Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MASTERTHESIS

Optimal Robot Human Handover Velocity

vorgelegt von

Jonas Tietz

Faculty of Mathematics, Informatics and Natural Sciences

Fachbereich Informatik

Studiengang: Master of Science Informatik

Matrikelnummer: 6702019

Erstgutachter: Prof. Dr. Jianwei Zhang

Zweitgutachter: Dr. Matthias Kerzel

Betreuung: Michael Görner

Abstract

Human-robot handover plays a central role in many human-robot collaboration tasks. Choosing the proper velocity for the handover trajectory of a robot giver is vital for the handover's performance. The trajectory could be slow and bore or even hinder the performance of its partner. Executing a too fast trajectory could lead to a less fluent handover. This thesis studied the effect of the Cartesian velocity of the handover trajectory of the robot giver.

Zusammenfassung

Human-robot handover ist ein zentraler Bestandteil von Mensch-Roboter-Kollaboration. Es ist wichtig die richtige Geschwindigkeit der Übergabetrajektorie des Übergebers zu wählen. Eine zu langsame Trajektorie könnte den Partner langweilen oder sogar die Gesamtleistung behindern. Eine zu schnelle Trajektorie könnte die Übergabe weniger fließend ablaufen lassen. Diese Arbeit beschäftigt sich mit dem Effekt der Kartesischen Geschwindigkeit der Übergabetrajektorie des Roboters.

Contents

List of Figures

1. Introduction

1.1. Motivation

Human-robot collaboration has become an important research area with growing interest in shared workspaces and industry 4.0. Other applications include nursing, general service robots, and eldercare. Human-robot handover is a crucial capability in human-robot collaboration, handing over tools or materials to complete a task or deliver an ordered food or beverage.

Making this process more fluent, comfortable, and safe is necessary to be accepted by humans in the previously mentioned applications. We are not aware of any research considering the Cartesian velocity of the robot's hand during the handover. Slow handover trajectories could make the handover less efficient and could lead to boredom of the receiver. Overly fast trajectories, on the other hand, could lead to less fluent handover. This thesis studies the effect of different Cartesian velocities and tries to determine if there is an optimal velocity.

1.2. Related Work

Human-robot handover has been of interest for studies for many years. The following is a selection of relevant papers concerning human-robot handover. These include human-robot interaction studies and also proposed methods for implementing subtasks of human-robot handover.

Edsinger et al. [Edsinger and Kemp, 2007] studied whether humans adjust their behaviour to the robots in a human to robot handover scenario. In their study, they instructed the human participants to place a box into the robot's hands once the robot performs a reaching motion towards the participant. Among other things, Edsinger et al. measured the grasp alignment of the box to the robot's hand. They found that all human participants matched the alignment of the box to the hand.

Huber et al. [Huber et al., 2009] proposed a trajectory generation method for handover tasks, based on their findings that velocity profiles observed during human to human handovers do not follow minimum jerk profiles and are dependent on the task. Their method decouples the z-component from the xy-components and combines the minimum jerk fits for both of those.

Mainprice et al. [Mainprice et al., 2010] proposed that including the human as a separate agent instead of an obstruction would improve the handover. They suggested three constraints for motion planning in handover tasks to make the interaction safer and more legible. The constraints are distance, visibility, and comfort. They describe how these can be integrated with standard motion planning algorithms.

Cakmak et al. [Cakmak et al., 2011] studied human preferences for handover configurations. They discovered that there is a common understanding of good handover configurations. Good configurations had in common that they were reachable, and the objects were in their default orientation. Likewise, participants preferred a natural configuration of the robot, which means that the robot's joints are in a configuration, which is likely for a human to perform.

Bohren et al. [Bohren et al., 2011] implemented an autonomous robotic butler on the Willow Garage PR2, the same robot used in this thesis. To implement the behaviour Bohren et al. introduced a new task-level architecture SMACH, based on hierarchical state machines. The implemented task included a handover of a beverage from the PR2 to a human.

Chan et al. [Chan et al., 2012] investigated the grip and load forces of the giver and receiver on the object during the physical handover in a human-to-human handover study. They found that the giver is responsible for the safety of the object. Additionally, the physical handover ends with an upward pulling force experienced by the giver.

Hendrich et al. [Hendrich et al., 2014, Hendrich et al., 2016] implemented a handover controller based on force measurements for physical handover detection and explored the force threshold's relation to the object weight. They discovered that participants preferred very low interaction forces but also tolerated higher forces for heavier objects.

Dragan et al. [Dragan et al., 2015] studied the effect of different trajectory types of the robot in human-robot collaboration tasks. They considered functional, predictable, and legible movements. Legible movements performed best in terms of coordination time and in terms of fluency.

Eguiluz et al. [Eguiluz et al., 2017] implemented a handover controller for the Shadow Dexterous Hand, which can differentiate between pulling forces initiating the physical handover and perturbation forces. If a perturbation is falsely identified as a handover, the robot could drop the object and potentially damage the object or itself.

Vannucci et al. [Vannucci et al., 2018] studied the effect of gentle and aggressive vital forms during human-robot handover. Their experiment tested aggressive and gentle reaching motions and aggressive and gentle voice commands and measured the effect on the peak velocity and peak acceleration of the human hand. Aggressive vital forms made the participants accelerate their hand faster than the gentle counterpart, but using voice commands instead of the reaching motion had a more significant effect on the human participant's peak acceleration and peak velocity.

Nemlekar et al. [Nemlekar et al., 2019] proposed an object transfer point (OTP) estimation method. Their method was based on an initial static OTP estimation based on a handover study, where they found that the OTP is in the middle between both agents. The second part dynamically refines the initial estimation with a Probabilistic Movement Primitive (ProMP) based approach, where they modeled the robot state together with the human hand position and the OTP in a single model.

Most of these cited works cover the topic of human-robot handover. For example, [Huber et al., 2009, Mainprice et al., 2010, Dragan et al., 2015] focused on trajectories for handover tasks, but they did not consider the effect of different velocities of the trajectory. [Vannucci et al., 2018] showed that vital forms affect the human partner. As how aggressive or gentle a robot will be perceived could, in part, be influenced by the velocity of the executed trajectory. However, they did not explicitly study different velocities. Instead, they asked an actor to perform aggressive and gentle handover movements and mapped them onto the robot. Other works inspired the implementation of the study. Even though we did not dynamically adapt the OTP to the human partner's actions, we used the result from [Nemlekar et al., 2019] for our static OTP position. [Hendrich et al., 2016] informed the decision to minimize the interaction forces during the physical handover.

1.3. Thesis Goal

This thesis aims to investigate the effect of different Cartesian velocities of the robot's hand during the trajectory towards the human receiver for robot-to-human handovers. We postulate that there might be an optimal Cartesian velocity above which the overall performance or perception of the handover worsens. In order to research this, a human-robot interaction study is implemented and executed.

1.4. Thesis Outline

Chapter 2 will present the fundamentals of handover. It will first go over the basic principles of human-robot handover. After that, a brief introduction to inverse kinematics is given. It follows an explanation of the topic of motion planning. It will then present the theory behind Probabilistic Movement Primitives, which are used to generate the robot's handover trajectory. Finally, fluency evaluation in human-robot collaboration will be explained. This is a central part of the evaluation of the study's results.

Chapter 3 will present the design of the study. The postulated hypotheses and the setup of the experiment are introduced.

In Chapter 4 explains how the experiment was implemented. Firstly, it describes how the object-picking was achieved. It will then go over the trajectory generation for the handover motion towards the receiver. And finally, how the physical handover is detected.

Chapter 5 presents the results of the experiment. Firstly, it examines how accurately the robot could reach and hold the desired Cartesian velocity. It will then take a look at one major mistake, which was made. And finally, it discusses the relevant data for each hypothesis, and the implications of said data.

And finally Chapter 6 summarizes this thesis's results and will give an outlook on possible future work and improvements to the study design.

2. Fundamentals

2.1. Object Handover

As described by Ortenzi et al. [Ortenzi et al., 2020], Object Handover is a joint action between two agents with the goal of passing an object from the giver to the receiver. In this thesis, only robot-to-human handover is considered. This means the giver is always a robot and the receiver is a human. The giver's goal is to offer the object to the receiver in a suitable manner and maintain a stable grasp during the transport until the receiver grasps the object. The receiver's goal is to grasp the object and perform the task for which the object is needed.

Ortenzi et al. split the handover process into two phases: The pre-handover and the physical handover. The pre-handover phase includes communication of the giver and receiver, the grasping of the object by the giver and the transfer of the object towards the receiver. Common ways of communication during handovers are oral cues, gaze, gestures, movements, and object grasps. These can help establish the what, when, and where of the handover.

The pre-handover phase either starts with a request for an object or with a request to perform a task. When an agent asks for an object, he then becomes the receiver and the other becomes the giver. When an agent asks for the task to be done with an object, he becomes the giver while the other becomes the receiver. The handover can either be direct, which means that the receiver grasps the object directly from the giver's hand, or it can be indirect. In that case, the giver places the object on an intermediary surface and the receiver grasps the object from the surface.

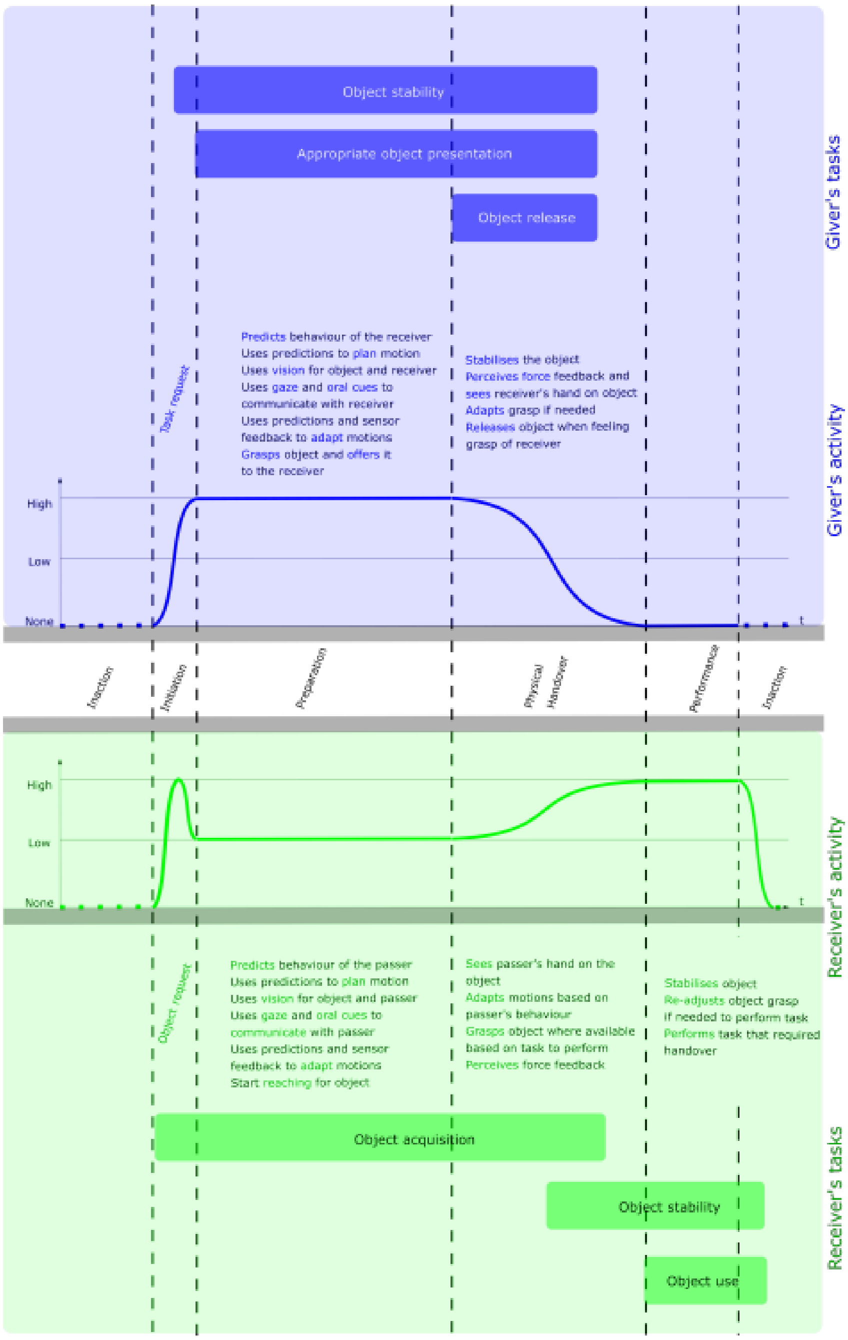


Figure 2.1.: The difference in activity for giver and receiver during the different phases of handover. [Ortenzi et al., 2020]

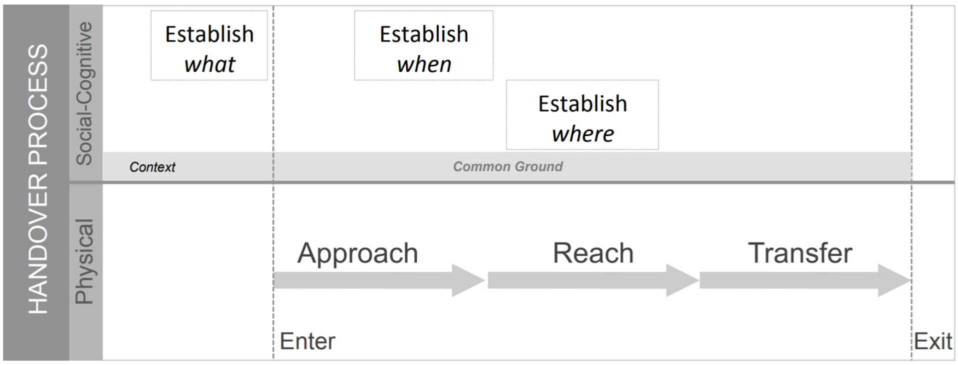


Figure 2.2.: The phases of a typical handover according to Strabala et al. [Strabala et al., 2013]

The pre-handover phase ends when the receiver touches the object. During the physical handover, the receiver obtains a stable grasp of the object. Once the object's load is transferred, the giver retracts its arm, and the receiver performs its task.

During the handover phases, both agents' activity is very different, as can be seen in figure 2.1. During the pre-handover phase most of the activity is done by the giver, while only some of the receiver's activity can be seen. In the physical handover phase, the giver's responsibility gets transferred to the receiver, and the activity goes to zero while the activity of the receiver increases.

Other ways of splitting the handover process exist. For example, Strabala et al. [Strabala et al., 2013] used three phases: The Approach, the Reach, and the Transfer. During the approach phase, information about the object, like weight and fragility, can be inferred by how the giver holds the object. The reach phase influences where and when the object is being transferred. In the transfer phase, the load of the object is transferred from the giver to the receiver. Both giver and receiver ensure that that object remains stable. Once the receiver holds the whole object load, the giver retracts its arm, ending the handover. These phases can be seen in Figure 2.2.

This thesis follows the definition of Ortenzi et al. Because the main focus of this thesis is the handover trajectory during the pre-handover phase, some of the communication between giver and receiver is simplified. We did not use any oral cues or gaze in this study to communicate where the physical handover will happen. The only information of where the physical handover will happen from the robot is its orientation and the reaching motion.

2.2. Inverse Kinematics

In robotics, it is very useful to specify points in Cartesian space, as it is the natural space we live in. To plan a trajectory, which ends up with, for example, the end-effector at that point, it is necessary to transform the point from Cartesian space to the joint space of the robot. The joint space represents all possible configurations of the robot with the robot's joint angles as the coordinates. This is not a one-to-one mapping. For robots with many degrees of freedom, multiple joint space positions for one Cartesian space position may exits. This is illustrated in Figure 2.3. The figure shows two possible solutions for a simple robot arm given a goal position. The problem of inverse kinematics is to find a joint space position for a given Cartesian space one.

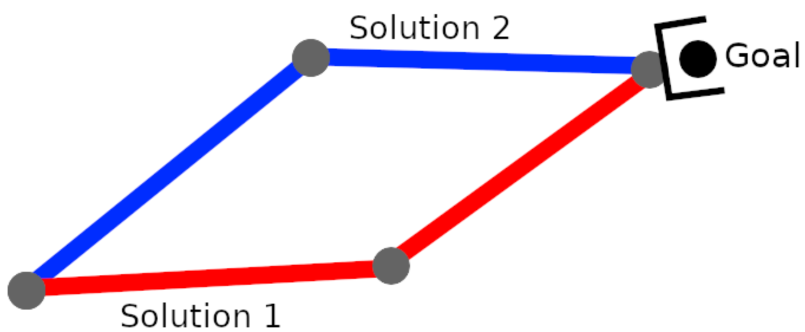


Figure 2.3.: An illustration of multiple possible inverse kinematics solutions given a goal.

For some robots, it is possible to solve this problem analytically. The standard approach is to use a numerical solver, which iteratively gets closer to the solution until the desired precision is reached.

One solver, which computes the inverse kinematics of a pose, is bio_ik [Ruppel et al., 2018]. It is a solver based on a memetic optimization algorithm. It allows one to compose a cost function from many different goal types. The genome of the evolutionary part is the robot joint positions. Each generation is mutated and the best individuals are selected. After some evolution steps, gradient-based optimization is used to improve the best solutions before resuming the evolutionary part.

The goal types include, among others, Pose Goals, which try to minimize the error to a position and orientation of an end effector, Minimal Displacement Goals, which penalize robot poses, which are far away from the last robot pose, and Look At Goals, which try to align the axis of a link towards a specified position.

2.3. Motion Planning

Motion planning is the problem of generating a collision-free trajectory between two states, based on the robot's shape, dynamics and the environment. The goal is to find a continuous path through the configuration space without colliding with any obstacle or itself.

In this thesis, motion planning was used for most of the robot's trajectories, which were not relevant to the main focus of this study. For example, it was used for some of the trajectories during the pick-up of the objects or when the robot retracts its arm.

One commonly used algorithm to solve the motion planning problem is RRT-Connect [Kuffner and LaValle, 2000] based on the rapidly-exploring random trees (RRT) algorithm [Lavalle, 1998]. RRT-Connect is a sampling-based algorithm. It

Algorithm 1: The RRT-Connect algorithm

```
tree_a = initTree(start_state);
tree_b = initTree(goal_state);
for i=0..n do
    random = SampleSpace();
    closest = FindClosestNode(tree_a, random);
    new = CreateNewNodeInDirection(closest, random);
    if CollisionFree(closest, new) then
        tree_a.addNode(new);
        tree_a.addEdge((closest,new));
        new_b = connect(tree_b, new);
        if new == new_b then
            return path(tree_a, tree_b);
        swap(tree_a, tree_b);
```

works by iteratively building two tree structures. Each node represents a robot state and each edge a collision-free trajectory between those states. The root of the first tree is initialized as the start state and the root of the second tree is initialized as the goal state. The idea is that one tree will grow randomly, exploring the configuration space, while the other tries to grow towards the other tree.

At each iteration, the first step is to sample a random point from the configuration space. For that sampled point, the nearest node of the first tree is then found. Then a point is calculated, which is in the direction from the previously found close node to the sampled point, but only a tiny step away from the found nearest node. If there is a collision-free trajectory between the new point and the found nearest node, the new point gets added as a node to the tree connected by an edge to the found nearest node. After that, the second tree tries to connect to the created node of the first tree by using the same method as the first tree. But this time, RRT-Connect tries to repeat this step until either the new node of the first tree is reached, or an obstacle is in the way. If this step has successfully connected both trees, the algorithm is done and the path can be returned, otherwise, the trees are swapped and the process repeats for a fixed number of iterations. Figure 2.4 shows some examples of the RRT-Connect algorithm exploring different setups.

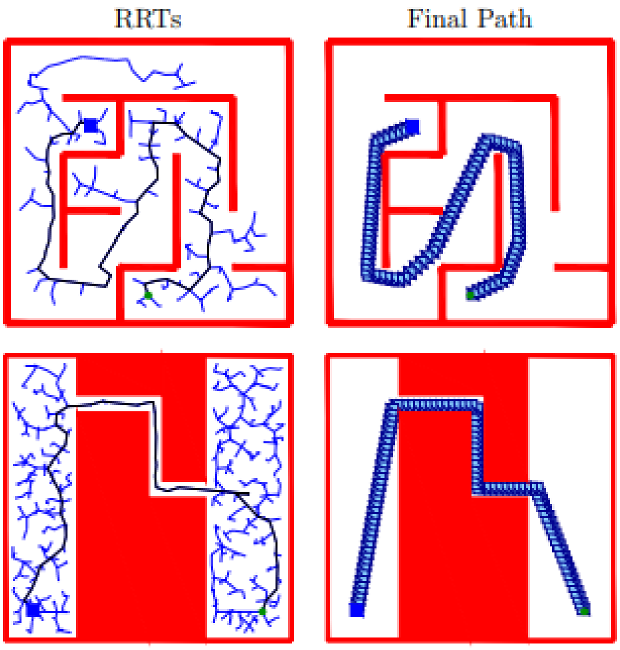


Figure 2.4.: Examples off the RRT-Connect algorithm. [Kuffner and LaValle, 2000]

2.4. Probabilistic Movement Primitives

For modeling the robot arm motion during the handover, Probabilistic Movement Primitives (ProMPs) were chosen. This ensures that the resulting trajectories are kinematically similar and gives one the ability to introduce some variation to counter the participants' learning effect. ProMPs were introduced by Paraschos et al. [Paraschos et al., 2013] as an alternative to other Movement Primitives approaches. In their paper, they first introduce some desirable properties of any Movement Primitives(MPs) framework. The first property is the ability to combine and blend multiple MPs to create complex motions. Smoothly blending from one MP into the other either in parallel or sequentially. The following important property is the ability to specify start, end, and via points. It is often necessary to ensure certain positions or velocities, which the trajectory has to run through. It is also helpful to change the execution speed and timing of the whole movement. Finally, the framework should support both stroke-based movements and periodic movements and should be able to be learned from demonstrations.

2.4.1. Representation

The goal of ProMPs is to create a compact probabilistic model for a movement execution $\tau = (y_0, y_1, \ldots, y_n)$ from multiple trajectory demonstrations. This allows one to have multiple trajectories for the same overall movement. A single demonstration is modeled as a weighted linear basis model with basis functions $\phi_j(z)$ and weight vector ω plus zero-mean Gaussian white noise ϵ. Let Φ_t be the basis function matrix and y_t a joint position and velocity at time t.

$$y_t = \Phi_t \omega + \epsilon_y \tag{2.1}$$

Then the probability of a trajectory τ is:

$$p(\tau|\omega) = \prod_t \mathcal{N}(y_t|\Phi_t\omega, \Sigma_y) \tag{2.2}$$

The weight vector ω can then be modeled by a Gaussian distribution with mean μ_ω and variance Σ_ω: $p(\omega;\theta)$ with $\theta = \{\mu_\omega, \Sigma_\omega\}$. By then marginalizing out the weight vector ω, the resulting distribution only depends on θ, giving us a Hierarchical Bayesian Model of the movement.

$$p(\tau;\theta) = \int p(\tau|\omega)p(\omega;\theta)d\omega \tag{2.3}$$

To model more degrees of freedom, y_t is expanded to the positions and velocities of all joints. The weight vector is expanded as well. The basis matrix gets extended to a block diagonal matrix Ψ_t.

$$\Psi_t = \begin{bmatrix} \Phi_t & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \Phi_t \end{bmatrix} \tag{2.4}$$

Typically Gaussian basis functions $b_j(z)$ with the center c_j and width h are used for stroke based movements. They are spaced uniformly in the interval $[-2h, 1+2h]$.

$$b_j(z) = \exp\left(-\frac{(z-c_j)^2}{2h}\right)$$

The basis functions for the model are normalized for a better regression performance.

$$\phi_j(z) = \frac{b_j(z)}{\sum_k b_k(z)} \tag{2.5}$$

The first step to learn the Hierarchical Bayesian Model is to use linear regression to estimate the weight vector ω_i for each demonstration of the movement.

$$\omega_i = (\Psi_t^T \Psi_t + \lambda \mathbf{I})^{-1} \Psi_t Y_i \tag{2.6}$$

Y_i are all the joint positions of the demonstrations and λ is the ridge factor. Paraschos et al. [Paraschos et al., 2013] suggest a small value arount 10^{-12}.

To then get θ the maximum likelihood method can be used to estimate the mean and variance of the weight vector.

$$\mu_\omega = \frac{1}{N} \sum_{i=1}^{N} \omega_i \tag{2.7}$$

$$\Sigma_\omega = \frac{1}{N} (\omega_i - \mu_\omega)(\omega_i - \mu_\omega)^T \tag{2.8}$$

The resulting Probabilistic Movement Primitives used in this thesis and the demonstrations they are based on can be seen in Figure 4.4 and Figure 4.3.

2.4.2. Viapoints

It is essential to have the ability to specify start, end, and via-points to apply a movement primitive in practice. There are many situations where it is required to be in a specific pose or move with a specific velocity. Because of the probabilistic model of ProMPs this can be achieved by adding the desired via-point as a new observation $x_t^* = \{y_t^*, \Sigma_y^*\}$. Where y_t^* is the desired state and Σ_y^* is the accuracy of the observation.

To get the updated model, Bayes theorem needs to be applied.

$$p(\omega | x_t^*) \propto \mathcal{N}(y_t^* | \Psi_t \omega, \Sigma_y^*) p(\omega) \tag{2.9}$$

If the distribution $p(\omega|x_t^*)$ is Gaussian, then the mean and variance are:

$$\mu_\omega^* = \mu_\omega + \Sigma_\omega \Psi_t (\Sigma_y^* + \Psi_t^T \Sigma_\omega \Psi_t)^{-1} (y_t^* - \Psi_t^T \mu_\omega) \tag{2.10}$$

$$\Sigma_\omega^* = \Sigma_\omega - \Sigma_\omega \Psi_t (\Sigma_y^* + \Psi_t^T \Sigma_\omega \Psi_t)^{-1} \Psi_t^T \Sigma \omega \tag{2.11}$$

The effect of specifying an endpoint can be seen in Figure 4.6.

2.5. Fluency Evaluation

To evaluate the effect of changing the velocity of the handover trajectory between different trials of the study, meaningful measures need to be chosen. One possible measure is fluency. The paper [Hoffman, 2019] by Hoffman collected different commonly used metrics for evaluating fluency in robotics, both objective and subjective. They then proceeded to study the correlation between the subjective and objective measures in an online study.

The subjective measures were grouped into seven different categories, which either were directly or indirectly associated with fluency of human-robot collaboration. The categories are as follows:

- Human-Robot Fluency
- Robot Relative Contribution
- Trust in Robot
- Positive Teammate Traits
- Improvement
- Working Alliance for H-R Teams
- Individual Measures

They also provide a list of questions associated with each of the categories from previous studies. The four objective measures, which were correlated with the subjective measures, are:

- Human Idle Time (H-IDLE)
- Robot Idle Time (R-IDLE)
- Concurrent activity (C-ACT)
- Functional Delay (F-DEL)

All of the above objective measures are ratios of the total time of the task or interaction. The Human Idle Time is the percentage of the time the human is not active. The idea behind this measure is that humans, in general, are still faster and more competent at the task and are consequently waiting for the robot to finish, which could be dull or seen as under utilization of the robot.

The Robot Idle Time measures exactly the same from the perspective of the robot. The percentage of time the robot is idle. Robots in human-robot collaboration are supposed to help humans. If a robot is not doing anything, it can be seen as not sufficiently using the robot's ability to help the human. Although this is only the case if the robot is not currently waiting for the human.

The concurrent activity (C-ACT) measures the percentage of time both the human and the robot are active at the same time. A high C-ACT could indicate that the human and the robot are working seamlessly together.

And lastly, the functional delay. F-DEL measures the percentage of time both agents are inactive and the total time. The higher the F-DEL, the longer both agents do nothing, which could, in turn, be perceived as not very efficient.

Hoffman used an online study to investigate the correlation between the here described objective and subjective measures. The results were that only H-IDLE and F-DEL were significantly correlated with subjective fluency. In contrast to the previous hypothesis, H-IDLE correlates positively with subjective fluency, meaning

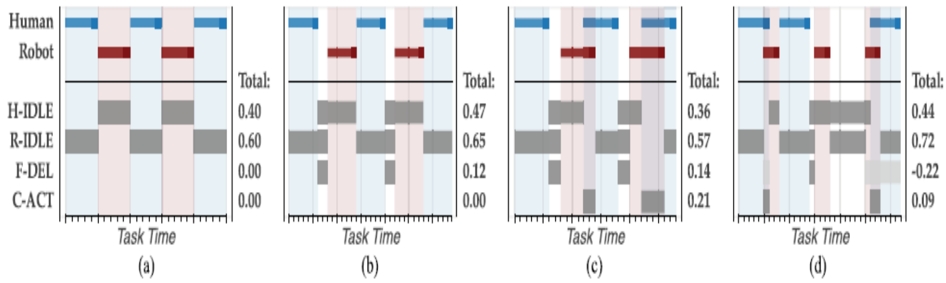


Figure 2.5.: Four examples of human robot colaboration. Subfigure (a) has no overlap between human and robot activity, but also no delay. In Subfigure (b) there is functional delay between the actions. Subfigure (c) and (d) both show examples with concurrent activity and functional delay. [Hoffman, 2019]

that handover is perceived as more fluent when the human is more inactive. F-DEL is reversely correlated with subjective fluency. The bigger the delay between one agent finishing an action and the other starting its action, the less fluent the task is perceived. It was also found out that R-IDLE did reversely correlated consistently with subjective fluency, but the results were not significant. C-ACT did not correlate with subjective fluency.

3. Study Design

3.1. Hypotheses

This thesis's primary goal is to investigate the influence of the Cartesian tool center point (TCP) velocity of the robot giver during the handover trajectory towards the human receiver. We chose to measure the Cartesian TCP velocity over the joint velocities because it is more universal between different robots. Additionally, the influence of a joint on the perceived motion can vary dramatically. For example, a joint, which rotates a link of the robot around the axis along the link, produces very little perceivable motion, while a joint far up the kinematic chain moving the whole robot arm can produce vastly more movement of the TCP.

We hypothesize that the trajectories can be executed too fast, which would lead the human to react negatively, resulting in overall less fluent and less efficient handovers. This effect will be studied with a human-robot interaction experiment. For this experiment, we postulated four hypotheses.

3.1.1. H1: Fast robot trajectories lead to shorter overall handover time.

The first hypothesis is reasonably obvious. If the robot executes its trajectory with a higher velocity, the trajectory is executed in a shorter amount of time, thus reducing the overall handover time.

3.1.2. H2: Object type affects the overall handover time.

The object type should have an influence on the overall handover time. Objects can differ, for example, in size and weight, making it more difficult for the robot to accelerate. However, they can also differ in ways, which could affect the human receiver. Heavier or fragile objects need to be handled more carefully to ensure they will not be dropped. Objects can be harder to grasp, making the handover slower. Some objects could also be more dangerous. A scissor, a knife or a pointy screwdriver need to be handled more carefully than other objects.

3.1.3. H3: Fast robot trajectories make humans act slower.

The idea behind this hypothesis is that if a robot executes the handover trajectory with a very high velocity, the human receiver might be intimidated or uncomfortable. Especially because the PR2's arms still look rigid and hard. The human might act more slowly, trying to avoid being hit. This should influence the overall time the human is active, making it longer for fast trajectories because of the more careful behaviour.

3.1.4. H4: Fast robot trajectories reduce fluency.

Fluency is a very subjective concept, but the idea behind this is very similar to the previous hypothesis H3. If the robot executes its trajectory too fast, the human might wait until the trajectory is executed and the arm is not moving anymore. He also might feel less safe and more uncomfortable. All this might lead to the handover being perceived as less fluent.

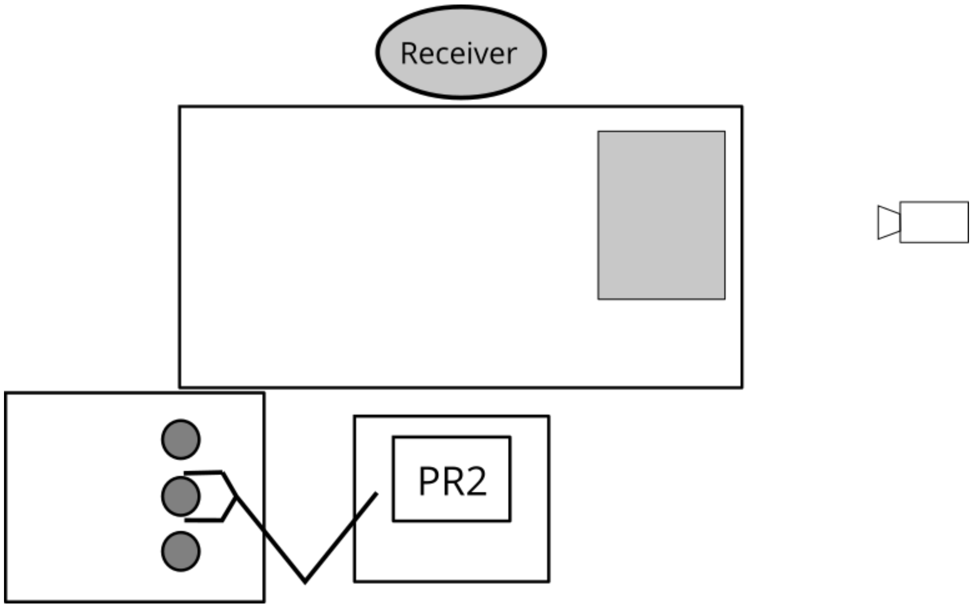


Figure 3.1.: A trial consists of the robot picking up one of the three objects from the table on the left and executing the trajectory with the specified Cartesian velocity. Once the human receiver grasps the object from the robot, the object will be placed in the area on the right. The camera on the right will record this process for labeling when the human receiver is active.

3.2. Experimental Setup

The setup consists of the PR2 robot as the giver, a table next to the giver with three objects, a table between giver and receiver, the human receiver and a camera pointed at the receiver and the place location for the objects. An overview of the setup can be seen in Figure 3.1.

Two of the three objects are chosen from the YCB Object and Model Set [Calli et al., 2015]. The Chips Can and the Flat Screwdriver. The last object is a juggling ball. The objects were chosen because of their different properties. The juggling ball is an easy to grasp and easy to place object. The chips can is also easily graspable, but due to its light weight and small bottom surface compared to its height, it is harder to place it down in a stable manner. The screwdriver is

Figure 3.2.: The objects used for the experiment. A Chips Can and a Flat Screwdriver and a juggling ball.

the hardest to grasp. Only one of the agents can grasp the handle; the other is presented with the shaft. The sharp tip of the shaft is making it more dangerous for fast trajectories. Figure 3.2 shows a photo of the used objects on the table next to the PR2.

The camera records the experiment. This is needed to annotate the start of human activity and the end of the handover. The annotations are done manually after the experiment is done. The camera is not used for any of the robot's actions. The table between giver and receiver acts as a convenient spacer to ensure that the participants are at the same distance from the PR2 and additionally to have a distinct endpoint to the handover, as the humans place the object onto the table.

The participant will be asked to stand directly in front of the table and receive the three objects from the PR2 one after another and place them in the dedicated area on the table. The robot giver will then start to pick the first object and lift it up. After that, it will execute the trajectory towards the human receiver, with the

Human-Robot Fluency
The human-robot team worked fluently together.
The robot contributed to the fluency of the interaction.
Robot Relative Contribution
I had to carry the weight to make the human-robot team better. (R)
The robot contributed equally to the team performance.
I was the most important team member on the team. (R)
Trust in Robot
I trusted the robot to do the right thing at the right time.
The robot was trustworthy.
Safety/Comfort
I feel uncomfortable with the robot. (R)
I feel safe working next to the robot.
I am confident the robot will not hit me as it is moving.
Individual Measures
The robot was committed to the success of the team.
The robot was uncooperative.

Table 3.1.: The statements used in the questionnaire. (R) means the scale is inverted. Participants were able to choose between five options ranging from *Strongly Disagree* to *Strongly Agree*

end pose being roughly between the receiver and the PR2. The trajectory velocity will be the same for all three objects and is chosen from a randomized list of desired Cartesian velocities. The velocities are in the range between $[0.4, 1.2]m/s$ with steps of 0.2. The robot will then wait for the participant to grasp the object. Once the object is grasped, the robot will retract its arm. The participant will then place the object in the dedicated area on the table.

This process will repeat another two times until each object was handed over. Once this is done, the participant is asked to fill out a questionnaire about the fluency of the handover while the setup is arranged again for the subsequent trial. The statements are picked from [Dragan et al., 2015] and [Hoffman, 2019] and are listed in Table 3.1. There are five options available ranging from *Strongly Disagree* to *Strongly Agree*. The statements cover the topics of fluency, robot contribution, trust in the robot, safety and comfort. The complete questionnaire can be seen in Figure A.1.

Each participant will do five trials which make it 15 handovers in total. The whole experiment is designed to not take longer than 25 minutes.

3.3. Pre-study

Due to the current Covid-19 pandemic, it was not clear if it was possible to conduct this study with an appropriate number of participants or at all. For that reason, a very similar pre-study was conducted. In this pre-study, only the author of this thesis participated, performing the experiment multiple times. The only major difference between the planned study and the pre-study is the range and resolution of the tested Cartesian velocities of the robot's handover trajectory. They are in in the range between $[0.2, 1.3] m/s$ in steps of 0.1. It was possible to increase the number of tested velocities because there was no set time limit for the experiment anymore. A too long experiment was viewed as not acceptable for random participants.

In total, there are 36 handovers per experiment. A questionnaire for only one participant, repeating the experiment repeatedly, does not reflect on the general population and is only one subjective view, which can not be compared to other views. For that reason, it was omitted from the pre-study.

4. Setup and Implementation

4.1. PR2 Robot Platform

For this study, the Willow Garage PR2 [Garage, 2021, Garage, 2012] robot platform was used. It is a mobile research robot with two arms. This particular one is equipped with the standard PR2 gripper on the left arm and a Shadow Dexterous Hand on the right. Only the standard gripper was used for this thesis. Figure 4.1 shows the exact PR2.

Figure 4.1.: The Willow Garage PR2 with the Shadow Dexterous Hand attached to the right arm and the standard PR2 gripper on the left arm.

4.2. Object picking

The first step for the PR2 is to pick up one of the objects. The objects are placed on a fixed and predefined position on the table next to the robot, so that there is no need for object detection and pose estimation.

The MoveIt task constructor [Görner et al., 2019] was used to implement the pick sequence. The MoveIt task constructor is a MoveIt [Coleman David, 2014] based framework, which allows one to build up complex tasks from simpler stages. These stages can be solved separately and the solution can be passed to the neighbouring stages if they depend on it. There are three basic stage types, which are classified by their interface.

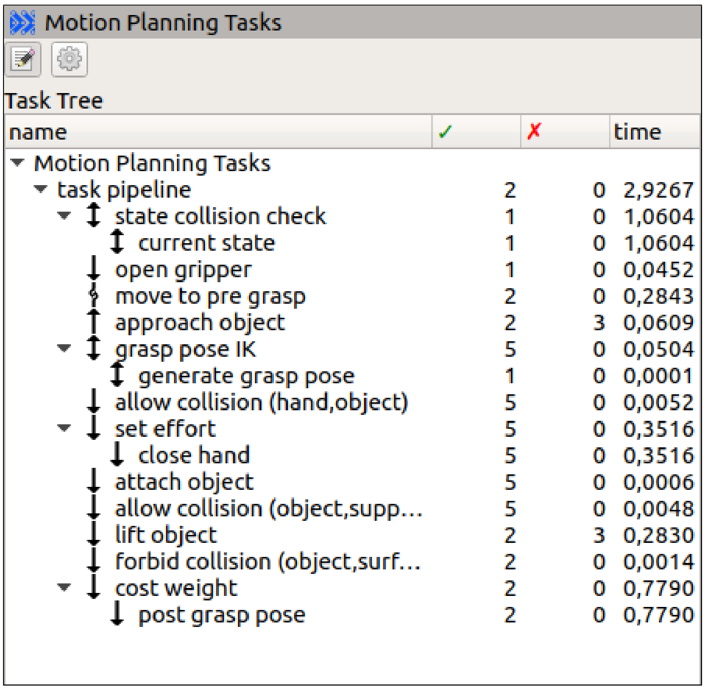


Figure 4.2.: The MoveIt Task Constructor panel in RVIZ showing the pick sequence. The arrows show the direction in which a solution is passed. Propagator stages only have one arrow. Generator stages have two arrows pointing outwards. The knot-like symbol indicates connectors.

Generator stages do not depend on any input from other stages and push their solution to both adjacent stages. Propagators take only one solution either from the previous stage or from the next stage and produce a new solution to push to the opposite site. Connector stages take the solutions from both neighbouring stages and find one or more valid trajectories between them.

The pick sequence is very typical. Firstly, the robot plans a trajectory to a pre grasp pose. Then executes a Cartesian approach trajectory and closes the gripper with an effort that is hardcoded per object. After that, a Cartesian lift-up motion is executed. Finally, the robot plans a trajectory to a hardcoded pose to later ensure handover trajectories of equal length. A visualization of the sequence in RVIZ can be seen in Figure 4.2. As one can see, the most often used basic stage is the propagator. Only the start stage, which is just the current robot state, and the grasp pose generation are generator stages. The grasp pose generation needs to be a generator because almost all previous and further motions depend on where the object is being grasped. If the object is in a different position, a different approach trajectory and lift trajectory are needed. This pick sequence has only one connecting stage. This is needed to plan a trajectory from the current robot state after opening the gripper to the state before the approach trajectory.

4.3. Trajectory Generation

For the generation of the handover trajectory ProMPs were used. They were implemented in C++ using the linear algebra library Eigen [Guennebaud et al., 2021].

The necessary demonstration trajectories were recorded beforehand. To record a demonstration, the robot first planned and executed a trajectory to the aforementioned hardcoded pose. This is the pose from which the later generated trajectories will always start. The robot was then set into mannequin mode. The mannequin mode allows an operator to move the joints of the robot freely. The robot will only try to resist the force of gravitation. Any other external force will move the

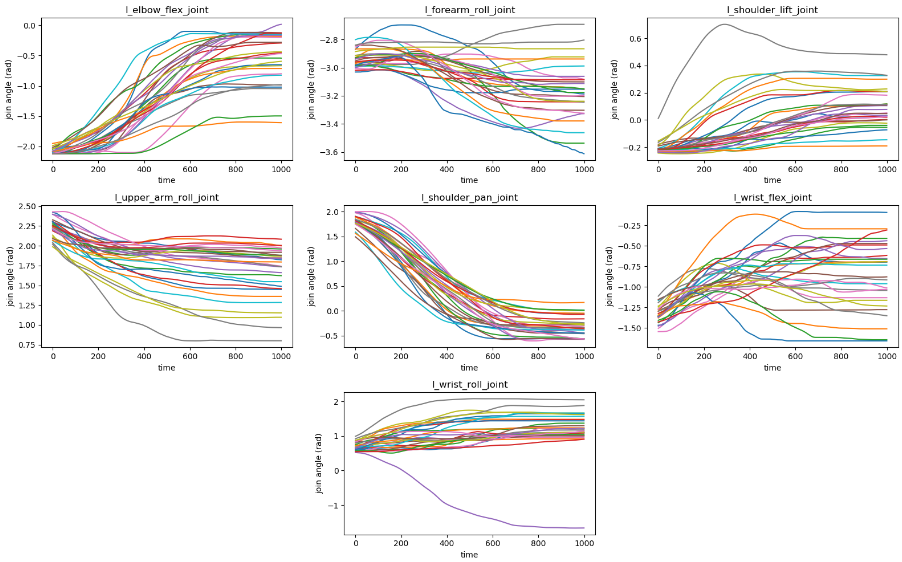


Figure 4.3.: This figure shows the recorded joint state of the relevant joints for each demonstration. Each line is a new demonstration. The y-axis is in radians and on the x-axis are the sample points.

robot joints. A human then guided the arm in a realistic handover motion towards another human receiver. The mannequin mode was then disabled again. This was repeated multiple times and recorded into a rosbag. In total, 29 demonstrations were recorded.

Later the demonstrations from the rosbag were hand-annotated using a visualization and annotation tool by Philipp Ruppel [Ruppel, 2021] determine the exact start and end of each demonstration. These demonstrations, which can be seen in Figure 4.3, were then used to generate the ProMP, which is shown in Figure 4.4. The ProMP used in this thesis only modeled the joint positions because the velocity at each timestep of the trajectory was later generated, with the goal of reaching and holding a specific velocity.

Many of the joints of the PR2 can rotate more than one complete revolution. Because of that, it was essential to normalize the joint values to a fixed range. This could not be done by simply mapping each joint position value in the range of $-\pi$ to π, because if the trajectory crossed the boundary of one revolution, it would create

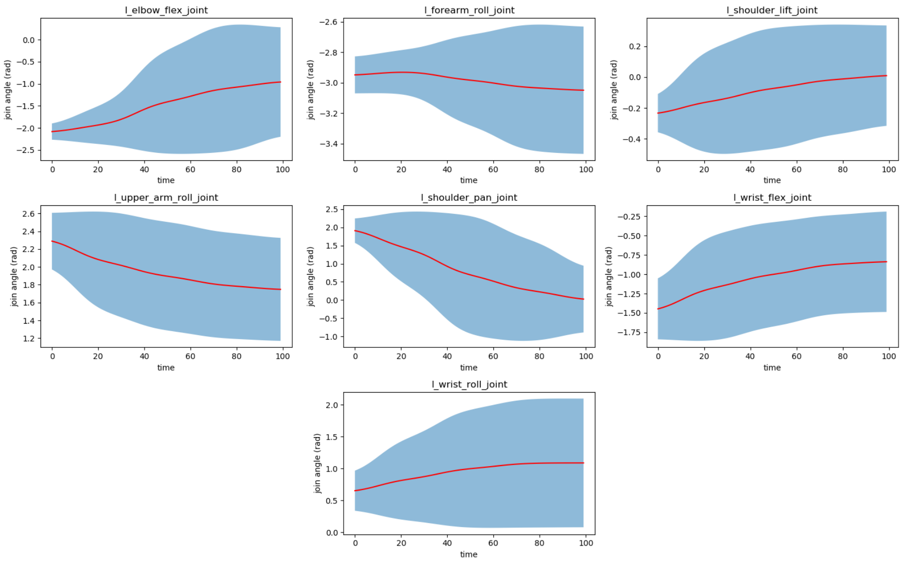


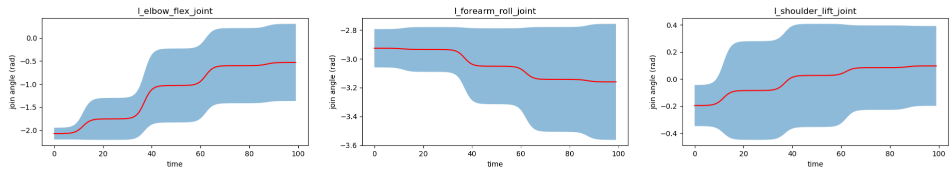
Figure 4.4.: A plot of the ProMp. The red line is the mean and the shaded area represents two times the standard deviation.

Algorithm 2: Generate ProMP from trajectory demonstrationss

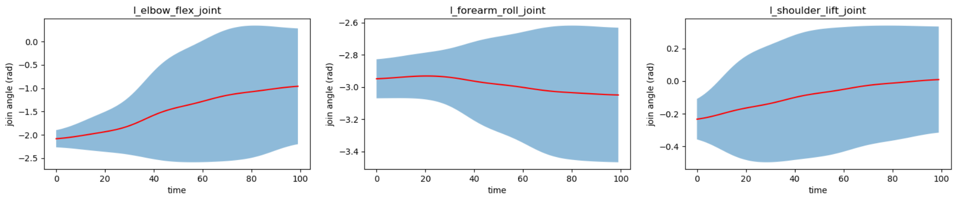
N = number_of_demonstrations;
foreach *demonstration* Y_i **do**
    enforcePositionBounds();
    resample(steps);
    Ψ_t = generateBasisMatrix(number_basis, width);
    $A = \Psi_t^T \Psi_t + \lambda \mathbf{I}$;
    $b = \Psi_t^T Y_i$;
    ω_i = solve(A, b);
    $\omega_\mu \mathrel{+}= \omega_i$;
ω_μ /= N;
$\Sigma_\mu = \frac{1}{N} \sum_{i=1}^{N} (\omega_i - \mu_\omega)(\omega_i - \mu_\omega)^T$;

---

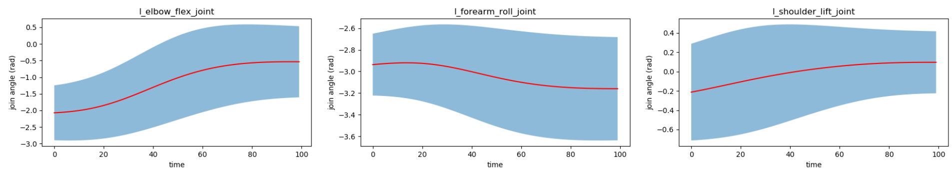
a discontinuity. Because of this, the whole trajectory is shifted by the same value. After that, the next step to generate the ProMP is to resample all demonstrations to be the same number of sample points. In the case of this thesis, 1000 sample points were chosen. If the new sample point was in between two points, linear interpolation was used.



(a) Basis function width of 0.004



(b) Basis function width of 0.016



(c) Basis function width of 0.05

Figure 4.5.: The resulting ProMP with different basis function widths for the joints l_elbow_flex_joint, l_forearm_roll_joint, l_shoulder_lift_joint

Once the demonstrations are prepared, the block diagonal basis matrix is generated for the specified number of basis functions and their width. Then the weight vector is computed for each trajectory demonstration using ridge regression. Finally, a Gaussian is fit over all weight vectors. The resulting mean and covariance matrix is the generated ProMP. The ProMP is then saved with the relevant metadata like the number of basis functions and the basis width and can be loaded to generate trajectories.

After some experimentation, it was found out that five basis functions with a width of 0.016 produced the best result for this specific setup. The resulting ProMP is smooth while still fitting close to the demonstrations. Figure 4.5 shows the ProMP for some of the joints with different widths for the basis functions. If the width is too low, the resulting ProMP has steps instead of being smooth. If the

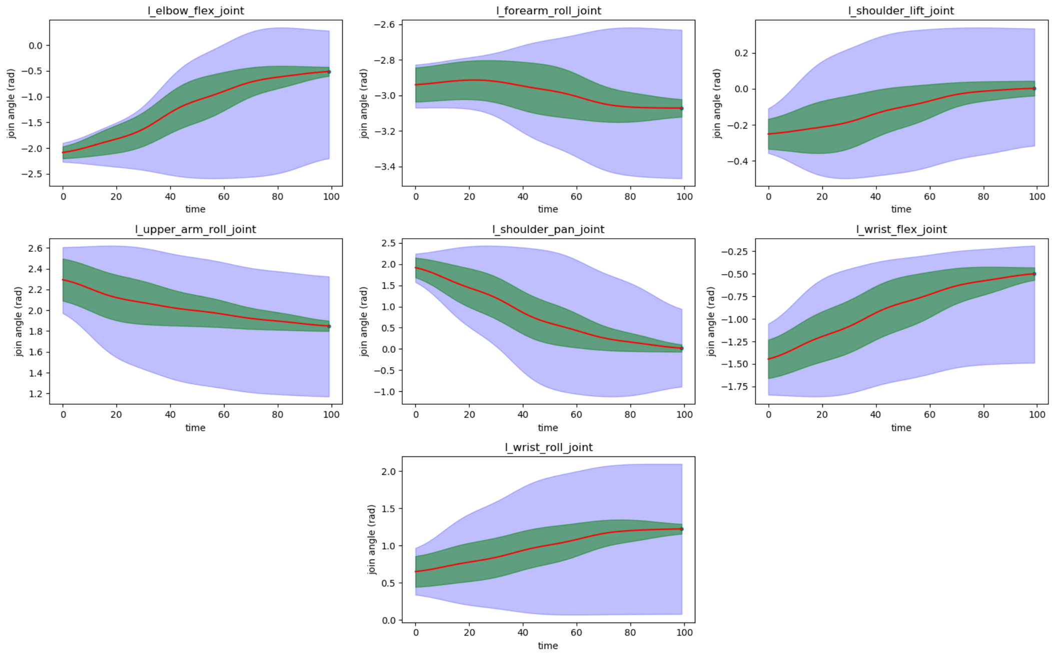


Figure 4.6.: A plot of the ProMp with an endpoint applied to it. The red line is the mean and the blue shaded area represents two times the standard deviation of the original ProMP and the green shaded area is two times the standard deviation after the endpoint is applied.

width is too large, the ProMP becomes too smooth and starts to lose detail. This is especially apparent at the beginning, where the variance should be very low because all demonstrations should start in the same state.

One of the advantages of ProMPs is that they allow one to use Bayesian inference to specify via points. This was used to specify the object transfer point (OTP), which was slightly randomized around a fixed point. This point was in the middle between the giver and receiver above the table. Because the ProMP is in joint space and the OTP is in Cartesian space, the inverse kinematics of the OTP needed to be calculated. The problem is, for one pose in Cartesian space, there may be many different poses in joint space, some of which may not lie inside the ProMP.

Joint Name	Max Velocity	Max Acceleration
l_elbow_flex_joint	7.0	6.5
l_forearm_roll_joint	7.0	6.5
l_shoulder_lift_joint	6.0	6.0
l_shoulder_pan_joint	6.0	6.0
l_upper_arm_roll_joint	6.3	6.3
l_wrist_flex_joint	6.0	6.0
l_wrist_roll_joint	7.0	6.0

Table 4.1.: Max velocity and max acceleration values for MoveIt

Therefore, the inverse kinematics solver needs to find a solution, which is close to the end poses of all possibly generated trajectories. For this, bio_ik [Ruppel et al., 2018] was used to get closer to the desired pose using a RegularizationGoal iteratively. Bio_ik was instantiated with the start state of the end pose of the mean trajectory of the ProMP. Once the solution is close enough, it is checked if the solution is too far outside of the ProMP. If it is too far, the OTP is rejected, and a new OTP is sampled. If not, the solution is used to update the ProMP using bayesian inference. Figure 4.6 shows how the ProMP changes after the viapoint is applied.

The next step is to use the ProMP to get the handover trajectory. For this, the distribution of the weight vector is sampled and the basis matrix it generated. Multiplying these produces the handover trajectory. To specify the Cartesian velocity of the TCP, the pull request to MoveIt by Scholz[Scholz, 2021] was used.

During testing, it was noticed that with the default joint velocity and acceleration limits, the desired velocities were not reached. Thus, these values needed to be adjusted in such a manner, that the arm reaches the specified Cartesian velocity and performs it in a stable way. Tuning these values too high resulted in shaky and stuttery trajectories. Surprisingly, higher values also sometimes led to slower trajectories. These joint limits can be seen in Table 4.1.

Finally, before executing the trajectory, it is checked for collisions with the environment and the robot itself. If it is not a valid trajectory, it is rejected and the ProMP is sampled again. If this is not the case, the trajectory is executed.

4.4. Physical Handover Detection

The physical handover detection starts once the handover trajectory is finished executing the generated trajectory. The tactile sensors of the PR2 gripper are used to detect the physical handover. If the sensors report a value higher than a specific threshold, the PR2 will open the gripper and release the object. The threshold was experimentally fine-tuned in such a way that the object is not dropped without any interaction with the receiver. It was also considered that the robot should not cause too much resistance once the receiver starts pulling on the object. One shortcoming of this approach is that any force acting on the object will lead to the robot releasing it even if it was hit by accident. The object could not be dropped by accident during the reaching motion because only after the PR2 finished the trajectory execution the data from the tactile sensors were considered. This also means that it is not possible to take the object out of the gripper before the trajectory is finished being executed by the robot.

4.5. Gaze

It is already known that gaze plays a vital role in human-robot collaboration. Although it is not a crucial part of this experiment, we wanted the robot to feel more interactive for the participants, especially in between the handovers, where the participants could quickly get bored. For this, we used the gazr package [Lemaignan et al., 2016] to let the PR2 look at the head of the participants and follow its movements.

An alternative idea was briefly considered. The idea was that the PR2 additionally looks at the object before grasping it and looking at the object transfer point when executing the handover trajectory. Moon et al. [Moon et al., 2014] showed that such a shared attention is highly effective at communicating the object transfer pose. Some of these motions took too long to execute. Because this was not a central part of the study, the idea was dropped.

5. Evaluation

Due to the current Covid-19 pandemic, it was not possible to conduct this study in a scope where statistical analysis might yield significant results. Only three participants, which all had prior experience in robotics and the robot used, could participate in the experiment. Nevertheless, the results will be presented because they may still provide some anecdotal evidence that could influence further studies.

The experiment took each participant on average 19:30 minutes for 15 handovers. Combining all handovers of all participants, there were 45 handovers in total. The pre-study was repeated four times, resulting in a total of 144 handovers. Each run of the pre-study took, on average, 32:30 minutes.

5.1. TCP velocity measurement

Because this thesis aims to investigate the influence of the Cartesian TCP velocity of the robot's handover trajectory, it is important to get a precise value. Unfortunately, the specified desired TCP velocity can not be used for this for multiple reasons. First of all, the robot may not be able to reach the desired velocity because of the physical limitations of the motors. And secondly, the robot can overshoot or undershoot the desired velocity. The finite-difference of the TCP position was used, to measure the velocity at each timestep of the trajectory. A ROS-node was written, which published the measured Cartesian velocity at 10hz.

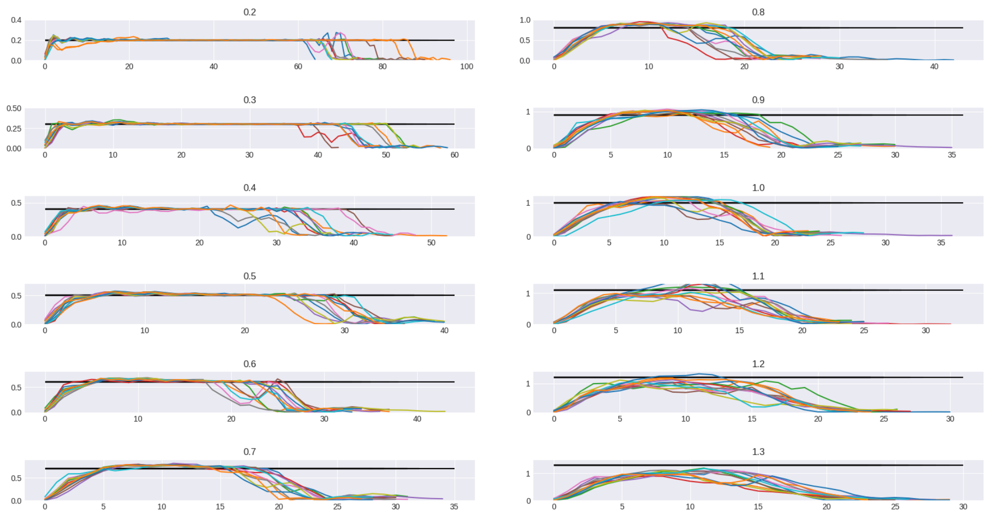
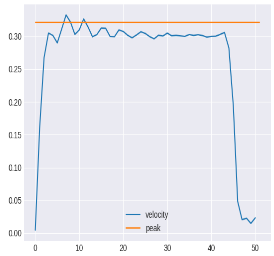


Figure 5.1.: Plots of the TCP velocities grouped by the desired velocity. The black horizontal line marks the desired Cartesian TCP velocity.

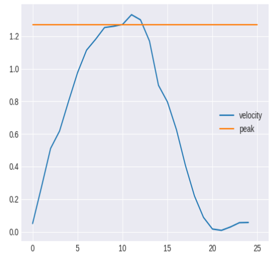
The measured velocity profiles can be seen in Figure 5.1. Each time step can potentially have a different measured velocity value. The trajectories need to be categorized to be able to compare them meaningfully. For this, we chose the peak Cartesian velocity as the category.

In Figure 5.1 it can be seen that the desired Cartesian TCP velocity matches the actual peak velocity very precisely for slow trajectories. Although there is some small overshoot at the end of some trajectories with the desired velocity of $0.2m/s$. At around $1.0m/s$, the trajectories start to vary more. Some overshoot and some undershoot the goal velocity. And for $1.2m/s$ $1.3m/s$ most trajectories do not even reach the desired velocity. This confirms that the peak Cartesian velocity needed to be taken from the measured velocities instead of simply using the specified desired velocity.

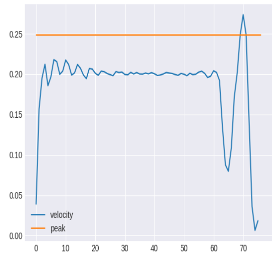
Simply taking the maximum would be too susceptible to noise, oscillations, or over-corrections of the robot. Therefore, the measured velocities were filtered using a median filter with a window of size five. Each time step represents an interval of 100ms. This window size was chosen to filter as much of the noise as possible



(a) Slow Trajectory



(b) Fast Trajectory



(c) Slow Trajectory

Figure 5.2.: Three examples of the measured peak TCP velocity using the median filter. Subfigure 5.2a and 5.2b show examples, where it gives a good solution. 5.2c gives an example where the result is not as desirable.

without flattening peaks of fast trajectories. Figure 5.2 shows a two good and one bad example of the resulting peak velocities. 5.2a and 5.2b show examples were the result coincides with the intuitive notion of the peak velocity. 5.2c shows an example for which the method failed. In 5.2c, there was a short oscillation at the end of a relatively slow trajectory. In this case, the desired result is not the actual peak of the measured velocities but the sustained peak velocity. For this, the window size is too small. A larger filter window is needed to filter this out, but a larger window would reduce the measured peak velocity for fast trajectories like 5.2b.

Figure 5.3 compares the desired Cartesian velocity against the measured Cartesian peak velocity. Until around $0.5m/s$, the desired velocity matches the measured velocity, although the measured velocity is slightly higher. This could be because the filter is not big enough for trajectories this slow. From $0.6m/s$ to a velocity of $0.8m/s$, the measured velocities start to overshoot noticeably. This is consistent with the visual findings from Figure 5.1. After $0.8m/s$, the variance of the measured velocity increases rapidly and for trajectories faster than $1.1m/s$, more often than not, it is below the desired velocity. This is consistent with Figure 5.1 as well.

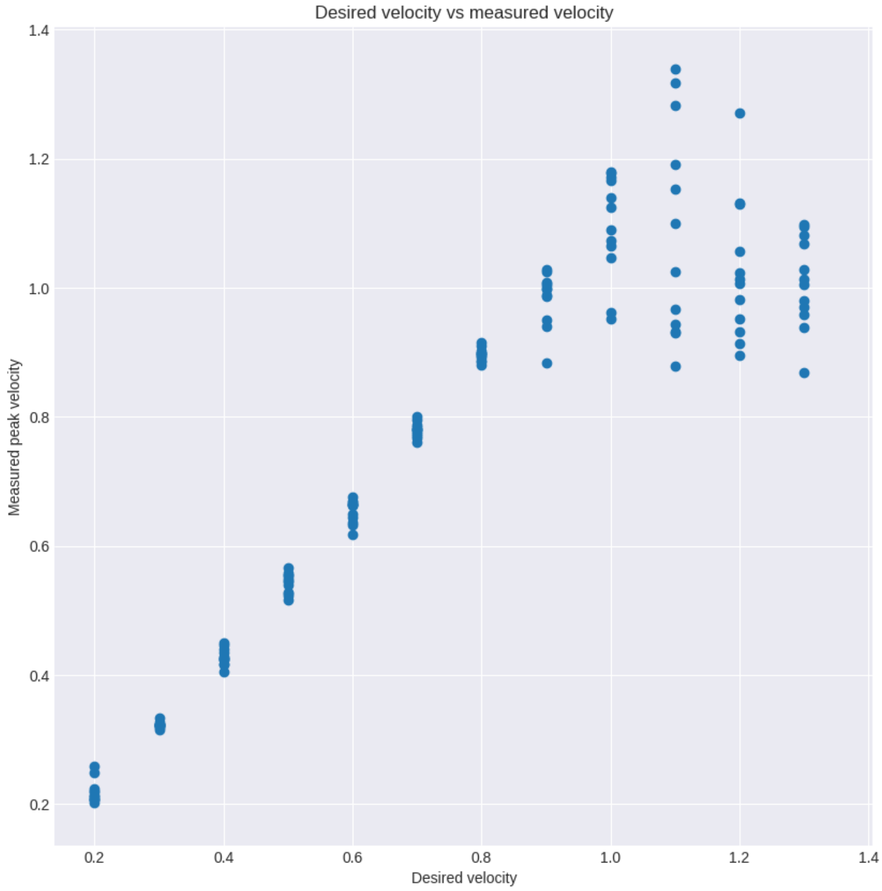


Figure 5.3.: This figure shows the desired velocity of the TCP against the measured peak TCP velocity.

5.2. Annotation

Due to how the experiment was set up, it was not possible to automatically detect the exact start and end of the human receiver's activity. Because of that, the recordings needed to be annotated manually. For this purpose, a visualization

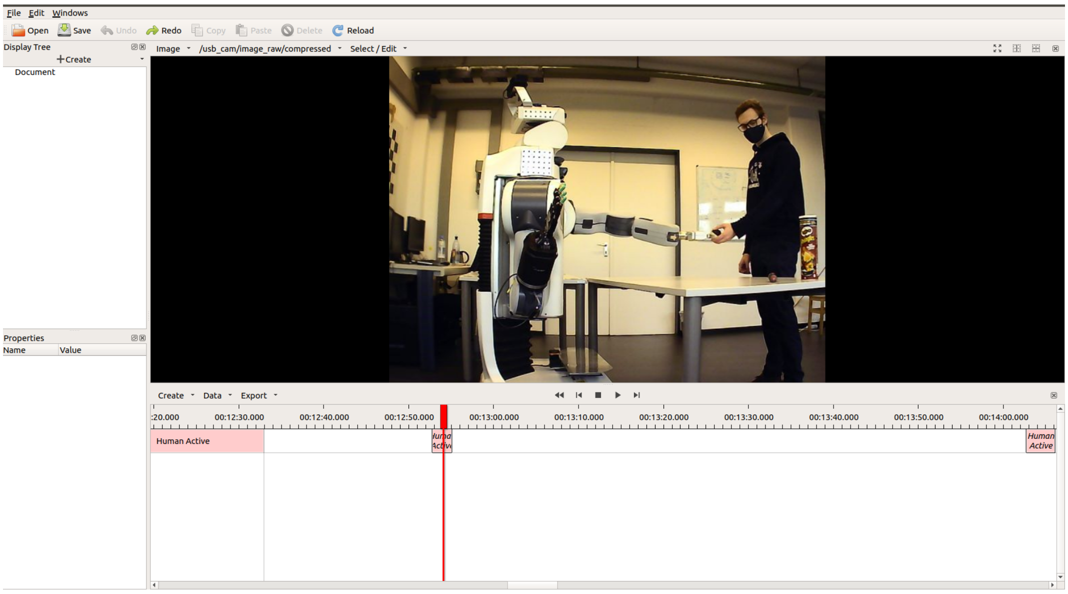


Figure 5.4.: The annotation tool used for labelling the start and end of human receiver activity.

and annotation tool by Philipp Ruppel [Ruppel, 2021] was used. It allows one to visualize many of the standard ROS messages, like JointStates or Image messages and navigate through them by using a timeline, place markers, and annotations.

Figure 5.4 shows a screenshot of this application visualizing the image topic of the side camera, of which the setup can be seen in Figure 3.1. Below the camera image, there is the annotation track with the annotated region of human activity. For this experiment, a human is considered active once the receiver starts to reach for the object in the giver's hand. This can overlap with the robot giver's reaching motion, although some participants waited until the robot finished executing its trajectory. It was not too difficult to differentiate between the idle movements of the participants and the reaching motion. Most participants either led their arms hang down or had their arms crossed while waiting for the robot.

The human was considered inactive again once the object had been placed on the table. For this, the object needed to stand stably on the table and not be grasped by the participant anymore. However, the hand did not need to be retracted. The camera recorded at a framerate of 27hz. This means that if the data is annotated frame-perfectly, an error of up to 37ms could be introduced to the total handover length and an error up to 74ms to the time the human is considered active. These annotations were then exported as a csv file for further evaluation.

5.3. Training Effect

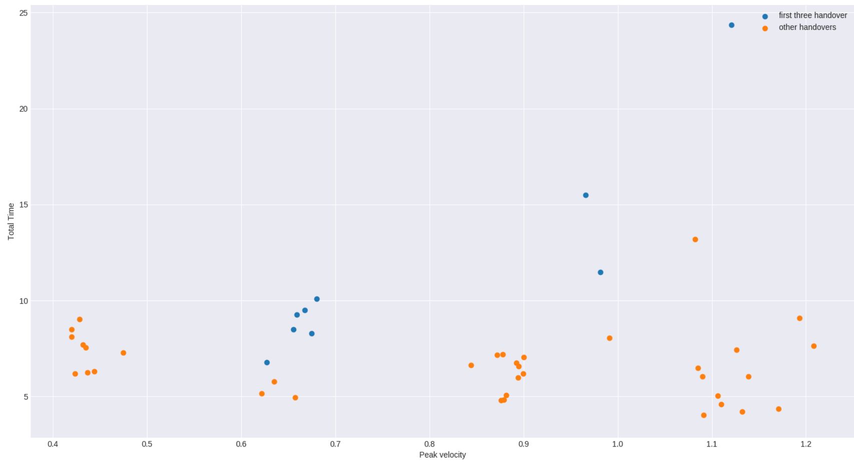


Figure 5.5.: Difference of the first three handovers vs the remaining handovers

One unfortunate shortcoming of the study was not including a training phase, in which each participant could practice the handover with some objects with different velocities to get used to the setup. It was clear while conducting the experiment that the first few handovers were very different from the later ones. For example, some participants had trouble knowing how hard to pull on the object in the gripper for the first handover. This is also clearly visible in the data. Figure 5.5 shows the total handover time for each run. The blue dots are the first three handovers, while

Figure 5.6.: Difference between prestudy results and study results.

the orange dots are the remaining ones. The first handovers clearly take longer to complete than the remaining ones. For that reason, they are removed from the later evaluation. This reduces the already small amount of data from 45 total handovers to 36 handovers.

5.4. Study Results

5.4.1. Comparing the pre-study to the study

Figure 5.6 compares the results of the pre-study with the result of this study. The blue dots represent the data from the pre-study and the orange ones the data from the study. The data follows a very similar curve, but in general, the handovers take longer to be completed. This is very likely due to the difference in experience with this specific setup. The study participants were completely unfamiliar with this experiment, while the author practiced the experiment many times before finally conducting the pre-study. Additionally, there is a greater variance in the data with more outliers, especially in the region of $1.0m/s$ to $1.2m/s$, where there are some very slow handovers compared to the pre-study.

To fill out the questionnaire multiple times occupied more time than expected for the participants of the study. This time could have been used to increase the range of tested values to be closer to the pre-study ones.

5.4.2. H1: Fast robot trajectories lead to shorter overall handover time.

For this experiment, the total handover time started as soon as the robot starts to execute its trajectory towards the human receiver and ends once the receiver places the object in the marked area on the table.

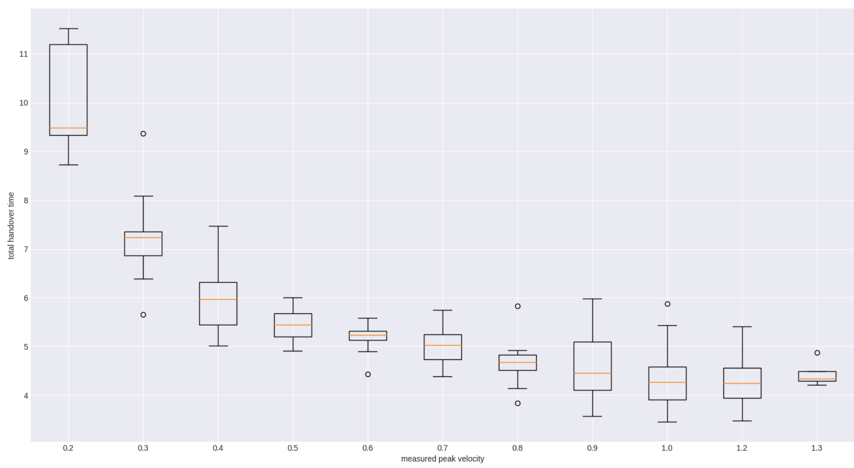


Figure 5.7.: The effect of Cartesian TCP velocity on total task time from the prestudy.

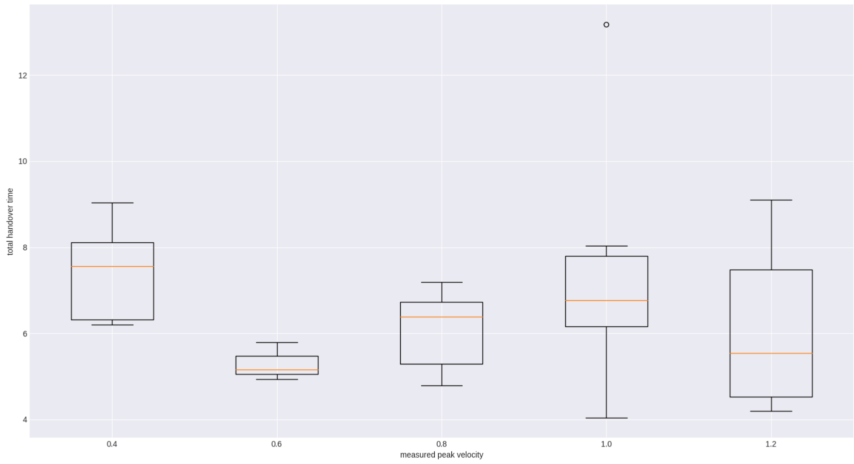


Figure 5.8.: The effect of Cartesian TCP velocity on total task time from the study.

Figure 5.7 shows the results from the pre-study. The measured total handover times were grouped by the measured velocities. The measured peak Cartesian TCP velocity is on the x-axis and the y-axis shows the total handover time.

Here its is clearly visible that the overall handover time gets shorter with higher Cartesian TCP velocities. One interesting observation is that the variance is lowest for trajectories around $0.6m/s$ and increases for both slower and faster trajectories.

The result of the study is not as clear as those of the pre-study. These can be seen in Figure 5.8. The longest handover times are still the slowest trajectories, but the large variances of especially the fastest trajectories, make it hard to make any significant conclusions. The small variance of the $0.6m/s$ bucket comes from the previously mentioned removal of the first three handovers. Nevertheless, the overall handover time goes down with faster trajectories.

The effect is more pronounced when going from slow trajectories to medium-fast trajectories compared to when going from medium-fast trajectories to fast trajectories. Both the study and the pre-study results are supporting the hypothesis.

5.4.3. H2: Object type affects the overall handover time.

Figure 5.10 shows the total handover time for each object grouped by the Cartesian TCP velocity. In this plot, it is visible that the flat screwdriver was the object, which led to the longest handover times for most of the velocities and the juggling ball consistently led to the shortest overall handover times. The reason for the flat screwdriver taking the longest time could be that the PR2 grasped it at the handle leading to a much more difficult grasp for the human receiver. Also, the screwdriver was the only pointy object, which was tested. It could be perceived to be more dangerous and thus handled more carefully, which would also lead to longer handover times.

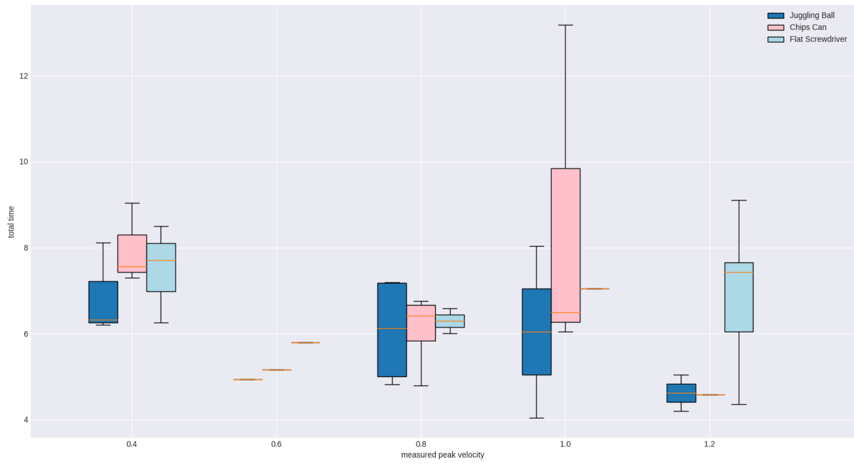


Figure 5.9.: The effect of Cartesian TCP velocity on total task time for each object from the study.

Both the chips can and the juggling ball were easy to grasp and lightweight. But still, the juggling ball consistently led to shorter handover times. The cause of that could be that the juggling ball is easier to place down. It does not matter how the ball is orientated and it is somewhat soft; that way, it does not roll away. The chips can, on the other hand, is quite tall. The receiver needs to place it down more carefully to prevent it from tipping over.

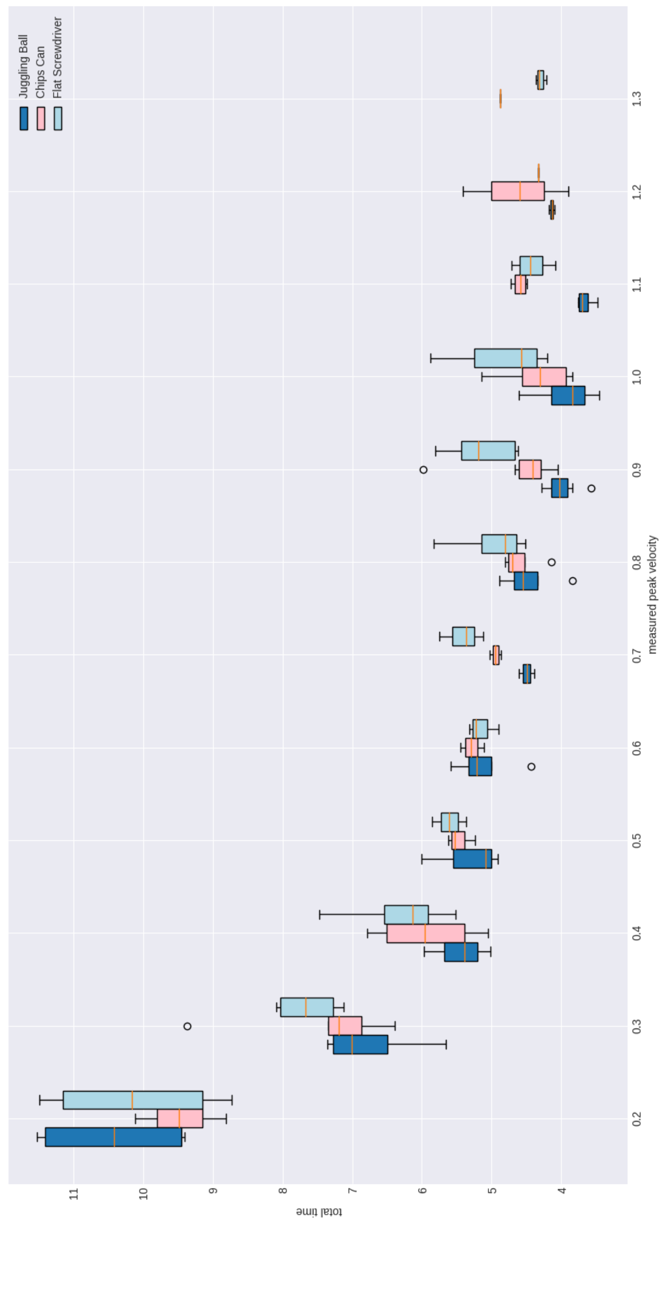


Figure 5.10.: The effect of Cartesian TCP velocity on total task for each object time from the prestudy.

Figure 5.9 shows the same plot for the study results. Again, the pattern is very similar but not as pronounced. The juggling ball led to the shortest handover times and the flat screwdriver led to the longest handovers most of the time. This supports the hypothesis that the object influences the total handover time.

5.4.4. H3: Fast robot trajectories make humans act slower.

To see if the human acts overall slower, we measured the overall time the human receiver is active during the handover. The participant is considered active once he starts reaching for the object in the PR2 gripper and is considered inactive again once the object is placed on the table. Figure 5.11 shows the human active time grouped by the Cartesian peak TCP velocity of the robot giver. It shows, similar to the total handover time, the human active time gets lower with faster robot trajectories. This disproves this hypothesis and instead shows the opposite effect.

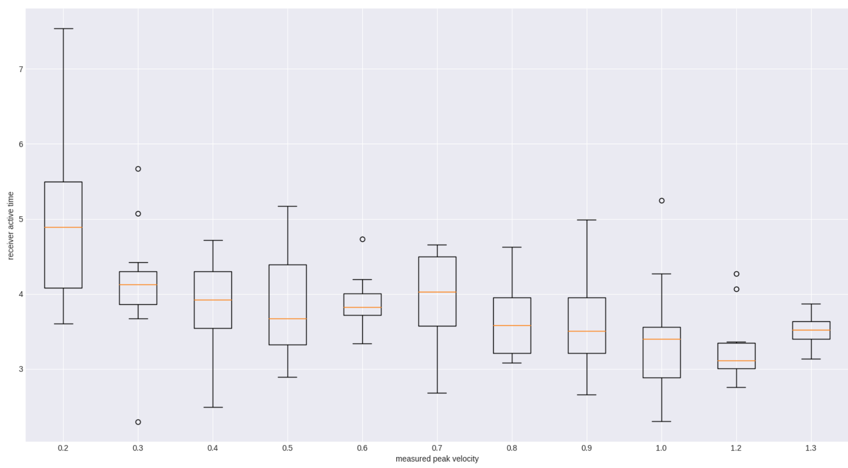


Figure 5.11.: The effect of Cartesian TCP velocity on the time the human receiver is active from the pre-study.

This could be due to a similar behaviour as described in [Vannucci et al., 2018] by Vannucci et al., where the human receivers accelerated their hands faster when the robot showed aggressive behaviour. The faster acceleration would mean that the participant reaches their peak velocity faster and thus completing the movement in a shorter time. It is also possible that movements were not fast enough to make the human feel uncomfortable and thus led to a more cautious reaction. In addition to that, the setup included a table between the robot and the human. This table could give a sense of safety and security to the participants. The table was used for two reasons. Firstly, to ensure that each participant is standing at the same distance to the robot and secondly, to have a well defined end to the handover by placing the object onto the table.

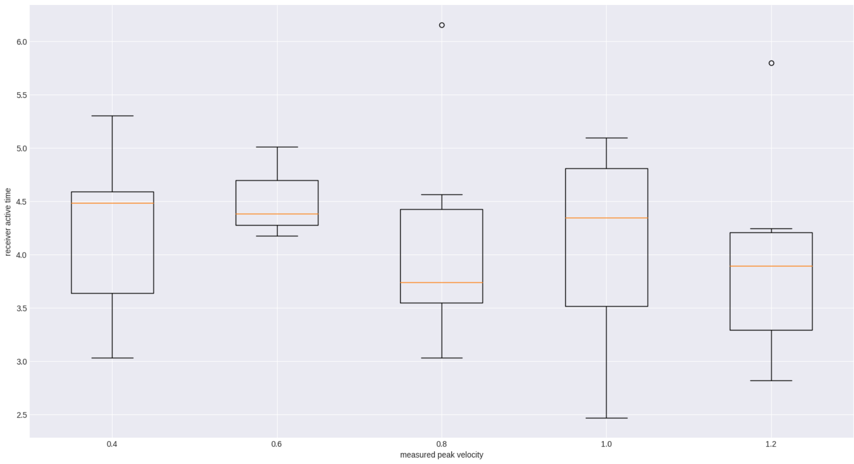


Figure 5.12.: The effect of Cartesian TCP velocity on the time the human receiver is active from the study.

Figure 5.12 shows the same data for the study. Again there is a downward trend instead of the hypothesized upwards trend giving more evidence against the hypothesis.

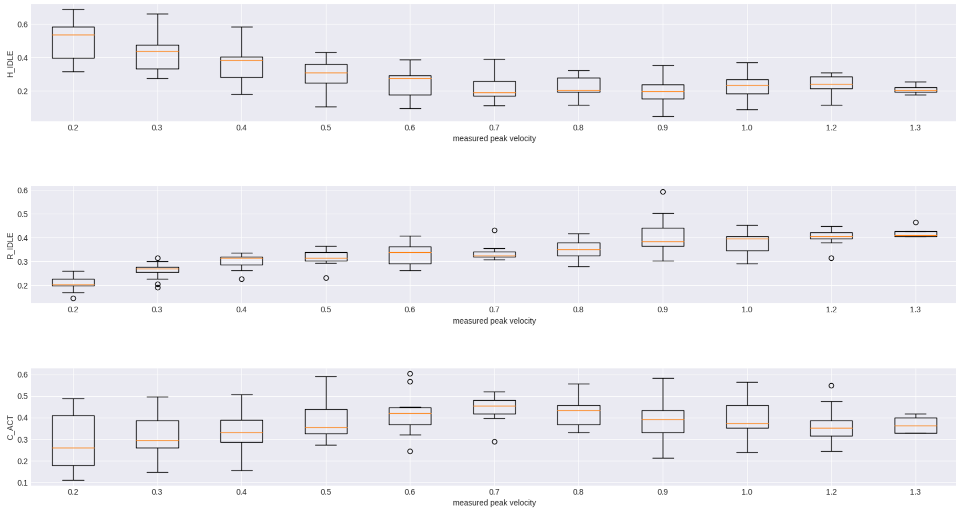


Figure 5.13.: The effect of Cartesian TCP velocity on H-IDLE, R-IDLE and C-ACT.

5.4.5. H4: Fast robot trajectories reduce fluency.

To evaluate the fluency, we mainly use the measures described in Section 2.5, namely:

- H-IDLE
- R-IDLE
- C-ACT
- F-DEL

The human idle time (H-IDLE) measures the percentage of time the human is not active during the handover, the robot idle time (R-IDLE) measures the percentage of time the robot is inactive, the concurrent activity (C-ACT) measures the percentage of time both agents are active at the same time and finally the functional delay (F-DEL) measures the percentage of time no agent is acting.

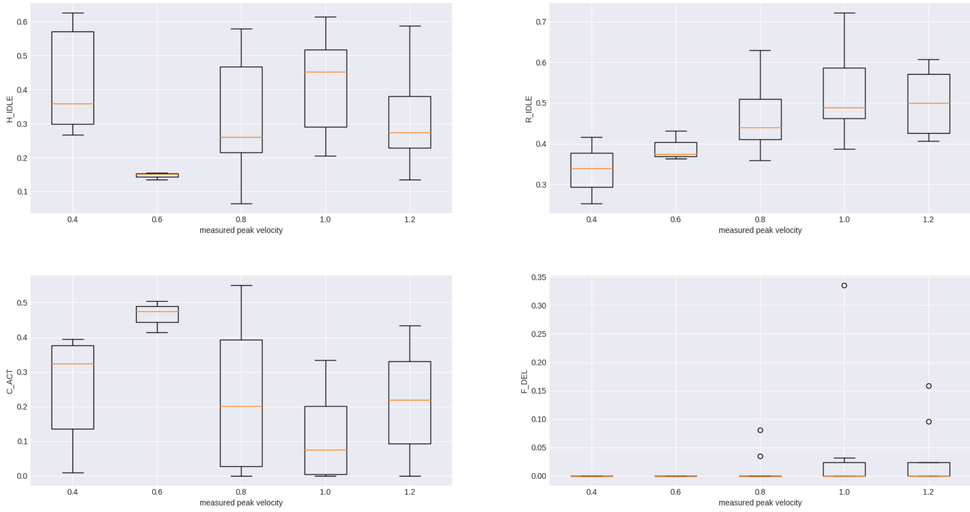


Figure 5.14.: The effect of Cartesian TCP velocity on H-IDLE, R-IDLE C-ACT and F-DEL.

Additionally a short questionnaire was used for each run of the three objects in the study. Although no statistically significant results can be expected for only three participants.

Figure 5.13 shows the before mentioned measures with the exception of F-DEL. F-DEL was omitted for the pre-study because there was no single handover with functional delay. The human idle time decreases for higher Cartesian peak TCP velocities until a velocity of $0.7m/s$. After that, it stays at around 20%. H-IDLE is positively correlated with subjective fluency. This indicates that fluency goes down for higher Cartesian peak TCP velocities of the robot giver.

The robot idle time goes up for higher Cartesian peak TCP velocities. R-IDLE did reversely correlate consistently with subjective fluency in [Hoffman, 2019]. Again indicating that higher velocities of the robot giver lead to a less fluent handover.

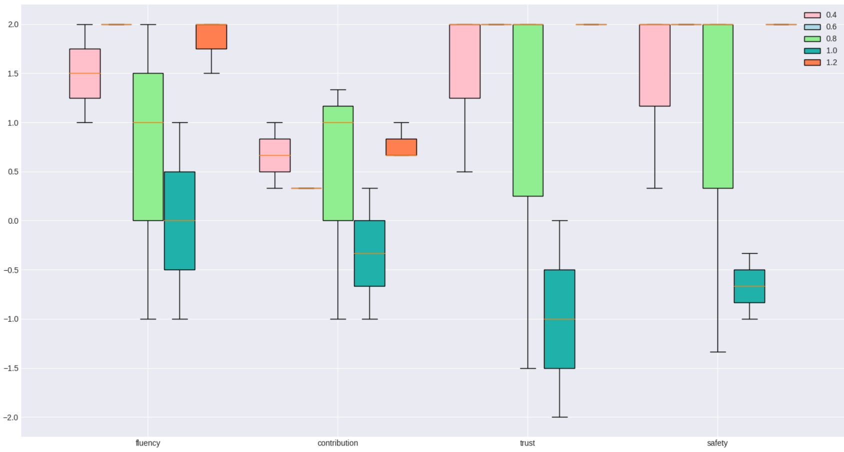


Figure 5.15.: The results of the questionnaire grouped by the Cartesian TCP velocity.

Although concurrent activity is not correlated with fluency, it is intuitively more efficient if both agents act at the same time. Interestingly C-ACT peaked at $0.7m/s$ and went down for slower and faster trajectories. Additionally, the variance is lowest for $0.7m/s$.

Figure 5.14 shows the H-IDLE, R-IDLE, C-ACT, and F-DEL results from the study. The very low variance for the $0.6m/s$ bucket for H-IDLE and C-ACT is due to the removal of the first handovers because of the training effect, as mentioned before. Furthermore, the variance of the remaining buckets is considerable and no clear trend is visible.

R-IDLE, on the other hand, shows a clear upward trend for faster robot trajectories. Again supporting the hypothesis that faster robot trajectories reduce the fluency of the handover.

Finally, F-DEL offers very interesting results. There are only a few handovers, which had functional delay. They only occurred for trajectories in the range of

$0.8m/s$ to $1.2m/s$. F-DEL is reversely correlated with perceived fluency, thus indicating that fast trajectories make the handover less fluent.

The results of the questionnaire are shown in Figure 5.15. As mentioned before the optioned ranged from *Strongly Disagree* to *Strongly Agree*. In Figure 5.15 *Strongly Disagree* is mapped to -2, *Agree* to 1, *Neutral* to 0, *Agree* to 1 and *Strongly Agree* to 2.

Unfortunately, the variances are too large to make any significant conclusions and should only be considered anecdotally. There seems to be a downwards trend for faster trajectories in all categories except the fastest velocity bucket $1.2m/s$. This is most likely because the robot did not always reach the desired $1.2m/s$ and the resulting slower handover trajectory then being grouped in the $1.0m/s$ bucket.

6. Conclusion

In this thesis, a study to investigate the effect of the Cartesian velocity of the robot giver's handover trajectory was implemented and conducted. Due to the current Covid-19 pandemic, the study could not take place in its originally planed form. Because of that, the results need to be considered cautiously.

Overall we could show that if only the total handover time is considered, faster trajectories result in shorter handover times in the tested range of Cartesian velocities. But when including the fluency of the handover, faster handover results in less fluent handovers. Additionally, when only considering concurrent activity, there was an optimal velocity. This indicates that the optimal velocity depends on the chosen metric. If a fast and fluent handover is wanted, a compromise must be made.

It was found out that the object type affects the overall handover time and there is some evidence that the human receiver adjusts the velocity of the hand to the robot's velocity.

6.1. Future Work

We believe that this topic is not yet fully explored. Repeating the study with a much larger number of participants would greatly improve the significance of the results. Also, a training period where the participants get used to the setup should be included.

As mentioned in Section 5.1 the PR2 had problems reaching the desired velocities. Either further fine-tuning the joint's velocity and acceleration limits to achieve faster trajectories or moving to a faster robot, like the Universal Robots UR5, would make the study more reliable. It would also be interesting to test higher velocities.

While observing the participants filling out the questionnaire it seemed, that there are a few too many questions for the number of times the questionnaire is filled out. For a future study, it should be considered shortening the questionnaire. It might also be a good idea to remove the table between the robot and the receiver, as it might act as a barrier making the participant feel safer. This would make it harder to measure if the human feels intimidated by the faster robot motions. The participant's safety should still be guaranteed.

Additionally, more complex objects with more varying weight and difficulty to grasp should be considered, maybe even including a task to be performed after the handover. And finally, the grasp position could be varied. In the case of this thesis, the screwdriver for example, was only grasped at the handle and the shaft was presented to the participants. One could compare whether there is a difference between presenting the handle or the shaft. Presenting the handle is less dangerous as well as easier to grasp, which would maybe make the handover more fluent and take less time overall.

7. Bibliography

[Bohren et al., 2011] Bohren, J., Rusu, R. B., Jones, E. G., Marder-Eppstein, E., Pantofaru, C., Wise, M., Mosenlechner, L., Meeussen, W., and Holzer, S. (2011). Towards autonomous robotic butlers: Lessons learned with the PR2. In *2011 IEEE International Conference on Robotics and Automation*. IEEE.

[Cakmak et al., 2011] Cakmak, M., Srinivasa, S. S., Lee, M. K., Forlizzi, J., and Kiesler, S. (2011). Human preferences for robot-human hand-over configurations. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE.

[Calli et al., 2015] Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., and Dollar, A. M. (2015). The YCB object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*. IEEE.

[Chan et al., 2012] Chan, W. P., Parker, C. A., der Loos, H. M. V., and Croft, E. A. (2012). Grip forces and load forces in handovers. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM Press.

[Coleman David, 2014] Coleman David, T. (2014). Reducing the barrier to entry of complex robotic software: a moveit! case study.

[Dragan et al., 2015] Dragan, A. D., Bauman, S., Forlizzi, J., and Srinivasa, S. S. (2015). Effects of robot motion on human-robot collaboration. In *Proceed-*

ings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction. ACM.

[Edsinger and Kemp, 2007] Edsinger, A. and Kemp, C. C. (2007). Human-robot interaction for cooperative manipulation: Handing objects to one another. In *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE.

[Eguiluz et al., 2017] Eguiluz, A. G., Rano, I., Coleman, S. A., and McGinnity, T. M. (2017). Reliable object handover through tactile force sensing and effort control in the shadow robot hand. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

[Garage, 2012] Garage, W. (2012). *PR2 User Manual*.

[Garage, 2021] Garage, W. ((accessed January 10, 2021)). Willow garage website. http://www.willowgarage.com/pages/pr2/overview.

[Görner et al., 2019] Görner, M., Haschke, R., Ritter, H., and Zhang, J. (2019). MoveIt! Task Constructor for Task-Level Motion Planning. In *IEEE International Conference on Robotics and Automation (ICRA)*.

[Guennebaud et al., 2021] Guennebaud, G., Jacob, B., et al. (2010 (accessed January 9, 2021)). Eigen v3. http://eigen.tuxfamily.org.

[Hendrich et al., 2014] Hendrich, N., Bistry, H., Liebrecht, J., and Zhang, J. (2014). Natural robot-human handover combining force and tactile sensors.

[Hendrich et al., 2016] Hendrich, N., Liebrecht, J., Bistry, H., and Zhang, J. (2016). Optimal interaction forces for natural robot-human object handover. Workshop on Personal Robot Interaction IROS.

[Hoffman, 2019] Hoffman, G. (2019). Evaluating fluency in human–robot collaboration. *IEEE Transactions on Human-Machine Systems*, 49(3):209–218.

[Huber et al., 2009] Huber, M., Radrich, H., Wendt, C., Rickert, M., Knoll, A., Brandt, T., and Glasauer, S. (2009). Evaluation of a novel biologically inspired trajectory generator in human-robot interaction. In *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE.

[Kuffner and LaValle, 2000] Kuffner, J. and LaValle, S. (2000). RRT-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*. IEEE.

[Lavalle, 1998] Lavalle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning.

[Lemaignan et al., 2016] Lemaignan, S., Garcia, F., Jacq, A., and Dillenbourg, P. (2016). From real-time attention assessment to "with-me-ness" in human-robot interaction. In *Proceedings of the 2016 ACM/IEEE Human-Robot Interaction Conference*.

[Mainprice et al., 2010] Mainprice, J., Sisbot, E., Siméon, T., and Alami, R. (2010). Planning safe and legible hand-over motions for human-robot interaction.

[Moon et al., 2014] Moon, A., Troniak, D. M., Gleeson, B., Pan, M. K., Zheng, M., Blumer, B. A., MacLean, K., and Croft, E. A. (2014). Meet me where i'm gazing. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. ACM.

[Nemlekar et al., 2019] Nemlekar, H., Dutia, D., and Li, Z. (2019). Object transfer point estimation for fluent human-robot handovers. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE.

[Ortenzi et al., 2020] Ortenzi, V., Cosgun, A., Pardi, T., Chan, W., Croft, E., and Kulic, D. (2020). Object handovers: a review for robotics.

[Paraschos et al., 2013] Paraschos, A., Daniel, C., Peters, J. R., and Neumann, G. (2013). Probabilistic movement primitives. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 2616–2624. Curran Associates, Inc.

[Ruppel, 2021] Ruppel, P. ((accessed March 06, 2021)). Tamsviz github. https://github.com/TAMS-Group/tamsviz.

[Ruppel et al., 2018] Ruppel, P., Hendrich, N., Starke, S., and Zhang, J. (2018). Cost functions to specify full-body motion and multi-goal manipulation tasks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

[Scholz, 2021] Scholz, B. (2019 (accessed January 9, 2021)). Moveit pr: add function to set cartesian speed of the end effector for a given trajectory. https://github.com/ros-planning/moveit/pull/1790.

[Strabala et al., 2013] Strabala, K. W., Lee, M. K., Dragan, A. D., Forlizzi, J. L., Srinivasa, S., Cakmak, M., and Micelli, V. (2013). Towards seamless human-robot handovers. *Journal of Human-Robot Interaction*, 2(1):112–132.

[Vannucci et al., 2018] Vannucci, F., Cesare, G. D., Rea, F., Sandini, G., and Sciutti, A. (2018). A robot with style: Can robotic attitudes influence human actions? In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE.

A. Appendix

Questionnaire

Participant: ____

Run: ____

	Strongly disagree	Disagree	Neutral	Agree	Strongly Agree
The human-robot team worked fluently together.	☐	☐	☐	☐	☐
The robot contributed to the fluency of the interaction.	☐	☐	☐	☐	☐
I had to carry the weight to make the human-robot team better.	☐	☐	☐	☐	☐
The robot contributed equally to the team performance.	☐	☐	☐	☐	☐
The robot was committed to the success of the team.	☐	☐	☐	☐	☐
I was the most important team member on the team.	☐	☐	☐	☐	☐
I trusted the robot to do the right thing at the right time.	☐	☐	☐	☐	☐
The robot was trustworthy.	☐	☐	☐	☐	☐
I feel uncomfortable with the robot.	☐	☐	☐	☐	☐
I feel safe working next to the robot.	☐	☐	☐	☐	☐
I am confident the robot will not hit me as it is moving.	☐	☐	☐	☐	☐
The robot was uncooperative.	☐	☐	☐	☐	☐

Figure A.1.: Questionnaire used in this study.

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Masterstudiengang Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel — insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen — benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Hamburg, den 31.03.2021

Jonas Tietz

Veröffentlichung

Ich stimme der Einstellung der Arbeit in die Bibliothek des Fachbereichs Informatik zu.

Hamburg, den 31.03.2021

Jonas Tietz