Grant agreement No: 101017008

Harmony

Assistive robots for healthcare

Enhancing Healthcare with Assistive Robotic Mobile Manipulation

(HARMONY) | H2020-ICT-2018-20| RIA

Start of the project: 01.01.2021
Duration: 42 months

Deliverable Number	8.4
Deliverable Name	Validated multimodal universal dictionary of robot intent locomotion behaviours
WP Number	8
Lead Beneficiary	UT
Dissemination Level	Public
Internal Reviewer	IDM
Due Date	31-12-2023
Date of Submission	22-12-2023
Version	1.0

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008

Revision History

Version	Date	Author(s)	Comments
0.1	20-12-2023	Evers, Schadenberg	First draft
1.0	21-12-2023	Evers, Schadenberg	Final version

Table of Contents

1. Summary

This deliverable concerns the validated multimodal universal dictionary of robot intent locomotion behaviours. As such it addresses three elements: The library of robot movement and multimodal behaviours; A validated set of response behaviours that the robot deploys to recover from social errors the robot has made during its navigation through the hospital; and the 'Universal Translator', which is the capability of the robot to learn over time which behaviours to adopt in different contexts. The efforts to develop each of these elements are

described, and when completed they are fully reported in the most current update of this deliverable.

2. Introduction

The Harmony project use cases concern a robot's deployment in a lab environment where it supports laboratory personnel in unpacking, handling, sorting and storing blood samples; and a robot's ability to pick up and deliver blood samples throughout a hospital while encountering and interacting with different target user groups (patients, staff, visitors).

It is a complicated challenge to determine when a robot should communicate what. If a robot has to urgently transport bioassay samples from the sampling room to the laboratory, it will need to do so in an efficient manner. However, while it should move efficiently through the hospital, it should not disturb or alarm staff or patients. A robot's appearance and behaviour need to intuitively convey to the people it encounters what its purpose is and not disrupt the workflow of the hospital. But when should a robot communicate what? If people are in the way should it sound a loud alarm? Probably not. If it needs to be quick, should it speed through the hospital corridors and cut other people off? This is also not ideal.

In the most ideal situation, a visitor, patient or staff member only needs to have one glance at the robot and feel that the situation is 'normal' or socially acceptable. It should be clear what the robot is trying to do and what it can be used for or not. In this, we try new and creative forms of interaction. For instance. If the robot can change its shape and colour it can subtly indicate that it needs to 'squeeze through' a busy area. If people would like to engage with the robot but it needs to carry out a task it could display some humour in its behaviour to show that it needs to do its task and cannot stop to play. However, when it is idle it could respond to people's wish to engage and have short or fun interactions with them.

This shows the importance of a robot being able to communicate its intentions effectively. This deliverable describes our efforts in the Harmony project to design such intentional behaviours for the robot, how we have validated the effectiveness of the behaviours and how we have realised the robot's capability to translate its behaviour for different people and places in the hospital.

In the current version of the deliverable, we describe our methods and goals. The updated deliverable in June will contain the results of the universal translator and behaviour library of the different communicative behaviours.

3. Harmony Robot Behavioural Library

In the multiple studies that were carried out as part of the Harmony project, an extensive library of possible robot behaviours was developed. Some of these responses use a single expressive modality (unimodal behaviour) and consist for instance of possible sounds a robot can adopt in different situations, other responses are multimodal and may combine changing its colour, shapes and movement behaviours to indicate to people that it needs to pass. In deliverable 8.3 and forthcoming 8.5, we detail the different studies that were carried out and behaviours that were designed for the Harmony robot's morphology and implemented either on the physical platform or in VR simulation. All these possible behaviours and combinations of modalities are collected in the **Harmony Behavioural Library** which consists of all the designed behaviours of the robot be it auditory, arm movements, full body movements, colour, shapeshifting and velocity. The Behaviour library is being updated and expanded constantly, it can be found in:

https://docs.google.com/spreadsheets/d/1uKyzhl69mATGntdShP79BJPkv1fzkoMTXr4cuBQpBgo/edit#gid=0.

Please note that an updated and complete Behaviour Library will be included in Deliverable 8.5 in June 2024.

4. Validated set of social recovery behaviours for the robot

A subset of these behaviours, namely the behaviours that were designed as part of the recovery strategies the robot deploys when it makes a mistake when interacting with people, are validated in user studies. Through the studies, of which some were completed in 2023 and some will be finalised in 2024, the behaviours are shown to effectively allow the robot to recover from social errors.

When a robot navigates through a hospital, it will traverse through many different areas, these can be hallways, public waiting areas, canteens, laboratories and so on. In different areas, the robot will encounter different people. As it travels the hallways of the laboratories the robot will encounter hospital staff who are carrying out their jobs and are moving from one room to the other. In public areas, the robot may encounter patients or visitors. Whereas hospital staff may be aware of the robot's function and could be trained to optimise hospital logistical processes that the robot is part of, this is different for patients and visitors. Patients who are waiting to be seen by a doctor or to have bioassay samples taken will not necessarily understand what the robot's purpose and function are. The same is true for people visiting patients in the hospital. Because of this, the robot will need to appropriately respond to different types of people in different areas of the hospital.

Just like any intelligent system, the robot will make mistakes. Sometimes these mistakes may be because of an error in perception (it may detect an obstacle that is not there), and sometimes this may be because of an error in execution (it may incorrectly grasp something). While all of these errors concern the robots' practical functioning, they will be experienced by people in a social way. When a robot doesn't grasp something right or accidentally bumps into something, people may see the robot as clumsy. When it approaches people too closely or blocks a hallway accidentally, people may feel the robot is rude or impolite. As such, the robot is sure to surprise, obstruct, or annoy people. Therefore, the robot must have the ability to make social reparations. When it makes a mistake, for instance by blocking a hallway, it should be able to recover from that mistake. There are many strategies that people deploy, they may say sorry, display some apologetic behaviour, blame the other, offer to pay for damage and so on. When a robot makes a social mistake, however, there is as yet no knowledge of what effective recovery strategies and behaviours are. In this deliverable, we report on a set of designed behaviours and offer support for the extent to which these are validated, and proven to be effective in making social reparations after making a mistake when encountering people in a hospital.

The social recovery behaviours were informed by a literature review (which can be found in D8.2), a study in which inspiration for robot movement behaviours was elicited from actors and dancers (which can be found in D8.3) as well as a study that explored appropriate robot sounds (also reported in D8.3). A set of vignettes that display the robot's possible recovery behaviours will be validated in studies to be carried out in the period January-April 2024. The results thereof will be reported in D8.5

5. The 'Universal Translator'

Even once we have shown which robot recovery behaviours effectively allow a robot to 'apologise' and repair the human-robot interaction, the robot will need to deploy different strategies in different situations. When blocking a person with a transport trolley in a hallway of the hospital where there are no patients, a different response by the robot is appropriate compared to when the robot blocks a person from entering the hospital shop. The right response is context-dependent.

Because there are not enough resources nor time available within the lifespan of the Harmony project, we instead develop an ability for the Harmony robot to learn over time (through reinforcement learning) which behaviours are effective in which contexts. We will show that it is possible for the robot to detect for instance a certain setting (for instance public parts of the hospital, work spaces etc); a type of mistake it made (it came too close or blocked someone, etc) or the type of user it is likely encountering (hospital staff or patient). For a limited set of contexts and user characteristics, we will show how the robot can try different error recovery strategies and automatically learn which are likely to have the desired positive effect. We call this the 'Universal translator' as the robot learns over time how to translate its behaviour to apply to different situations.

6. First iteration of the Universal Translator

In this section, we will describe the first efforts in realising the universal translator, embedded in learning social navigation in hallways. To this end, we aim to consider people's theory of mind in terms of what people predict how the robot will behave in the near future, as well as the behavioural responses of people to the robot (distance to the robot, and (predicted) trajectory of people). Based on this information, the robot is to learn when certain hallway behaviours are to be performed (e.g., "squeeze through", or indicate a turn). To achieve this, a generative model generates anticipated trajectories based on observed data and contextual information, while the deep learning model infers the actions individuals are likely to take based on these trajectories.

This first iteration of a universal translator is currently in development and we will report on the results in D8.5, along with the results of the final universal translator. In the remainder of the section, we outline how we aim to approach the first iteration.

6.1 Data Collection

The initial step in developing a socially aware navigation system for robots involves collecting human trajectory data. The THOR dataset (Rudenko et al., 2020) provides indoor footage of a fixed space. In some scenes, additional obstacles were placed in the environment, allowing for variations in the environmental conditions. There were also people placed in the scene, some of whom had fixed paths and tasks, whereas others were more freely roaming through the scene. Both video and Ros Bag files are included in the dataset and give plenty of data to work with. An additional remark is that the data was recorded through a small forklift Linde CitiTruck robot with a footprint of 1.56 x 0.55 meters and 1.17 meters high. The main focus is on using this dataset for the fine-tuning of the generative model for indoor HRI. In case this dataset is not sufficient for any reason, there are quite a few alternatives: The OpenTraj (Amirian et al., 2020) dataset contains comprehensive information regarding human trajectories across various outdoor spatial contexts. The MuSoHu dataset (Nguyen et al., 2023), which contains scenes from a first-person point-of-view video and lidar data, is compatible with ROS. Or the SCAND collection (Karnan et al., 2022), which contains footage that was recorded by a remote-controlled Clearpath Jackal and a legged Boston Dynamics Spot.

6.2 Defining Social and Cognitive Rules

To enhance the predictive capabilities of our model, a set of fundamental social and cognitive rules related to navigation will be defined. These do not need to be explicitly defined, as they can be learned from training data. While in many cases, hand-crafted rules may show promising results, it is hard to capture the complexity and intricacies of human navigation. These rules provide essential context and guidance for the generative network. By using AIRL (Fu et al., 2018) algorithms to learn the fundamental rules of human navigation, this model can be trained on people’s trajectories to yield the reward function that governs their behaviour.

6.3 Building the Generative Network

6.3.1 Preprocessing

Before inputting data into the generative network, some preprocessing is performed on the human trajectory dataset. This includes feature extraction and the creation of a structured dataset. Assuming that there are coordinates of each person's trajectory in the selected dataset, each one of these points should be able to be defined as a potential goal point depending on the time of observation of the trajectory. This claim is valid during training and inference of the model, with the aim being, the inclusion of trajectory mid-points across the overall trajectory.

Although the goal points don't need to be explicitly defined in the dataset, it is important to be able to locate the people in the scene. By locating where people are in the scene, you can trace their trajectory through the environment and arbitrarily set a frame to such a state. Moreover, it is critical to be able to determine the kinds of information that can be used as input for the model. An individual's position in the scene can be one of the data points, but more can be extracted. Velocities can also be determined by looking at the distance travelled between frames.

6.3.2 Generative Network Architecture

We construct a Deep Neural Network (DNN) based on an encoder-decoder architecture. This network takes the preprocessed data from the human trajectory dataset and the defined social and cognitive rules, as input. The rules would not necessarily be hard coded into the architecture, but given the inferred reward function from the AIRL model, it can be used to assist the model's forecasted trajectories. The encoder component processes the input data, capturing critical features of human behaviour. In contrast, the decoder generates predicted trajectories based on these features and rules.

6.4 Robot Policy Learning

After constructing and training the human policy generative network, we can assume that we have a reliable way of accounting for the human policy in the robot's responses and reactions. The goal is not to make the robot purely reactionary, but socially aware instead. It is possible to have a model that takes into account the goals of the robot, as well as the presence and behaviours of other agents in the environment.

The generative network serves as a human policy generator and describes how a human is likely to move in a given space based on their previous actions and the established social and cognitive rules. Therefore we can expect our robot to take human social norms into account.

A separate deep-learning model for robot policy learning will be implemented. This model aims to optimize the robot's behaviour within a social context. It employs multi-objective optimization techniques, where the robot's policy generation takes the human's policy into account before generating actions for the robot. This approach aims to align the robot's behaviour with both task objectives and social norms. Deliverable 8.5 due June 2024 will report the finalised universal behaviour translator module based on DNN and the robot's ability to learn which behaviours to deploy over time.

7. Learning from Demonstration (LfD)

7.1 Human-robot teaming: Application of LfD

In human-robot cooperation and collaboration, workers work with a robot on a shared task in a shared collaborative space [,]. Collaborative robots (or cobots in short) are specifically designed to work safely in close spatial and temporal proximity with workers []. The benefit of these robots is that a worker can work alongside, or together with, the cobot, and leverage the strengths of both; e.g., the repeatability and precision of robots, and the creativity, adaptability, and flexibility of people.

To enable more flexible use of cobots, they often feature programming interfaces that require little programming knowledge. A worker with little programming knowledge can then program the cobot without the intervention of the cobot's supplier. For instance, to program the cobot to perform a new (sub)task, or to adapt the cobot to the preferences of that specific worker. Workers then can delegate subtasks to the cobot that they find physically or mentally stressful, whilst doing the subtasks that the cobot cannot do, they find enjoyable, or that give meaning to their work.

There are a variety of methods for workers with limited knowledge of programming to program robots to program. One such way is through the learning from demonstration (also referred to as apprenticeship learning, programming by demonstration, learning by demonstration, behavioural cloning, or imitation learning) paradigm, where a person provides demonstrations to the robot on the task from which the robot then derives a policy to execute the task itself [, ,].

LfD is particularly compelling for tasks that are difficult to script or frame as an optimization problem but can be demonstrated []. The demonstration techniques used within this paradigm broadly fall within one of three categories []. Firstly, *kinesthetic teaching*, where the worker physically moves the robot through the desired motions. Secondly, demonstrating the desired motions through *teleoperation* of the robot through an external device (e.g., a joystick). And lastly, the robot could passively *observe* a worker performing the task through its sensors. The resulting motion trajectories generated through any of the three techniques can be further optimised by smoothening the motions or using other post-processing techniques.

Given that LfD relies on demonstrations by a person, the paradigm is sensitive to who is teaching the robot. One teacher may be better at teaching the robot than another, but teachers may also encode their preferences in how the robot should execute the task. [] compared teaching the robot with observing the robot learn by itself. The robot had to learn a reaching task, and the authors found that teaching a robot improves the teacher's trust in the robot and they considered the robot to have more humanlike motions compared to when observing the robot learn. [] found that when a robot that has been taught by a person fails at its task, those teachers then have a lower impression of both the robot and themselves, and trust themselves and the robot less as well as believe that others will trust them less.

Compared to tasks where robots and humans have separate work spaces, the role of the teacher in the execution of the task may be exacerbated in human-robot cooperative or collaborative tasks, as the worker may not have been the one who taught the task to the cobot. The preferences of the teacher then can have a direct effect on how other workers will need to perform the task with the cobot. This can result in physical mismatches where the cobot's motions are relative to the teacher's bodily dimensions, the speed at which the task needs to be performed, or the order of subtasks. In particular in tasks that allow for the encoding of preferences (i.e., for which there is no single optimal way to perform the task).

7.1.1 Problem statement

A technological solution could be that each worker trains the cobot on their own so that it knows their preferred way of working. Or that the cobot learns the preferences (e.g., working speed of a worker) of individual workers throughout the interaction and adapts accordingly.

However, from a practical point of view, changes in cobot behaviour are problematic for the risk assessment, which is currently the largest barrier to the adoption of cobots []. While the cobot itself may still be safe to use, the process in which it's embedded may need to be reevaluated. For instance, while the cobot may not severely harm a person directly (e.g., stop when it comes into contact with a person), it could drop an object that then hurts a person, interact with its surrounding environment to create a hazard or cause a failure in the process down the chain. Whenever the cobot's behaviour is changed, a new risk assessment may be required. A risk assessment is a time-consuming and complex matter that may in turn undermine the benefits of an LfD system.

Because a continuously changing cobot may be problematic, it may not be desirable to have workers program their own way of working with the cobot, or have the cobot adapt its behaviour to the worker. Thus, we need to understand the effect of who is teaching the cobot has on the task execution by the workers, including the teachers, thereafter.

7.1.2 This study: Self vs Other training

In this study, we investigate the effect of who taught the cobot on the task performance, the participants' trust in the cobot's ability to provide aid in the task, and fluency of the participant in performing the task. Our study is carried out within a hospital laboratory setting where they process test samples for bioassay. These test samples need to be unpacked, registered, sorted and pre-processed so that they are ready for the respective clinical test []. This is a task that is associated with causing work-related musculoskeletal disorders due to the repetitiveness of the task, as well as the potential for coming into contact with the, potentially hazardous, contents of the test samples []. For instance, when a lid is not closed correctly or there is a spill.

While a cobot could alleviate these issues, it is currently a difficult task to fully automate due to the complexity of the required manipulations and the need to deal with non-standard samples. Cooperation between worker and cobot could also address the issues experienced by the laboratory workers, where the cobot carries out the repetitive tasks and hands over the test samples for further inspection and sorting on priority by the worker. Within this setting, LfD could be utilised to deal with changes in the shapes of the test samples and the boxes that contain them or to readjust the cobot's effort to a different task within the laboratory that requires additional labour.

7.2 Materials and Methods

7.2.1 Experiment Design

The study was set up as a 1 by 2 within-participants design. The independent variable was the **training**, which was either done by the participants themselves or by somebody else (i.e., another participant). Participants participated in one session, divided in two parts, where the first part constituted training the robot to do a cooperative task for handling test samples. In the second part, participants would perform the task with the robot that they trained, or the same robot that was trained by another participant (training from participant 1). The order for the second part was randomised, with six participants starting with the training of themselves, and the other five, with the training of another.

7.2.2 Participants

Participants were recruited through convenience sampling at the University in Twente. This resulted in 11 participants (2 men, 9 women), aged between 25 and 53 (M = 38.8, SD = 11.8). Of the participants, 8 were right-handed, 2 left-handed, 1 mixed-handed (it differs), and none ambidextrous. The distribution of participants' frequency of hand dominance was: 4 right-handed and 1 left-handed per group and 1 mixed handed in the group of self-training first. On average, participants were 171 cm (SD = 6.54). Out of 11 participants, 6 had a technical background, 4 had a non-technical (i.e business management, communication) and 1 had neither. On average, participants gave a score of 3.91 (SD = 2.21) when asked about their experience in programming in general and a score of 1.73 (SD = 1.85) when asked about their experience in programming robots (1 = None to 7 = High experience). The ethical committee of the University of Twente reviewed this study, filed under reference number 240336.

7.2.3 Materials

For this study, we used the Franka Emika Research 3 robotic arm, equipped with a tray that could hold test samples (see Figure 10). Participants could press an interactive button that would either start/stop recording a trajectory, or play back a stored trajectory. The Point-To-Point method was used for the trajectories of the participants. It consists of recovering the last position registered for each training motion and sending it to the robot arm which will move to that position by itself without taking account of the participant trajectory. This method was selected due to connectivity issues and some limitations in the robot programming as it was still in the process of being tuned. We used an overhead camera to record the interaction of the participants with the robot arm.

[Picture franka robot (from overleaf)]

7.2.4 Experimental task and setup

Participants had to perform a test sample processing task, inspired by the preprocessing of test samples in a hospital laboratory (also see [,]). The task starts with a tray filled with two types of test samples. The tray was the same for each participant. Participants could see what type a sample belonged to by looking at the colour of the sticker on the lid. The goal of the task was to process each of the test samples in this initial tray. Blue samples had to be scanned at the scanner and then stored in a tray behind the participants. Green samples had to be disposed of in a box for internal transportation. We asked the participants to stand so that individual differences in height could play a role.

[Picture setup (from overleaf) with caption : "Experimental setup with the trays with mixed test samples on a brown table, red and blue samples that have already been preprocessed by the participant on the bottom table, and processed green samples deposited in the box on the robots' table. The red dot is the interaction button, and left of it was the scanner."]

7.2.5 Procedure

Teaching

Participants were first verbally informed about the study, asked to read the information letter and had to sign the informed consent form. Then, the experimenter had the participants read a short explanation of the context and watch two videos: the current processing of test samples in a hospital lab and a robot that handles test samples coming in through the internal tube system of the hospital. Because the task was not very realistic, this was to get participants in a frame of mind that this is not a purely fictional task. Next, the participants were asked to take place in the setup. The experimenter explained the task in more detail, the constraints for the task and the fact that they would replay their training and the training of another without telling them which one they would start with.

Next, the experimenter would show participants how they could teach the robot. First, the participants had to learn how to handle the robot controls with the experimenter giving some recommendations regarding safety and manipulation of the robot. Once they felt comfortable, the participants were given one trial to show how the recording works. Then, the training started. Participants had to demonstrate each of the actions in sequence. Each sequence was independent from the others so that each action could be repeated and replaced if necessary. To start or stop the recording of the motion, they could press the interaction button. The button would change colour from green to orange, meaning that the robot was currently working. To finalise the teaching, participants were given the options to either keep their training or redo some or the totality of the trained actions. After the training, participants were asked to fill a short survey.

Task Application

For the second part of the session, the experimenter would inform the participants whether they would first perform the task with the robot that they taught or that somebody else taught. For the other training, the experimenter would mention that the robot was trained for the same task, but that its actions may be different from the ones they trained. This was so that participants could anticipate the unpredictability of the robot's motions and actions. Before the other training, the experimenter would also explain in detail to the participants the order of the actions and the method used in the other training to handle the vials to reduce any confounding effect of unpredictability. After each training, the participants filled in a questionnaire. At the end, the experimenter debriefed the participants and answered additional questions that may have come from the participants.

7.2.6 Measures

System Trustworthiness

We measured the trustworthiness of the robot through the System Trustworthiness Scale [STS], []. The scale consists of 15 items for three subscales: performance, purpose, and process. The order of the items was randomised and participants responded to them using a 5-point response scale (1 = Strongly Disagree to 5 = Strongly Agree). The STS was used after performing the first and after the second task.

Preference

On a 7-point Likert scale, we asked participants which training they preferred, along with an open question on why.

Training experience

Self-confidence and *experience in programming* can influence how participants feel after programming the robot []. Those with low self-confidence or low experience may be more afraid of the robot after teaching it than those with higher scores. After the training session, we therefore asked a couple of questions related to their experience, including their self-confidence in their training of the robot and their experience in programming in general.

7.4 Results

7.4.1 Training

On average, participants answered the question `How confident are you in training this robot?` with a score of 5.27 (*SD* = 1.01), showing confidence in manipulating the robot.

To the open question `How did the training of the robot go?`, most participants found the experience interesting. Most answers such as "went very well", "intuitive" or "great" showed that participants were able to understand easily how to manage the robot. One participant also answered: "the robot could be of very help when working in a laboratory". However, a few participants also had difficulties with the scenario with answers such as "it is difficult to imagine what is reasonable training" and "Requires maybe additional perception of the world constraints and objects".

7.4.2 Trustworthiness

On average, participants have scored performance scales for their own training a little higher than for the other (For self, M = 3.71, SD = 1.10; For other, M = 3.64, SD = .737). Purpose scales got an average score of 4.2 (SD = .732) for self training and 3.76 (SD = .742) for the other training. The process scales got an average score of 3.8 (SD = .759) for self training against 3.65 (SD = .877) for the other training.

In the comparison between the use of self training and the use of the other training, the Paired T-test analysis didn't show any significant results for any of the three scales: Performance, Purpose and Process ($p > 0.6$).

7.4.3 Preference

To the question `How different did you experience the two training? (1 = no difference), participants gave an average score of 5.18 (SD = 1.33). The T-test analysis showed significant results ($p <$ 1e-06), meaning that participants really did feel a difference between the two training sessions. The cohen analysis showed a small effect size (*effsize* = .228). The T-test analysis to compare the training order gave no significant result ($p > 0.9$), showing that training order during the experiment didn't affect the results.

To the question "Which training would you prefer to work with ?" (1 = self, 7 = other), participants gave an average score of 3.09 (SD = 2.51), showing a tendency to favour their own training. When separated per group, the self-first group showed an average score of 2.33 (SD = 1.97) and the other-first group an average of 4.0 (SD = 3). The *non-parametric t-test* SignTest analysis (with μ = 4) didn't show significant results ($p > 0.2$). The cohen analysis showed a moderate effect size (*effsize* = .657). The Wilcox test analysis was used to investigate the effect of the training order during the experiment. It showed no significant result (*p* = .388), meaning that the training order had no effect on the results.

7.4.4 Open Questions

In general, to the question `Can you describe your experience performing the task with the robot?`, most participants tended to be more positive towards their own training.

Self-training

Most participants considered their own training as interesting and easier to use. They used words such as "better understandable", "more logical", "exciting" or "fun". However, a few participants answered that the experience was "not good"or that the programming could use some tweaking". This was due to some incidents with the motions and the joints angle. But one of those participants also indicated that with "some more trial and error", it could be better. Finally, two participants said their training was "slow".

Other-training

Some participants seemed to show some confusion and/or considered the other training as "not handy" or "inconvenient". Out of them, a few didn't like the movements or the robot in their personal working space and its orientation. One of them considered it "of no help". However, some participants also affirmed that the robot's performance to manage its task was okay. One particularly answered "Robot performed all the tasks seamlessly and in the correct sequence as instructed".

Preference

To the question `Could you elaborate on your preference?`, most participants preferred their own training in terms of predictability, positioning and efficiency. However, a few participants also considered that the other training motions seemed more efficient than theirs. One participant doesn't seem to show any preference with the answer: "one can adapt easily also to the programming of others".

7.5 Discussion, Limitation & Future Research

The investigation into the role of the teacher in Learning from Demonstration (LfD) for human-robot cooperative tasks has yielded several notable insights. However, it is important to note that most survey data did not provide significant quantitative results. Therefore, most of the conclusions and assumptions are derived from qualitative data obtained through open-ended questions. Additionally, connectivity issues between the robot and the program posed significant challenges. This section delves into these findings, their implications, potential impacts, and areas for future research.

7.5.1 Key findings

The study found qualitative indications that individuals seemed to trust and prefer their own training over the training from another individual in terms of performance and safety, though the survey did not yield statistically significant data. Observations also showed that the different robot trainings showed variations in task execution, due to subjective differences in teaching styles and preferences. This would suggest that training is highly dependent on individuals, and familiarity with the training process can enhance trust in the robot's performance. As trust is vital for the acceptance of robotic systems in work environments, future systems should consider integrating individual's preferences as well as feedback mechanisms to foster trust, even more so when users are not directly involved in training the robot.

[relation experience and training]

However, participants' observations also showed that experience and knowledge in handling tasks had a real impact on the participants' robot training, with some individuals having difficulties to imagine the tasks or remembering them during the replay, probably due to the lack of experience in an unusual task. This would suggest that qualified and experienced people in the trained tasks should be incorporated in the training process.

Finally, qualitative data indicated that LfD could handle variations in task, though no significant quantitative evidence supported this adaptability. Participants observed that LfD allowed robots to adapt to task variations, reflecting its potential flexibility. This adaptability could be crucial for dynamic environments requiring frequent adjustments. Further research is necessary to quantitatively validate the effectiveness of LfD in handling such task variations and reducing the need for reprogramming.

7.5.2 Limitations

First and foremost, given the absence of significant survey results, most conclusions in this study are drawn from qualitative responses to open-ended questions. This study provided rich insights into participants' experiences and perceptions but lacked the necessary quantitative results. Future studies might benefit from a more robust approach, with more participants and further quantitative measures to derive more robust conclusions on the effectiveness, applicability of this LfD method

LfD assumes the robot learns from observations and in that sense is an agent themselves. In our case, the robot was merely playing back what the user did during training and had little agency there. It also did go back to its home position by itself between each action and the trajectory designed by the participants couldn’t be replayed. This Point-to-Point (PtP) approach was decided after communication troubles (drops of data) and programming limitations caused interruptions in robot operation, inconsistencies in the execution of tasks, and impacting the data collection, affecting the overall experience and evaluation of the experiment. To be precise, connectivity issues led to difficulties in ensuring stable communication between the robot and the programming side. This affected the robot’s ability to perform tasks accurately and consistently, and the participants’ ability to manipulate the robot arm due to a loss of robustness in the robot’s motions. It also made it necessary for the experimenter to re-launch the programs multiple times per participant due to the many drops of robot’s operations.

From this, we could conclude that it may be difficult to encode personalised motions in learning from demonstration. You may need to only store endpoints and the robot generates its motion itself, rather than use the motion generated by a person. However, that would limit how much they can personalise the LfD themselves. For instance, the robot could generate motions that may feel unsafe for the person by going right in front of a person's nose, rather than use the motion made by the person. In the end, these issues highlight the importance of reliable communication frameworks in LfD applications. Effective connectivity is crucial for real-time control and monitoring, ensuring that robots can perform tasks as expected and that data collected is accurate.

References

J. Amirian, B. Zhang, F. V. Castro, J. J. Baldelomar, J.-B. Hayet, and J. Pettre, “Opentraj: Assessing prediction complexity in human trajectories datasets,” in Asian Conference on Computer Vision (ACCV), no. CONF, Springer, 2020.

A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilienthal, “Th¨or: Human-robot navigation data collection and accurate motion trajectories dataset,” IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 676–682, 2020.

D. M. Nguyen, M. Nazeri, A. Payandeh, A. Datar, and X. Xiao, “Toward human-like social robot navigation: A large-scale, multi-modal, social human navigation dataset,” 2023.

H. Karnan, A. Nair, X. Xiao, G.Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, “Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation,” 2022.

J. Fu, K. Luo, and S. Levine, “Learning robust rewards with adversarial inverse reinforcement learning,” 2018.