector with the axis around which the desired rotation is to be conveyed. This alignment can be achieved through two available options: the end effector can either face the side of the object located in the positive direction of the axis or the side of the object situated in the negative direction of the axis, as depicted in Figure 3.13. Given the choice between the two options, the approach that has the end effector closest to the arm's first joint is tested first for attainability, following the same process as the "Point To Object Side" cue. If the closest approach proves to be unattainable, the second and final approach is tested. In the case that one of the approaches is valid, then is it is possible to convey around which axis the rotation is intended. In the event that both approaches fail, it is considered impossible to convey or achieve the cue, and the motion planning is terminated.

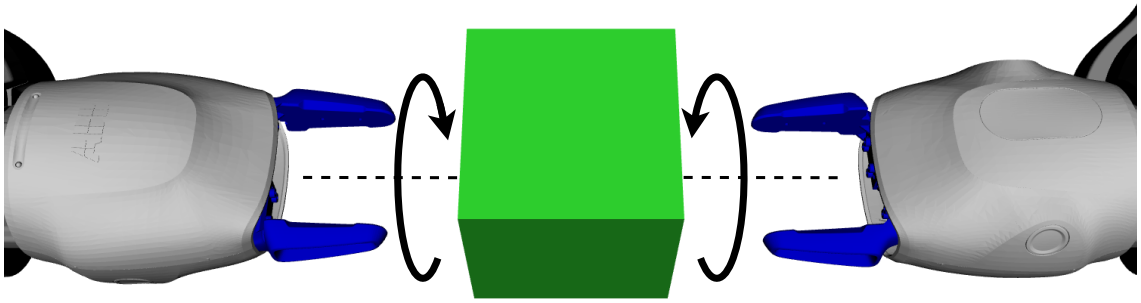


Figure 3.13: Approach options for the Rotate Object cue.

After completing the first step, there is still a need to convey the direction of rotation. To achieve this, the next step in this cue involves rotating the end effector of the robotic arm in the intended direction. However, as shown in Figure 3.13, the actual direction of rotation of the end effector depends on which approach side it's positioned on. The actual direction of rotation can be determined through the following steps:

1. Determine the vector that represents the direction of rotation of the object.

2. Determine the vector that represents the pointing direction of the end effector.

3. Calculate the dot product between the two vectors.

4. If the dot product results in a number greater than zero, both vectors are pointing in the same direction, and therefore the end effector should rotate around its z-axis in the positive direction. If the dot product results in a number less than zero, both vectors are pointing in opposite directions, and therefore the end effector should rotate around its z-axis in the negative direction.

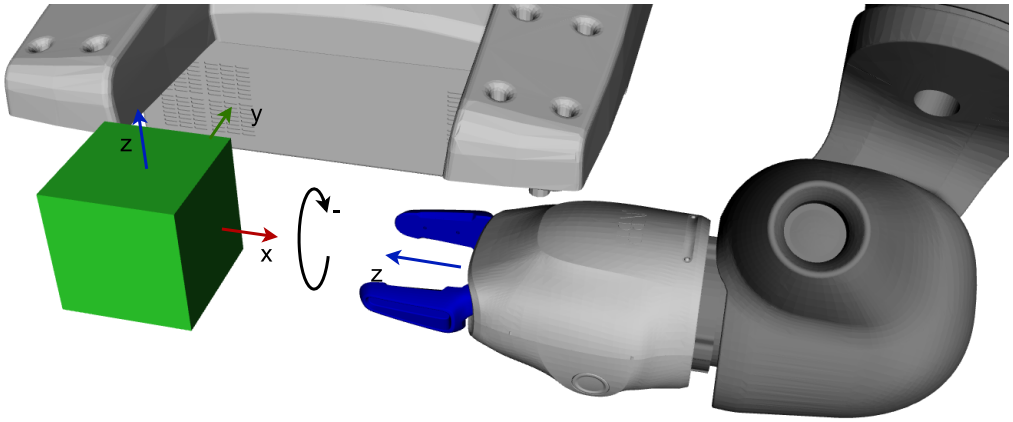



Figure 3.14: Example scenario of Rotate Object cue execution.

To clarify the approach, let's consider the scenario depicted in Figure 3.14. In this example, the objective is to convey the rotation of an object around its x-axis in the negative direction. With the first step of the cue, the end effector is positioned to face the side that points in the positive direction of the axis. Upon examining the image, it becomes evident that the end effector must perform a rotation around its z-axis in the positive direction. That inference can be confirmed by performing the steps previously described, which can be depicted in Figure 3.15.

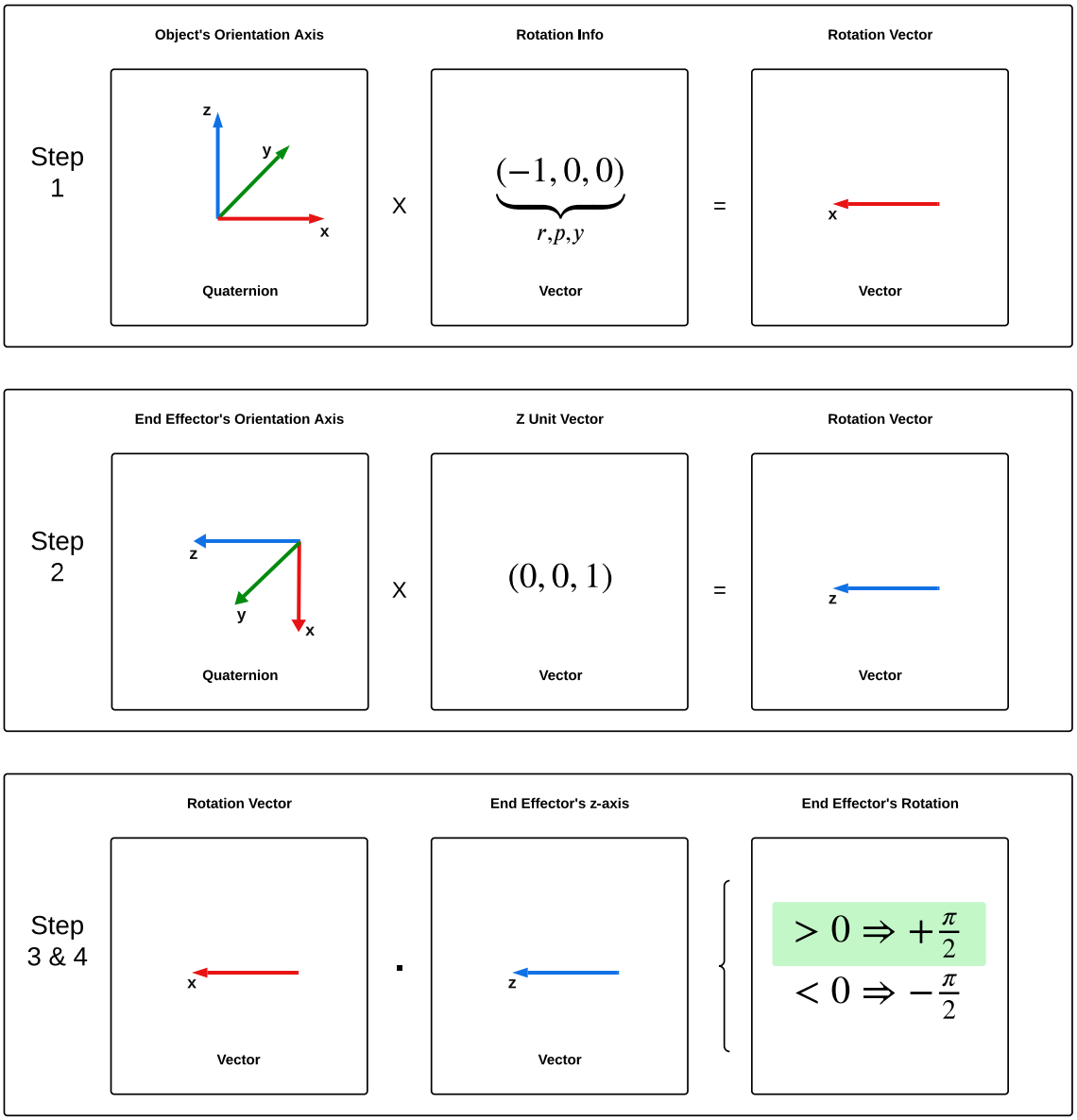



Figure 3.15: Diagram of the steps taken to determine the rotation direction of the wrist of the robotic manipulator during the execution of the Rotate Object cue in an example scenario.

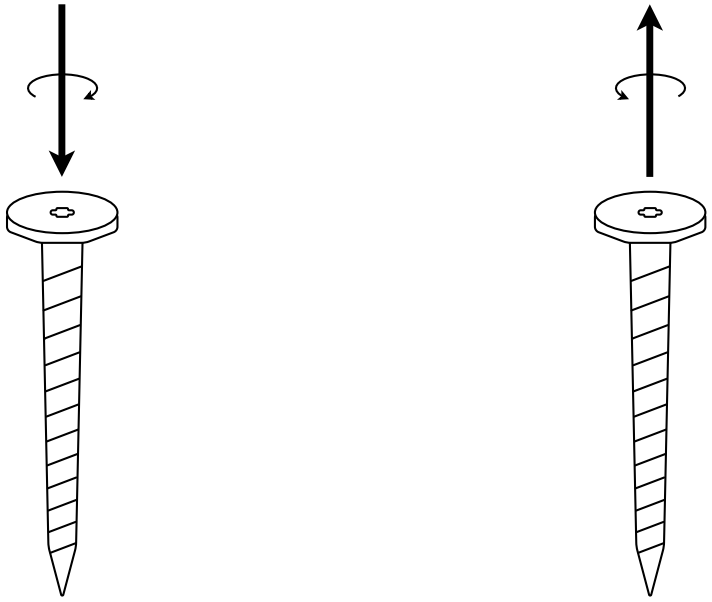
After determining the direction of rotation for the end effector to convey the cue, the end effector is rotated by $\pi/2$ radians in that direction and then by $\pi/2$ radians in the opposite direction, thus completing the execution of the cue.

#### 3.3.2.3 Screw and Unscrew

The screw and unscrew cue was chosen with the intention of allowing the robotic manipulator to indicate when something needs to be pushed or pulled, or when it needs to become loose or tightened.

For this cue, the design was based on the motion of a screw being screwed (Figure 3.16a) and unscrewed (Figure 3.16b). Both motions are similar and can be described as

a linear trajectory along the screw's axis while simultaneously rotating around that axis. The linear trajectory for the screwing motion is directed toward the screw, whereas the linear trajectory for the unscrewing motion is directed away from the screw. The direction of rotation in both movements is given by the right-hand rule along the direction of the linear trajectory.



(a) Screw (b) Unscrew

Figure 3.16: Motion concept for screw and unscrew cues.

Considering the previous description of the motions, the implementation of this cue can be divided into the following steps:

1. Define a pose that is close to the target, aligns with its axis, and points toward it.
2. Define a travel distance.
3. Plan and execute a linear trajectory between the two points while simultaneously performing a rotation motion around the axis.

To define the close pose, the same method as in the "Point To Object Side" cue was employed. In this approach, the end effector's pose, if feasible, is positioned close to the target and oriented toward the side of the object where the cue should be executed.

The travel distance is the desired distance between the starting pose and the final pose of the trajectory. This should be sufficient to create a noticeable linear motion between the two points.

Unlike the other cues developed, in which it was only needed a way to plan a feasible trajectory between determined poses, in this cue, the planned trajectory is required to be

constrained to ensure that the intended linear and rotation components of the movement are met. As a result, for the planning of this cue's motion, the MoveIt Cartesian Interpolator was used. This tool enables the planning of Cartesian paths, allowing the end effector to move between poses following a linear trajectory. It requires the following data to be used: a list of waypoints, step resolution, and jump threshold.

The list of waypoints is needed to specify both the poses the end effector needs to reach and the order in which they should be achieved. Due to the rotational aspect associated with the motion, ensuring a smooth and directional rotation throughout the translation between the start and end points is crucial. Therefore, the following approach was used to set up the waypoints for conveying the cue.

The main consideration was to make the most of the available range of rotation for the end effector, which for the chosen robotic manipulator stands at a total of 460 degrees. When applying rotation from one waypoint to the next, during path planning, the MoveIt Cartesian Interpolator will plan for the rotation direction that minimizes the amount of rotation needed to transition from the current pose to the next. Therefore, in an application like this where the direction of rotation is important, it is recommended to keep the rotation between consecutive waypoints below 180 degrees. For this approach, a rotation of 45 degrees was chosen between each waypoint.

Considering the chosen 45 degrees of rotation to be applied between waypoints and the maximum rotation range allowed by the robotic manipulator, it is possible to have a maximum of 10 transitions between the waypoints, resulting in a total of 11 waypoints.

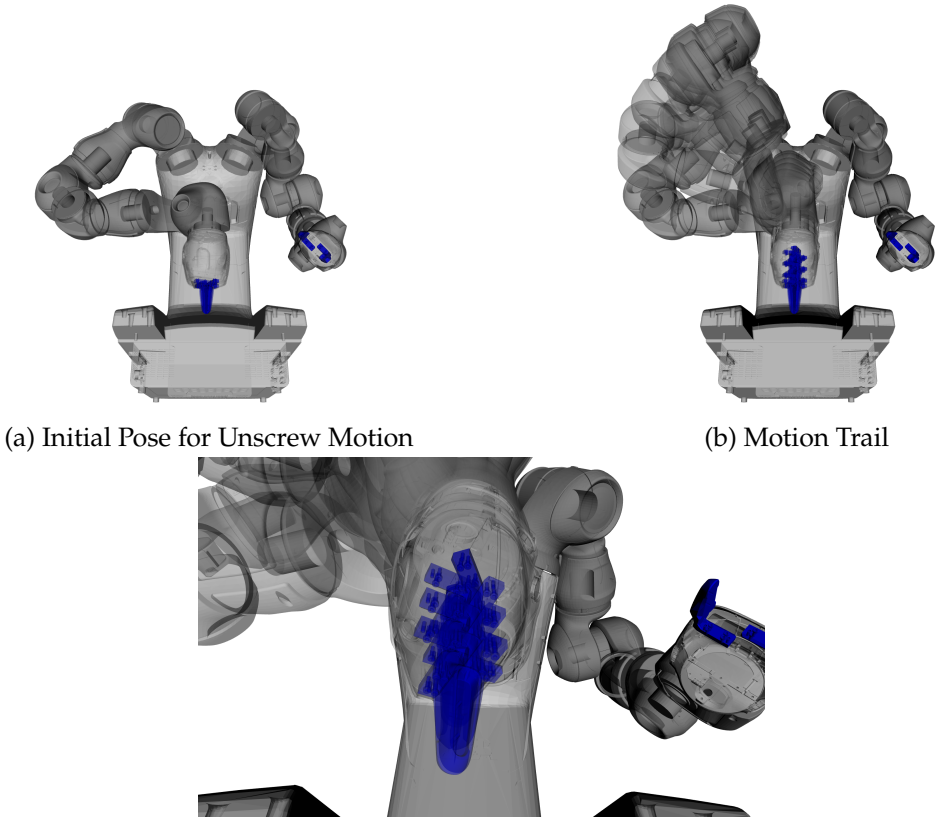
The list of waypoints is constructed following the trajectory of the unscrewing motion (Figure 3.16b). It begins with the previously defined close pose, and given that the z-axis of the pose points towards the target, for each successive waypoint it's applied a translation in the negative direction of its z-axis, calculated as the desired travel distance divided by the number of transitions, along with a rotation in the negative direction around its z-axis. This process continues until the total number of waypoints is reached. For the screwing motion (Figure 3.16a), the same method is used, but the order of the list is reversed at the end.

The step resolution is the maximum translation distance allowed in between generated waypoints. As a result, if its value is small, the generated trajectory will be denser with waypoints, and if its value is high, the generated trajectory will be more scattered with fewer waypoints.

The jump threshold is a parameter used to prevent abrupt changes in inverse kinematic solutions between consecutive waypoints. It represents the maximum cumulative change in joint values between consecutive waypoints. If set too low, the trajectory may become impossible to execute, while setting it too high may result in sudden jumps. Consequently, this parameter is dependent on the specific robotic manipulator and should be experimented with to determine the optimal value.

With the list of waypoints, step resolution, and jump threshold defined, the MoveIt Cartesian Interpolator is then emplyed to plan the motion of the cue. Due to collision with

the environment or with the itself, the interpolator might not find a feasible plan for the cue. To mitigate that possibility, but not be stuck in planning attempts, a maximum of four additional attempts is provided in the event that the initial planning proves unsuccessful. Each subsequent attempt will have a new traveling distance which equals a reduced percentage of travel distance of the previous attempt. If it is still not possible to plan the motion, under those circumstances, the cue is deemed impossible and the execution is aborted. A successful unscrewing cue might resemble the one shown in Figure 3.17.



(a) Initial Pose for Unscrew Motion

(b) Motion Trail

(c) Gripper Rotation Close-up

Figure 3.17: Example of a successful Unscrew Cue motion.

## 3.4 Human Position Perception

The "Human Position Perception" component of the system was primarily implemented using the *hri_fullbody* package from ROS4HRI. This package provides a ready to use ROS node, built on top of Google Mediapipe, that is capable of 3D pose estimation for detected humans in the environment. To achieve this, the node relies on image data, which, in this specific implementation, is provided by the Intel® RealSense™ Depth Camera D435. This camera, in addition to providing RGB images, is also capable of providing image depth information using stereoscopic sensors and an infrared projector. Although the node can perform the estimation using only the RGB image data, this depth information can then

be utilized for a more accurate pose estimation.

When activated, the node detects a total of 15 body points per human detected, as shown in Figure 3.18. These points encompass the major joints of both legs and arms, waist, torso, and head. Each human detected in the environment is assigned a unique body identifier, which is also integrated into the labeling of each body point, following the structure <bodyPoint>_<bodyId>. The current position of these points is provided in both an image coordinate system and a 3D position relative to the camera, as shown in Figure 3.19. From the both, the 3D data was the one selected to be used in two applications.

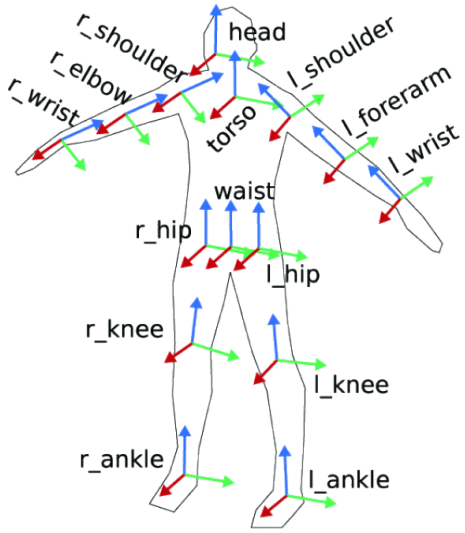



Figure 3.18: The 15 body points identified on the human body by hri_fullbody and their nomenclature [55].

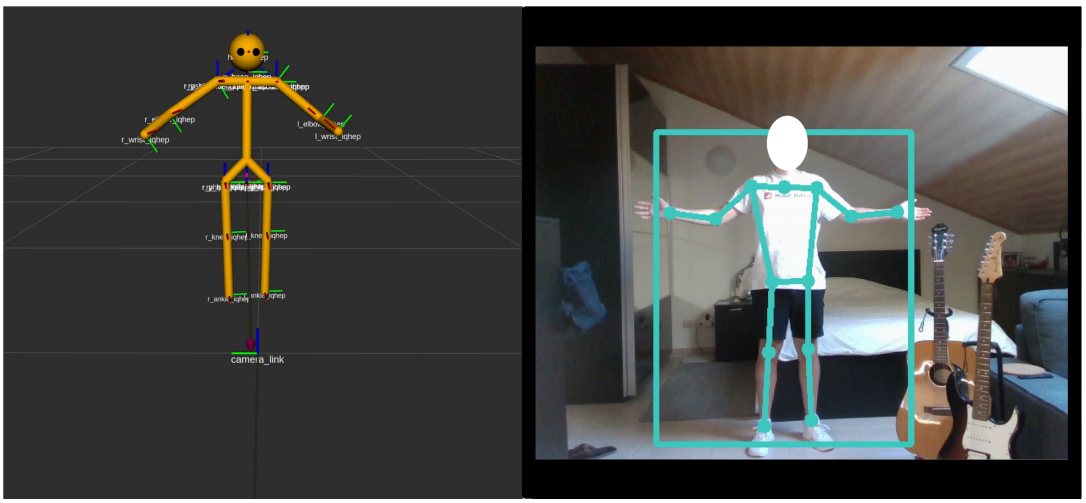


Figure 3.19: 3D (left) and 2D (right) representation of the body points detected by hri_fullbody.

### 3.4.1 Reference Target

The first application of the 3D pose data of the human was to establish a reference target for the "Point To Human" cue. Out of all the available body points, the human's head was the one chosen as the target for the cue to reference.

The 3D position of the body point is represented within the ROS framework using Transform frames from the *tf2* library. These frames enable the tracking of the body point's pose over time and facilitate the conversion of 3D coordinates between different reference frames. To obtain the position of the body point, a Transform Listener needs to be employed. To use this listener in this case, three pieces of information are required:

1. The name of the body point frame, following the structure <bodyPoint>_<bodyId>.

2. The name of the frame to which the body point frame coordinates should be relative, which depends on individual implementation and choice.

3. The timestamp of the published information, which was chosen to be the most recently available data.

To gather the name of the body point frame, it is needed to retrieve the unique identifier associated with the human in the environment. That was done using the *getBodies()* method from the *HRIListener* class defined in the *libhri* ROS package. This class subscribes to the ROS4HRI's published topics and makes the retrieval of information about the humans in the environment easier.

### 3.4.2 Human Collision Avoidance

The last application that required the use of the 3D pose data of the human was to enable a safer collaborative environment. Given MoveIt's capabilities to plan around obstacles and avoid collisions, this was done by considering the human itself as a collision object in the motion planning.

The process begins by collecting the current positions of the body points for every body tracked by the *hri_fullbody*, using the same method as described in section 3.4.1. That information is then used to consider the human as a collision object in one of two modes that can be selected during its initial execution via ROS parameters: Box Collision and Full Body Collision.

In Box Collision mode, the entire volume occupied by the human in the environment is enclosed within a box collision object. This is done by finding the limit values for the x, y, and z axes amongst all the body points of the human, which are then used for defining both the dimensions and position of the box collision object. This mode can be depicted in Figure 3.20.

(a) Small Volume (b) Large Volume

Figure 3.20: Box Collision Mode.

In Full Body Collision mode, each individual part of the human body is encapsulated by a primitive collision object. The body is divided into ten sections: the head, the torso, the upper and lower arms, and the upper and lower legs, with the head encapsulated by a sphere collision object and the rest by cylinder collision objects. This mode can be depicted in Figure 3.21.

(a) Small Volume (b) Large Volume

Figure 3.21: Full Body Collision Mode.

## 3.5 Manipulation Execution

The manipulation execution component of this dissertation is intended to enable the robotic manipulator to interact with the objects in its environment through the performance of manipulation actions such as pick and place. MoveIt offers three tools that have been developed over time to facilitate the execution of manipulation actions on objects: the Pick and Place Pipeline, MoveIt Grasps, and the MoveIt Task Constructor.

The Pick and Place Pipeline allows for the planning and execution of pick and place motions through the specification of a standardized message format designated as moveit_msgs::Grasp. This message must be composed by the following fields:

- **trajectory_msgs/JointTrajectory pre_grasp_posture:** This field specifies the position of the end effector's joints before grasping the object, for example, the joints values

to have the gripper considered open.

- **trajectory_msgs/JointTrajectory grasp_posture:** This field specifies the position of the end effector's joints that allow the grasping of the object, for example, the joints values to have the gripper considered closed.
- **geometry_msgs/PoseStamped grasp_pose:** This field specifies the position and orientation of the end effector that should be used when attempting to grasp the object.
- **moveit_msgs/GripperTranslation pre_grasp_approach:** This field specifies the axis, direction and distance from which the end effector must approach the grasp posture.
- **moveit_msgs/GripperTranslation post_grasp_retreat:** This field specifies the axis, direction and distance that the end effector must move after the grasping of the object.
- **trajectory_msgs/JointTrajectory post_place_posture:** This field specifies the position of the end effector's joints after placing the object, for example, the joints values to have the gripper considered open.
- **geometry_msgs/PoseStamped place_pose:** This field specifies the placing position and orientation of the picked object.
- **moveit_msgs/GripperTranslation pre_place_approach:** This field specifies the axis, direction and distance from which the end effector must approach the place pose.
- **moveit_msgs/GripperTranslation post_place_retreat:** This field specifies the axis, direction and distance that the end effector must move after placing the object.

The MoveIt Grasps tool serves as a grasp generator that can be used with simple primitive objects such as block or cylinders. This tool was design to be the replacement of the Pick and Place Pipeline and provides an automatic generation and selection of the best grasp pose by filtering them based on reachability and feasibility of Cartesian planning for approach, lift and retreat motions. This works by invoking three components in the following order:

1. **Grasp Generator:** This component generates a set of possible approach positions and orientations based on the object's shape geometry and end effector kinematics.
2. **Grasp Filter:** This component tests the attainability of each grasp generated by searching inverse kinematic solutions.
3. **Grasp Planner:** This component plans all Cartesian paths for approach, lift, and retreat motions.

This set of features and the degree of automation of the process makes the MoveIt Grasps tool a great choice to implement individual manipulation actions of tasks.

The MoveIt Task Constructor is, at the time of the development of this dissertation, the current recommended way to plan manipulation actions within the framework. This tool was build around the concept that complex motion planning problems can divided into Tasks and those into sub-problems designated as Stages. Depending on the order and hierarchy of the stage, one can be considered one of these types:

- **Generators:** This stage generates a result that doesn't depend on adjacent stages and then propagates that result to both the previous and next neighbors.
- **Propagators:** This stage receives the result from an adjacent stage, performs some operation to solve a subproblem, and then propagates the result to the remaining neighbor.
- **Connectors:** This stage generates solutions that attempt to connect the resulting states of neighbor states.

This implementation enables the construction of complex behaviors using only a set of discrete stages.

Although all three tools allow for the implementation of manipulation actions for the task design that will be talked about in section 4, the Pick and Place Pipeline was chosen. This decision derived from the fact that this component is not the main focus of the dissertation, and both MoveIt Grasps and MoveIt Task Constructor required a longer setup process compared to the Pick and Place Pipeline, which allowed for a quick and easy setup of different pick and place actions. However, the use of the Pick and Place Pipeline is not recommended for larger applications given that, compared to the MoveIt Task Constructor, it is not an easily scalable option.

## 3.6 Motion Planner

The MoveIt framework supports a variety of motion planning techniques. This can be done since MoveIt integrates with motion planners via a plugin interface. This enables MoveIt to interface with and employ motion planners from various libraries. This also adds the possibility of implementing and using custom made planners. Having said that, for this project, three different planners commonly used with MoveIt were considered: Open Motion Planning Library (OMPL), Covariant Hamiltonian Optimization for Motion Planning (CHOMP), and Stochastic Trajectory Optimization for Motion Planning (STOMP).

The OMPL is a set of primarily sampling based/randomized motion planning algorithms. Because these algorithms are probabilistic complete, if a solution exists, given enought time, it will be found. MoveIt uses OMPL algorithms as the default planners on new MoveIt Config packages, making any planner from this library readly available to

use with the robot since the beginning. These algorithms usually find a solution quickly, but they frequently compromise path quality.

The CHOMP algorithm works around optimizing a given initial trajectory by quickly trying to pull that trajectory out of collisions given the current environment. Comparatively to OMPL algorithms, CHOMP is now also readly available to use when configuring a new robot and can also come up with a solution rather quickly. On the other hand, given a bad initial trajectory, the algorithm can become stuck in the local minimum and fail to generate a trajectory. On collision free environments, this algorithm produces quickly smooth trajectory between the intended poses, but on cluttered environments the trajectories found are usually the opposite given the addition of noise on factors such as acceleration and velocity.

The STOMP algorithm is designed to find smooth, well-behaved trajectories. Similarly to CHOMP, it works around a given initial trajectory, but in this case it generates noisy trajectories around it to explore the space around it before combining them to create a new lower cost trajectory. By using a stochastic method, STOMP can avoid local minimums, unlike CHOMP.

The three planners were tested by executing the various cues implemented, and the default planner from OMPL was subsequently chosen. In terms of path quality, the STOMP algorithm produced the simplest trajectories of the three planners, and despite not being a determinist algorithm, it produced the most similar trajectories when tested repeatedly from the same starting and end pose. The CHOMP algorithm presented the previously described problem in that when presented with collision objects in the environment, it consistently produced jittery trajectories, which let to the decision to not use this planner. The OMPL algorithms presented trajectories that were overall smooth, but the paths they produced varied greatly from attempt to attempt, with some generating a reasonably simple path and others requiring the arm to perform extremely exaggerated and unnecessary movements to reach the end pose. If it were only for this factor, the STOMP algorithm would have been chosen, but some aspects prevented the choice. One of the reasons was that this algorithm took consistently more time to find solutions when compared with the others in the same scenario and conditions. In general, the difference in times was not considerable, with both algorithms finding solutions in under a second, with OMPL usually finding them faster, but in some scenarios, STOMP consistently needed upwards of three seconds to find a solution. The other reason has to do with solution finding consistency. Even in scenarios in which the spatial complexity was not too high, like the one described in section 4, the STOMP algorithm in some iterations of testing could not find a solution, even though it was able to find one in previous attempts in the same conditions. On the other hand, the OMPL algorithms did not fail to find a solution in any of the testing attempts, even though each solution rarely reseambled each other. Some of these problems may be related to the processing power of the system used to run these simulations and should be tested with different hardware in future work. However, some of these problems were also partially found in Liu and Liu [56] research.

# 4

# Test Design

This chapter will go over the process of designing and planning the experimental tests. This component is essential to the study, as the effectiveness of the cues will be validated through the results obtained from these tests. It starts with section 4.1 which provides a brief overview of the layout of the task's environment. It concludes with section 4.2, which describes the desired workflow for completing the task correctly.

## 4.1 General Layout

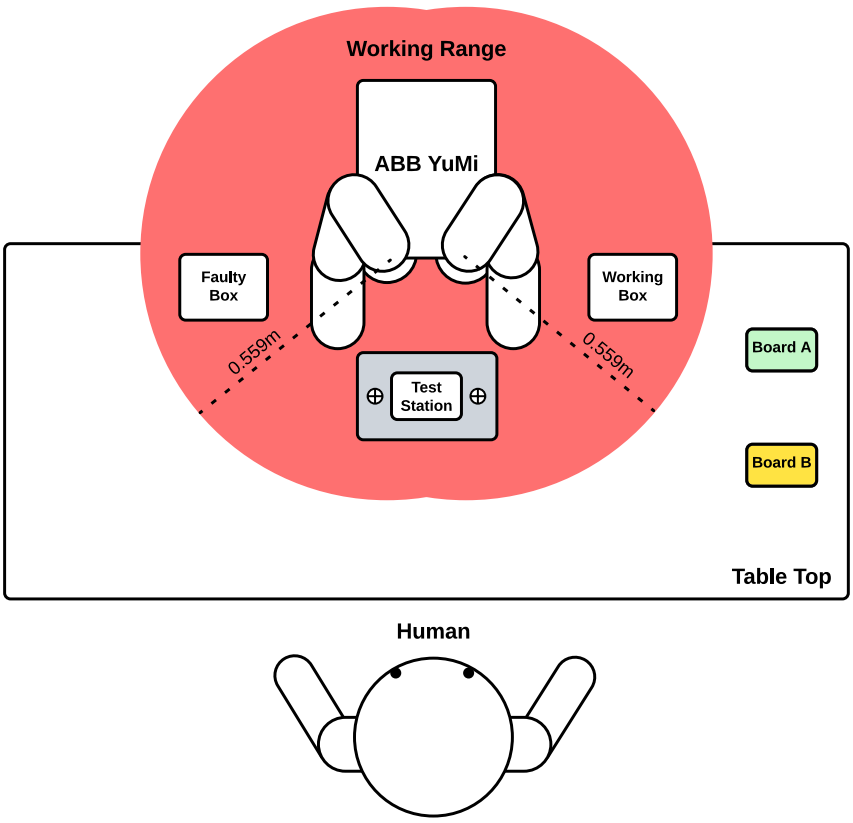



Figure 4.1: Layout of the testing task's working station.

The layout of the task is illustrated in Figure 4.1. A dual-arm collaborative robot (ABB YuMi) is positioned at the center of one of the long edges of a table. This robot has a working range of 0.559 meters for each arm, which is represented by the red area. As manipulation objects, there are displayed two boards (Figure 4.2), labeled A and B, which are positioned, relatively close to each other on the right side of the tabletop, just outside the robot's working range. A testing station (Figure 4.3) is located in the center of the table, well within the robot's reach. This one is made up of a slot for one board and a locking mechanism that secures the board in place which the robot cannot operate. On each side of the robot, there is a box that will receive the board based on the test results. The human operator is positioned at the opposite end of the table relative to the robot, where there is access to the entire table area. This layout is intended for the execution of a manipulation task that requires the effective use of nonverbal cues in order to be executed successfully.

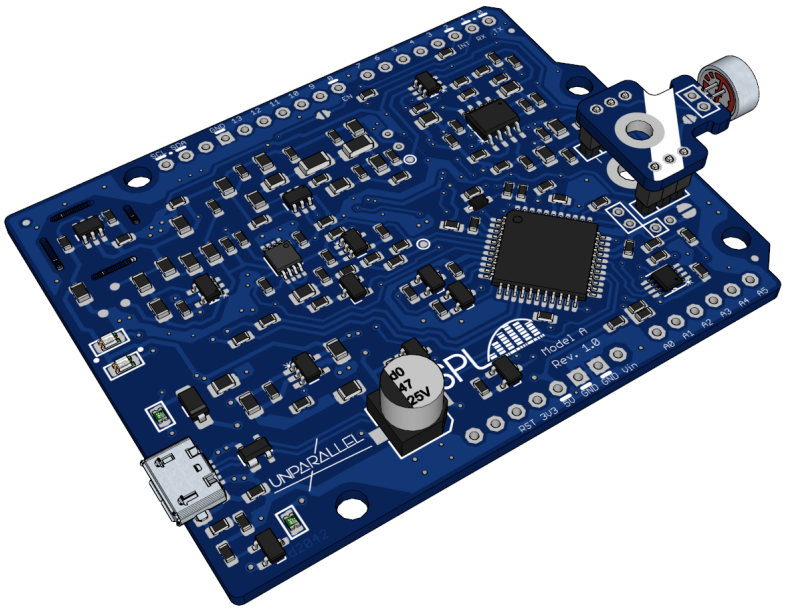



Figure 4.2: 3D model of the board used in the task.

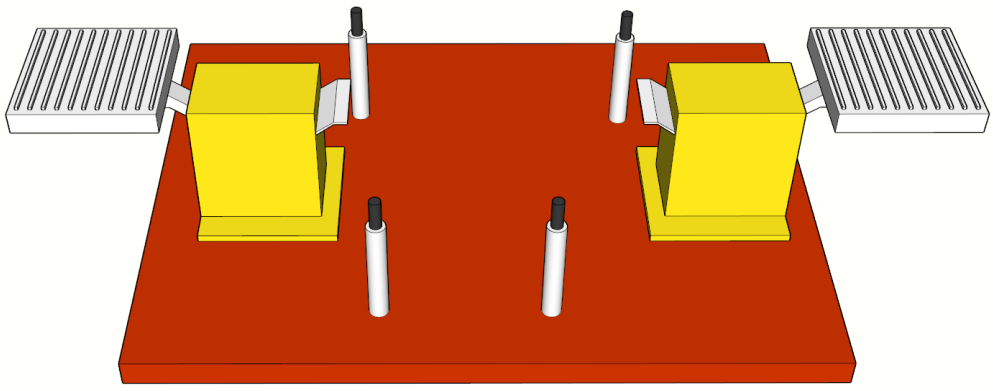


Figure 4.3: 3D model of the testing station used in the task.

## 4.2 Task Design

The main objective of the task is to make an assessment of the functioning capacity of the board of type A and store it in its respective box. To achieve this, the board has to be placed on the testing station located at the center of the workbench. Given that, in this scenario, only the robot knows both the context of the task and the steps required to complete it, and that it lacks the capabilities to do so on its own, the workflow depicted in Figure 4.4 was designed.

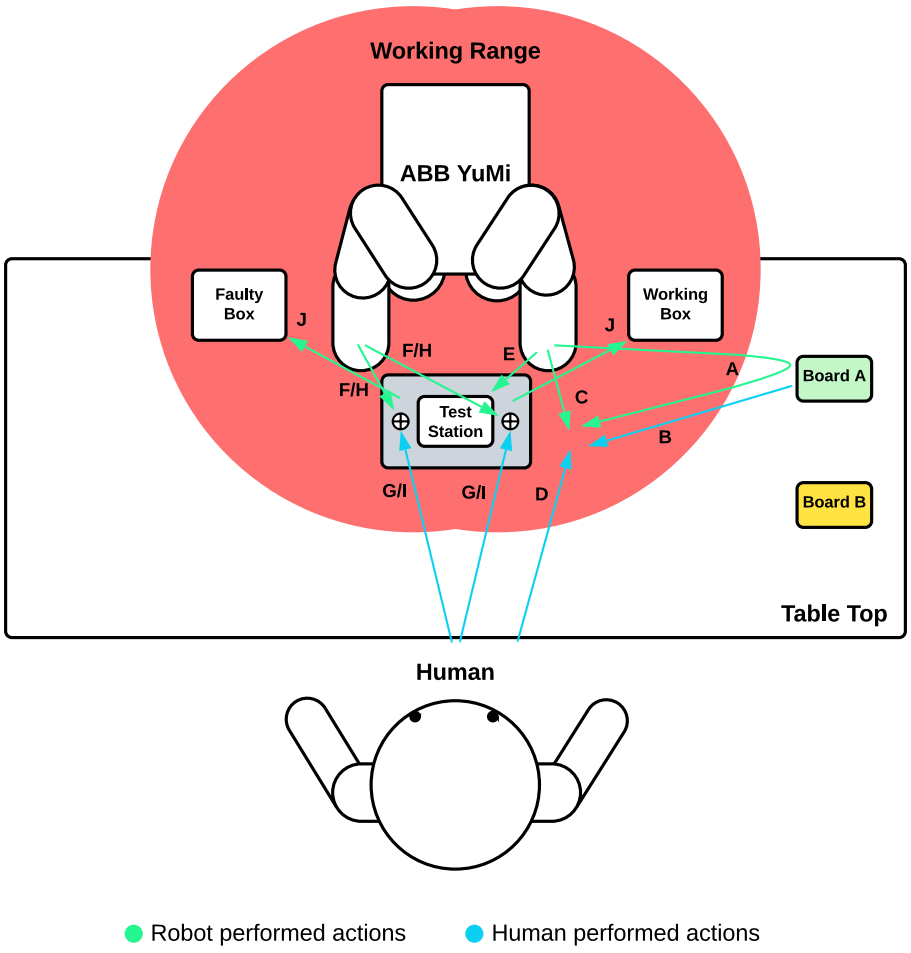



Figure 4.4: Desired task flow. Green arrows represent the robot's actions, blue arrows represent the actions of the human operator.

The workflow of the task consists of the execution of three types of movements: communication movements by the robot, manipulation movements by the robot, and manipulation movements by the human operator. In this work, the robot's movements were hardcoded, which means that the robot will only perform the movements that were defined in advance and will lack decision-making capabilities. This was done in this manner to provide a low complexity implementation of the task, which was required due to time constraints for the test execution. A categorization of each movement and a

summary of the desired task flow can be found in Table 4.1, which is then followed by a brief description of each movement and a visual representation of its execution within the task context.

Table 4.1: Description of the individual actions that make up the desired task flow.

| Move Code | Actor | Type | Description |
|---|---|---|---|
| A | Robot | Cue | Points to board A, signals to pick up, points to desired location on the table |
| B | Human | Manipulation | Picks up board A and places it in the designated location |
| C | Robot | Cue | Signals rotation action to be performed on board A |
| D | Human | Manipulation | Performs rotation action |
| E | Robot | Manipulation | Picks up board A and places it on the test station |
| F | Robot | Cue | Signals screwing action on locking mechanism |
| G | Human | Manipulation | Performs screwing on locking mechanism |
| H | Robot | Cue | Signals unscrewing action on locking mechanism |
| I | Human | Manipulation | Performs unscrewing on locking mechanism |
| J | Robot | Manipulation | Picks up board A and places it on the designated box |

As a context mechanism, whenever the robot performs a nonverbal cue that suggests human actions, the robot makes a pointing gesture towards the human beforehand, similar to the one depicted in Figure 4.5.

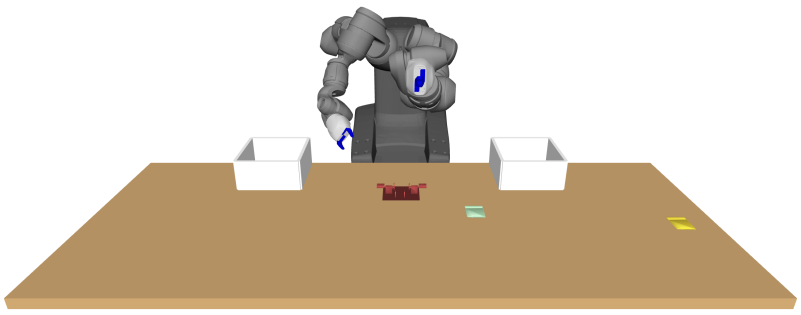
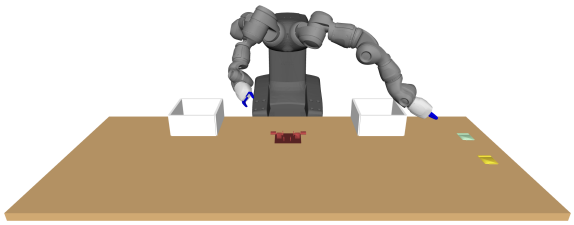
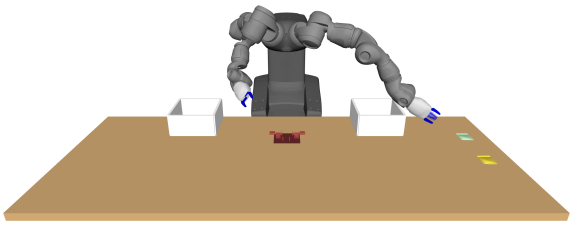


Figure 4.5: Point to Human.

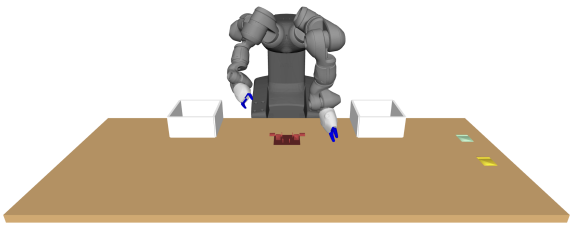
In the state in which the task begins, the robot cannot reach board A to execute its functions. Therefore, the task starts off with a sequence of three cues performed by the robot [A]. The robot begins with a pointing gesture towards board A (Figure 4.6a), then a picking cue (Figure 4.6b), and finally a pointing gesture indicating the desired location on the table to set the board (Figure 4.6c).



(a) Point to object



(b) Pick object

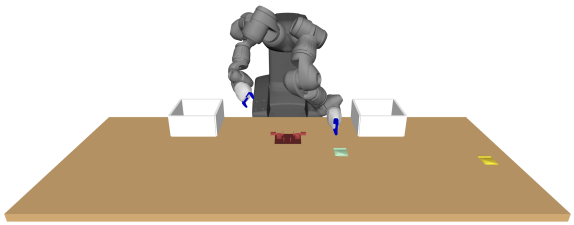


(c) Point to location

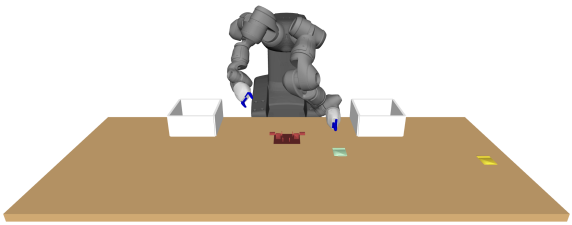
Figure 4.6: Task Movement A.

Subsequently, the human operator is expected to carry out the action that they believe the robot intends, which is to identify, between the two boards, the reference to board A, pick it up, and place it in the desired location.

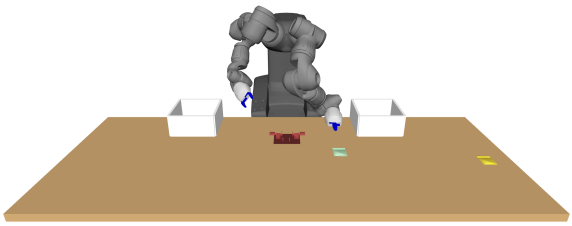
The slot on the testing station only allows the board to be placed in a specific manner, so the robot must first verify if the board was placed in an orientation suitable to be picked and placed effectively. If the orientation is undesirable, the robot uses the rotate cue (Figure 4.7) to indicate how to rotate the board so that it ends up in the correct orientation [C].



(a) Signal Rotate (Step 1)



(b) Signal Rotate (Step 2)

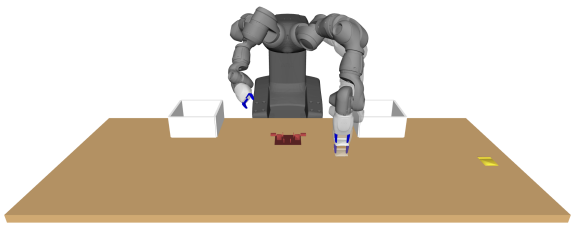


(c) Signal Rotate (Step 3)

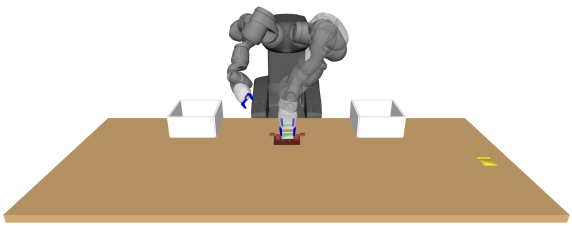
Figure 4.7: Task Movement C.

The human operator in this situation would have to identify the action of rotating the object as well as the direction of rotation instructed [D].

With the board in the desired place and orientation, the robot then proceeds to pick up the board and place it down on the testing station in the desired orientation (Figure 4.8) [E].



(a) Pick

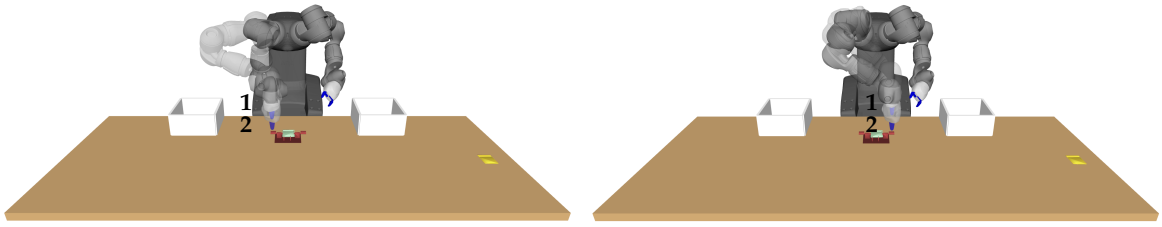


(b) Place

Figure 4.8: Task Movement E.

As previously stated, to secure the board in place, a locking mechanism is present, which the robot cannot operate. Consequently, the robot must communicate to the human operator the necessary action to take. As so, the robot performs a screwing cue (Figure

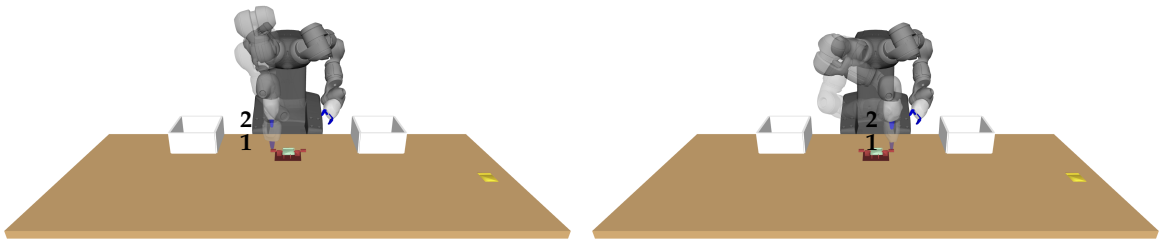
4.9) near both locking mechanisms [F].




(a) Left Screw cue (b) Right Screw cue

Figure 4.9: Task Movement F.

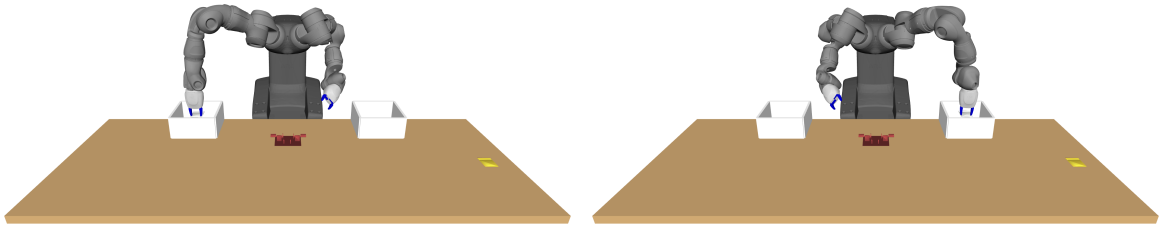
After that, it is expected for the human operator to perform the desired action [G]. A performance evaluation is conducted on the board, and depending on the evaluation outcome, the robot places the board into one of two designated boxes. However, before the robot can do so, the locking mechanism must be released by the human operator. As a result, the robot performs an unscrewing cue near both locking mechanisms, similarly as before (Figure 4.10) [H].




(a) Left Unscrew cue (b) Right Unscrew cue

Figure 4.10: Task Movement H.

Similarly to the previous movement, it is expected for the human operator to release the board from the testing station [I]. After the release of the board, the robot completes the task by picking up the board and placing it in the appropriate box (Figure 4.11) [J].



(a) Faulty (b) Working

Figure 4.11: Task Movement J.

As can be seen from the prior description, this task offers a few opportunities to take advantage of almost all of the robot's nonverbal cues, namely pointing to the human operator, pointing to an object, picking an object, pointing to a location, rotating an object, and screwing or unscrewing.

# 5

# Experimental Results

This chapter will describe in section 5.1 the test's experimental setup and the acquisition method of the cues conveying performance, followed by section 5.2 with an overview of the results and their corresponding analysis.

## 5.1 Experimental Setup and Acquisition Method

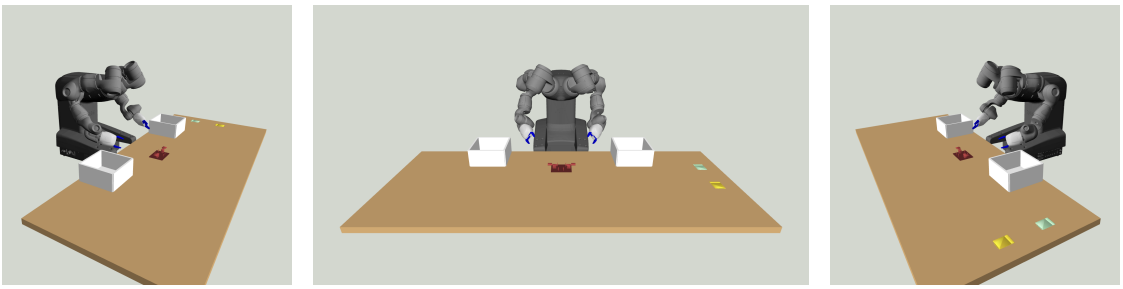
As described in the previous chapter 4, the testing component of this study is made up of one collaborative manipulation task that takes advantage of the majority of the nonverbal cues developed, and was designed with the objective of evaluating the conveying effectiveness of these nonverbal cues.

For the execution of the test, two approaches were considered: using a real robot in which a person physically interacts with it and the environment, or using a simulated robot in which a person only observes the robot and the environment. The latter approach was the one chosen due to the delayed availability of the real robot, which wouldn't allow for the integration and testing of the implemented method within the available timeframe. Given this, the execution of the test and data collection were done in the following manner.

The simulated robot and environment for the human participant to observe the test task execution were captured using MoveIt's representations of the robot model and collision objects on RViz. This choice stems from the fact that MoveIt allows a representation of both the robot model and the collision object geometry with a high degree of visual accuracy when compared to the real versions of each, as well as a smooth execution of the arms' movements with control over its velocity and acceleration. This approach does not simulate physics. This is not an issue because the objects are never positioned in such a way that external forces such as gravity have an effect, as they are either stationary on the table top or test station, or attached to the robot when performing a manipulation action.

With that in mind, the test task execution was recorded in video format on RViz using three different camera angles: left (Figure 5.1a), front (Figure 5.1b), and right (Figure 5.1c). Following that, the three video angles were combined and edited to play simultaneously in a configuration resembling the one shown in Figure 5.2. This was done in order to

mitigate the viewpoint issues pointed out by Holladay, Dragan, and Srinivasa [29] and Bodden et al. [33].



(a) Left Camera (b) Front Camera (c) Right Camera

Figure 5.1: Angles recorded for the test task videos.

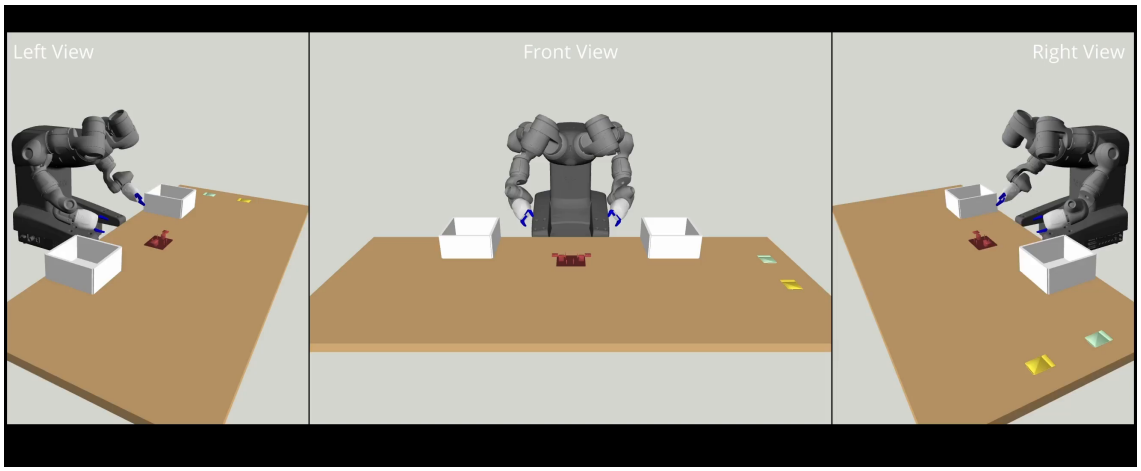



Figure 5.2: Configuration template of the videos depicting the test task. The left view angle occupies the left portion, the front view angle occupies the middle portion, and the left view angle occupies the right portion.

The video was subsequently split into four distinct parts. Each part shows a different cue move from the ones described in Table 4.1, with part one displaying move code A, part two move code C, part three move code F, and part four move code H. The videos were used as visualization content in a Google Form questionnaire used to gather data about the conveying effectiveness of the nonverbal cues used in the task. The conveying quality of these cues is intended to be acceptable to the average person. As a result, this questionnaire aims to be filled out by people from various professional backgrounds.

The questionnaire begins by gathering information about the human participant through the following five questions:

- What is your age?
- In which industry do you now work in?
- Do you consider yourself to be someone who uses and adapts to technology easily?

- What level of experience do you have with handling and assembling electronic devices?

- Have you ever interacted with a collaborative robot?

After that, the human participant will be presesented with four similar sections. First, they will be asked to watch a video in which the robot executes one of the four movement codes. After watching the video, the human participant is asked to provide a brief description of the action they believe they are supposed to perform based on the movements they observed (Figure 5.3).

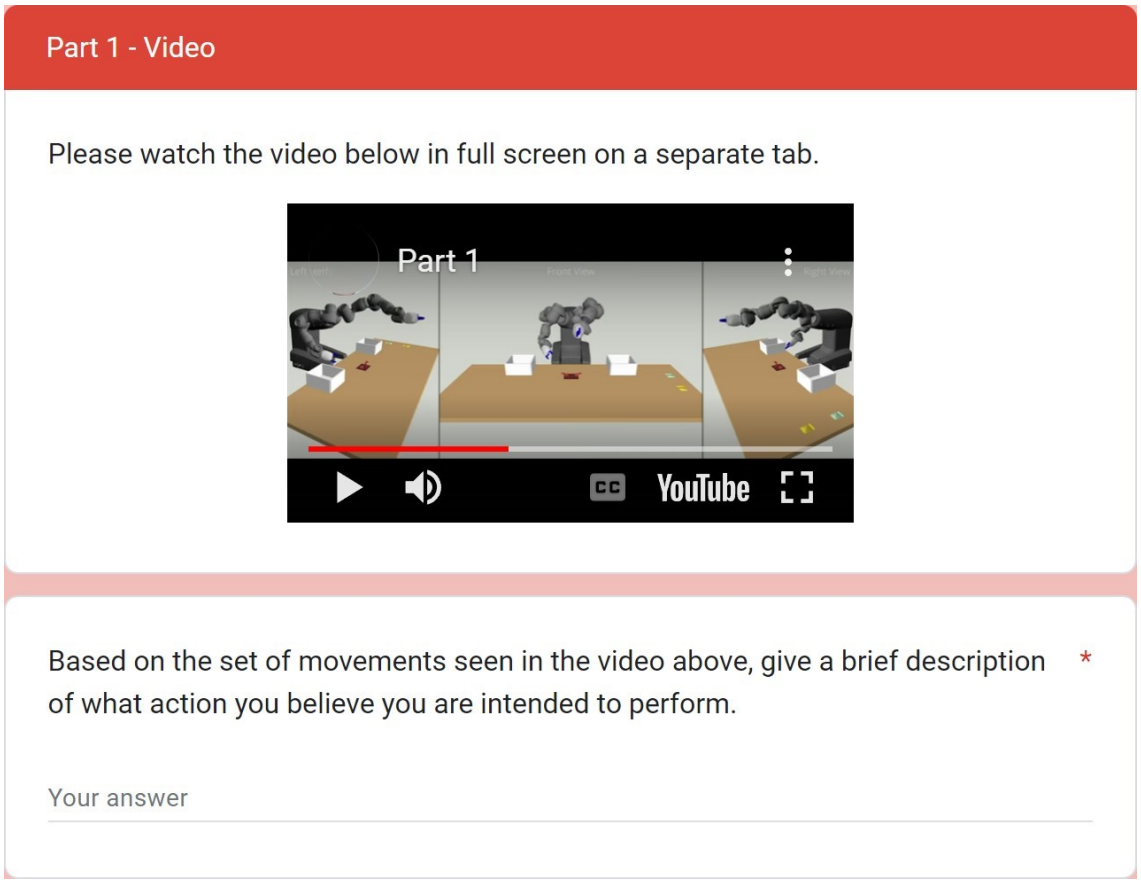



Figure 5.3: Section of the questionnaire containing the video content and input text field for the participants anwser.

Following that, the true conveying intention of the set of movements is revealed to them, and the participant is asked to rate how well the true intent of the movements compares to the perceived intent on a scale of 1 to 10, with 1 representing that the movements failed completely to convey the true intent and 10 representing that the movements effectively conveyed the true intent. Furthermore, it is asked of the human participant to select, if applicable, the individual cues from the complex movement that were more challenging to comprehend (Figure 5.4).

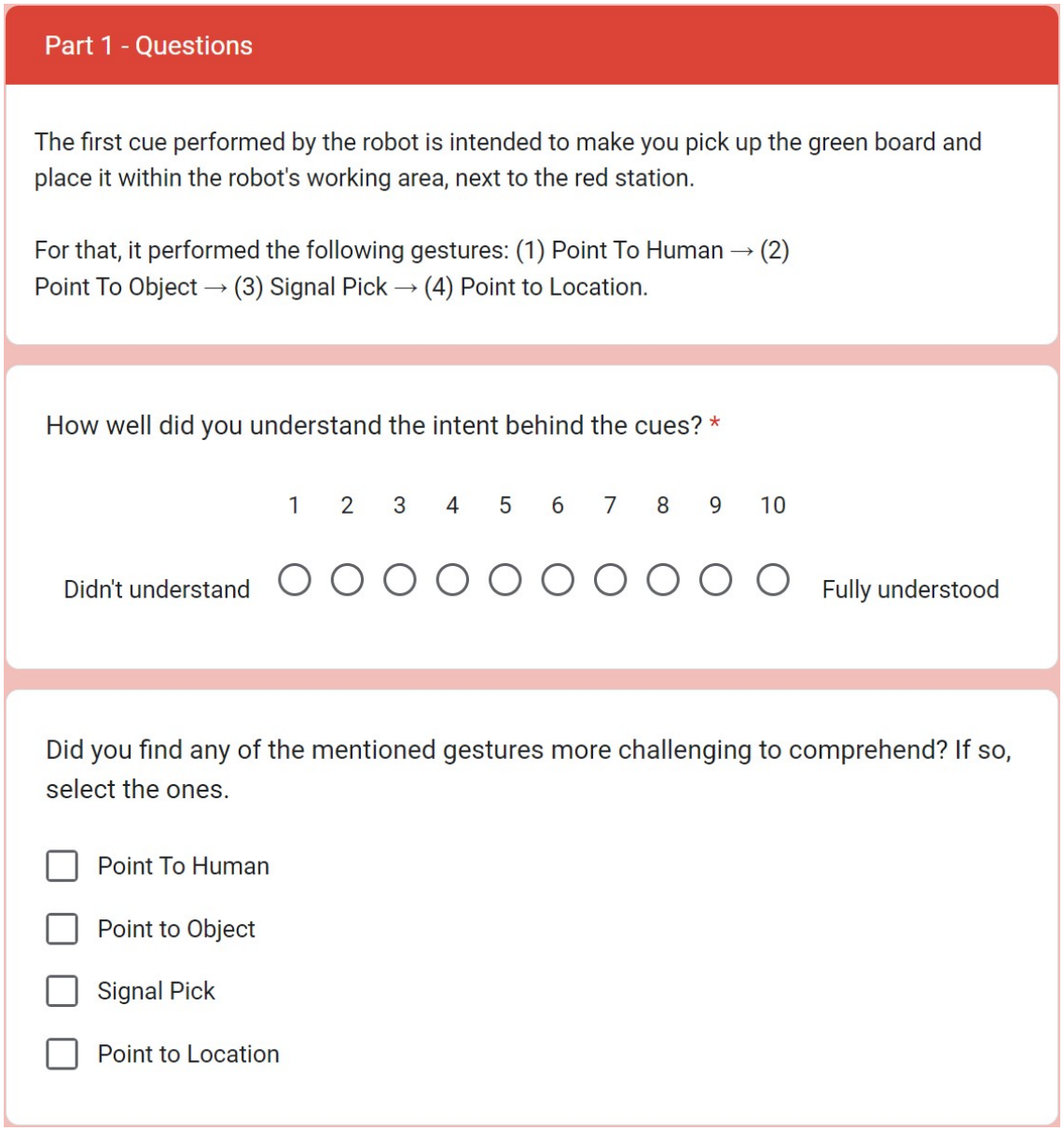



Figure 5.4: Section of the questionnaire containing an explanation of the cue's intent followed by an evaluating component made of a rating scale for the conveying quality and a multiple selection.

At last, after watching the four videos and answering their respective questions, the human participant is asked two additional questions. The first one has to do with the influence that the MoveIt's path planning for each movement has on the overall perception of the cue's intent, in which the human participant is asked to classify its influence as either positive, neutral, or negative. The final question prompts the person to describe the motions they expected this robot model to perform to convey the cues that they found more difficult to understand (Figure 5.5).

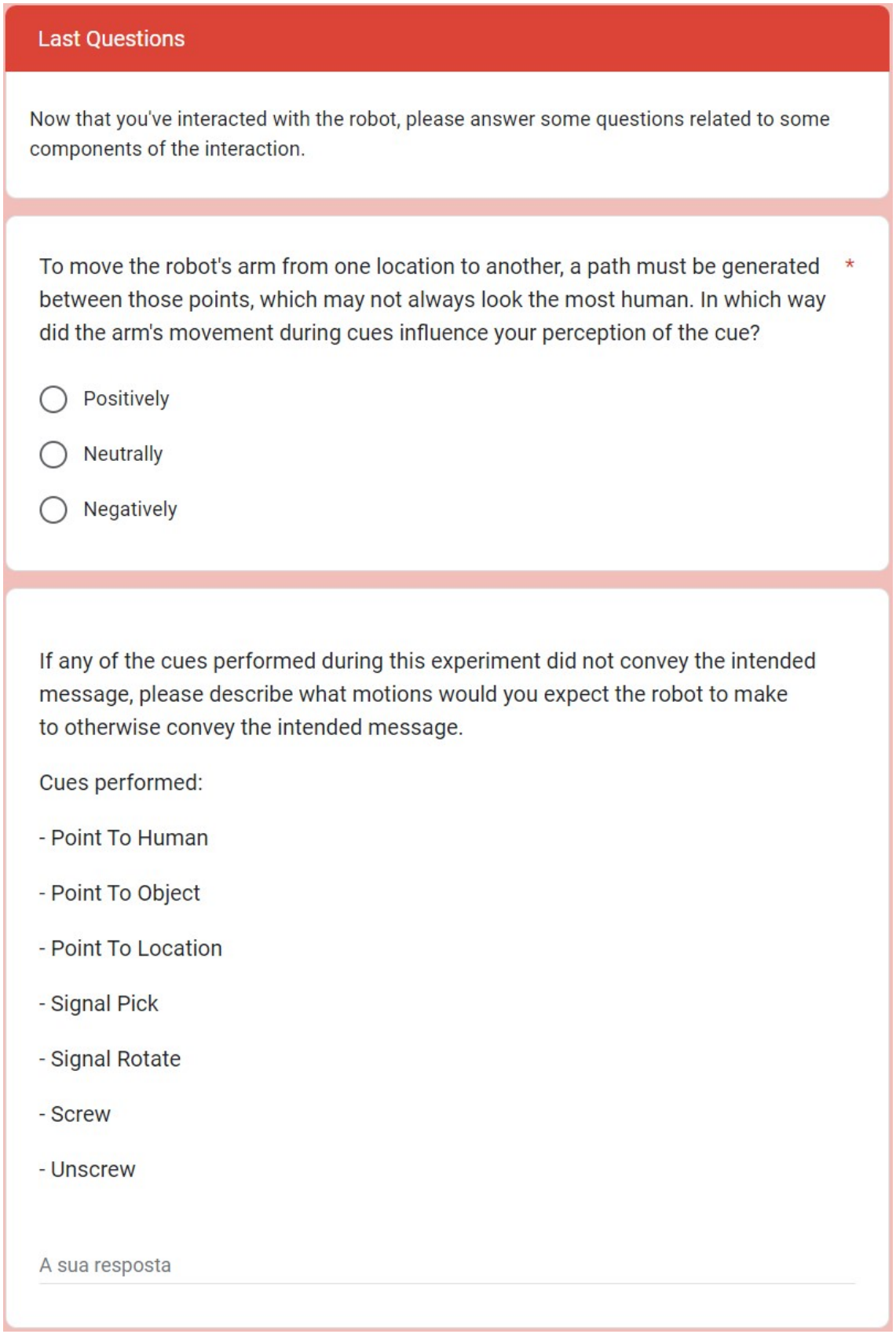



Figure 5.5: Last section of the questionnaire in which the participant rates the influence of the motion planning on the conveying quality of the cues and can also provide implementation suggestions.

## 5.2 Results

The previously described questionnaire was sent and filled out by 21 people. Table 5.1 shows the age distribution of the participants, with an average of 25.33. Figure 5.6 shows the work industry distribution of the participants, with 9 identified as a *Student*, 4 in *Engineering or Manufacturing*, 5 in *Computing or IT*, 1 in *Healthcare*, 1 in *Energy and Utilities*, and 1 in *Transport and Logistics*. All participants reported being easily adaptable to technology . When asked about their experience handling and assembling electronic devices, 8 participants said they had none, 9 said they had hobbyist experience, and 4 said they handle and assemble electronic devices professionally (Figure 5.7). Only two of the 21 participants reported having previously interacted with a collaborative robot (Figure 5.8).

Table 5.1: Age distribution of the questionnaire participants.

| Age | 19 | 21 | 22 | 23 | 24 | 26 | 29 | 31 | 33 | 50 |
|---|---|---|---|---|---|---|---|---|---|---|
| Count | 1 | 1 | 1 | 11 | 2 | 1 | 1 | 1 | 1 | 1 |

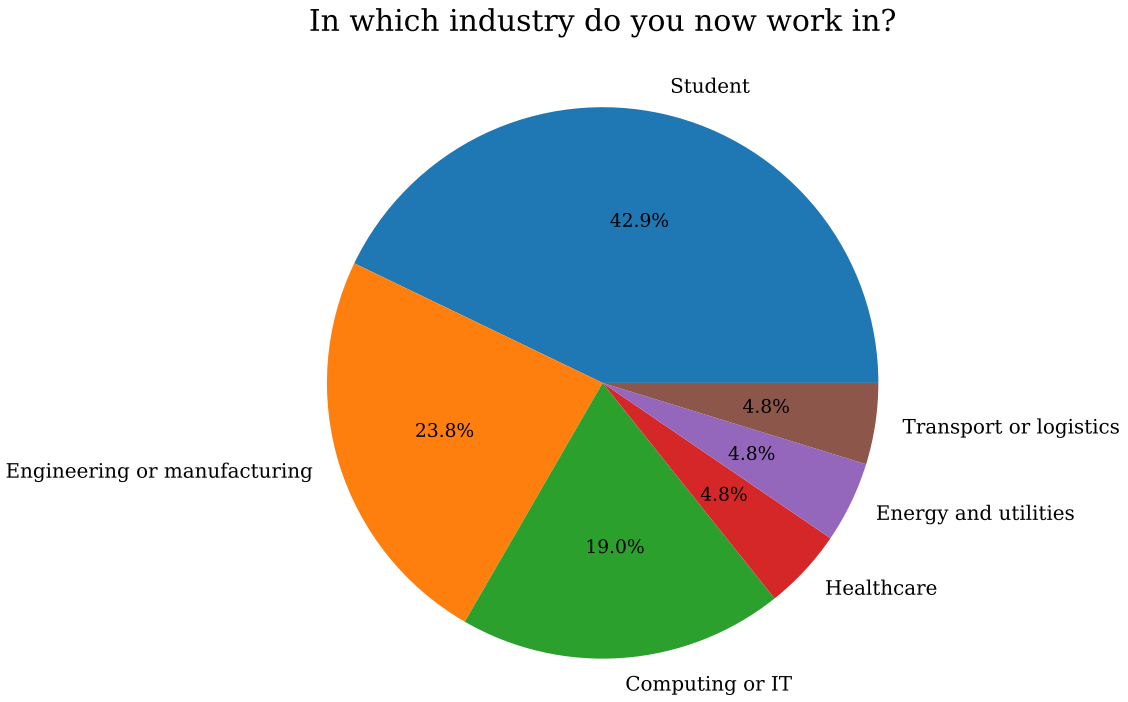



Figure 5.6: Industry distribution of the questionnaire participants.

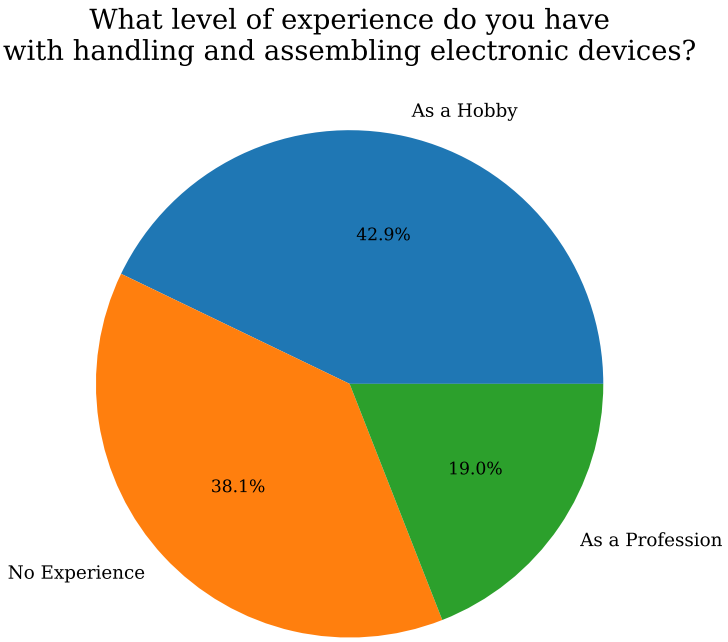



Figure 5.7: Experience in handling and assembling electronic devices of the questionnaire participants.

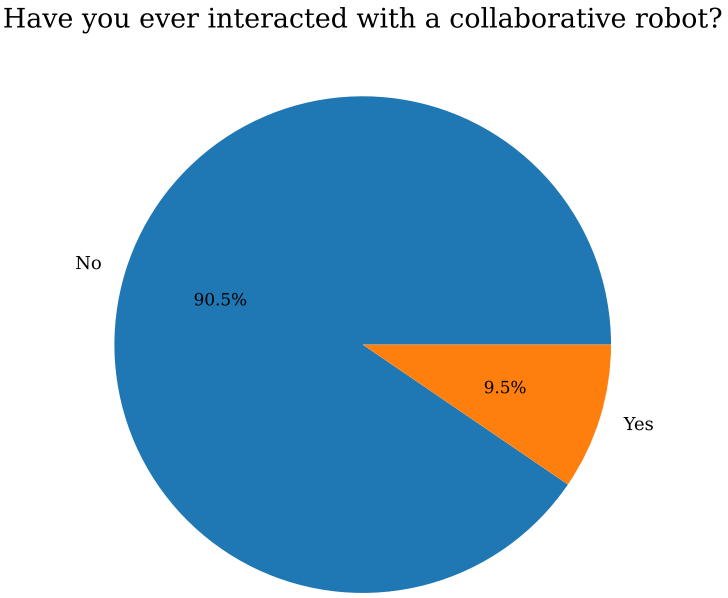
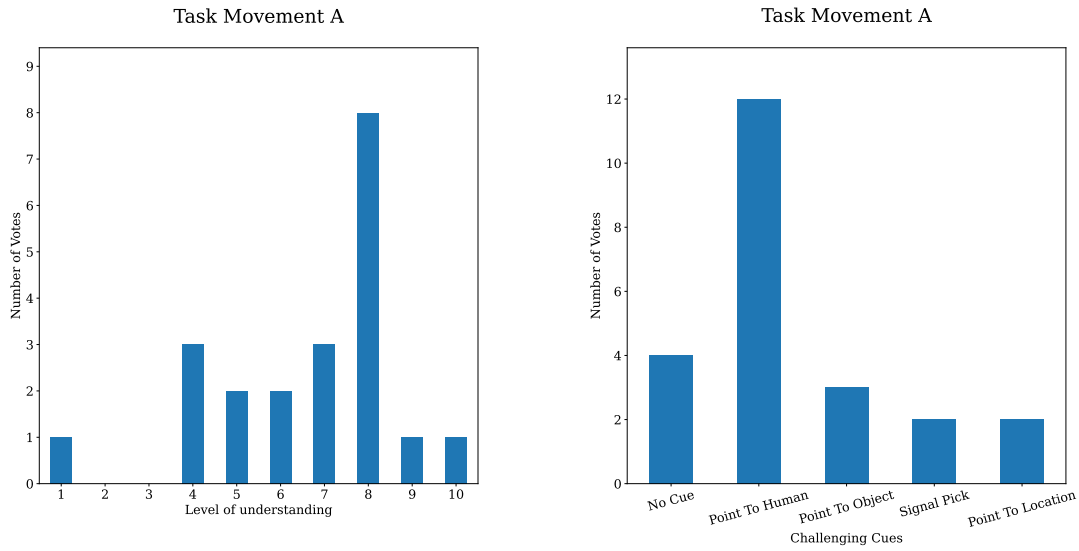



Figure 5.8: Experience with collaborative robots of the questionnaire participants.

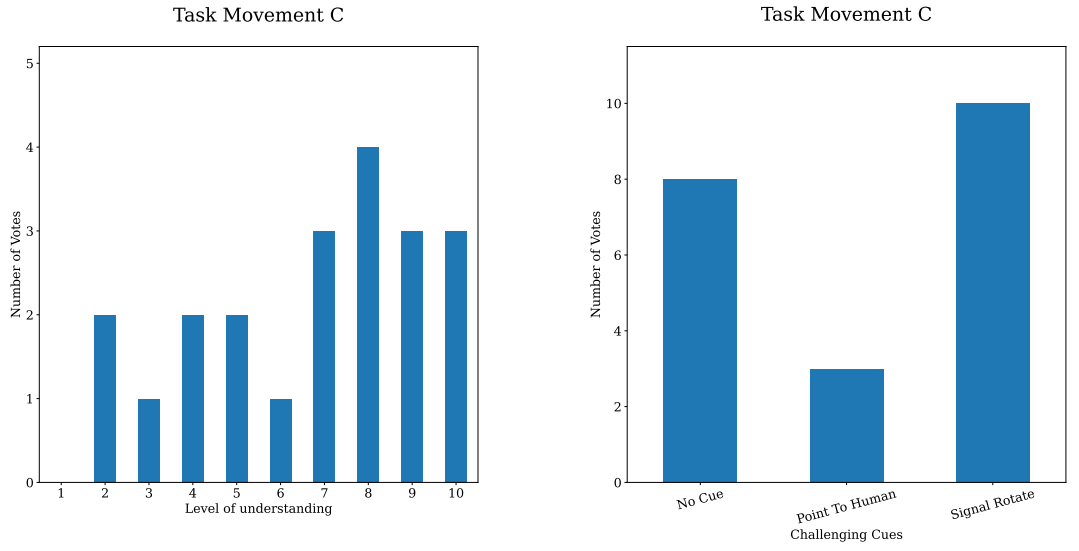
In a first analysis, the data extracted from the questionnaire was analysed without categorization based on the participants' profile in order to gather a general conclusion on the effectiveness of the movements.

The results of Move A, Move C, Move F, and Move H without categorization are represented in Figures 5.9, 5.10, 5.11, and 5.12, respectively, in which a) depicts the vote distribution of the level of understanding of the participants for the whole move, and b) depicts the vote count for the individual challenging cues. Table 5.2 provides a summary of the data provided from the graphics referenced previously. Following that, Figure 5.13 displays the votes of the participants in regard to the influence of the arm's movement on cue perception.



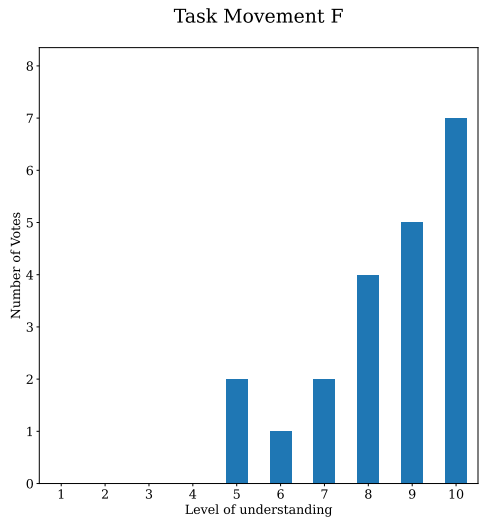

(a) Level of Understanding (b) Challenging Cues

Figure 5.9: Move A data without categorization.

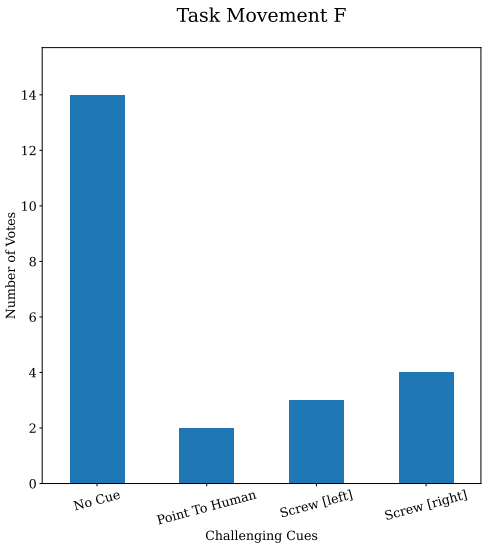



(a) Level of Understanding (b) Challenging Cues

Figure 5.10: Move C data without categorization.

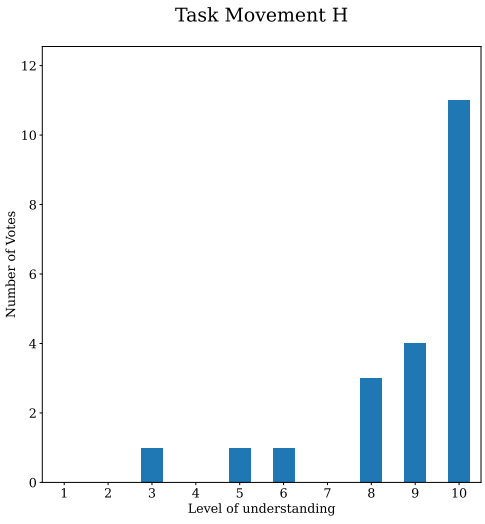



(a) Level of Understanding

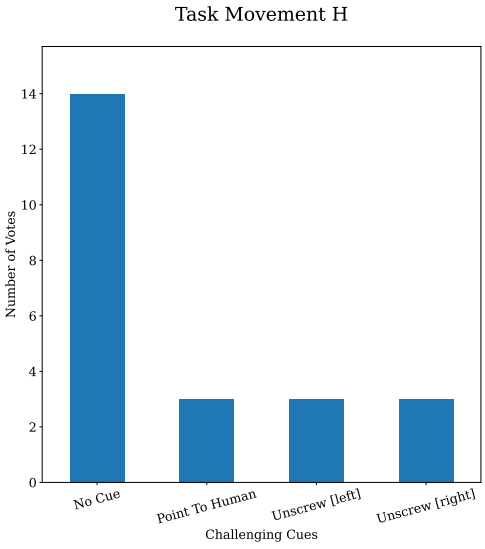



(b) Challenging Cues

Figure 5.11: Move F data without categorization.




(a) Level of Understanding




(b) Challenging Cues

Figure 5.12: Move H data without categorization.

Table 5.2: Summary of results from "Level of Understanding" graphics considering no categorization.

| | Vote Count | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Move Code** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **Mean Vote** |
| A | 1 | 0 | 0 | 3 | 2 | 2 | 2 | 8 | 1 | 1 | 6.62 |
| C | 0 | 2 | 1 | 2 | 2 | 0 | 3 | 4 | 3 | 3 | 6.71 |
| F | 0 | 0 | 0 | 0 | 2 | 1 | 2 | 4 | 5 | 6 | 8.43 |
| H | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 3 | 4 | 10 | 8.76 |

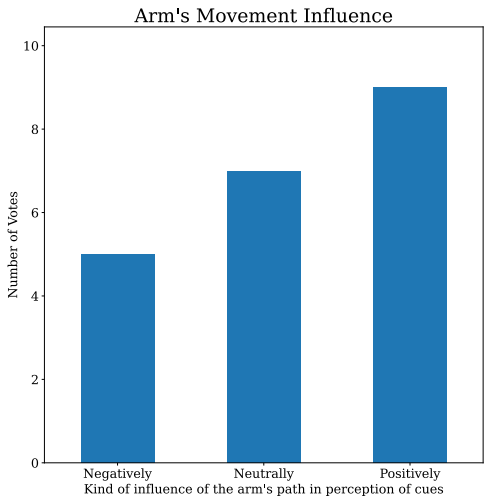



Figure 5.13: Arm's Movement Influence on Cue Perception.

By analyzing the data provided by the graphics above and the written responses provided by the participants, some conclusions can be reached:

- Move A received a mean vote of 6.62, which considering the rating scale, can be considered a reasonable result. The written responses of the lower half of the "Level of Understanding" scale (1-5) revealed that half of them did not understand that the robot was communicating and was instead performing random movements. Aside from one case in which a rotating cue was interpreted, the rest correctly identified the movements as communicative actions as well as described their true intent. In this latter case, all participants only rated the "Point To Human" cue as challenging, which could explain the low level of understanding vote. According to the responses on the upper half of the "Level of Understanding" scale (6-10), all participants perceived the robot's movements as communicative actions. The majority accurately described all of the actions the robot attempted to convey, while a couple of cases could only understand that the robot pointed to/selected the green object.

- Move C received a mean vote 6.71, which can also be considered a reasonable result. When compared to the others, this move had the most divided results, as nearly half of the participants struggled to understand the "Signal Rotate" cue, which conveyed the intent of the move, while nearly the other half found none of the cues difficult to understand. The written responses from the lower half of the "Level of Understanding" scale showed that the grand majority of the participants from this half either did not understand what the robot was doing or interpreted it as a failed attempt of a manipulation action by the robot. These interpretations could have been influenced by the arm's position, which was not the most natural and may have diverted attention away from the cue itself. Similarly to Move A, a singular case from this half described the correct intent of the cue, with only the "Point To Human" cue being rated as challenging. The responses of the upper half of the "Level of Understanding" scale showed that all participants from this half correctly interpreted that the robot asked them to rotate the green object. However, a couple of participants wrongly interpreted that the object should have been rotated back to its original orientation given the last rotation of the wrist.

- Move F received a mean vote of 8.43, which can be considered a good result. This move was perceived as the true intent by the participants when analyzing the written responses, who identified that they needed to perform the task of locking the green object into the red one, though some believed they could do so by screwing and others by pressing on the sides of the red object. Both interpretations could be considered correct, as mentioned in section 3.3.2.3, since the exact meaning of this cue depends on the context in which it is used, like the way the locking mechanism is physically operated. This communicative move was preceded by a manipulation action by the robot, also shown to the participants, with the majority of the participants being able to distinguish between the manipulation action and the communication action, correctly identifying in which component of the task they were asked to intervene.

- Move H received a mean vote of 8.76, which can also be considered a good result. This move was also correctly perceived by most of the participants when analyzing the written responses. However, given that the previous cue (screw) presented a similar execution with the one present in this move (unscrew), 3 out of the 21 participants perceived this cue also as screwing.

- The "Point To Human" cue was the first one performed by the robot, which was not identified by the majority of the participants as a cue but rather simply a movement the robot had to do in order to place the arm in a certain position. After receiving the context of the cue, the number of participants reporting the cue as challenging reduced significantly. Some participants reported that the generated path that preceded the pointing pose generated confusion, while others stated that they didn't think they would be referenced when observing a recording of a virtual environment.

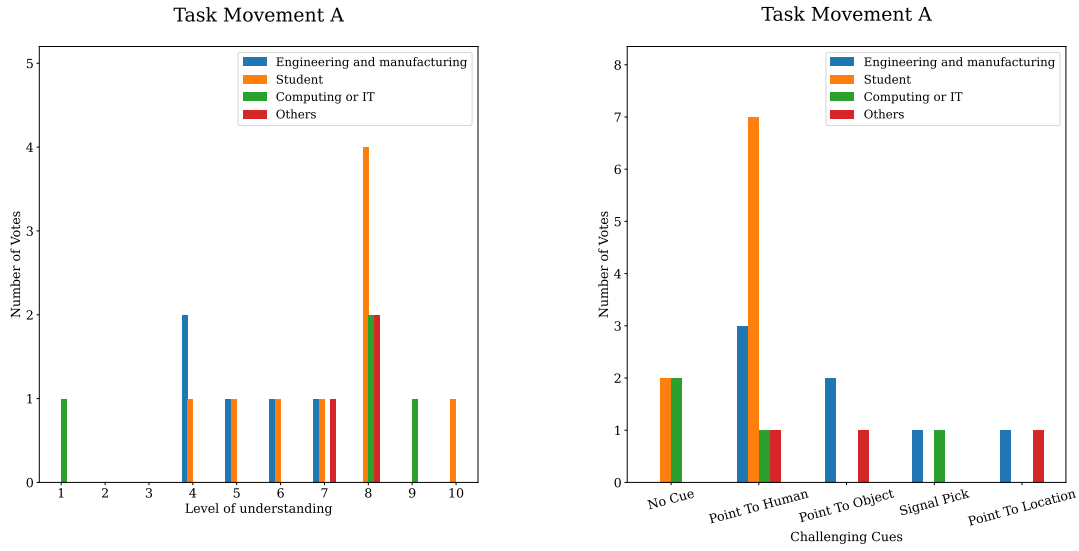
Given that this cue referenced the human participant's position, it is possible that the video format had a significant impact on the negative perception of this cue. If the robot had pointed at the human participant in a real-world setting, these issues might not have been as prominent.

- Approximately 24% of participants reported a negative impact of arm's movement on cue perception, while the rest reported either a neutral or positive impact. According to the group of participants who found the impact negative, the movement of the arm had the greatest impact on the perception of the cue "Point To Human", whereas the impact was not viewed negatively in the other cues. Although the implementation approach is the same, this impact was not reported to be as significant for other reference/pointing cues such as "Point To Object" and "Point To Location". This could be because, in this task, the "Point To Human" cue was used in situations where the path planning needed to account for more collision constraints, causing more complex looking trajectories.

On a second analysis, the data extracted from the questionnaire was analysed based on the participants' profile in order to understand if certain profiles had more influence on the level of understanding of the cues.

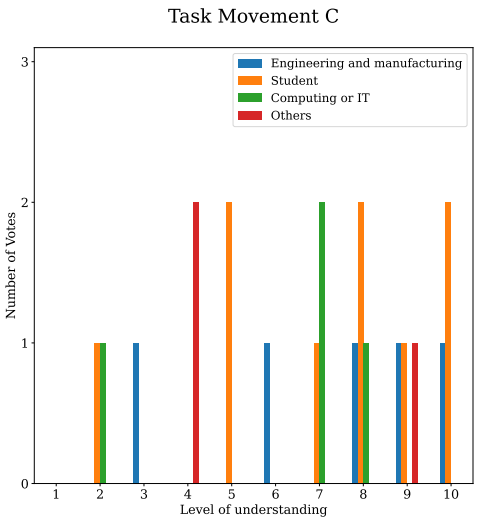
Two categories were created: by industry and by level of experience assembling and handling electronic devices. Categorizations such as age, adaptability to technology, and level of interaction with collaborative robot were not considered due to the significant difference in participant numbers between comparison groups.

The results of Move A, Move C, Move F, and Move H considering by industry categorization are represented in Figures 5.14, 5.15, 5.16, and 5.17. Table 5.3 displays the mean vote values for each move based on the industry category of the participants.

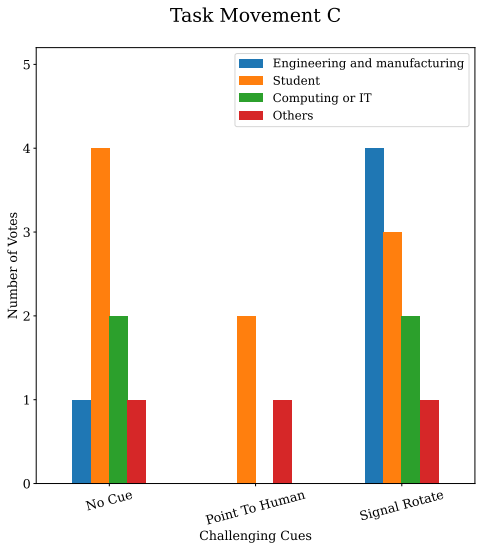



(a) Level of Understanding (b) Challenging Cues

Figure 5.14: Move A data by industry categorization.

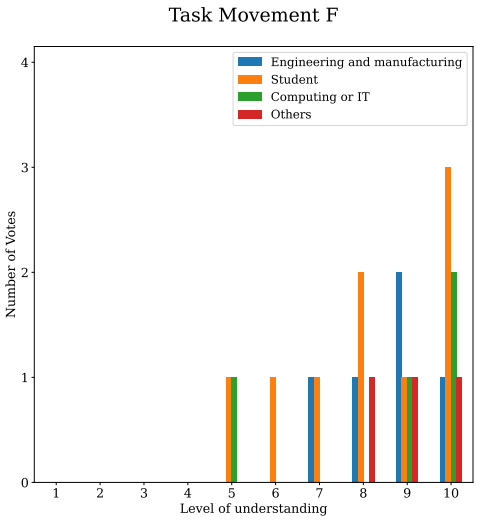



(a) Level of Understanding

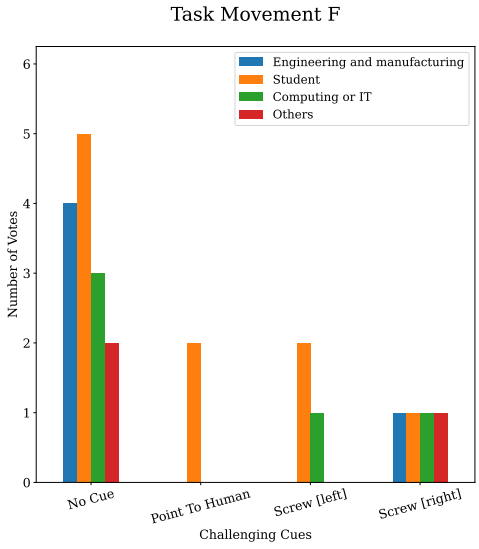



(b) Challenging Cues

Figure 5.15: Move C data by industry categorization.

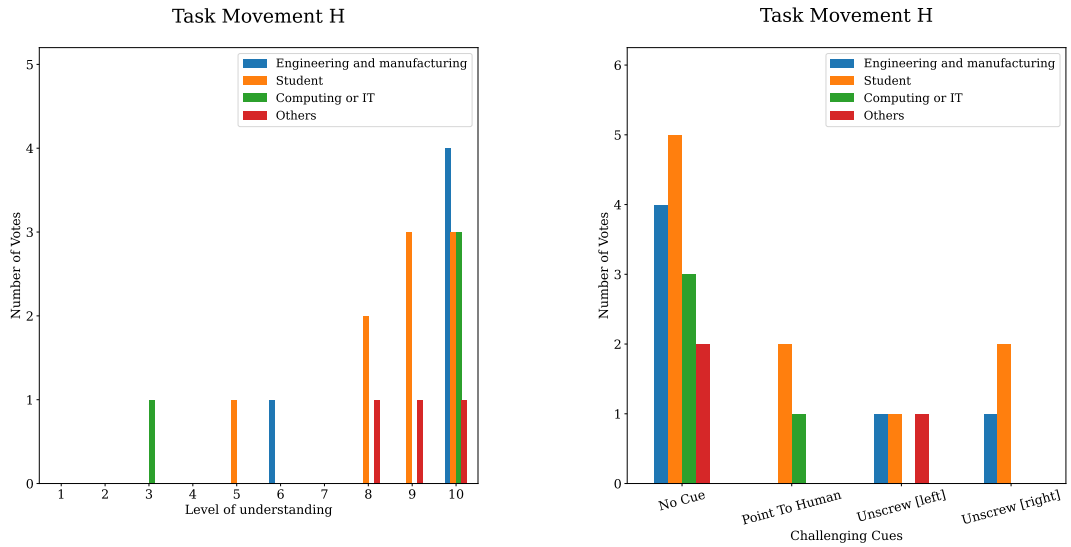



(a) Level of Understanding




(b) Challenging Cues

Figure 5.16: Move F data by industry categorization.



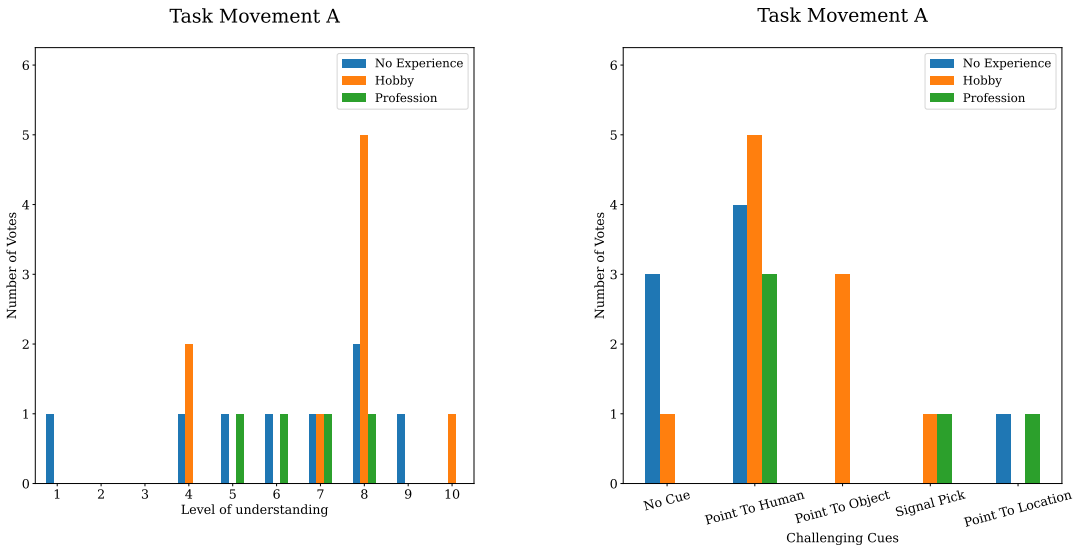

(a) Level of Understanding (b) Challenging Cues

Figure 5.17: Move H data by industry categorization.

Table 5.3: Summary of results from "Level of Understanding" graphics when considering by industry categorization.

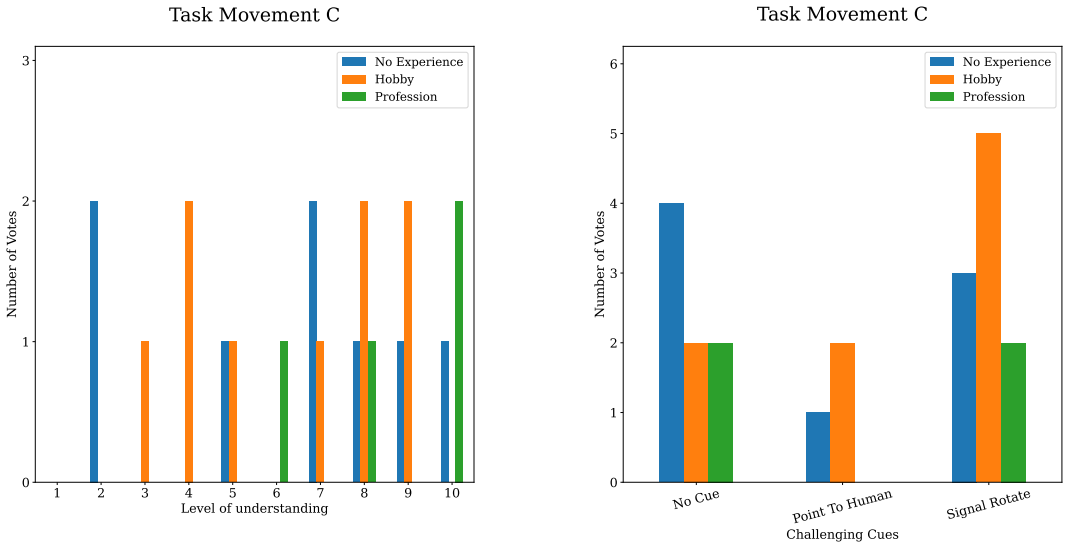
| | **Mean Vote** | | | |
|---|---|---|---|---|
| **Move Code** | Engineering and manufacturing | Student | Computing or IT | Others |
| A | 5.20 | 7.11 | 6.50 | 7.67 |
| C | 7.20 | 7.11 | 6.00 | 5.66 |
| F | 8.60 | 8.11 | 8.50 | 9.00 |
| H | 9.20 | 8.66 | 8.25 | 9.00 |

The results of Move A, Move C, Move F, and Move H considering by experience level in assembling and manipulation of electronic devices are represented in Figures 5.18, 5.19, 5.20, and 5.21. Table 5.4 displays the mean vote values for each move based experience level in assembling and handling electronic devices groups.



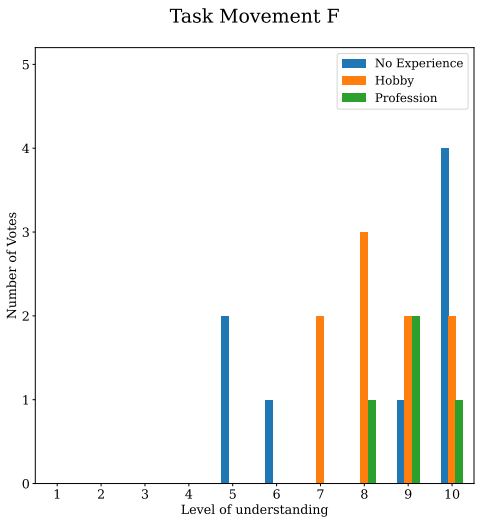

(a) Level of Understanding (b) Challenging Cues

Figure 5.18: Move A data by level of experience handling and assembling of eletronic devices.

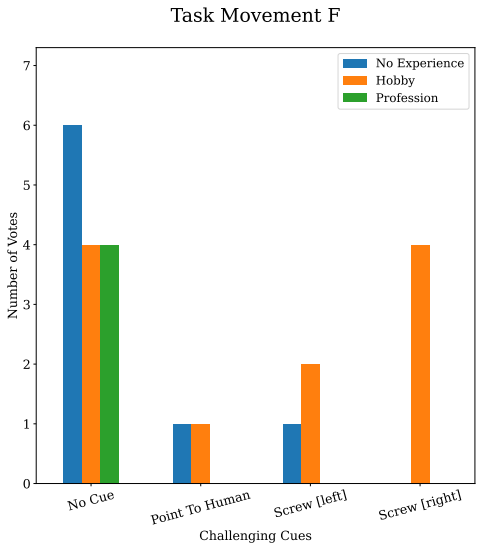



(a) Level of Understanding (b) Challenging Cues

Figure 5.19: Move C data by level of experience handling and assembling of eletronic devices.

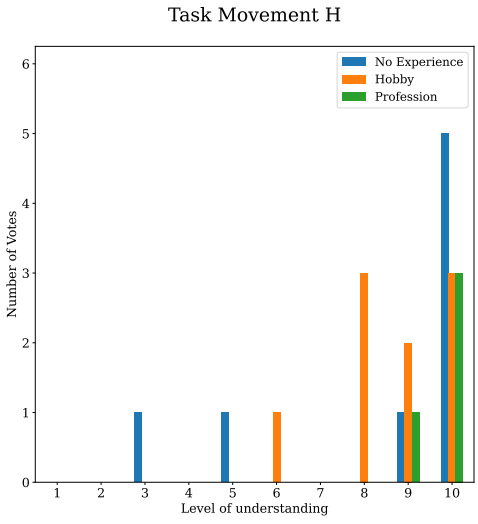



(a) Level of Understanding

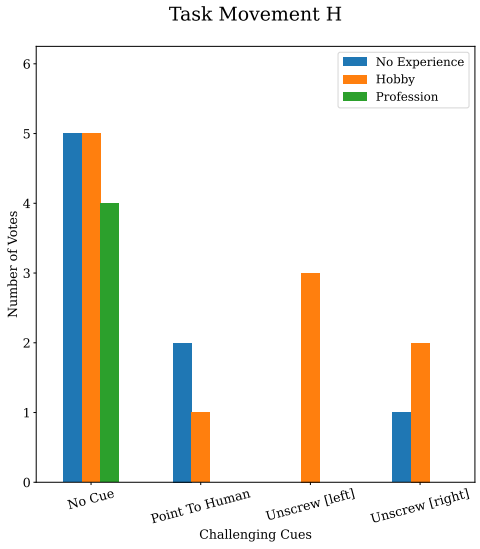



(b) Challenging Cues

Figure 5.20: Move F data by level of experience handling and assembling of eletronic devices.




(a) Level of Understanding




(b) Challenging Cues

Figure 5.21: Move H data by level of experience handling and assembling of eletronic devices.

Table 5.4: Summary of results from "Level of Understanding" graphics when considering by level of experience in handling and assembling of eletronic devices categorization.

| | **Mean Vote** | | |
|---|---|---|---|
| **Move Code** | No Experience | Hobby | Profession |
| A | 6.00 | 7.22 | 6.50 |
| C | 6.25 | 6.33 | 8.50 |
| F | 8.13 | 8.44 | 9.00 |
| H | 8.38 | 8.67 | 9.75 |

Although the number of participants per group is insufficient to draw an accurate conclusion, certain patterns can be identified when categorizing the participants:

- The average vote value per industry is very similar, with *Engineering and manufacturing* at 7.55, *Student* at 7.75, *Computing or IT* at 7.31, and *Others* at 7.83, with a maximum absolute deviation of 0.52. This means that no industry background consistently understood all of the moves much better than others.

- On the other hand, the average vote value per level of experience in handling and assembling of eletronic devices differs more, with *No Experience* at 7.19, as a *Hobby* at 7.67, and as a *Profession* at 8.44. The minimum absolute deviation between the average vote values is 0.48 between the groups *No Experience* and *Hobby*, which stands at a very similar value to the maximum absolute deviation when considering the categorization by working industry. In this case, the second largest absolute deviation is between the groups *Hobby* and *Profession* with a value of 0.77, and the maximum absolute deviation is between the groups *No Experience* and *Profession* with a value of 1.25. In this case, it could be said that there is a correlation between a higher level of experience in handling and assembling electronic devices and a higher level of understanding of the cues.

To summarise, the data revealed that, in general, the cues were perceived to be of reasonable quality, allowing the robot to instruct its human counterpart on the necessary actions to be taken in the task. Only the "Point To Human" and "Signal Rotate" cues were mostly reported as challenging to perceive correctly in the testing task. It is hypothesised that the lack of interaction with a real robot in a real environment, as well as the unnatural arm poses generated, are the culprits that contribute to the negative perception of the respective cues.

The arm's movement of the robot, which was generated by the default OMPL algorithm, presented a low negative influence on the cue's perception, though some reports of negative influence were received. The negative impact was primarily reported on referential cues, which appear to be the type of cue that would benefit the most from research conducted on motion design.

When the questionnaire participants' profiles were considered, the results revealed that for this task, the level of experience in handling and assembling electronic devices was related to the level of understanding of the cues. Given the small number of individuals in each category, this conclusion cannot be reached with certainty, but it does suggest that the human counterpart's knowledge of the task topic influences the ease with which the cues intent is identified.

# 6

# Conclusions and Future Work

## 6.1 Conclusions

In the manufacturing industry, humans and robots are typically assigned tasks that benefit from their respective strengths and capabilities, with robots usually performing dull, physically demanding, and repetitive operations and humans performing tasks that require more dexterity. However, some tasks might require the use of both skills sets. When participating in a collaborative task, in order to have an intuitive and safe interaction, both humans and robots must be able to understand each other's intentions. As a result, communication methods must be implement. This dissertation proposes an approach of implementing nonverbal communication capabilities on a robotic manipulator to aid in the execution of manipulation collaborative tasks. With this goal in mind, a review of previous and current approaches and studies on this topic was conducted, depicted in section 2.

The state of the art research focused on communication techniques and approaches, specifically the robot-to-human communication component used in collaborative tasks. Three distinct modalities were identified: verbal communication, nonverbal communication, and extended reality techniques.

- It was demonstrated that verbal communication, when compared to nonverbal communication, resulted in higher levels of understanding from humans during collaborative tasks and typically does not require training or getting accustomed on the part of humans. Nevertheless, this communication approach is not viable in high ambient noise level environments, is hindered by language barriers, becomes more descriptively costly in high complexity scenarios, and, depending on how the sentences are constructed, can affect the recall accuracy of instructions and the level of trust in the robot.

- The use of nonverbal communication seems to be the most ideal for this kind of environments since it is robust against ambient noise and doesn't require external components to implement. Nonetheless, the majority of nonverbal communication

research found focuses on movement predictability or legibility, resulting in a lack of understanding about the applicability of other cues in collaborative task scenarios. Regardless, these studies have shown that the implementation of nonverbal communication capabilities on robots varies depending on the robot's morphology and characteristics, the perception of them can be influenced by the receiver's perspective, and the robot's movements can have an impact on the well-being of its human counterpart.

- Finally, there have been some recent studies that use extended reality techniques as a communication modality. These methods involve the use of devices such as displays, projectors, and augmented reality headsets to transmit information to humans. The use of these concepts and technologies for these purposes appears to be effective in conveying information about the task as well as the robot's intentions, resulting in a safer and more productive environment. At the moment, the implementation of this modality has the disadvantage of being more expensive than other modalities, and in some applications, it is unable to provide a comfortable user experience.

This work intended to provide an approach for implementing a set of nonverbal cues for use in collaborative manipulation and assembly tasks, primarily using the ROS-based motion planning framework MoveIt. In total, eight nonverbal communication cues were developed, four of which are reference/pointing cues and the other four are symbolic cues. Furthermore, the ROS4HRI framework was used in conjunction with the MoveIt framework to implement a potential approach to human collision safety in human-robot collaborative environments.

To assess the effectiveness of the implemented cues in conveying their true intent, a task was designed that required the robot to instruct the human operator to perform actions that it itself could not perform. The task was recorded in a video format using a virtual environment and divided into four parts based on the individual instructions provided by the robot. Twenty-one human participants watched the task recording and completed a questionnaire to assess their understanding of the cues.

The questionnaire responses indicate that the cues can convey their intent with reasonable quality. Move A, which instructed the human operator to pick up the object "board A" and place it in the specified location, received the lowest average level of understanding rating of 6.62 on a scale of 1 to 10. The majority of participants correctly identified the cue's true intent, so the rating is thought to be heavily influenced by the misidentification of the cue "Point To Human" given the higher "Challenging Cue" reports compared to the other cues in the instruction. Move B, which required the human operator to rotate the object "board A" in the intended direction, received a slightly higher average rating than Move A of 6.71, but had the most divided/spread votes about their level of understanding of the cues. Given the fact that at least half of the participants correctly identified the true intent of the cue while half reported having difficulty perceiving it, it is believed that the robot's unnatural arm position during the cue execution could have played a

significant role in influencing the perception of the cue. Move F, which instructed the human operator to secure "board A" in the testing station by pressing its side buttons, and Move H, which instructed the human operator to release "board A" from the testing station by pressing its side buttons, received average ratings of 8.43 and 8.76, respectively. Both of these cues were very well perceived in comparison to the previous two, and they were also the only ones inspired by tool behaviour and performed by the robot in a way that a human could not, by leveraging the robot's attributes such as the wrist's rotation degrees. This demonstrates that, while humans are a great source of inspiration for implementing nonverbal cues, referencing behaviour from other sources, such as tools, can lead to positive communication results. Only 5 of the 21 participants reported that the arm's movement influenced negatively the perception of the cues, with a greater impact on reference/pointing cues and negligible on the rest. This suggests that, while not ideal, the OMPL default planner can generate trajectories that could be used in these types of applications. These results, however, do not account for the impact of the path on the time it takes the human to identify the intent of the cues.

These results, combined with the fact that the participants were given no context about the task other than the fact that it would be a collaborative task with a robot capable of performing undisclosed nonverbal cues, demonstrate that the tools used in this work can, to some degree, provide the resources and capabilities for implementing understandable means of communication for a robot. That being said, while the results obtained with the use of the implemented cues in this study were not negative, they cannot be attributed solely to the cue because the perception of the cues using the same implementation approach may differ when using for example a robot with a different morphology than the one used in this case. Furthermore, the results were also analysed in terms of the participants' working industries as well as their proficiency in manipulating and assembling electronic devices. Although the number of participants in each category is insufficient to draw firm conclusions, they do indicate that the greater the familiarity with the subject matter of the task at hand, the greater the understanding of the cues performed.

## 6.2 Future Work

This approach, while functional as suggested by the results, can still be vastly improved. Here are some considerations for future work:

- Some questionnaire participants reported that only after answering the questions about the robot's first instruction did they begin to understand how they were supposed to analyse the robot's movements. This fact may have had a negative impact on the reported level of cue understanding. This problem was anticipated and attempted to be avoided by including a description at the start of the questionnaire that vaguely explained how the questionnaire worked without providing specifics

about the task and the cues. This approach did not appear to be sufficient. To address this issue in future tests, the questionnaire description must be revised and improved. One extra approach would be to create more tasks in which the order in which the cues are performed varies, allowing ratings from the first performed cues to be discarded if they were influenced.

- Due to delays in availability, a real robot could not be used in this study. This made it impossible to test the effectiveness of the cues in a real-world environment, which could yield different results than those obtained in a virtual environment, as well as to test the implemented "Human Collision Avoidance" component in order to understand its potential and limitations.

- The motion planner used in this approach produced good enough trajectories for the cues to be understandable, but they do not have the best characteristics for this application, in large part to their randomness and unpredictability. The STOMP planner might generate better testing results if the previously encountered issues are sorted out. Given that the MoveIt framework supports the use of custom motion planners, there can also be implemented a motion planner that is more appropriate for communication applications.

- Some cues do not present the best design approach, as evidenced by the questionnaire results, such as the cue "Rotate Object", which generated more reports of "Challenging Cue" than the other symbolic cues. More research would be required for this cue to determine how humans expect the action to be conveyed by humans and possibly by the robot. Another example would be the pointing cues approach. Because of the way it chooses an end pose only considering the collision state and the fact that the end effector is oriented towards the target, it completely disregards other aspects such as if there is an object in the way between the end effector and the target, misreferencing the target in this case, or consider the perspective of the receiver of the cue in order to improve the chance of conveying the intent correctly.

- Although the cues can be considered dynamic since they can be used regardless of the object's location, for example, the task, however, was hardcoded considering its various stages. This may be a viable option if the environment is static or highly predictable, but the presence of humans in the task makes it unsuitable. The human may not understand what the robot communicated, requiring it to repeat the actions; the human may adapt quickly to the task and perform actions that no longer require explanation or execution by the robot; or, on the other hand, the human may perform the incorrect action, which may require additional indications by the robot. In light of these factors, research must be done about possible decision-making mechanisms and approaches.

- The state of the art research demonstrates that the robot's characteristics can influence the conveying quality of specific cues. In this study, a single robot was used to implement and test the cues, preventing the possibility of determining whether some of the approaches used would result in viable approaches when used by different robots. Applying and testing the same methods described in this work on various robots is required to fully understand the cues' true effectiveness.

# Bibliography


[1] Y. Choi et al. "Service robots in hotels: understanding the service quality perceptions of human-robot interaction". In: *Journal of Hospitality Marketing & Management* 29 (2020), pp. 613–635 (cit. on p. 1).

[2] J. Holland et al. "Service Robots in the Healthcare Sector". In: *Robotics* 10 (2021), p. 47 (cit. on p. 1).

[3] L. Tian and S. L. Oviatt. "A Taxonomy of Social Errors in Human-Robot Interaction". In: *ACM Transactions on Human-Robot Interaction (THRI)* 10 (2021), pp. 1–32 (cit. on p. 1).

[4] G. Litzenberger. *IFR publishes collaborative industrial robot definition and estimates supply*. 2019-01. URL: https://ifr.org/post/international-federation-of-robotics-publishes-collaborative-industrial-rob (cit. on p. 1).

[5] R. Salehzadeh, J. Gong, and N. Jalili. "Purposeful Communication in Human–Robot Collaboration: A Review of Modern Approaches in Manufacturing". In: *IEEE Access* 10 (2022). Conference Name: IEEE Access, pp. 129344–129361. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3227049 (cit. on pp. 2, 5).

[6] Y. Kong and Y. R. Fu. "Human Action Recognition and Prediction: A Survey". In: *International Journal of Computer Vision* 130 (2018), pp. 1366–1401 (cit. on p. 2).

[7] I. Horswill. "Polly: A vision-based artificial agent". In: *AAAI*. 1993, pp. 824–829 (cit. on p. 5).

[8] A. Giuliano et al. "Experiencing Real-Life Interactions with the Mobile Platform of MAIA". In: *The Biology and Technology of Intelligent Autonomous Agents: Proceedings of the NATO Advanced Study on the Biology and Technology of Intelligent Autonomous Agents*. Springer. 1995, pp. 296–311 (cit. on p. 5).

[9] N. Mavridis. "A review of verbal and non-verbal human–robot interactive communication". In: *Robotics and Autonomous Systems* 63 (2015), pp. 22–35 (cit. on p. 5).

[10] N. C. Krämer, A. v. d. Pütten, and S. Eimler. "Human-agent and human-robot interaction theory: Similarities to and differences from human-human interaction". In: *Human-computer interaction: The agency perspective*. Springer, 2012, pp. 215–240 (cit. on p. 5).

[11] S. P. Parikh, J. M. Esposito, and J. Searock. "The role of verbal and nonverbal communication in a two-person, cooperative manipulation task". In: *Advances in Human-Computer Interaction* 2014 (2014) (cit. on p. 6).

[12] J. Wang, A. Chellali, and C. G. Cao. "A study of communication modalities in a virtual collaborative task". In: *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE. 2013, pp. 542–546 (cit. on p. 6).

[13] H. H. Clark and S. E. Brennan. "Grounding in communication." In: (1991) (cit. on p. 6).

[14] A. Bonarini. "Communication in human-robot interaction". In: *Current Robotics Reports* 1.4 (2020), pp. 279–285 (cit. on p. 6).

[15] G. Buisan, G. Sarthou, and R. Alami. "Human aware task planning using verbal communication feasibility and costs". In: *International Conference on Social Robotics*. Springer. 2020, pp. 554–565 (cit. on p. 6).

[16] D. Silvera-Tawil, D. Bradford, and C. Roberts-Yates. "Talk to me: The role of human-robot interaction in improving verbal communication skills in students with autism or intellectual disability". In: *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2018, pp. 1–6 (cit. on p. 6).

[17] A. Taheri et al. "Human–robot interaction in autism treatment: a case study on three pairs of autistic children as twins, siblings, and classmates". In: *International Journal of Social Robotics* 10 (2018), pp. 93–113 (cit. on p. 6).

[18] S. Nikolaidis et al. "Planning with verbal communication for human-robot collaboration". In: *ACM Transactions on Human-Robot Interaction (THRI)* 7.3 (2018), pp. 1–21 (cit. on p. 7).

[19] R. Younes et al. "Automatic Verbal Depiction of a Brick Assembly for a Robot Instructing Humans". In: *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 2022, pp. 159–171 (cit. on p. 7).

[20] A. K. Singh et al. "Verbal explanations by collaborating robot teams". In: *Paladyn, Journal of Behavioral Robotics* 12 (2020), pp. 47–57. URL: `https://api.semanticscholar.org/CorpusID:227129129` (cit. on p. 8).

[21] H. Grice. "Logic and Conversation". In: *Syntax and Semantics* 3 (1975), pp. 43–58 (cit. on p. 8).

[22] Ș.-D. Ciocirlan, R. Agrigoroaie, and A. Tapus. "Human-Robot Team: Effects of Communication in Analyzing Trust". In: *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)* (2019), pp. 1–7 (cit. on p. 8).

[23] A. Mehrabian. *Nonverbal communication*. Routledge, 2017 (cit. on p. 8).

[24] J. Zhou et al. "Occupational noise-induced hearing loss in China: a systematic review and meta-analysis". In: *BMJ open* 10.9 (2020), e039576 (cit. on p. 8).

[25] C. L. Nehaniv et al. "A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction". In: *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.* IEEE. 2005, pp. 371–377 (cit. on pp. 8, 9).

[26] B. Gleeson et al. "Gestures for industry intuitive human-robot communication from human observation". In: *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE. 2013, pp. 349–356 (cit. on pp. 9, 16).

[27] S. Sheikholeslami, A. Moon, and E. A. Croft. "Cooperative gestures for industry: Exploring the efficacy of robot hand configurations in expression of instructional gestures for human–robot interaction". In: *The International Journal of Robotics Research* 36.5-7 (2017), pp. 699–720 (cit. on pp. 10, 16).

[28] A. D. Dragan, K. C. Lee, and S. S. Srinivasa. "Legibility and predictability of robot motion". In: *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE. 2013, pp. 301–308 (cit. on pp. 10–12).

[29] R. M. Holladay, A. D. Dragan, and S. S. Srinivasa. "Legible robot pointing". In: *The 23rd IEEE International Symposium on robot and human interactive communication*. IEEE. 2014, pp. 217–223 (cit. on pp. 11, 54).

[30] F. Stulp et al. "Facilitating intention prediction for humans by optimizing robot motions". In: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2015, pp. 1249–1255 (cit. on p. 11).

[31] D. Bortot, M. Born, and K. Bengler. "Directly or on detours? How should industrial robots approximate humans?" In: *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE. 2013, pp. 89–90 (cit. on pp. 11, 12).

[32] M. Huber et al. "Human-robot interaction in handing-over tasks". In: *RO-MAN 2008-The 17th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE. 2008, pp. 107–112 (cit. on p. 12).

[33] C. Bodden et al. "Evaluating intent-expressive robot arm motion". In: *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2016, pp. 658–663 (cit. on pp. 12, 54).

[34] A. Saran et al. "Human Gaze Following for Human-Robot Interaction". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ISSN: 2153-0866. 2018-10, pp. 8615–8621. DOI: 10.1109/IROS.2018.8593580 (cit. on p. 13).

[35] L. Paletta et al. "Estimation of situation awareness score and performance using eye and head gaze for human-robot collaboration". In: *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. 2019, pp. 1–3 (cit. on p. 13).

[36] O. Palinko et al. "Robot reading human gaze: Why eye tracking is better than head tracking for human-robot collaboration". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 5048–5054 (cit. on p. 13).

[37] B. Amaro et al. "An Approach to Behavioural Distraction Patterns Detection and Classification in a Human-Robot Interaction". en. In: (2018) (cit. on p. 13).

[38] H. Ling, G. Liu, and G. Tian. "Motion Planning Combines Psychological Safety and Motion Prediction for a Sense Motive Robot". In: *ArXiv* (2020-09). URL: https://www.semanticscholar.org/paper/36026cc22afdc3e0e2fbcd86df158d33c7d4df38 (visited on 2022-12-23) (cit. on p. 13).

[39] A. Kshirsagar et al. "Robot gaze behaviors in human-to-robot handovers". In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6552–6558 (cit. on p. 13).

[40] T. Faibish et al. "Human preferences for robot eye gaze in human-to-robot handovers". In: *International Journal of Social Robotics* 14.4 (2022), pp. 995–1012 (cit. on p. 13).

[41] Y. Terzioğlu, B. Mutlu, and E. Şahin. "Designing social cues for collaborative robots: the role of gaze and breathing in human-robot collaboration". In: *Proceedings of the 2020 ACM/IEEE international conference on human-robot interaction*. 2020, pp. 343–357 (cit. on pp. 13, 14).

[42] M. Dianatfar, J. Latokartano, and M. Lanz. "Review on existing VR/AR solutions in human–robot collaboration". In: *Procedia CIRP* 97 (2021), pp. 407–411 (cit. on p. 14).

[43] G. Bolano, A. Roennau, and R. Dillmann. "Transparent robot behavior by adding intuitive visual and acoustic feedback to motion replanning". In: *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2018, pp. 1075–1080 (cit. on p. 14).

[44] G. Bolano et al. "Transparent robot behavior using augmented reality in close human-robot interaction". In: *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2019, pp. 1–7 (cit. on pp. 14, 15).

[45] R. K. Ganesan et al. "Better teaming through visual cues: how projecting imagery in a workspace can improve human-robot collaboration". In: *IEEE Robotics & Automation Magazine* 25.2 (2018), pp. 59–71 (cit. on p. 15).

[46] G. Tsamis et al. "Intuitive and Safe Interaction in Multi-User Human Robot Collaboration Environments through Augmented Reality Displays". In: *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. IEEE. 2021, pp. 520–526 (cit. on pp. 15, 16).

[47] M. Walker et al. "Communicating robot motion intent with augmented reality". In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 2018, pp. 316–324 (cit. on p. 16).

[48] M. Quigley et al. "ROS: an open-source Robot Operating System". In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009, p. 5 (cit. on p. 17).

[49] S. Macenski et al. "Robot Operating System 2: Design, architecture, and uses in the wild". In: *Science Robotics* 7.66 (2022), eabm6074 (cit. on p. 17).

[50] Github. *moveit*. `https://github.com/ros-planning/moveit` (2023/04/01) (cit. on p. 18).

[51] *MoveIt Robots*. URL: `https://moveit.ros.org/robots/` (cit. on p. 18).

[52] R. Ros et al. "ROS4HRI: Standardising an Interface for Human-Robot Interaction". In: () (cit. on p. 19).

[53] Github. *abb_robot_driver*. `https://github.com/ros-industrial/abb_robot_driver` (2023/11/28) (cit. on p. 24).

[54] *Dual-arm YuMi® - IRB 14000*. URL: `https://new.abb.com/products/robotics/robots/collaborative-robots/yumi/dual-arm` (cit. on p. 24).

[55] Y. Mohamed and S. Lemaignan. "Ros for human-robot interaction". In: *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2021, pp. 3020–3027 (cit. on p. 39).

[56] S. Liu and P. Liu. "Benchmarking and optimization of robot motion planning with motion planning pipeline". In: *The International Journal of Advanced Manufacturing Technology* 118.3 (2022), pp. 949–961 (cit. on p. 44).