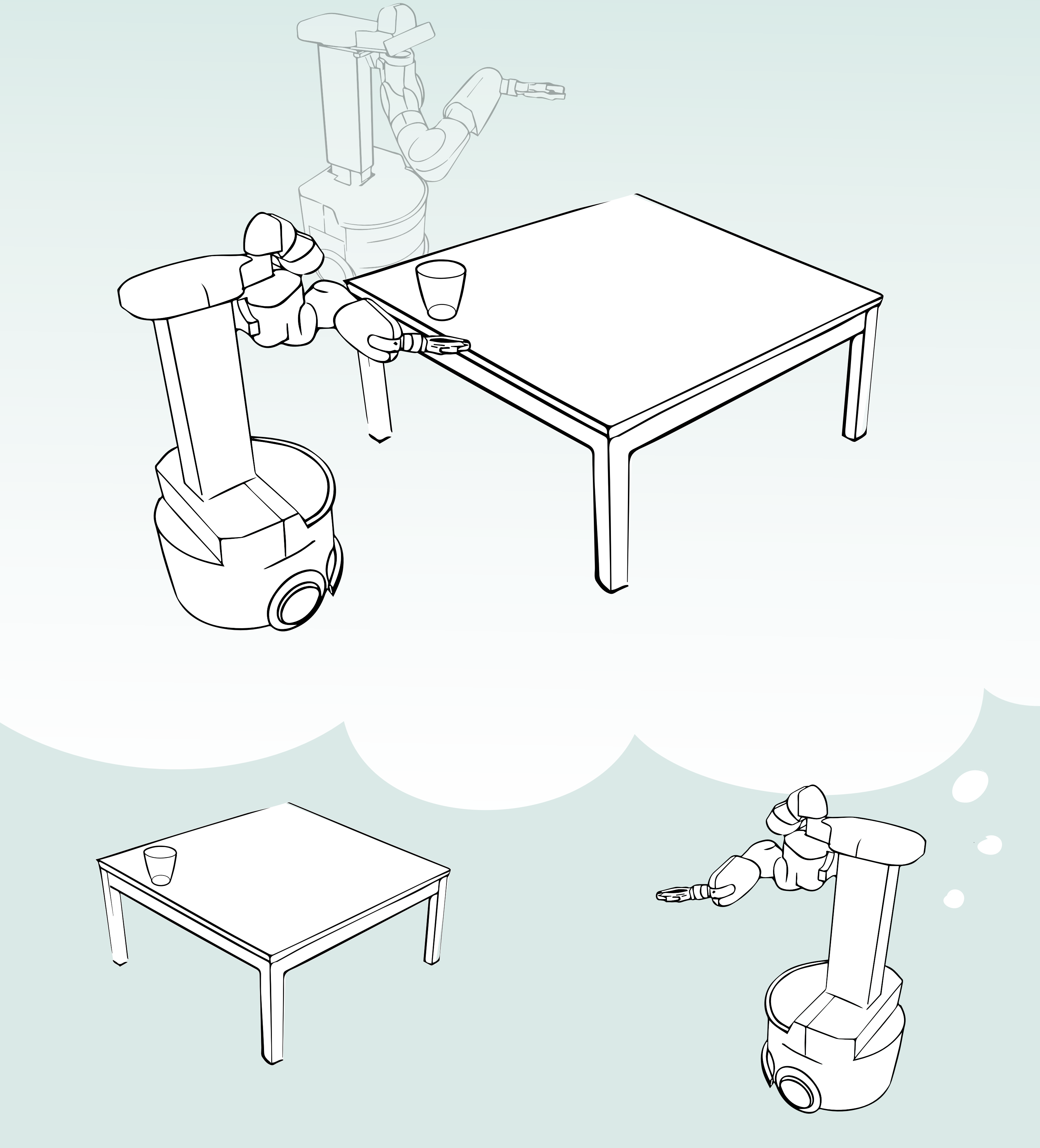
Which position do I pick, if I don't know how to do the task?

A model to predict advantageous base poses for semi-autonomous robots.

Karel Sirks



Which position do I pick, if I don't know how to do the task?

A model to predict advantageous base poses for semi-autonomous robots.

by

K. A. Sirks

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday November 27, 2018 at 10:00 AM.

Student number:	4096886	
Project duration:	December 22, 2017 – November 27, 2018	
Thesis committee:	Prof. dr. ir. D. A. Abbink,	TU Delft, supervisor
	Ir. J. G. W. Wildenbeest,	Heemskerk Innovative Technology

TUDelft

HiT
Heemskerk Innovative Technology

Contents

Which position do I pick, if I don't know how to do the task?

A model to predict advantageous base poses for semi-autonomous robots.

Karel A. Sirks*

Abstract

In our ageing society, the demand for care is increasing. Therefore, it is foreseen that robots can help elderly to keep living independently. However, due to the unstructured and dynamic environment, human assistance will often be required for manipulation tasks. While the robot can navigate autonomously, it does not know where to go for telemanipulation tasks.

Methods have been developed that position a robot for autonomous manipulation. But with telemanipulation, there are fewer acceptable base poses due to human and telemanipulation limitations. The objective of this study is to design and evaluate a positioning model, the Inverse Telemanipulation Capability Map (ITCM), which includes these constraints. A scoring metric is used based on manipulability and task visibility. Six participants did a pick-and-place task with the robot in a base pose used by an expert (Expert), in the lowest (ITCM-low) and the highest scoring base pose (ITCM-high).

The results show an increase in completion time of 71% from the ITCM-high to ITCM-low condition and the effort metrics are around one and a half times higher. Between the ITCM-high and Expert conditions no differences were found in the metrics. But participants also reported that the task was less or equally difficult in the ITCM-high condition.

It is concluded that the ITCM model is a good predictor for the telemanipulation advantageousness of a base pose and even the lowest scoring pose is acceptable. It can thus successfully be used to select navigational goals for semi-autonomous mobile manipulators.

— semi-autonomous – telemanipulation – mobile manipulators – positioning —

I. Introduction

People are getting older, but elderly are not getting less dependent on care [1]. To alleviate caregivers from mundane tasks, a mobile manipulator could be utilised as a home-care robot. Multiple of these type of robots are being developed, for example Cosero [2] and ROSE [3]. These robots have in common that they have one or two arms with seven degrees of freedom mounted on a height-adjustable platform and can operate semi-autonomously.

Due to the domestic environment, humans need to assist with manipulation tasks [3, 4]. To meet the care demand, a home-care robot must be able to manipulate a diverse set of objects and varying tasks. Although automation of manipulation tasks in domestic environments is being researched for more than a decade [5], not all possible domestic tasks have been automated. And would this be the case, then automation can still fail. Because humans intuitively understand how to do a task, they will remain needed to assist home-care robots by telemanipulation.

The robot's pose[1] is of great influence on the success of a telemanipulation task. It determines if the task can be reached and how the task is seen by the operator. Although robots can navigate autonomously in a domestic environment [6], for telemanipulation tasks the operator positions the robot. He or she selects a base pose on a map or drives the robot towards the task. To be successful, the human needs to know which regions of the robot's workspace are suitable for this task and to estimate how far away the robot is from the task. Robots are better suited for this task because they can measure the exact distance to objects and exactly remember workspace properties. Therefore a positioning method is needed that conforms to limitations of both the operator and the robot.

In previous work, mappings of the reachability and capability of the arm have been used to position the robot for autonomous manipulation. To our best knowledge, no research has been done regarding positioning a mobile manipulator for telemanipulation. A difference between autonomous and manual execution is that the task execution is not pre-determined with telemanipulation. Consequently, it is helpful if the robot's pose provides a variety of possible execution tactics. Methods to position robots have first been developed to position robots optimally in an assembly line [7]. In extension of these methods for fixed robots, methods have been developed to position mobile manipulators in order to to autonomously execute a task. Zacharias et al. [8] introduced the reachability map in

[1] position and orientation

which base positions were ranked. Their score was computed with Inverse Kinematics (IK) as the amount of possible orientations the end effector could reach a position. Because the amount of possible orientations determine the score, the orientation of the base is not variable. Based on the work of Zacharias et al., a method was developed by Abolghasemi et al. [9] that generates an approximation of the map that can take obstacles into account. Other methods extended on the work with on-line kinematic chain extension (e.g. with a tool) [10] and a task specific positioning method called Action Related Place (ARPlace) [11].

Besides inverse kinematics, ARPlace used an accurate simulation to determine if the task was practically reachable. Because care was taken in the simulation to accurately model the controllers, the robot's limitations were accurately represented in this method. However, the generated model was based on position input. It was thus only able to find a position of the robot with no specified orientation.

The amount of end effector movement options can be described by the manipulability score [12]. Using the manipulability to score base poses, the orientation of the robot can be selected as well. Whereas with the reachability score, the gripper orientations are used to score a position. Therefore no score is appointed to the grippers orientations, only the position. The Oriented Reachability Map (hereafter ICM, for inverse capability map), as presented by Vahrenkamp et al. [13], uses the same IK sampling as Zacharias et al. but with the manipulability score. The ICM is thus also able to pick an orientation for the robot. However, from tests with our implementation of the ICM, it is concluded that not all base poses in it result in a successful task, see Fig. 1 (and Appendix B for more). According to the IK analysis, the base pose results in a kinematically reachable task. This is not the case because the master and slave controllers are not taken into account.

The positioning methods for mobile manipulators have been developed for autonomous manipulation tasks. For telemanipulation a kinematically reachable pose, might not be reachable. Similar to the ARPlace method, a more accurate representation of the poses reachable with telemanipulation, can be obtained if the representation is obtained with the telemanipulation setup. Furthermore, the manipulability score can be a good indicator for the amount of tactic options.

The base pose determines task perception, which can influence the operators limited situational awareness. The lack of depth information via two-dimensional camera based visual feedback has been seen to reduce operator performance with a factor of two [14]. Because the camera images are displayed two-dimensionally, the operator is dependent on depth cues to perceive depth, e.g.: shadows of objects, motion parallax, relative height and occlusion [15]. By tilting the camera with 45° downwards, the operator can use these depth cues in both the robot's forwards and downwards direction. Concerning the camera's *yaw* angle, it has been found that discrepancies between the operators horizontal viewing angle and the steering input can cause errors in telemanipulation tasks [16]. Therefore base poses which enable the camera to look straight ahead avoid such errors. A visibility scoring method can be used in conjunction with the manipulability to prefer base poses with depth cues in two directions and with a congruent horizontal viewing and steering angle.

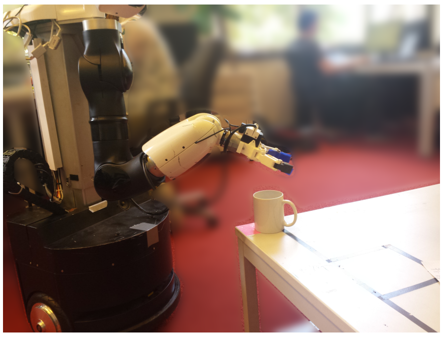


Fig. 1: Base pose option of the ICM that resulted in a task that was not executable with telemanipulation, see Appendix B for our implementation. (Irrelevant background blurred.)

The objective of this research is to develop and test a positioning method that can select a base pose from which an operator can perform the task fast and with little effort without having to reposition the robot. The possible base poses of this method will be based on the whole telemanipulation setup to ensure task execution, therefore it is called the Inverse Telemanipulation Capability Map (ITCM). These poses will be scored by the manipulability as well as the visibility of the task. This scoring method will favour poses from which a larger number of execution tactics are possible, have depth cues in two directions and with as less discrepancy between the viewing and the input angle. To investigate if the designed method is effective, a human factors experiment was conducted with a real world slave. Participants were asked to execute a pick-and-place task from three different base poses. The best of four base poses selected by an experienced operator, the Expert condition, the base pose with the highest score of the task specific ITCM (ITCM-high) and the base pose with the lowest score (ITCM-low). It is hypothesised that:

- the performance in the ITCM-high condition will

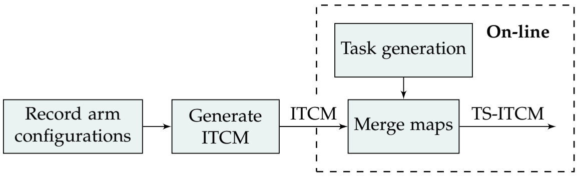


Fig. 2: Generation of task specific inverse telemanipulation capability map (TS-ITCM).

be greater and operator effort will be less than in the ITCM-low condition.

- the performance and effort will be equal or better in the ITCM-high condition than in the Expert condition.

In section II it is described how the ITCM model is designed, in section III the design of the experiment is detailed, in section IV the results of the experiment are reported, in section V these results are interpreted and discussed, in section VI the conclusions of the research are given.

II. Design of the positioning model

The goal is to have a model which can predict from which position the task can best be executed with telemanipulation. Because the robot's kinematics and telemanipulation settings do not change in operation, a pre-computed map can be used to represent the robot's capabilities to reach a fixed End Effector (EE) pose. Thereafter, the robot can use this pre-calculated map while on duty (on-line) to generate Task Specific maps (TS-ITCM) for any number of tasks, see Fig. 2.

To generate the pre-computed ITCM, the manipulability and visibility per End Effector (EE) pose is calculated based on the recorded configurations. But to know from which base positions the fixed EE pose can be reached and which combined manipulability and visibility score they have, the EE poses are inverted to base poses and rounded to a discretisation grid in the 'generate ITCM' step.

Then, on-line, EE poses are generated as a representation of the task. For each EE pose of the task, an ITCM is transformed according to that EE pose. The transformed ITCMs of all the EE poses are then merged to a TS-ITCM as described in the 'Merge maps' step. The base pose with the highest score is then selected as base pose for the task.

A. Record arm configurations

To get a map of the arm's capabilities, an operator used the master device to sample the workspace of the slave. The same real world telemanipulation setup was used for gathering the data as for the experiment of this research. Because the experiment uses one task, only a part of the workspace needed to be sampled: the task can be described solely with side grasps at a height of 41 cm. To reduce the number of not relevant configurations, the commands of the master device were fixed at a height such that the slave's EE would be 41 cm high and the roll and pitch of the EE were fixed to be zero. Also the height of the robot's lift was fixed to 91 cm, because that height resulted in the highest score with the IHCM model (see Appendix B). During 1 h, samples of possible configurations were recorded with the rosbag [17] system. The result is seen in Fig. 4

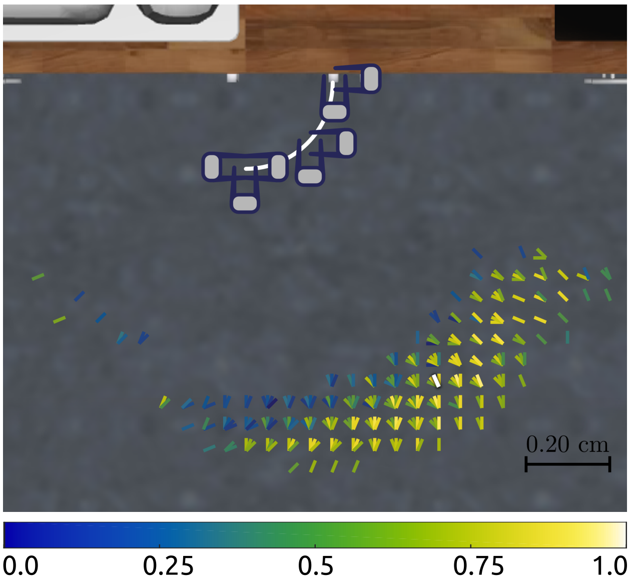


Fig. 3: A kitchen cabinet needs to be opened (20 cm radius). Along the path of the cabinet's handle (white) EE poses are generated at three positions with varying orientations. Each EE pose is a purple 'U' shape with a grey area near the wrist. Based on the generated task and the pre-computed ITCM, the Task Specific Telemanipulation Map (TS-ITCM) is displayed. The possible base poses are denoted with a short line, coloured according to their score. Where the lines meet is the possible position. The direction of the line is the robot's forward direction.

B. Generate ITCM

In this section it is explained how the recorded arm configurations are used to generate the ITCM.

Based on the kinematic model represented in the URDF[2] of Marco[3], a transform was calculated per set of recorded joint angles. Each transform represents the pose of the EE in Marco's base frame. Poses of the EE which are 2.5 cm higher or lower than the reference height, were removed. Using the MATLAB toolbox of Corke [19], the Jacobian[4] was calculated per pose,

[2] universal robot description format

[3] prototype robot of PAL's TIAGo robot [18]

[4] The Jacobian is the derivative of the arm's Kinematic Model with respect to the generalised coordinates and is also called the Instantaneous Kinematic Model.

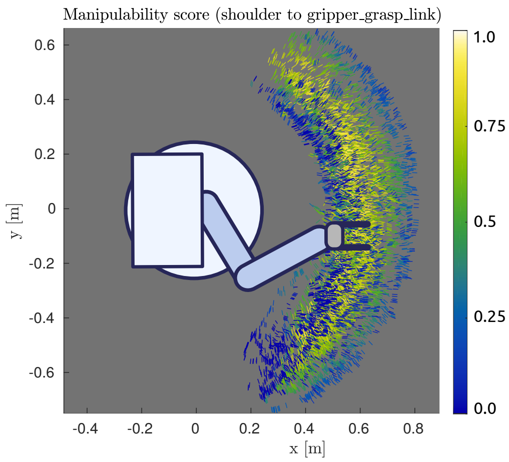


Fig. 4: Recorded end effector poses rated with the joint limit penalized manipulability score. For clarity, only every tenth pose is shown.

using the joint angles and the kinematic model.

The fraction of the smallest (σ_{min}) over the largest singular value (σ_{max}) of the singular value decomposition of the configurations Jacobian is used as the manipulability score as presented by Yoshikawa [12].

$$\text{score}_{\text{manip}} = \frac{\sigma_{\text{min}}}{\sigma_{\text{max}}} \quad [12] \tag{1}$$

The manipulability score is penalised if the configuration is close to joint-limits, the implemented penalization was presented by Tsai [20, p. 114] and used $k = 10^6$.

The visibility score indicates the closeness of the head camera's orientation to a goal orientation. The goal orientation is the *yaw* and *pitch* in which the head camera would look at the EE. To calculate the visibility score, a direction vector (*dir*) between the head's camera and the configuration's EE is calculated in the shoulder frame. The shoulders x-axis points forward, the z-axis points upward and was chosen because it does not translate relative to the camera and does not rotate relative to the base. It is assumed that the pitch and yaw joints coincide with the position of the camera, thus no displacement is taken into account as a result of the rotation. Looking downward at the task with a 45° angle allows the operator to perceive depth cues in the horizontal as well as a vertical plane, as such $pitch_{\text{goal}} = 45^o$. A *yaw* angle of 0 is advantageous because the master input is relative to robot's base frame. With a large yaw angle, operators need to rotate the master input mentally to match the rotated view of the scene. The lower and upper bounds of the *pitch* and *yaw* are the joint limits of the camera's head with as extra margin the field of view of the camera.

$$yaw = \text{atan}\left(\frac{dir_y}{dir_x}\right) \quad pitch = \text{atan}\left(\frac{dir_z}{\sqrt{dir_x^2 + dir_y^2}}\right) \tag{2}$$

$$\text{score}_{\text{yaw}} = \begin{cases} 1 - \left|\frac{yaw - yaw_{\text{goal}}}{yaw_{\text{ub}} - yaw_{\text{goal}}}\right| & \text{if within bounds} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$\text{score}_{\text{pitch}} = \begin{cases} 1 - \left|\frac{pitch - pitch_{\text{goal}}}{pitch_{\text{ub}} - pitch_{\text{goal}}}\right| & \text{if within bounds} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$\text{score}_{\text{visib}} = \left(\text{score}_{\text{yaw}} + \text{score}_{\text{pitch}}\right) \cdot \frac{1}{2} \tag{5}$$

The manipulability and visibility score need to be combined in one metric. Because the base poses can only be sorted according to one metric at a time. Both the visibility and the manipulability score are scaled independently such that the base poses have values from 0 to 1. Therefore, an average of both scores can be made without biasing one score over the other.

$$\text{score} = \left(\text{score}_{\text{manip}} + \text{score}_{\text{visib}}\right) \cdot \frac{1}{2} \tag{6}$$

The EE poses in the base frame are then inverted to base poses in the EE frame. Thereafter the inverted poses are discretised to positions in a square grid of 5 cm and orientations divisible by $\frac{1}{8}\pi$ rad. This discretization is necessary to let base positions overlap precisely. Each pose and corresponding score was stored as a row in a two-dimensional array. This array is the Inverse Telemanipulation Capability Map.

C. Task generation

As first part of the on-line process, a task representation is generated. Although it is not known how the task is to be executed, it is known where the gripper needs to be. For example, if a cupboard needs to be opened, positions can be selected along the handles path. These positions need to be on a similar grid as the base poses have been discretised on. The distance between the gripper positions thus need to be a multitude of the discretization grid size. In that way if the maps overlap during merging, the overlapping poses are at the exact same position. In Fig. 3, it can be seen that the positions of the gripper do not coincide in the middle of the fridge door's arc.

Because it is not known from which direction the operator will approach the grasp, all possible orientations are generated. These are however limited to be either perfectly perpendicular to or in the direction of the grid of gripper positions as can be seen in Fig. 3.

Additionally a pre-grasp pose can be used as well. Because it is often not possible to directly grasp the object, the object is approached in a linear manner. The pre-grasp is placed 5 cm (the discretisation distance) before the grasp.

Due to the constraints on selecting gripper poses, the task may be a couple of centimeters different or actually use other gripper orientation. But it does enable an computationally efficient combination of the ITCMs and a precise representation of the task is not that useful as it will probably not be executed in that way.

D. Merge maps

With the gripper poses generated and the ITCM calculated off-line (see Fig. 2), the task specific ITCM (TS-ITCM) can be generated.

From the ITCM it is checked which poses are feasible, these can only have an angle around the vertical axis of the world frame (otherwise the robot would not be vertical) and the vertical axes of the robot and the world frame should be parallel. The height of the base should be zero, i.e. on the floor. But the lift of Marco gives a margin of 35 cm.

For each gripper pose, all the possible base poses are transformed to be relative to a single fixed coordinate frame. Then it can be compared how many grasps are possible from which base pose.

First, for each gripper position (*pos*) and their orientations (*or*) the grasp and pre-grasp gripper pose are merged. Only the base poses are kept which occur in both the grasp and pre-grasp ITCM-set (intersection), see equation 7. The lowest score is selected because that represents the minimum telemanipulation capability from that base pose to the EE.

$$\mathrm{ITCM}_{pos_{or}} = \mathrm{ITCM}_{\mathrm{grasp}} \cap \mathrm{ITCM}_{\mathrm{pre\text{-}grasp}} \tag{7}$$

Secondly for each grasp position (*pos* of *m*), all (*n*) grasp orientations are merged. Since not all the orientations need to be reachable from the base poses (candidate base poses), all base poses are kept (union) and for equal base poses the highest score is kept (equation 8).

$$\mathrm{ITCM}_{pos} = \bigcup_{or=1}^{n} \mathrm{ITCM}_{pos_{or}} \tag{8}$$

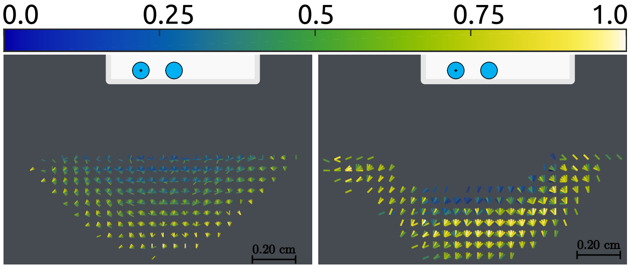
Lastly, to generate the TS-ITCM base poses that are present in ITCM_{pos} of all (*m*) positions are selected. For the remaining base poses, the lowest score is used. This is because the most difficult to reach EE position will be the most noticeable.

$$\text{TS-ITCM} = \bigcap_{pos=1}^{m} \mathrm{ITCM}_{pos} \tag{9}$$

The base pose with the highest score is selected as the best base pose, demonstrated in Fig. 3.

E. Case study

In Fig. 5, a comparison is shown between the IHCM (see Appendix B) and ITCM method. The IHCM is an implementation of the ICM with the visibility score. The task is a pick-and-place task, with thus two positions 15 cm apart. Per position three orientations are used in the task, frontal, left and right side. The case study in the next part of this paper uses the same task as shown in the figure.



(a) TS-IHCM **(b)** TS-ITCM

Fig. 5: The same pick-and-place task used to generate a map with the IHCM and ITCM method. The pick and place positions are denoted with the blue circles. The coloured lines can be interpreted as in Fig. 3.

It can be seen that the TS-IHCM shows a lot more base pose options than the TS-ITCM. The main difference between the IHCM and ITCM is the way in which they gather reachable EE poses. It can thus be concluded from Fig. 5 that a large number of kinematically reachable poses, are not reachable with telemanipulation.

III. Methods

A. Participants

Eight participants volunteered to take part in the experiment, of which two were excluded during the experiment. One of the two excluded participants continued to hit the table quite hard. The other was excluded because the robot had various problems and the experiment could not be rescheduled. The six participants which completed the experiment were aged 23, 23, 23, 27, 28 and 65. Of these participants, one was female and one other had extensive experience with this telemanipulation setup.

No compensation was given to the participants for partaking in the experiment. The participants were briefed about the purpose of the experiment, see Appendix J, they could also see that the pose of the robot with respect to the task changed per condition. All participants gave their informed consent prior to the experiment. The Human Research Ethics Committee of the Delft University of Technology approved of this research.

B. Apparatus

The experiments were performed on a telemanipulation setup consisting of a Geomagic Touch Omni (Omni) on the master and a prototype of the TIAGo robot (Marco) on the slave side, see Fig. 6. The Omni

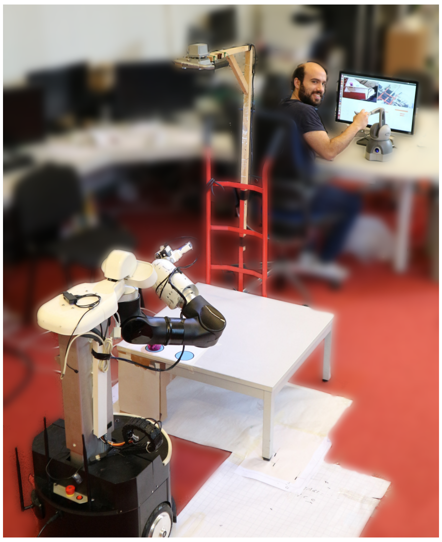


Fig. 6: Experiment setup with the operator holding the master device, Marco in front of the table with the cup and the markers and the table placed on the template. (Irrelevant background blurred.)

gave position and orientation commands to Marco and Marco's force-torque sensor was used to give translational haptic feedback to the participant with the Omni. The master and the slave were connected over WiFi and communicated over ROS topics. Data was recorded with the rosbag system [17].

Marco has a seven degrees of freedom arm, which sits on a lift together with the overview camera, on the gripper an additional camera is mounted. Images of both cameras are displayed simultaneously.

The table used is 37 cm high and 75.5 cm wide, the cup is 9 cm high, the height of the middle of the cup is thus at 41.5 cm from the floor. The cup itself is 50 g, weights are added to the bottom of the cup to make the cup less prone to tumble. This resulted in an total weight of 325 g. As can be seen in Fig. 6, a lamp (LED, 10 W, 750 lm) is added above the cup.

C. Experimental design

1) Task definition

The participants were asked to pick up an object and place it on a marked surface, with the telemanipulation setup, see Fig. 6. From the starting configuration of the arm, the participants picked the cup and placed it 15 cm along the table's edge. After releasing the cup, the participants retreated from the cup towards the original position. The participants were asked to do the task as fast as possible while ensuring the object ends up within the goal area. If the cup was placed partly outside of the black circle, the researcher asked the participant to reposition the cup after the cup left the gripper. The task failed when the cup was no longer in reach of the robot (e.g. it has fallen or it was pushed away while grasping).

2) Experimental conditions

The independent variable was the method with which the robot was positioned, it is either:

- Expert: see paragraph below
- ITCM-high: the highest scoring TS-ITCM base pose
- ITCM-low: the lowest scoring TS-ITCM base pose

The order in which the conditions were presented was permuted per participant, to counterbalance learning effects and carry-over effects between conditions. The distance and angle of the robot as seen from the pick up location is shown in Table 1.

The participants were not aware in which order the positioning methods were used to position the robot. The height of the robot's lift is the same for all conditions, the *yaw* and *pitch* of the overview camera is fixed per condition to show the pick and place markers at the same position on the cockpit's screen. In Appendix I these views are displayed.

Table 1: Distance in the forward (x) and left (y) direction and rotation of the task frame .

Condition	x [m]	y [m]	θ [^{o}]
Expert	0.88	0.20	2.9
ITCM-high	0.75	0.15	0
ITCM-low	0.55	−0.45	−90

The Expert condition represents manual position selection. The base pose of the Expert condition is fixed such that it is not dependent on the navigation skill of inexperienced participants and there is no risk of the participants copying positions from the ITCM model. An operator – experienced with this setup – was asked to perform the task and drive the robot as well. The robot started 2 m away from the table, the operator drove the robot to a position from where the operator thought he would be able to perform the task with telemanipulation. The operator was not allowed to reposition the robot. Out of four attempts, one failed, because the robot was to far from the task to reach it. From the three successful attempts, the position with the fastest execution time was chosen (28.73 s) to represent the Expert position. From the other positions the operator took 29.75 s and 31.98 s. To get a precise measurement of the Expert condition's base pose, the table pose measurements were averaged over 35 s.

3) Experimental protocol

In Fig. 8, the experimental protocol is depicted. Prior to the experiment, the participants were familiarised with the telemanipulation setup and the task. For

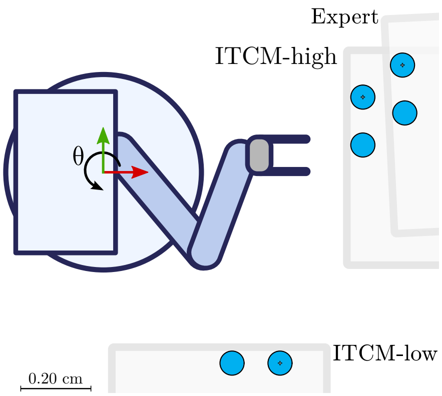


Fig. 7: In the experiment, the task moved instead of the robot. The positions of the table are shown for each condition. The x-axis of the `base_footprint` frame is displayed with the red arrow and the y-axis with the green arrow. The black arrow displays the positive rotational direction of θ.

the familiarisation phase, the table was positioned between the table positions belonging to the ITCM-high and Expert conditions. The robot's two-dimensional lidar was used to evaluate if the correct base pose was attained (see Appendix A).

The workspace was explored by getting force feedback from contact with the table and by seeing the response when the edge of the robot's workspace is approached. Thereafter the researcher performed the task as example for the participant, repeating the strategy as briefed in the participant instruction (Appendix J). Then the participant performed the task, while receiving strategy tips from the researcher, until seven repetitions were successful (not failed).

After the familiarisation phase, each condition was administered. The table was positioned for each condition with the help of a template. The researcher performed the task, while repeating the strategy verbally. Thereafter the participant got seven repetitions to practice. After the practice, seven successful repetitions were recorded for analysis, the amount of failed trials was noted down. Lastly, for each condition the NASA-TLX [21], difficulty score and open questions (Appendix H) were administered.

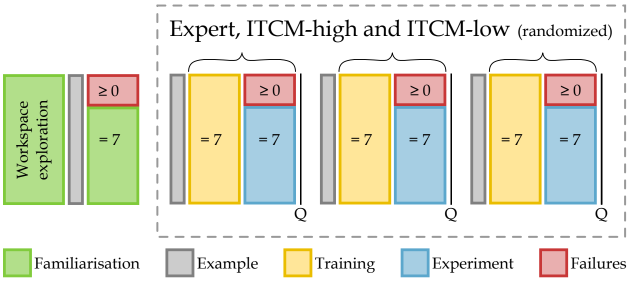


Fig. 8: Experimental protocol. The letter 'Q' denotes when the NASA-TLX, difficulty question and open questions are administered to the participant.

4) Metrics

Metrics are used to interpret the measured data and look for effects of the different positioning methods on task performance and operator effort.

The task performance is defined as the task completion time (*tct*): the time from the moment the coupling button is pressed up until the gripper is 0.30 m away from the cup-place-position before the gripper is reset.

The following metrics are used to evaluate the operator effort of which the first two are subjective and the last three are objective measures:

– Self reported difficulty score based on the agreement or disagreement to the statement whether 'it was difficult to execute the task'.

wwl Self reported weighted workload [%] as measured with NASA-TLX [21].

csa Cumulative steering angle [rad]. The total amount of rotation that the master device was steered by the human. As used by [22].

cst Cumulative steering translation [m]. The sum of differences in consecutive master positions.

sal Spectral arc length [-] as presented by [23], *sal* is used to evaluate the smoothness of the translational velocity of the master device.

5) Data analysis

A within subject design has been used for the experiment. Experimental conditions were compared on basis of populations, assuming a normal distribution and homogeneity of variance. For all metrics, except the difficulty score, statistical analysis has been done on the means of all subjects' repetitions per condition.

The input the participant gives on the master device is initially sampled at around 1 kHz. It is downsampled by first filtering with a second order Butterworth filter with a cut-off frequency of 50 Hz, whereafter it is interpolated to a 50 Hz signal. A cut-off frequency of 50 Hz is selected because voluntary human contractions are reported to go up to 25 Hz [24].

The kolmogorov-Smirnovtest did not reject any of the metrics to be normally distributed. A one-way repeated measures anova was used to find if there is a difference between conditions. Differences were deemed significant if the probability that the difference could come from the same population was smaller than 5% ($p < 0.05$). A one-tailed paired sample t-test was used as post hoc test with a Bonferroni correction for multiple testing error.

IV. Results

Expert and ITCM-low conditions resulted in three failed repetitions and the ITCM-high condition re-

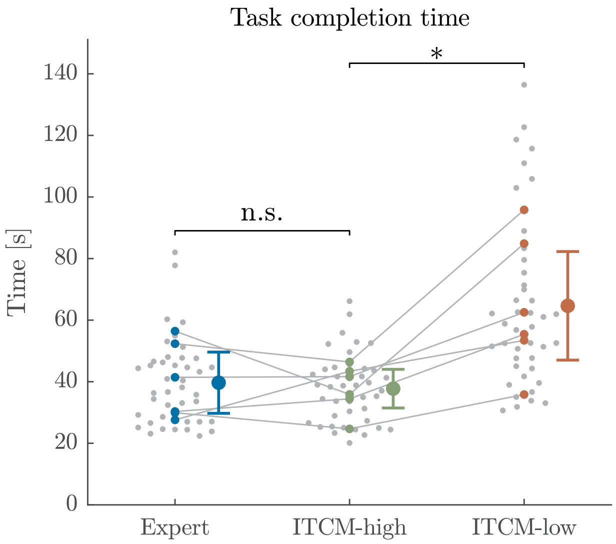


Fig. 9: Task completion time (*tct*) for six participants, seven repetitions and all three conditions. The *tct* is significantly longer in the ITCM-low condition as compared to the ITCM. No significant difference has been found between the Expert and ITCM-high condition. The large dot with the whiskers is the mean of the participants means with the 95 % confidence interval. The coloured dots represent the mean per participant and the grey lines link means of one participant and each grey dot represents a repetition. The horizontal bars denote a post-hoc tests, where 'n.s.' stands for not significant, '*' for $p < 0.05$, '**' for $p < 0.01$ and '***' for $p < 0.001$

sulted in one failed repetition during the experiment. During training, there was one failed repetition in the Expert condition and eleven in the ITCM-low condition.

A. Performance

As performance metric the task completion time (*tct*) is used, see section III.C.4. In Fig. 9, the mean *tct* is almost two times as high in the ITCM-low compared to the ITCM-high condition, this difference is significant. Additionally, the *tct* increased for all participants in the ITCM-low condition.

No significant difference has been found between the *tct* of the ITCM-high and of the Expert condition. The standard deviation of the ITCM-high condition is almost half that of the Expert condition, as can be seen in Table 2.

B. Effort

In this section the metrics concerning operator effort as presented in section III.C.4 and the responses on the closed and open questions are discribed. These can be divided in subjective (*wwl*, reported difficulty and responses to open questions) and objective measures (*sal*, *csa*, *cst*).

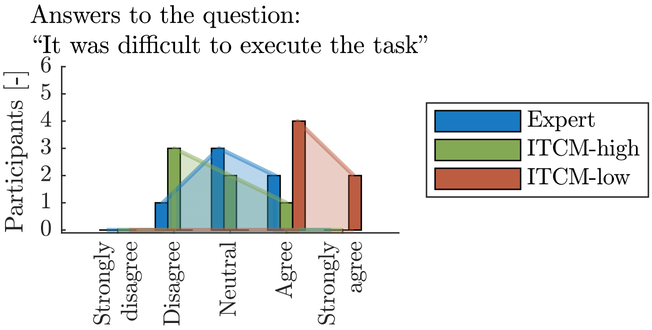


Fig. 10: All six participants reported the difficulty of the task for each condition. The shaded areas and lines are added for illustrative purposes. The task is rated more difficult from the ITCM-low base pose than from the ITCM-high base pose. From the ITCM-high base pose the task is rated less difficult than from the Expert base pose.

1) Subjective effort measures

The participants' agreement to the statement 'It was difficult to execute the task' is shown in Fig. 10. All participants reported that the ITCM-low was more difficult than the ITCM-high condition. Between the Expert and ITCM-high condition, half of the participants reported similar difficulty and the other half reported that Expert was more difficult.

The reported weighted workload shows no difference between conditions, see Table 2.

Responses to the open questions are aggregated in Appendix H. A positive response was that two participants reported that the ITCM-high condition resulted in an easier task. In the other responses two participants said that the cup is easily pushed outside the robot's workspace in the Expert condition. Of two participants, one found in the Expert and the other in the ITCM-low condition that the depth information was limited because the arm blocked the view on the arm's shadow. Lastly, four participants responded that the ITCM-low condition was made hard because the rotation between the master input and the camera's viewpoint.

2) Objective effort measures

The amount of steering input in rotational (*csa*) and translational (*cst*) sense is significantly larger in the ITCM-low condition than in the ITCM-high condition (see Fig. 11 and Table 2). No significant difference has been found between the *csa* and *cst* of the Expert and ITCM-high condition.

The smoothness of the steering input, as calculated with the spectral arc length (*sal*) shows no difference for the different conditions, see Table 2.

Table 2: Descriptive and inferential statistics of all metrics except the reported difficulty score. Significant differences are printed in bold.

Metric	Descriptive				ANOVA	Post-hoc	
	Expert	ITCM-high	ITCM-low			ITCM-high – ITCM-low	Expert – ITCM-high
tct	39.68(12.43)	37.74 (7.84)	64.65(22.03)	[s]	$F(15) = 5.78$, $\mathbf{p < .05}$	$\mathbf{p < .01}$ $r = .67$	$p = .62$ $r = .10$
wwl	52.11(21.64)	46.33(18.21)	68.00(12.94)	[%]	$F(15) = 2.34$, $p = .13$	–	–
csa	7.55 (2.22)	7.96 (1.45)	14.20 (2.80)	[rad]	$F(15) = 16.8$, $\mathbf{p < .001}$	$\mathbf{p < .001}$ $r = .84$	$p = .36$ $r = .12$
cst	208.27(43.94)	226.46(31.26)	304.82(70.45)	[m]	$F(15) = 6.02$, $\mathbf{p < .05}$	$\mathbf{p < .05}$ $r = .62$	$p = .21$ $r = .25$
sal	-18.93 (5.95)	-18.74 (3.05)	-20.99 (6.05)	[-]	$F(15) = 0.35$, $p = .71$	–	–

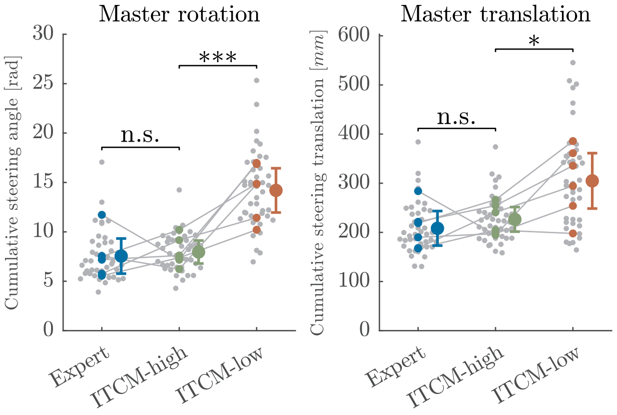


Fig. 11: Rotational and translational steering input of the master. Significant differences can be seen between the ITCM-high and ITCM-low conditions. See Fig. 9 how to read the figure.

V. Discussion

To enable short term introduction of home-care robots [4], a positioning model for telemanipulation tasks has been designed for semi-autonomous mobile manipulators. The designed method uses recorded data from the telemanipulation setup in order to accurately predict the task's reachability. This model is called the Inverse Telemanipulation Capability Map (ITCM). Based on a collection of gripper poses a Task Specific ITCM (TS-ITCM) can be calculated on-line.

The designed ITCM method is based on the ARPlace method [11] and the ICM method [13]. Like the simulation ARPlace uses, the ITCM uses an recording of the slave's workspace to get an accurate map. However, since the map representation is similar to the ICM, the ITCM is not task specific and can select a base orientation and lift height. To cater telemanipulation tasks' unpredictability, the ITCM accounts for multiple EE orientations per position in the task and even pre-grasps can be taken into account. Furthermore, the ITCM scores base poses using the visibility of the task and manipulability.

Contrary to related work, this research designed and tested a positioning method for telemanipulation. In this related work, autonomous manipulation was tested in the real world [11] or in simulation [8, 9, 10, 13]. The performance of these methods was evaluated with computation time and success rate. However, with an operator, different metrics become relevant.

An experiment was set up to test if the ITCM method can predict suitable positions for telemanipulation. It was investigated if this method is able to differentiate advantageous from less advantageous base poses. Also the execution with telemanipulation was compared between the ITCM selected and the manually selected pose. Three conditions are compared: an expert selected pose (Expert), the lowest (ITCM-low) and highest (ITCM-high) scoring pose of the TS-ITCM.

The results confirm the hypothesis that the ITCM-high condition has higher performance and lower effort than the ITCM-low condition. The task completion time increased with 71% from the ITCM-high to the ITCM-low condition. The operator effort was lower in the ITCM-high than in the ITCM-low condition: cumulative steering angle 78% and cumulative steering translation 34%. No difference has been found in the weighted workload or spectral arc length between the two conditions. Although the steering input in cumulative steering angles and translations seems to vary as much as the task completion time (see Appendix F), no difference has been found in the steering smoothness. It seems that similar steering tactics were used across conditions. Thus the increase of control effort stems from a longer duration of control. It can be concluded that the combined manipulability and visibility score accurately predicts which base pose is better suited. Furthermore, it can be deduced that the task is possible from all base poses in the TS-ITCM as it is possible to do the task from the worst scoring base pose.

The hypothesis can be confirmed that performance and effort are equal or better from the ITCH-high than the Expert condition. Because the results show no differences between any of the metrics of these conditions. It was conspicuous that the base poses of both conditions were just just 14 cm apart and the *yaw* angle difference was only 2.8°, while the TS-ITCM was 1.2 m long and 40 cm wide. That both conditions poses are close, suggests that the used scoring method results in a similar pose as human preferences. It can be concluded that the ITCM method is able to predict a base pose which is a good alternative for manual placement.

In this experiment some factors introduced variance in the resulting metrics. Firstly, the participants were instructed to correct the cup if the researcher stated that it was not in the circle, not all repetitions of the task were the same. Secondly, some participants had none and some participants had telemanipulation experience. Lastly, varying latency in the WiFi signal and variability in the force-torque sensor could not be controlled during the experiment. Nonetheless, the results clearly indicate the importance of base pose selection and the capability of the designed ITCM model.

Overall, it can be concluded that the ITCM method predicts advantageous base poses for telemanipulation.

This means that for semi-autonomous mobile manipulators, the operator does not need to position the robot anymore. It saves time and prevents operator error. The applicability of the method is not limited to domestic environments. However, the robot must be semi-autonomous, otherwise the robot's workspace could not be recorded. In that case a method for autonomous grasping would be better suited.

The ITCM model is also beneficial for autonomous tasks on a semi-autonomous robot. Because if the autonomous task fails, the robot is already in a good position for telemanipulation. With the ITCM model the need to position and reposition the robot for a telemanipulation task is thus removed.

With the ITCM model, the robot and the human supplement each other's skills. A robot is good in measuring its position and computing the TS-ITCM and a human is good in doing the manipulation task.

A limitation of the way the ITCM is merged in a TS-ITCM is that it constitutes of discretised grids. Therefore points only really are at the same points if rotated with 90° and/or translated with a factor of the grid's distance. Therefore, only side and top grasps can currently be merged, sampling more EE poses is thus not useful with this merge method.

To be able to generate a task, a path and a starting location are needed. While this information is needed for autonomous manipulation, it does not mean that the task can be done autonomously.

The calculation of the visibility score is under the assumption that the camera's position is fixed. The consequential error in position is not more than 10 cm, but it does not influence which base pose has the highest score.

Because the ITCM is based on recorded configurations, changes in the robot setup require re-recording of the workspace. The same applies to telemanipulation settings; the size of the usable workspace of the slave can then increase or decrease.

A. Future work

In future work, a task generation method could be made based on sensor data and natural human assignments. In this experiment parameters were used to generate a task on which the TS-ITCM was based, but in domestic environments the robot needs to rely on sensor input. To give tasks to a home-care robot interfaces have been developed that use speech [25], a point-and-click interface [26, 27] or gesture recognition [28]. To make these methods suitable for telemanipulation, instead of selecting one orientation the not possible orientations need to be identified per grasp position.

For this experiment the robot's workspace was recorded manually. However, it could also be done entirely in simulation, with accurate joint-limits and self collisions of the master and slave. Or with the the real slave and master, but aided by the master devices actuators. In this way the workspace can be ensured to be sampled uniformly and completely.

The maximum load of a robot arm is dependent on its configuration, by using the dynamic manipulability [29, 30] to calculate base poses' scores for different loads, maps could be made. When a task is generated the anticipated load could then be taken into account when selecting the weight specific ITCM and generating the TS-ITCM. The limitation of four grasp orientations for side and top grasps per task position can be expanded to six by using an equilateral triangle instead of a square discretisation grid. In that way tasks could be represented more naturally. If the robot is capable of detecting the object and the grasp, a scoring model for grasps (see for example: [31, 32]) could be used and the score could propagate to the base options.

For mobile manipulators which have a non prismatic lift joint, taking the lift height into account is not as straightforward. Due to the additional horizontal displacement, the orientations of the base poses in the map will not meet in positions as ordered in a grid. Therefore the merge method used here will not work. To prevent this, the tool extention method presented by [10] can be used. Instead of attaching a tool to the EE, a variable translation is added to the base of the arm. This will come at computational cost, as a six dimensional array is implemented.

Other use of the ITCM could be to aid the operator in picking a good base position. Or the other way around, visualising the workspace from the current base pose.

For sequential tasks, the ITCM method can be further developed to find if there is a set of base poses from which both tasks can be executed. In this way repositioning between tasks can be negated.

For a future experiment, it is interesting to compare the robustness of expert positioning to positioning with the ITCM model. For tasks with a lower amount of base pose options, it could be tested if an expert is able to find a possible position at all and if the base pose predicted by the ITCM model to be advantageous results in an executable task.

The presented model is only applied with a telemanipulation setup in which the master's position is statically linked to the slave's position. Meaning that the same master pose always results in the same slave goal pose. Although the configurations can be recorded in this way excellently, the resulting map will contain more base poses because of the less strict telemanipulation constraints. For future research, it can be investigated if other master slave relations have influence on the predictive capabilities of the ITCM model.

VI. Conclusion

In this study, a model (TS-ITCM) to select an advantageous mobile manipulator position for telemanipulation has been designed. It has been investigated if this model can replace positioning by an operator and if there actually is a difference within the base poses from which the task is reachable. Participants executed a task with a physical slave in three conditions: driven by an expert operator to a position, the highest and lowest scoring pose in the TS-ITCM. The presented model can be used to position a mobile manipulator such that an operator can do a task via telemanipulation.

Based on the experimental conditions tested, we can conclude that:

- The highest scoring base pose of the TS-ITCM resulted in similar performance and operator effort as the position selected by the expert. Participants regarded this pose as easier than the expert selected pose.
- The highest scoring base pose of the TS-ITCM resulted in significantly better performance and effort than the lowest scoring. The ITCM model is thus effective in predicting how suitable a base pose will be for telemanipulation.
- The lowest scoring TS-ITCM base pose resulted in a do-able task. This suggests, together with the validity of the scoring model, that from all base poses in the TS-ITCM the task is do-able.

The experiment shows that the base pose influences the telemanipulation performance and operator effort. The base poses selected by the ITCM result in a reachable task and that the scoring model of the ITCM accurately predicts advantageous and non advantageous poses.

The ITCM model is thus useful for semi-autonomous mobile manipulators. With it, such robots can navigate to an advantageous base pose from which the operator can do the task as good as from a human selected position without repositioning the robot.

References

[1] W. H. Organization, *World report on ageing and health*. World Health Organization, 2015.

[2] J. Stückler, M. Schwarz, and S. Behnke, "Mobile Manipulation, Tool Use, and Intuitive Interaction for Cognitive Service Robot Cosero," *Frontiers in Robotics and AI*, vol. 3, no. November, 2016.

[3] M. Van Osch, D. Bera, K. Van Hee, Y. Koks, and H. Zeegers, "Tele-operated service robots: ROSE," *Automation in Construction*, vol. 39, pp. 152–160, 2014.

[4] M. Mast, Z. Materna, M. Španěl, F. Weisshardt, G. Arbeiter, M. Burmester, P. Smrž, and B. Graf, "Semi-autonomous domestic service robots: Evaluation of a user interface for remote manipulation and navigation with focus on effects of stereoscopic display," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 183–202, 2015.

[5] M. Matamoros, V. Seib, R. Memmesheimer, and D. Paulus, "Robocup@ home: Summarizing achievements in over eleven years of competition," in *Autonomous Robot Systems and Competitions (ICARSC), 2018 IEEE International Conference on*, pp. 186–191, IEEE, 2018.

[6] R. Ventura and A. Ahmad, "Towards optimal robot navigation in domestic spaces," in *Robot Soccer World Cup*, pp. 318–331, Springer, 2014.

[7] G. Pamanes and S. Zeghloul, "Optimal placement of robotic manipulators using multiple kinematic criteria," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pp. 933–938, IEEE, 1991.

[8] F. Zacharias, W. Sepp, C. Borst, and G. Hirzinger, "Using a model of the reachable workspace to position mobile manipulators for 3-d trajectories," *9th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS09*, pp. 55–61, 2009.

[9] P. Abolghasemi, R. Rahmatizadeh, A. Behal, and B. Ladislau, "A real-time technique for positioning a wheelchair-mounted robotic arm for household manipulation tasks," *Workshop on Artificial Intelligence Applied to Assistive Technologies and Smart Environments (ATSE-16) at AAAI-2016*, pp. 2–7, 2016.

[10] J. Dong and J. C. Trinkle, "Orientation-based reachability map for robot base placement," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, pp. 1488–1493, 2015.

[11] F. Stulp, A. Fedrizzi, L. Mösenlechner, and M. Beetz, "Learning and reasoning with action-related places for robust mobile manipulation," *Journal of Artificial Intelligence Research*, vol. 43, pp. 1–42, 2012.

[12] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[13] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," *Proceedings - IEEE International Conference on Robotics and Automation*, no. 2, pp. 1970–1975, 2013.

[14] G. Y. Schropp, C. J. Heemskerk, A. M. Kappers, W. M. B. Tiest, B. S. Elzendoorn, and D. Bult, "Influence of visual feedback on human task performance in iter remote handling," *Fusion Engineering and Design*, vol. 87, no. 5-6, pp. 808–812, 2012.

[15] J. V. Draper, "Human factors in telemanipulation: perspectives from the oak ridge national laboratory experience," in *Telemanipulator Technology and Space Telerobotics*, vol. 2057, pp. 162–175, International Society for Optics and Photonics, 1993.

[16] J. A. Macedo, D. B. Kaber, M. R. Endsley, P. Powanusorn, and S. Myung, "The effect of automated compensation for incongruent axes on teleoperator performance," *Human Factors*, vol. 40, no. 4, pp. 541–553, 1998.

[17] D. Thomas, T. Field, J. Leibs, and J. Bowman, "Rosbag-ros wiki," *URL: http://wiki. ros. org/rosbag*, 2014.

[18] P. R. Tiago, "Tiago. pal-robotics. com," *available at least as early as Jul*, vol. 6, 2015.

[19] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised*, vol. 118. Springer, 2017.

[20] M.-J. Tsai, *Workspace geometric characterization and manipulability of industrial robots*. PhD thesis, The Ohio State University, 1986.

[21] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*, vol. 52, pp. 139–183, Elsevier, 1988.

[22] J. Van Oosterhout, J. G. W. Wildenbeest, H. Boessenkool, C. J. M. Heemskerk, M. R. De Baar, F. C. T. Van Der Helm, and D. A. Abbink, "Haptic shared control in tele-manipulation: Effects of inaccuracies in guidance on task execution," *IEEE Transactions on Haptics*, vol. 8, no. 2, pp. 164–175, 2015.

[23] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet, "A robust and sensitive metric for quantifying movement smoothness," *IEEE transactions on biomedical engineering*, vol. 59, no. 8, pp. 2126–2136, 2012.

[24] J. McAuley, J. Rothwell, and C. Marsden, "Frequency peaks of tremor, muscle vibration and electromyographic activity at 10 hz, 20 hz and 40 hz during human finger muscle contraction may reflect rhythmicities of central neural firing," *Experimental Brain Research*, vol. 114, no. 3, pp. 525–541, 1997.

[25] J. Stückler, I. Badami, D. Droeschel, K. Gräve, D. Holz, M. McElhone, M. Nieuwenhuisen, M. Schreiber, M. Schwarz, and S. Behnke, "Nimbro@ home: Winning team of the robocup@ home competition 2012," in *RoboCup 2012: Robot soccer world cup XVI*, pp. 94–105, Springer, 2013.

[26] S. Muszynski, J. Stückler, and S. Behnke, "Adjustable autonomy for mobile teleoperation of personal service robots," in *RO-MAN, 2012 IEEE*, pp. 933–940, IEEE, 2012.

[27] E. Liscio, "Learning autonomous grasping strategies for a care robot: A machine learning approach," *TU Delft repository*, 2017.

[28] D. Fischinger, P. Einramhof, K. Papoutsakis, W. Wohlkinger, P. Mayer, P. Panek, S. Hofmann, T. Koertner, A. Weiss, A. Argyros, *et al.*, "Hobbit, a care robot supporting independent living at home: First prototype and lessons learned," *Robotics and Autonomous Systems*, vol. 75, pp. 60–78, 2016.

[29] T. Yoshikawa, "Dynamic manipulability of robot manipulators," *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1033–1038, 1985.

[30] R. Koeppe and T. Yoshikawa, "Dynamic manipulability analysis of compliant motion," in *International Conference on Intelligent Robots and Systems (IROS)*, pp. 1472–1478, 1997.

[31] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, "Grasp planning in complex scenes," in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pp. 42–48, IEEE, 2007.

[32] D. Morrison, A. W. Tow, M. McTaggart, R. Smith, N. Kelly-Boxall, S. Wade-McCue, J. Erskine, R. Grinover, A. Gurman, T. Hunn, *et al.*, "Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7757–7764, IEEE, 2018.

[33] P. Kovesi, "Good colour maps: How to design them," *arXiv preprint arXiv:1509.03700*, 2015.

Appendices

A. On-line part

To create a task specific map of the IHCM (Appendix B) and ITCM (Appendix C), a task needs to be generated. Maps need to be merged and those maps need to be interpreted by the robot relative to a certain task in real life, as shown in Figure 12. These general on-line aspects of these methods are discussed in this appendix.

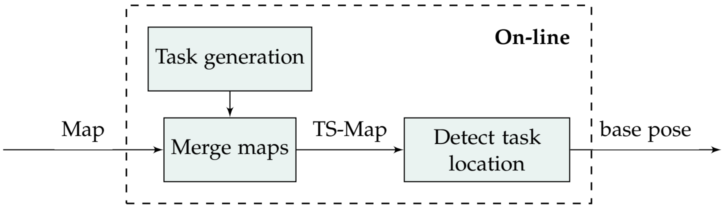


Figure 12: A map and a specified task is used to generate maps that are merged to a task specific map (TS-Map). To find where this goal position relative to the current base pose, the origin of the task is detected.

1. Task generation

The positioning methods: Inverse Human Capability Map (IHCM) and the Inverse Telemanipulation Capability Map (ITCM), need a way to describe the task in order to generate maps that can then be merged to find a suitable base pose. In this section it is shown how tasks are generated and what should be taken into account in that process. For this research project, tasks are generated based on parameters collected from the user via an interface. These generated poses can resemble opening a drawer, a sliding door cabinet or a rotating door cabinet and pick and place tasks.

Task generation for autonomous grasping is done by for example Vahrenkamp et al. [1]. They generate a path of Tool Centre Points (TCP) poses a long a curve, see Figure 13. For each TCP, a transformation of the ICM was made to match the generated TCP. The intersection between the maps (overlap of the maps) is used by Vahrenkamp et al. to constrain the on-line search for a suitable base pose. But because map merging in the ITCM project is done by reusing the pre-computed data with the aim of limiting on-line computation, the TCP positions and orientations are constrained. The TCP's must be placed in the same grid as used for the discretisation of the base poses in the IHCM and ITCM.

Because the robot Marco as used in this project has a lift on which the arm is mounted, the shoulder height can be selected by the positioning method as well. Instead of searching for base poses with a vertical robot on a specific height, a height range is used corresponding with the lift's range. And due to the telemanipulation

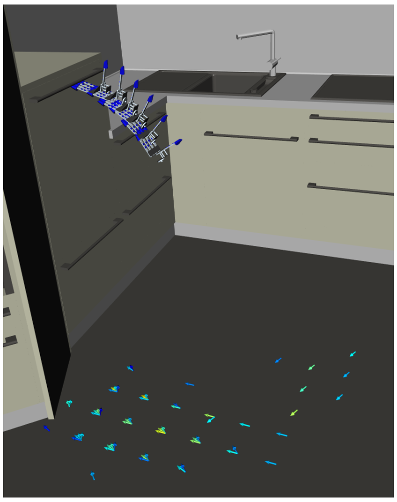
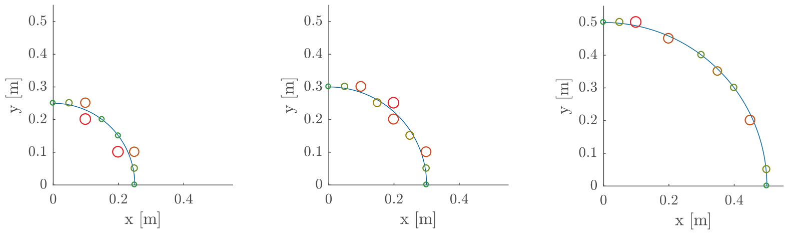
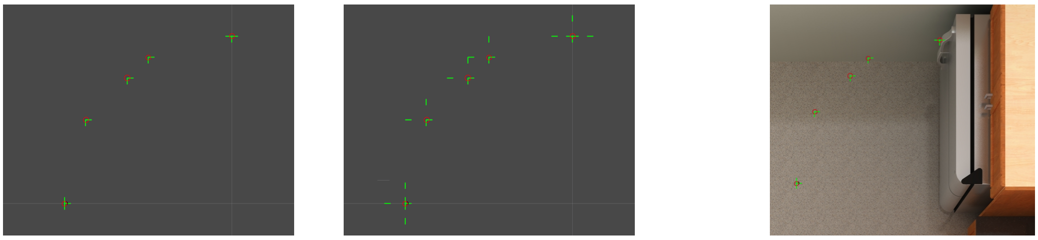


Figure 13: A path for opening a dishwasher, with the resulting Inverse Capability Map of Vahrenkamp et al. [1].



(a) Radius: 25 cm **(b)** Radius: 30 cm **(c)** Radius: 50 cm

Figure 14: Comparison of position error due to 5 cm discretisation grid for three different curved paths. Each circle is a position on the grid near the curved path. Both the colour and the size of the circle represent the distance to the path (i.e. the error). A green small circle is a low error, a big red circle is a large error.



(a) Without pre-grasp. **(b)** With pre-grasp **(c)** The same curved path used on a fridge.

Figure 15: The same task with and without pre-grasp. The red circles to denote the selected positions. The green lines start from the TCP position and direction denotes the orientation of the gripper. The green lines corresponds with the green y-axis on the gripper, see Figure 47 (p. 43).

context, the task as generated by the robot might not be executed in that way by the operator. The orientation of the selected grasp is not known, therefore it would be nice to select multiple options per grasp position. To further limit the amount of possible base poses, the implemented task generation method can generate pre-grasps as well.

By searching for not one, but for a number of possible grasp orientations, the amount of possible base-positions is increased. For example, a cup can be picked up from multiple directions with a side grasp, it should not mean that the robot should stand somewhere where it can reach all possible grasps the best.

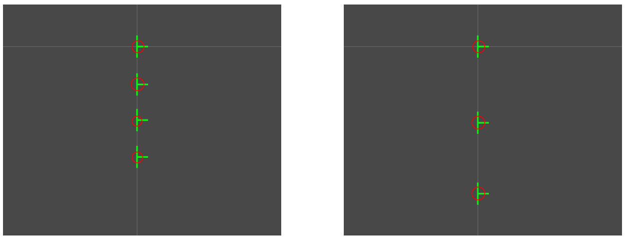
To incorporate these extra possibilities instead of an intersection, a union of the maps belonging to the possible grasps is made per position. In this union, the maximum score is used to represent the best configuration possible from that position. Thereafter the maps of all positions are combined by making taking the intersection of the possible Thereafter an intersection is made of the ICM's for all positions as described earlier to get the ICM for the task. Additionally, a pre-grasp option has been implemented in this projects task generation method. Because the approach and retreat from a grasp should be reachable as well.

Curved paths are generated based on the radius, the number of samples, the angle at which the path starts and ends and the height of the task. As can be seen in Figure 14 the positions with zero x or y distance are almost limited to those exactly on the curve, at least if the radius is a multiple of 5 cm. For the selected curved path, positions are generated around it. The distance to the path (dist $= \sqrt{x^2 + y^2}$) is calculated for all positions. If m samples are needed, the first m points on the discretiation grid with the lowest distance to the curved path are selected.

The orientation options for the curved path are not static however. Because this path is envisioned to be used for opening and closing doors, care is taken that no orientations are used that will let the gripper collide with the door. For different positions, different orientations are selected, this is shown in Figure 15.

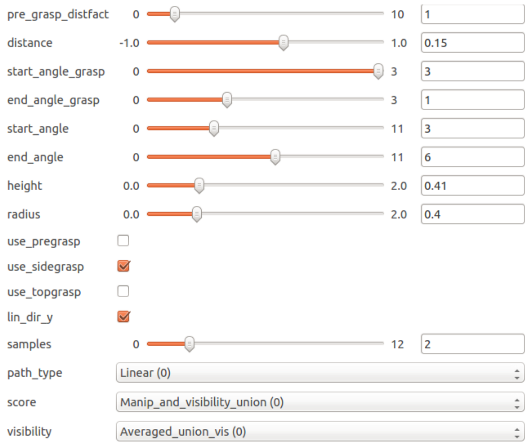
Note that the task generation of curved paths is only implemented in the horizontal plane.

Linear paths are generated based on distance, the supplied distance and the height of the task. But positions can only be used if they are on the grid. The distance between every position should thus be the product of a natural number and the discretisation distance (in this case 5 cm). This could lead to selecting more or less



(a) 3 samples, distance 0.15 cm **(b)** 3 samples, distance 0.20 cm

Figure 16: Two linear tasks with 3 samples. To get the task positions on the discretisation grid with a distance of 0.15 cm, the distance between each position is 5 cm and 4 positions are used. For a distance of 20 cm, placing the task positions 10 cm apart results in three task positions. The green lines and red circles are explained in the caption of Figure 15.



pre_grasp_distfact 0 10 1
distance -1.0 1.0 0.15
start_angle_grasp 0 3 3
end_angle_grasp 0 3 1
start_angle 0 11 3
end_angle 0 11 6
height 0.0 2.0 0.41
radius 0.0 2.0 0.4
use_pregrasp □
use_sidegrasp ☑
use_topgrasp □
lin_dir_y ☑
samples 0 12 2
path_type Linear (0)
score Manip_and_visibility_union (0)
visibility Averaged_union_vis (0)

Figure 17: Interface to set parameters as used in this project. This is made with the dynamic reconfigure package.

positions than the selected amount of samples, an example is shown in Figure 16. The orientations are the same for all positions and are set with the options `start_angle_grasp` and `end_angle_grasp` in the menu shown in Figure 17.

Pre-grasp positions are added if the option is selected in the menu (Figure 17). These can be added any multiple of the discretisation distance in the negative direction of the grasp. An example is shown in Figure 15b.

2. Merge maps

As positioning methods have been developed by among others: Zacharias et al. [2], Vahrenkamp et al. [1] and Abolghasemi et al. [3], each has introduced a way of combining a set of capability/reachability maps. The reachability map of Zacharias et al. is combined by doing a cross-correlation following the spheres in which the task is discretised [2]. As this method does not give a lift height or base orientation as a result it is not suitable for our robot. The method of Abolghasemi et al. created their own score and approximated the description of that score with Gaussians, because that might influence the trueness of the scores it is not chosen to follow such a method [4]. Vahrenkamp et al. used their original maps to deduce were there was an overlap between all maps. In this overlapping region they applied a new discretisation of possible base poses. Thereafter for all new base poses it was checked if all task's TCPs were kinematically reachable.

None of these methods were used in this project. In this project the computation time is reduced by reusing the original sets of maps, this was possible because the task generation could be constrained to fit to the same grid the maps were discretised to. This was deemed a valid task simplification because the task execution will be done by a telemanipulation operator and could thus be done differently than the generated task.

As can be seen in Figure 18, the exact placement of the EE positions ensures that if two base options overlap, they overlap exactly. There is thus no need to interpolate with Gaussians or re-sample in the overlapping area. For each gripper pose, all the possible base poses are transformed to be relative to a single fixed coordinate

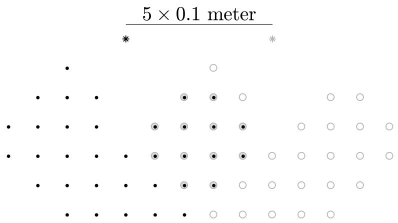


Figure 18: Overlap between two map's belonging to two end effector positions '*'. One map is denoted with a '·', the other with 'o'. The overlapping area has grey circles. To simplify no orientations are taken into account.

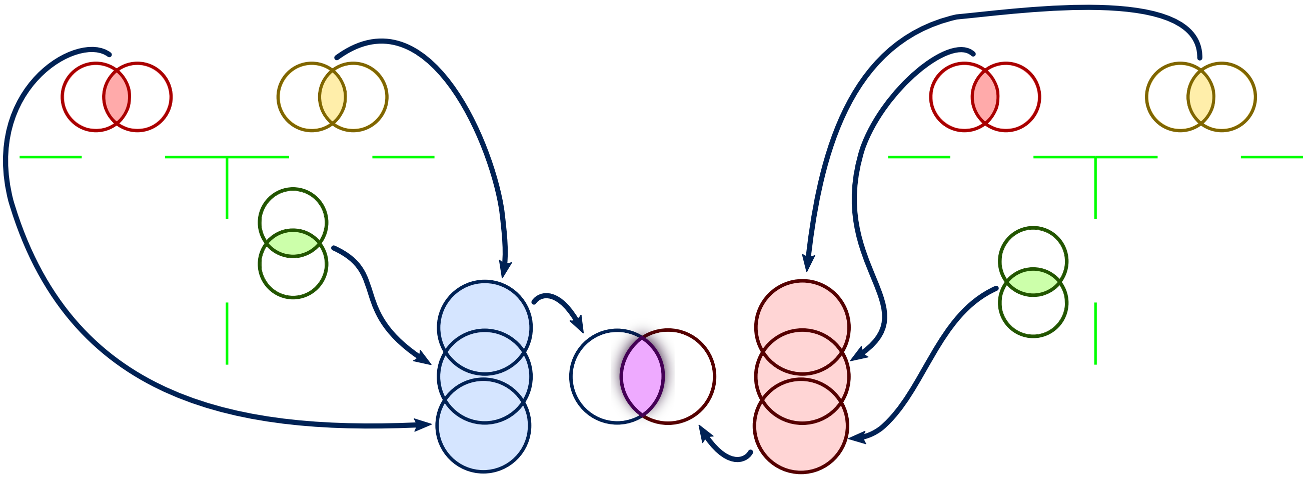


Figure 19: The green lines represent a pick-and-place task and is generated linearly. Each position has three possible orientations and a pre-grasp. See Figure 15 how these green lines represent possible gripper poses. For each gripper (green line) the pre computed ITCM is transformed according to its pose. Each circle thus represents a collection of base poses. For the three orientations per position, the intersection of base poses of the grasp and pre-grasp gripper For each gripper orientation, the intersection of grasp and pre-grasp base poses is made. Per position (pick or place in this case), a union of these intersections is made. The union of these three intersections represents the ITCM for that position (blue and red). The intersection of these ITCMs is the TS-ITCM and is represented by the purple area.

frame, the task frame. These transformed base poses belonging to a gripper pose are called a set of base poses. In three steps all the separate sets of base poses are combined into a single task specific map.

First, if the gripper orientations have a pre-grasp, the grasp as well as the pre grasp should be reached for all base positions for that grasp pose. The set of base poses suitable for that grasp pose is thus the intersection[5] of the base poses of the grasp and pre-grasp EE pose. When the intersection of two sets of base poses is taken, the lowest score of the same base pose is used in the new set. Because the when manipulating at two points, the point that makes the manipulation hard will have more influence on the performance, effort and difficulty of the manipulation task.

Secondly, per position, extra gripper orientations are added because the eventual grasp orientation is not known by the robot. These extra orientations have additional base pose options. Because it is not the goal to find a base pose from which all orientations can be used, the intersection method is not suitable. Thus a union[6] is made of the base pose sets of all orientations of the same position. If there are multiple base poses with the same pose in the original sets, the highest score is used for the unioned set. Because the gripper orientations belonging to the lower scores will probably not be used because the operator will find them less intuitive.

Lastly to combine all base poses per position, an intersection of the base pose sets per position is taken. Again the lowest score is used for the remaining base poses. An overview for a pick-and-place task of this process is shown in Figure 19.

[5] In set theory, the intersection is the set that contains only elements from both original sets.
[6] In set theory, the union is the set in which all elements from the other sets are present.

3. Laser scan for task detection

In the experiments, the laser scanner of the robot was used to measure the position and orientation of the robot with respect to the task. Previously, the RGBD camera was used for this same purpose, but due to specifics of the setup used, there were to many deviations from the real world to be accurately used.

In Figure 21 the experiment setup is depicted. The task is always in the same place on the table. Thus if the table is detected correctly, the pose of the task with respect to the robot is known (or vise versa). To this end, one of the table's legs is widened to be able to distinguish it from the other legs.

The laser scan data is published at 10 Hz. Per scan, the two closest blobs are interpreted as the table legs. Of which the left one has a lot more data points than the right table leg. The average location of the data points is used to determine the locations of the table legs for this time-step. An moving average of the last ten leg locations is used to determine the table location. Because the table was stationary this did not cause any problems in positioning the robot.

From these leg positions a fixed transform to the pick position of the task is known. This pick position is then used as task frame and as origin of the task specific Inverse Telemanipulation Capability Map. The result is shown in Figure 20.

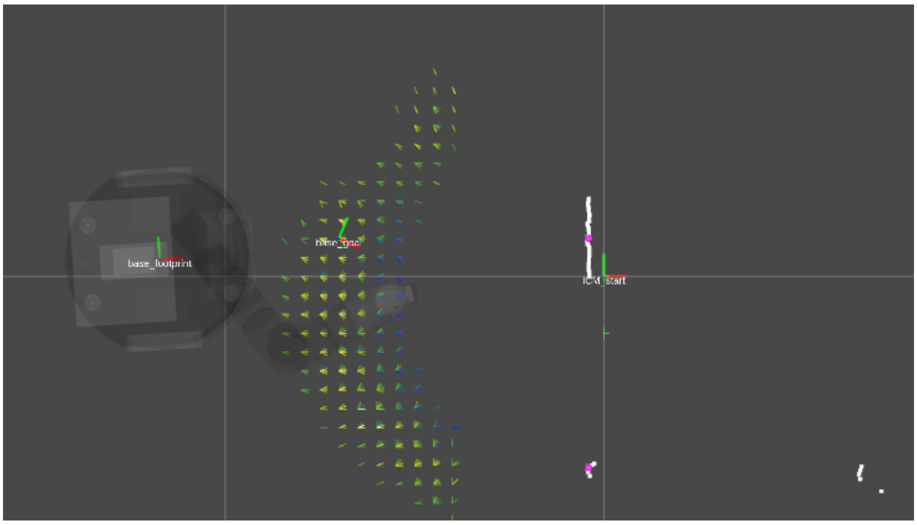


Figure 20: Visualisation in Rviz of table legs, robot, and TS-ITCM. The white dots are the measurements from the laser scanner, the wide part is the left table leg, the smaller one is the right table leg and one of the hinter table legs can also be seen. The pink dots denote the position and time averaged tabel leg locations. The green and red lines with 'ICM_start' denote the fixed task frame on which the TS-ITCM is based. A model of the robot is visualized as well with the `base_footprint` frame, this frame needs to be collocated with corresponding orientation to the frame in the TS-ITCM.

References

[1] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," *Proceedings - IEEE International Conference on Robotics and Automation*, no. 2, pp. 1970–1975, 2013.

[2] F. Zacharias, W. Sepp, C. Borst, and G. Hirzinger, "Using a model of the reachable workspace to position mobile manipulators for 3-d trajectories," *9th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS09*, pp. 55–61, 2009.

[3] P. Abolghasemi, R. Rahmatizadeh, A. Behal, and B. Ladislau, "A real-time technique for positioning a wheelchair-mounted robotic arm for household manipulation tasks," *Workshop on Artificial Intelligence Applied to Assistive Technologies and Smart Environments (ATSE-16) at AAAI-2016*, pp. 2–7, 2016.

[4] S. O. Adalgeirsson and C. Breazeal, "MeBot A Robotic Platform for Socially Embodied Telepresence," pp. 15–22, 2010.

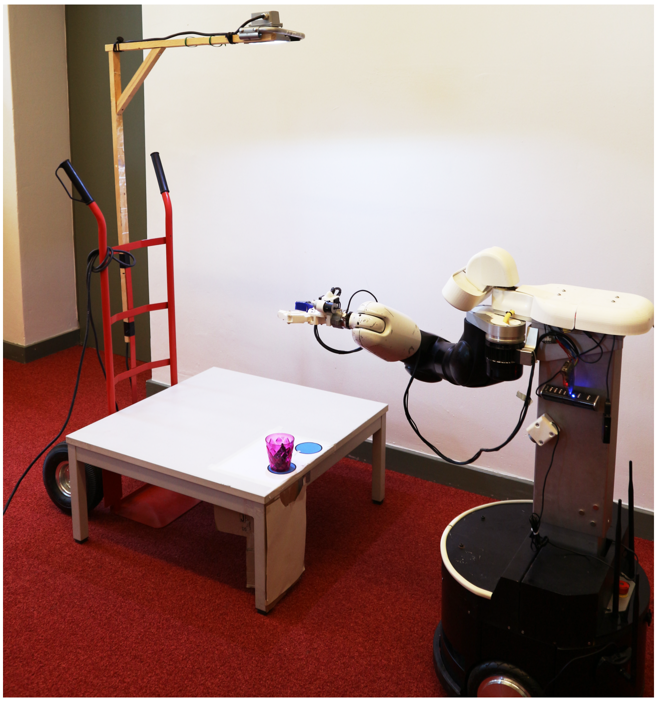


Figure 21: Experiment setup with cardboard attached to the table to differentiate the left from the right table leg.

B. Inverse Human Capability Map (IHCM)

The Inverse Human Capability Map is based on the Inverse Capability Map. Which is presented as the Oriented Reachability Map by Vahrenkamp et al. [1]. The IHCM differs from the ICM in how the maps are merged to as is discussed in Appendix A. The method to score base poses is complemented by a task visibility score to make the map more human operator compatible. The task generation part is extended as well with pre grasps and multiple orientations per position, but as those are additions to the online part, these are discussed in Appendix A as well. In Figure 22 an overview is given how the task specific Inverse Human Capability Map (TS-IHCM) is constructed.

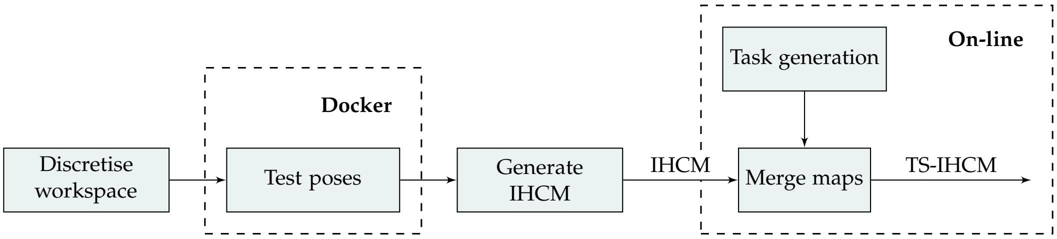


Figure 22: A discretisation of the robot's workspace is generated and tested to be reachable with a simulation in a docker [2] container. Based on these results the general Inverse Human Capability Map (IHCM) is generated. This map is then used on-line to create a task specific IHCM (TS-IHCM) as is discussed in Appendix A.

1. Off line part IHCM

Because the capabilities of the robot's arm do not change, the map can be calculated off-line. As has been shown in Figure 22 this consists out of three separate parts. After these parts the computation of the visibility score is discussed.

1) Discretise workspace

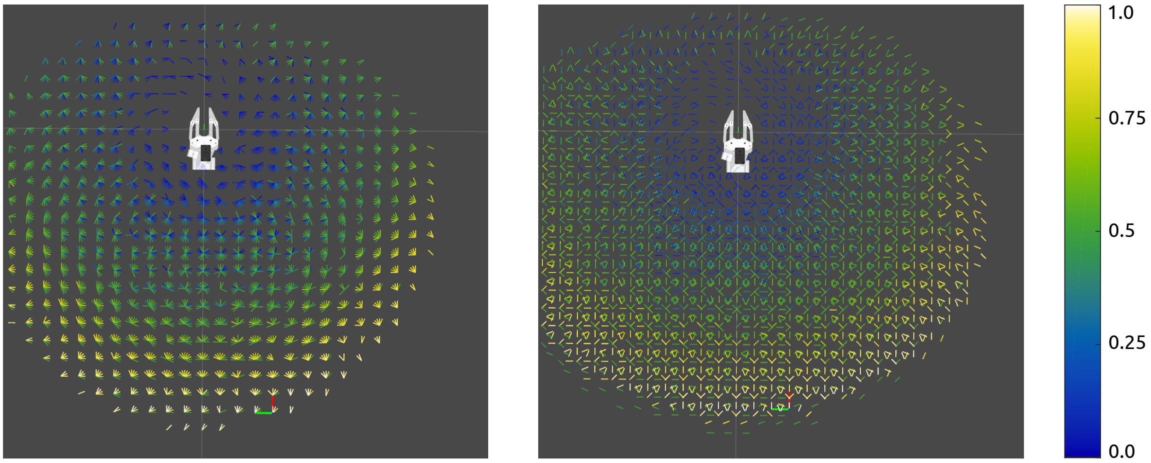
Because the mapping cannot be done for all possible position and orientations (hereafter poses), a discretization must be made. For human interpretation it is beneficial if for each base position all orientations are tested, in contrast to scattered combinations of positions and orientations, see for this difference Figure 23.

To ensure that from the same position multiple orientations are tested to end up with a map as in Figure 23a. The poses are discretized from the `gripper_grasp_link` to the `base_footprint` as follows: A three dimensional grid of `base_footprint` positions around the `gripper_grasp_link` frame (for simplicity used as origin) with an equal distance in x y and z from each other is created. The robot Marco with the relevant frames is shown in Figure 47 (Appendix E p. 43). For this research a resolution of 5 cm is used and because the maximum length of the arm is 87 cm, the cube is 2 by 2 by 2 meters. The `shoulder` frame is the first joint of the arm (`arm_1_link`), but with zero rotation with respect to the base of the robot. The `base_footprint` poses are then transformed to `shoulder` poses as seen from the static `gripper_grasp_link` frame, this results in the transform depicted in Figure 24a. This is useful because we can than more accurately dismiss poses based on the distance to the `gripper_grasp_link` frame. Only the `shoulder` positions that are within a sphere of 90 cm (a bit longer than the maximum reach of the arm) from the `gripper_grasp_link` are considered for evaluation with inverse kinematics, in order to reduce unnecessary IK computations.

Then the possible orientations of the end effector are generated. To get a complete mapping, orientations can be varied in all directions or if the application will ask only for side and/or top grasps the discretization can be limited to those as well.

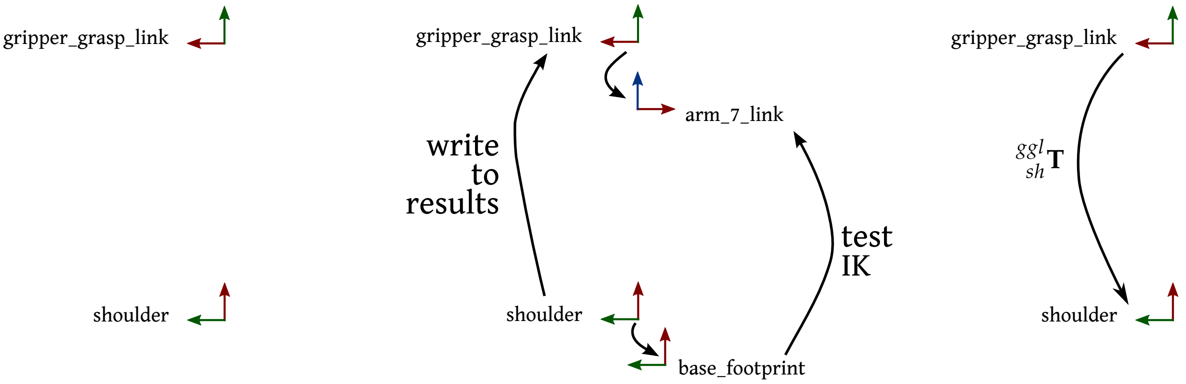
Finally the transform from the `gripper_grasp_link` to the `shoulder` frame is written to a `.txt` file as illustrated in Figure 24.

Because of memory limitations of MATLAB, the complete set of samples is split per position in separate files. These are combined and split by a bash script to a number of files corresponding with the amount of nodes in the launch file. By using 20 nodes to test the poses, better use can be made of the computers processing power.



(a) `base_footprint` **(b)** `shoulder`

Figure 23: IHCM with samples generated from the `base_footprint` frame or from the `shoulder` frame. The left side of the map has been cut of better see the small lines. If the different base orientations are grouped per base position, the map is easy to interpret by humans and easier to merge by the computer.



(a) Discretise workspace **(b)** Test poses **(c)** Generate IHCM

Figure 24: Frame transformations used for the discretization, pose testing and IHCM generation. See Figure 47 (Appendix E p. 43) for the relative location of these coordinate frames in the robot.

2) Test poses

To find a configuration of the robot arm for a given position and orientation, an Inverse Kinematic equation must be solved. For redundant robot arms, there could be found multiple solutions or none at all. Solving an Inverse Kinematic (IK) equation for a redundant kinematic chain, is thus partly stochastic. Although finding a solution for one pose can be quite fast, for an 7 degree of freedom arm around 0.5 ms [3], doing it for 32 million samples would take 4 h. That is the reason for exploring the workspace off-line as opposed to on-line.

As can be seen in Figure 22, the `base_footprint` poses are tested in a `docker` container. In this docker container the robot can be simulated with gazebo and the same software as would be running on the real robot. For the Inverse Kinematics (IK) tests, trac-ik [3] is used because of the high solve rate and quick solve time.

The transforms from the `.txt` file generated by the 'discretise workspace' part are read by the program to test if there is an arm configuration. As shown in Figure 24b, the `gripper_grasp_link` frame is transformed to the `arm_7_link` frame. This is because there simulated robot did not have a gripper and thus no `gripper_grasp_link` frame in the arm's kinematic chain. There are no moving joints between the `arm_7_link` and the `gripper_grasp_link` frame, therefore the transform is constant over all poses. Also the `shoulder` frame is transformed back to the `base_footprint` frame, taking the current lift height into account of the simulated robot.

The read pose is thus now represented as a pose of the `arm_7_link` in the `base_footprint` frame. This pose is tested to be reachable or not. If it is reachable, the found configuration is used to compute the manipulability as presented by [4] with the joint limit penalization as presented by [5, p. 114]. How this is done can be read in Appendix 1), except that the Jacobian is generated by the MoveIt! ROS package. Only the poses which could be reached are written to the result file. These are not the tested poses (from the `base_footprint` frame), but the poses from the `shoulder` to the `gripper_grasp_link` frame (${}^{sh}_{ggl}T$). In this file each line represents one pose and corresponding score. score x y z q_x q_y q_z q_w

3) Generate IHCM

The result files produced by all 20 nodes are then merged into one `.mat` file. This mat file is then used to

Because the samples are first generated in the grid we would like to seem them in (`base_footprint` frames expressed in the `gripper_grasp_link` frame: ${}^{ggl}_{basef}T$), to test the validity of the poses they are inverted to ${}^{sh}_{ggl}T$. To get the poses back in the grid we would like to seem them in, the shoulder poses need to be inverted back and thus be defined with respect to the `gripper_grasp_link` frame, as shown in Figure 24c.

The poses are then saved with the rotation expressed as a rotation matrix instead of a quaternion. Because with rotation matrices, it is easier to check if a rotation is unique than with quaternions. And when using the collection of rated possible base positions, it can quickly be evaluated if a possible shoulder pose has an orientation that enables the robot to stand upright.

As last step double shoulder poses are removed (a margin of 0.0001 is used), the maximum manipulability score is kept. The suitable shoulder poses are stored in a `.txt` file per row, with in each row: score x y z $R_{1,1}$ $R_{1,2}$ $R_{1,3}$ $R_{2,1}$ $R_{2,2}$ $R_{2,3}$ $R_{3,1}$ $R_{3,2}$ $R_{3,3}$ An additional `.txt` file is saved with the size of the array. In that way the `cpp` program can preallocate memory before reading the file.

2. visibility score

The visibility score is calculated the combination of a yaw score and a pitch score. The best visibility of the grasp is assumed to be with zero yaw and a pitch of 45° downwards. The score is designed to be maximum (1) at those goals and minimum (0) outside of the visible limit. The visible limit is seen as the joint limits of the camera with the field of view added. Below this text, the bounds are calculated as the joint limit plus or minus half of the field of view. The joint limits are in the yaw direction 75° and the field of view is 58° both from the neutral in both ways. In the pitch direction the joint limit is downwards 60° and 45° upwards with a field of view of 45° from the neutral in both ways. These scoring formulas result in the scoring landscape as seen in Figure 25.

$$yaw_{\text{goal}} = 0^o \qquad yaw_{\text{lb}} = -75^o - \frac{1}{2}58^o \qquad yaw_{\text{ub}} = 75 + \frac{1}{2}58^o \tag{10}$$

$$pitch_{\text{goal}} = -45^o \qquad pitch_{\text{lb}} = -60^o - \frac{1}{2}45^o \qquad pitch_{\text{ub}} = 45 + \frac{1}{2}45^o \tag{11}$$

$$yaw = \text{atan}\left(\frac{dir_y}{dir_x}\right) \qquad pitch = \text{atan}\left(\frac{dir_z}{\sqrt{dir_x^2 + dir_y^2}}\right) \tag{12}$$

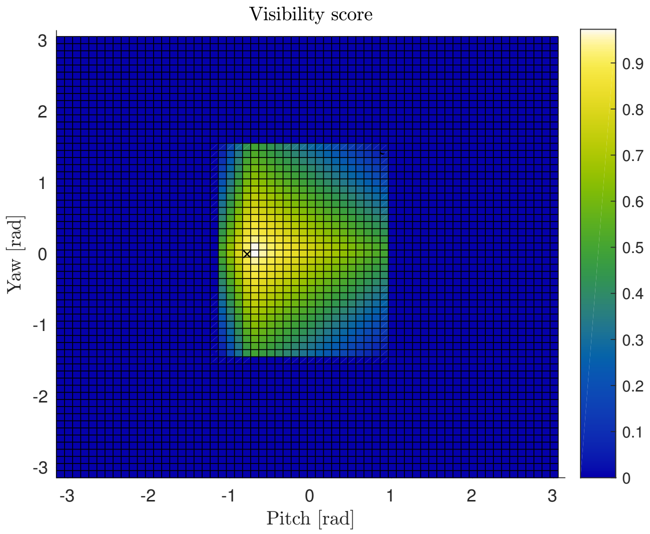


Figure 25: Visibility score is depicted in colour as shown by the colour bar as function of *yaw* and *pitch*. The lowest score is 0 and is thus represented in blue, these are mostly view angles outside the reach of the head. The highest reachable score is white and is near the view-goal (x).

$$\text{yaw}_{\text{score}} = \begin{cases} 1 - \left|\frac{yaw - yaw_{\text{goal}}}{yaw_{\text{ub}} - yaw_{\text{goal}}}\right| & \text{if} yaw_{\text{lb}} \leq yaw \leq yaw_{\text{ub}} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

$$\text{pitch}_{\text{score}} = \begin{cases} 1 - \left|\frac{pitch - pitch_{\text{goal}}}{pitch_{\text{ub}} - pitch_{\text{goal}}}\right| & \text{if} pitch_{\text{lb}} \leq pitch \leq pitch_{\text{ub}} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$\text{visib}_{\text{score}} = \left(\text{yaw}_{\text{score}} + \text{pitch}_{\text{score}}\right) \cdot \frac{1}{2} \tag{15}$$

The yaw and pitch are calculated not from the actual pose of the robot's head but the position of the `xtion_link` frame when the `head_1_joint` and `head_2_joint` have an angle of 0°. In other words, if the head is in the horizontal, straightforward position as depicted in figure 26. The difference in camera position to the `gripper_grasp_link` due to the links after the head joints, is thus neglected in calculating the *yaw* and *pitch* angles. From the shoulder frame (`arm_1_link` in the same orientation as `torso_lift_link`) the position of the camera is calculated with the fixed transform ${}^{sh}_{cam}T$. Subtracting the tool centre point's (tcp) location from that of the camera (cam) in the frame of the shoulder, results in a vector describing the direction the camera should look to have the tcp in the middle of the sensor.

$${}^{sh}P_{cam} = [0.136,\ 0.043,\ 0.155]^T \tag{16}$$

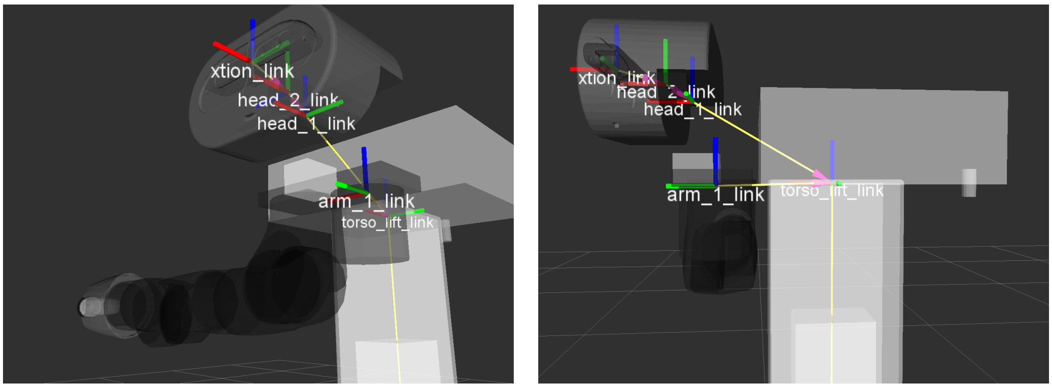
$$dir = {}^{sh}P_{tcp} - {}^{sh}P_{cam} \tag{17}$$

This direction vector in normalized, such that the length of the vector equals 1.0. Combining manipulability and visibility is done similarly to how the yaw and the pitch scores are combined. However, to ensure that the manipulability (manip) and the visibility (visib) have the same weighing in the average total score. The visibility score and the manipulability score are first normalized such that the lowest value of each score vector is zero and the highest value of each scoring vector is 1. Thereafter the average is calculated and used to further evaluate the shoulder poses.

$$\text{score} = \frac{\text{visib\_score} + \text{manip\_score}}{2} \tag{18}$$

3. Results & discussion

In Figure 27 the resulting task specific IHCMs are shown for various kitchen related tasks. Conspicuous about these maps is that base options are very close to the gripper poses of the task. It can be imagined that from



(a) Perspective view

(b) Side view

Figure 26: Transformation frames from `torso_lift_link` to `xtion_link` and the `arm_1_link` as seen from the side and in perspective.

these positions, the gripper will be to close to the base and a self collision seems inevitable.

While testing the IHCM, this suspicion was confirmed. It occurred that the task was not reachable from a base pose in the TS-IHCM, see Figure 29. This could be due to the fact that telemanipulation constraints are not taken into account in the generation of the IHCM. Although the trac-ik and MoveIt! combination said to be taking joint limits and self collisions into account, it could be that it is not the case. To create a map which contains only the possible base poses, the ITCM is developed, see Appendix C.

Referring back to the visibility score as introduced in the IHCM method in this Appendix, a comparison is made between scoring method with manipulability, the manipulability and yaw score and the combined visibility and manipulability score. This is shown in Figure 28. It can be seen that no clear better regions can be seen with the manipulability score, 28. The yaw and manipulability score result in the pose which was used in the failed manipulation in Figure 29.

By taking into account the visibility of the object, a bias is created for base poses which point towards the task positions. This can be seen in Figure 28.

References

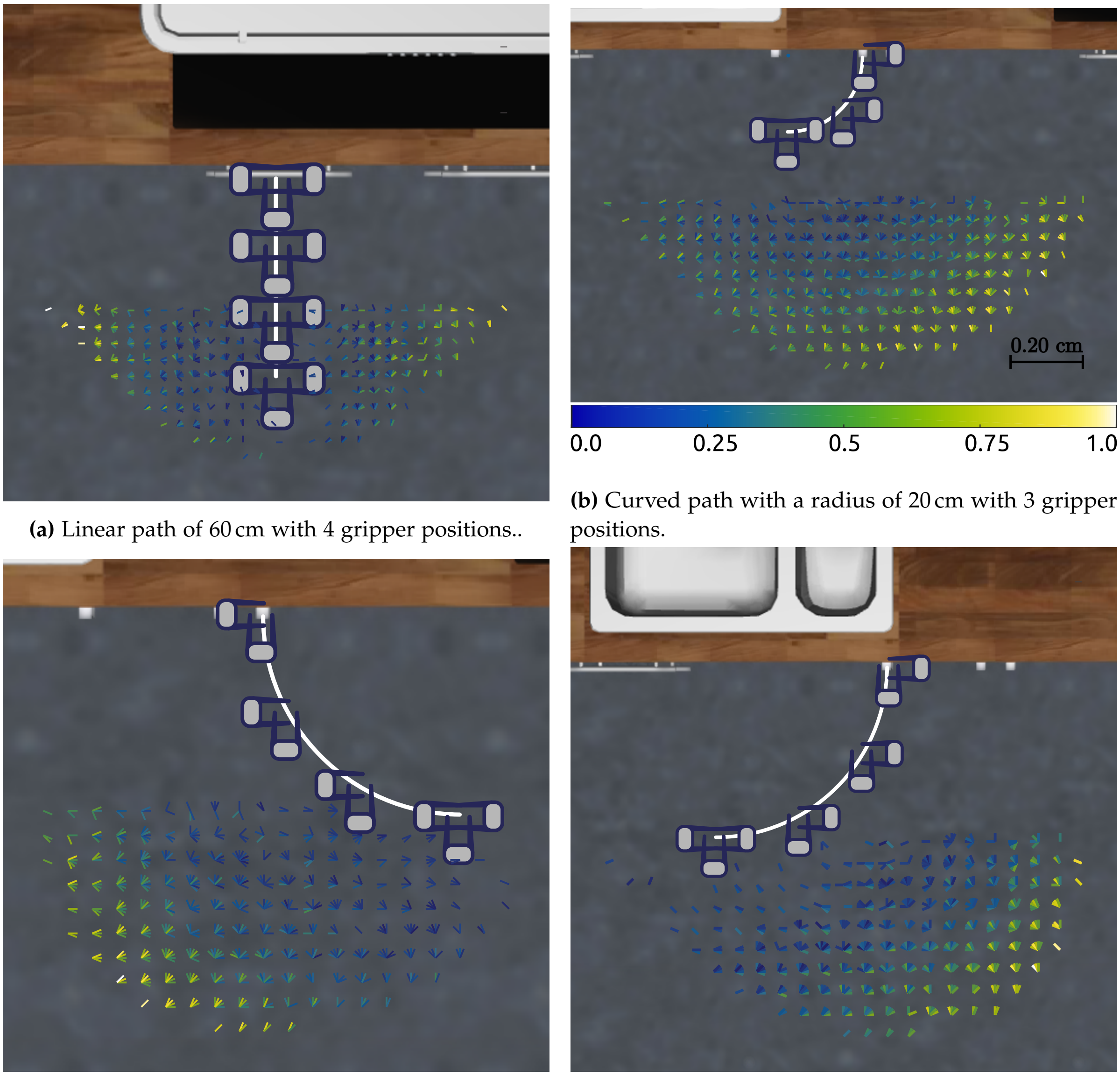
[1] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," *Proceedings - IEEE International Conference on Robotics and Automation*, no. 2, pp. 1970–1975, 2013.

[2] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux Journal*, vol. 2014, no. 239, p. 2, 2014.

[3] P. Beeson and B. Ames, "Trac-ik: An open-source library for improved solving of generic inverse kinematics," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pp. 928–935, IEEE, 2015.

[4] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[5] M.-J. Tsai, *Workspace geometric characterization and manipulability of industrial robots*. PhD thesis, The Ohio State University, 1986.



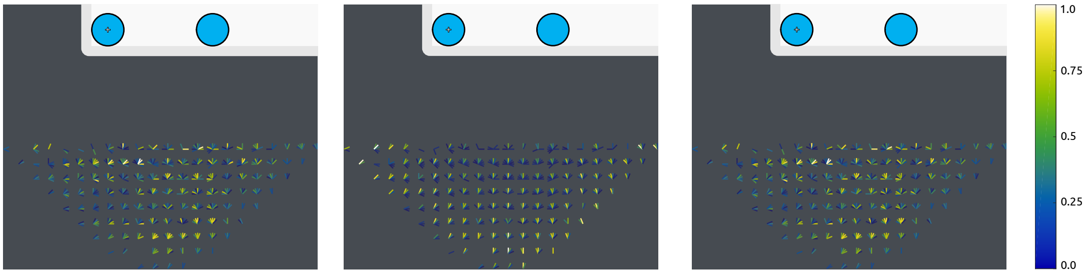
(a) Linear path of 60 cm with 4 gripper positions..

(b) Curved path with a radius of 20 cm with 3 gripper positions.

(c) Curved path with a radius of 40 cm with 4 gripper positions.

(d) Curved path with a radius of 40 cm with 4 gripper positions.

Figure 27: The linear path has the same gripper orientation for each gripper position, and the curved path has different gripper orientations per position.



(a) Manipulability

(b) Manipulability and yaw score

(c) Manipulability and visibility score

Figure 28: Pick and place task, generated with a two position linear path of 35 cm.

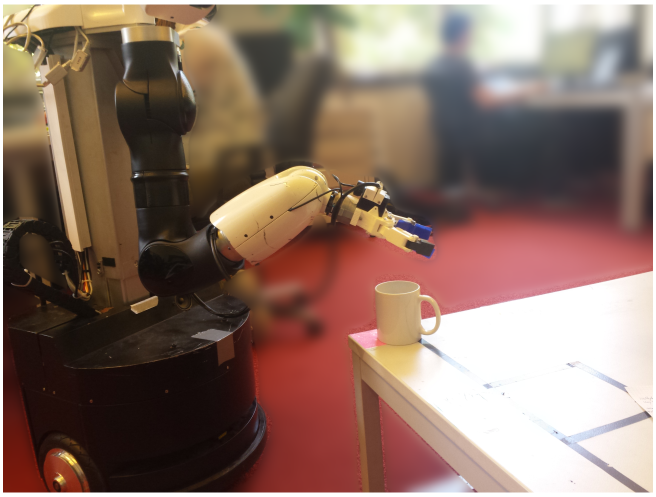


Figure 29: Base pose selected with the IHCM that resulted in a task that was not executable with telemanipulation. Irrelevant background blurred.

C. Inverse Telemanipulation Capability Map (ITCM)

After implementing the Inverse Capability Map (ICM) as described by Vahrenkamp et al. [1] and extending it to the IHCM (see Appendix B). It became clear during tests that constraints of the telemanipulation setup should be considered too. With a base pose selected with either the ICM or the IHCM, it would occur that the the task would be carried out with a highest manipulability possible. The task could however not be reached because the workspace of the master device is limited and not the entire workspace of the robot can thus be explored. Because Vahrenkamp's ICM is designed for autonomous grasping, it suffices to only regard the kinematic properties of the robot's arm.

This caused inspiration to use data from the actual telemanipulation setup in order to take all the limitations into account. The scheme to create a task specific ITCM is shown in 30, in this appendix only the map generation is discussed. This is the off-line part as shown in the figure. The online part is discussed in Appendix A. How the data for the ITCM map is gathered is described in section 1. The recorded joint states are then used to compute the robot's Inverse Telemanipulation Capability Map (ITCM). The joint states are also used to compute the manipulability of base poses with respect to the gripper, see section 3). The visibility score is calculated online because it was already implemented for the IHCM and is discussed in Appendix B. The ITCM is saved as a `.txt` files which is then interpreted by the same on-line part as used for the ICM and IHCM (Appendix B) methods, see Appendix A (p. 13) for the implementation. With the generated grasps resembling the task, the task specific ITCM (TS-ITCM) can be constructed for various tasks. These can be seen for various tasks in section 3.

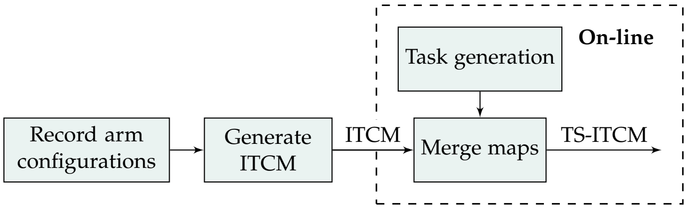


Figure 30: Schematic of the generation of a task specific ITCM. A distinction is made between off-line processes (the white blocks) and on-line processes (the coloured blocks).

1. Gather data

In order to take all constraints of the telemanipulation setup into account, the workspace was explored using the same master device (a geomagic touch omni [2]) and with the same settings as in normal operation. The telemanipulation setup is described in the paper's methods section, see page 5.

To limit the amount of data used in the ITCM for the experiment, only the grasp poses that could be used in the experiment were sampled. This means that the lift height of the Marco robot was fixed at 91 cm (the ICM and IHCM both selected it as the best lift-height for this task). The task was picking and placing a cup on the same table. To describe this task, only a side grasps were used and because the table does not vary in height all grasps were on the same height (41 cm). To get a proof of concept map for this task, the commands of the master device were edited to keep the roll, pitch and height of the gripper constant. To this end the rostopic `/geo_haptic_state_m` was re-maped to `/geo_haptic_state_prefilter` and a new node was put in between these topics to edit all messages, see Figure 31.

Although the height command of the master side was fixed, that did not mean that the end effector of the slave was not going to go up or down. The slave controller calculates a path towards the new pose, this path is not optimised to keep the end effector on the same height. Therefore excursions were made outside of prescribed plane during the recording.

During 53 min and 17 s the topics `/joint_states` and `/tf` are saved in three consecutive `.bag` files. The `/tf` topic was recorded as a back-up, this topic consists of the so called transform tree. The joint states are more important than just the transform between the `shoulder` frame and the `arm_7_link`, because the manipulability computation is dependent on the configuration of the arm. The `/joint_states` topic contains messages of type `sensor_msgs/JointState`, publishes at 50 Hz and has besides the arm joints of interest also all other joints of

Marco, see Listing 1. These messages thus resemble the state of the generalized coordinates of the robot. The arm joints we are interested in are revolute, therefore the position is in rad and the velocity is in $\frac{\text{rad}}{\text{s}}$. The arm joint positions are in the range between $-\pi$ and π. For the ITCM only the position field is used, of which only the values corresponding to the arm joints are used.

Listing 1: Example of a `sensor_msgs/JointState` message on the `/joint_states` topic.

```
header:
  seq: 54567
  stamp:
    secs: 1531842924
    nsecs: 956858817
  frame_id: ''
name: ['arm_1_joint', 'arm_2_joint', 'arm_3_joint', 'arm_4_joint', 'arm_5_joint', '
    ↪ arm_6_joint', 'arm_7_joint', 'head_1_joint', 'head_2_joint', 'torso_lift_joint', '
    ↪ wheel_left_joint', 'wheel_right_joint', 'caster_back_left_1_joint', '
    ↪ caster_back_left_2_joint', 'caster_front_left_1_joint', 'caster_front_left_2_joint', '
    ↪ caster_back_right_1_joint', 'caster_back_right_2_joint', 'caster_front_right_1_joint',
    ↪  'caster_front_right_2_joint']
position: [1.0058956265627637, -0.6668913466995965, -1.354832333675927, 0.7812407815765756,
    ↪ -0.6853028725871546, 0.591213316532442, 1.7869163059953614, -0.010122909649999998,
    ↪ -0.07086036754999998, 0.34834603143986886, 0.028439153439153438, 0.02513227513227513,
    ↪ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
velocity: [0.11516672210821845, 0.017456504899373546, -2.93139096977664e-08,
    ↪ -0.18246975967800516, -0.8855937909277602, 0.22133122678905787, 0.8436615362030537,
    ↪ 0.0, 0.0, -0.0009925644529267174, -2.4405701631994282e-15, 0.002955700326565297, 0.0,
    ↪ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
effort: [0.301, 1.325, 0.666, 0.018, 0.016, -0.085, -0.085, 0.0, 0.0, 0.043, 1.002, -0.272,
    ↪ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

2. Generation of map (off-line)

This section describes how the recorded configurations were imported into MATLAB, the manipulability was calculated and how the poses were inverted to get all the base positions as seen from one end effector. The inverted poses are then binned to the same discretisation grid as has been used for the ICM and IHCM methods, 5 cm in translational and 1/8 rad in rotational dimensions.

The discretization is done to:

- enable the search to a base position from which multiple tool centre points can be reached.
- reduce memory size and computation time on the robot
- increase interpretability of the ITCM (see Figure 32 (p. 28))

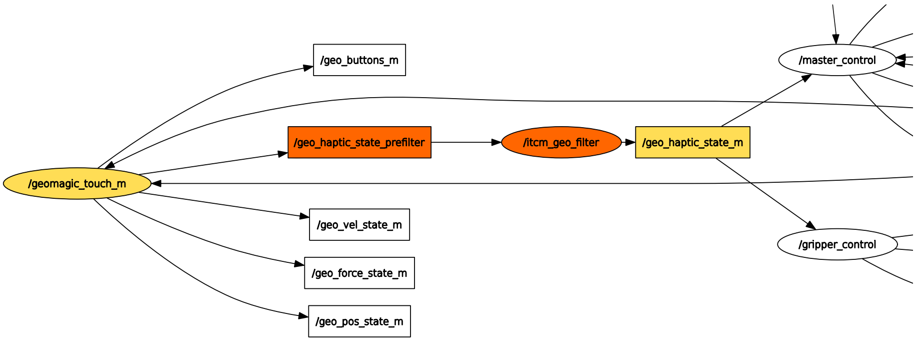
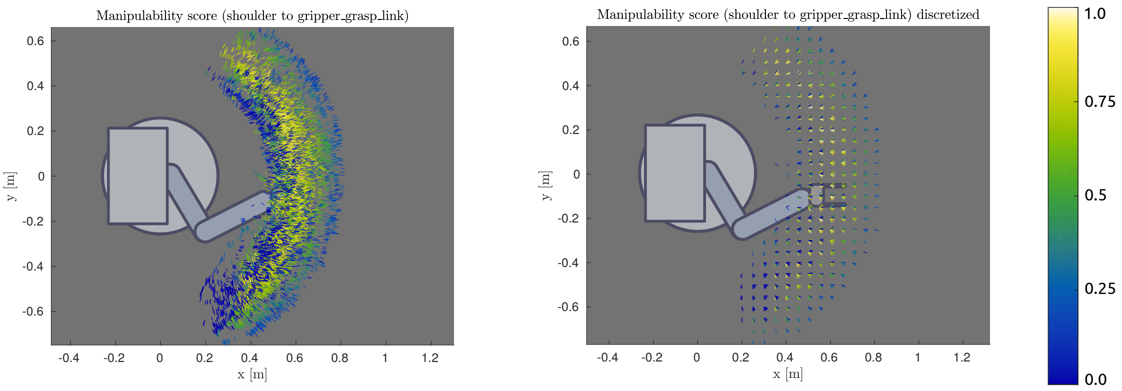


Figure 31: ROS nodes (elipses) and topics (rectangles) relevant to the ITCM data gathering filter. The orange topic has been remapped from `/geo_haptic_state_m` to `/geo_haptic_state_prefilter`, the orange node changes the contents of the messages to comply to relevant search space.



(a) Raw poses

(b) Discretized poses

Figure 32: Comparison of raw and discretized poses and their manipulability. Each line in the graph represents the y-axis of the `gripper_grasp_link` frame, see Figure 47 (Appendix E p. 43). The lines colour represents the manipulability score (normalized between zero and one). The robot is placed in the picture for context. In Figure 32a one tenth of the data is used to be able to plot it.

The visibility score is calculated on the robot during operation (on-line), see Appendix section 2 (p. 21).

1) Manipulability

Yoshikawa [3] presented a method to calculate the homogeneity of the possible directions of movement of a robot arm based on the kinematics of the robot's arm. A kinematic model (KM) of the arm is used with generalized coordinates (q). This describes the end effector state (ξ) as a function of the armâĂŹs model which is dependent on the generalized coordinates.

$$\xi = \text{kinematic model}(q) \tag{19}$$

The Jacobian of the kinematic model, or Instaneous Kinematic Model (IKM), describes the velocity the end effector can obtain in all directions for equal changes in joint angles. This is called the instantaneous velocity of the end effector.

$$w = \frac{\sigma_{\text{min}}}{\sigma_{\text{max}}} \quad [3] \tag{20}$$

The homogeneity is evaluated using a singular value decomposition (SVD) ($J(q) = U\,\Sigma\,T$). The positive diagonal matrix Σ stands for the scaling of the singular vectors, i.e. the singular values. If the minimal and maximum scaling are more equal, in that case the instantaneous velocity in all translational and rotational directions is more homogeneous. The manipulability (w) is calculated as fraction of the smallest singular value divided by the largest singular value.

2) Joint limit penalization

Vahrenkamp et al. [1] said that the same analysis of the manipulability is used as presented in [4]. There Vahrenkamp et al. presented a method of taking the joint limits into account, where it is also considered if the joint is moving towards or away from the nearest limit and if the joint near the limit is important or that it can be compensated by a redundant joint. This would complicate the manipulability calculation tremendously, and was not deemed necessary to test if positioning the robot autonomously would have merits. Therefore the same joint limit calculation is used, without taking the direction of movement into account. This joint limit penalization method has been presented in [5, p. 114].

$$P = 1 - \exp^{-k \cdot \prod_{i=1}^{7} \frac{(\theta_i - \theta_{i,lb})\,(\theta_{i,ub} - \theta_i)}{(\theta_{i,ub} - \theta_{i,lb})^2}} \quad [5, \text{p. } 114] \tag{21}$$

Because of the seven degrees of freedom a value of is used for k. The joint limit correct manipulability is calculated by multiplying the manipulability with the joint limit penalty.

$$w_{\text{corrected}} = w \cdot P \quad [5, \text{p. } 114] \tag{22}$$

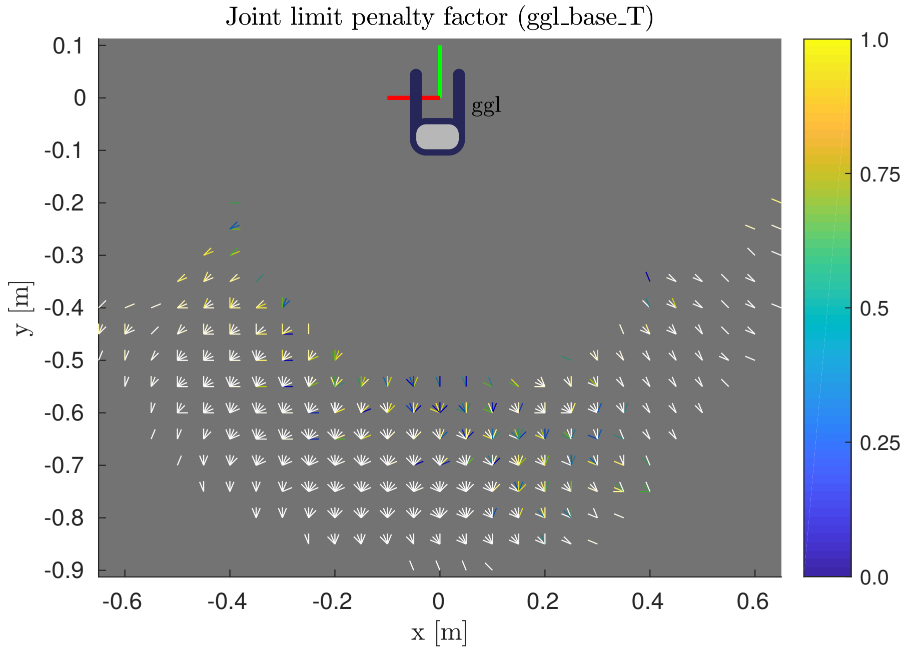


Figure 33: Joint limit penalization factors per base pose as seen from the `gripper_grasp_link`. Each line corresponds with the x-axis of a possible `base_footprint` frame as seen from the `gripper_grasp_link` frame. The lines are coloured according to the joint limit penalization factor with which the manipulability is multiplied. A penalization factor 1 hence does not influence the eventual manipulability, whereas a low penalization factor lowers the eventual manipulability.

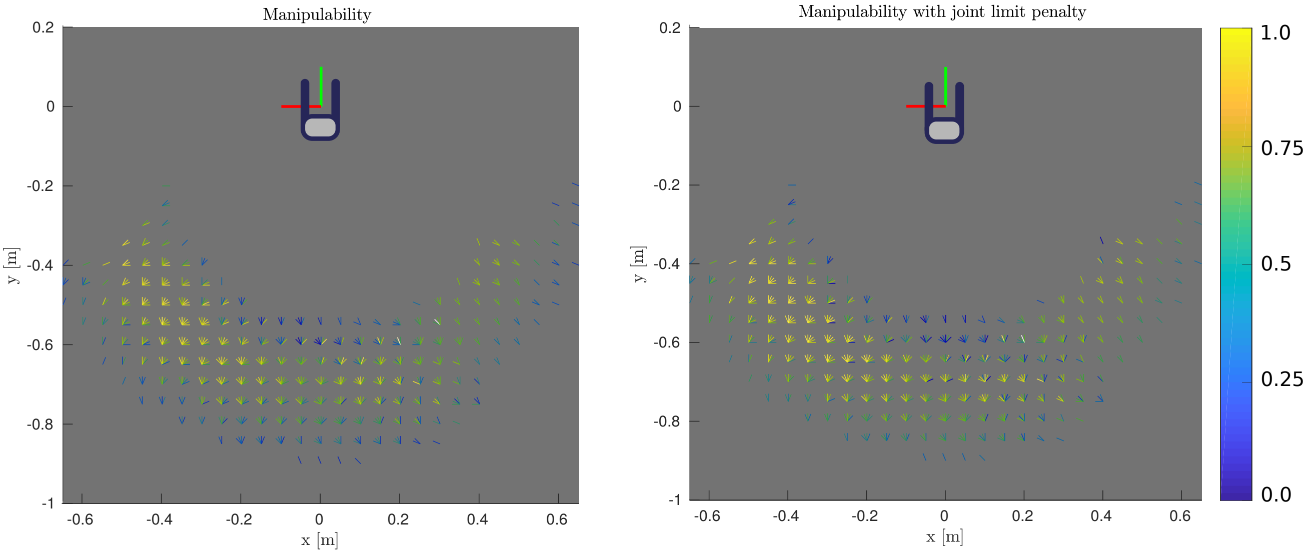


Figure 34: Comparison between the ITCM with and without joint limit penalization. The base poses which are blue in Figure 33 can be seen to be lower scoring in the joint limit penalized ITCM.

The penalization factor per base pose is shown in Figure 33. The effect on the manipulability per base pose is shown in Figure 34 as an comparison between with and without joint-limit penalization.

3) From joint states to ITCM

From the recorded joint states and the kinematic model, it can be calculated what the positions of each link of the arm was. In MATLAB, the Robotic Systems Toolbox of Mathworks is used to import the model of the arm of the Marco robot and to calculate link locations for all the joint states. The Robotics Toolbox of Peter Corke [6] is used to calculate the Jacobian (Inverse Kinematic Model) of the arm. This Jacobian is needed to calculate the manipulability, see above. With the joint states, the transform from the shoulder to the `arm_7_link` can be calculated (${}^{sh}_{a7l}T$, which is written in code as: `sh_ggl_T`). Because ${}^{a7l}_{ggl}T$ is constant, we can calculate ${}^{sh}_{ggl}T$ by

Table 3: Difference mean(std) of used recorded pose and discretized pose for translation ([m]) and rotation ([rad]).

	Translation [m]	Rotation [rad]
x	0.0112(0.0078)	0.0107(0.0253)
y	0.0109(0.0079)	0.0129(0.0167)
z	0.0100(0.0070)	0.0604(0.0649)

multiplying ${}^{sh}_{a7l}T$ and ${}^{a7l}_{ggl}T$.

$$ {}^{sh}_{ggl}T = {}^{sh}_{a7l}T \cdot {}^{a7l}_{ggl}T \tag{23}$$

Because we are only interested in the `gripper_grasp_link` poses on a certain height (for the purpose of the experiment), only poses within a margin of half the descretization resolution either way are used. Thereafter the poses are inverted from `gripper_grasp_link` poses in the shoulder frame, to `base_footprint` poses in the `world` frame. Besides inverting the ${}^{sh}_{ggl}T$ pose, the fixed transforms of ${}^{w}_{ggl}T$ and ${}^{base}_{sh}T^T$ are also needed.

$$ {}^{ggl}_{sh}T = {}^{sh}_{ggl}T^T \qquad {}^{sh}_{base}T = {}^{base}_{sh}T^T \tag{24}$$

$$ {}^{w}_{base}T = {}^{w}_{ggl}T \cdot {}^{ggl}_{sh}T \cdot {}^{sh}_{base}T \tag{25}$$

The inverted poses are then fitted to a six dimensional grid with a resolution of 5 cm in translational dimensions and 1/8 π rad in rotational dimensions. To let the grid be an as close enough match to the recorded poses, the distance in position and orientation to the nearest grid pose are calculated. A distance score is made that combines the position and orientation distance, which enables sorting on one metric. First for all recorded poses the distance in translational and rotational sense are calculated, these are normalized between zero and one and subsequently multiplied with each other per pose. The pose with the lowest distance score is selected, and it's joint-limit corrected manipulability is used for that grid pose.

The differences in translational dimensions between the original and the grid position are shown in Figure 35 (p. 31) and in Figure 36 (p. 32) the differences in rotational dimensions are shown. The values are never higher then half of the grid's resolution (0.025 m or 0.1963 rad), the mean is around 0.01 m (x, y, z), 0.01 rad (roll and pitch), 0.06 rad (yaw), which can be seen together with the standard deviation in Table 3. In translational dimensions, no particular thing can be noted about the spread of the distance between the selected and the measured pose. In rotational dimensions it is notable that the mean deviation is 0.01 rad which is a sixth of the mean deviation in yaw. This could be due to the fact that only the yaw angle of the gripper was steered by the master device, whereas the roll and pitch directions had the command to stay stationary. However, the z axis was also fixed during the recording. That the differences between the sampled and grid poses are not less in the x axis might be explainable due to optimisation of the controller of the slave. As has been suggested in section 1.

4) Data format

The resulting `base_footprint` poses in the `gripper_grasp_link` frame (${}^{ggl}_{base}T$) are written to a `.txt` file with the joint limit corrected manipulability score. In this file each line represents one pose and corresponding score.

score x y z $R_{1,1}$ $R_{1,2}$ $R_{1,3}$ $R_{2,1}$ $R_{2,2}$ $R_{2,3}$ $R_{3,1}$ $R_{3,2}$ $R_{3,3}$

$$ R = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{bmatrix} \tag{26}$$

This can than be interpreted on the robot (on-line) by the software described in Appendix A (p. 13).

3. Example tasks

As has been shown in the appendix regarding the online part (Appendix A), tasks can be generated to resemble opening doors, drawers or picking and placing objects. This can be seen in Figure 37.

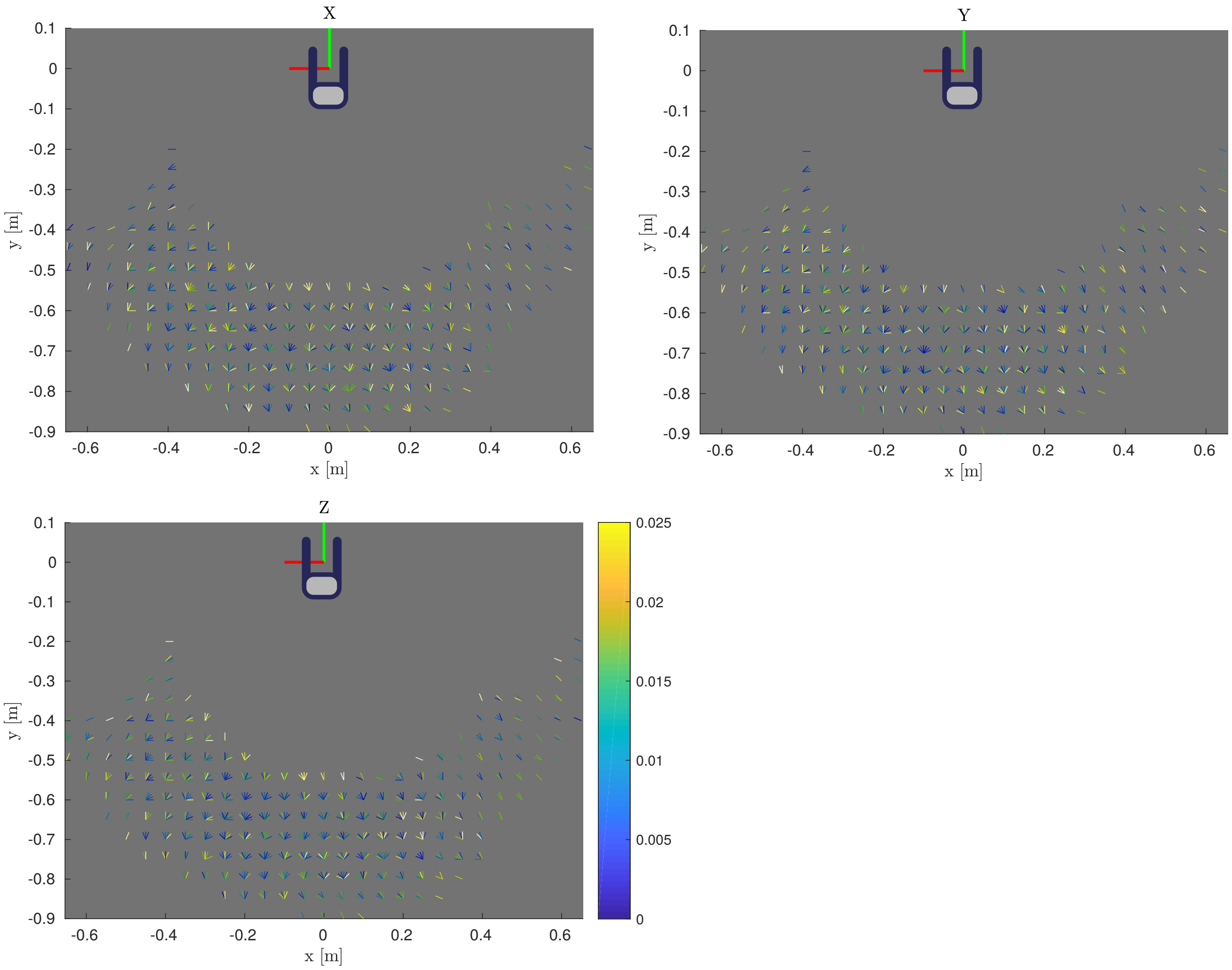


Figure 35: Difference between the grid position and selected sampled pose in meters. Gripper added for illustrative purposes.

4. Discussion and recommendations

For my research, the data was gathered in a plane, with a fixed lift height and only one rotational degree of freedom. To apply this method for tasks with other grasp orientations or lift heights, the workspace sampling should be expanded. An option to take the variability of the lift into account is to do the exploration at the lowest lift level, such that the base of this robot obstructs as much of the workspace that it possibly could, and the highest lift level such that the base obstructs a negligible part of the arm's workspace. For tasks that are on the floor, the lowest level map could be used and for tasks higher than a certain level the map made with the highest lift level could be used, because the base would not be a problem. Furthermore gripper poses close to the base do not have good manipulability and visibility which makes it unlikely that they would be chosen base poses unless very few options are available.

Care should be taken not to sample to much orientations. Due to the task generation and map merge method I implemented, only side and top grasps can be combined. Otherwise the points of the square grids do not coincide and values would not be able to be compared without interpolating. To be able to use the ITCM for all tasks that can be generated with the presented task generator, an ITCM must be generated that also covers the height of the entire workspace and top (and if need be bottom) grasps. To do an exhaustive workspace exploration to gather the data needed for a complete ITCM, three methods are envisioned:

- Complete simulation
  A high fidelity simulation of the telemanipulation setup is then needed, especially workspace scaling between the master and the slave device, the whole body controller, and the joint limits should be modelled

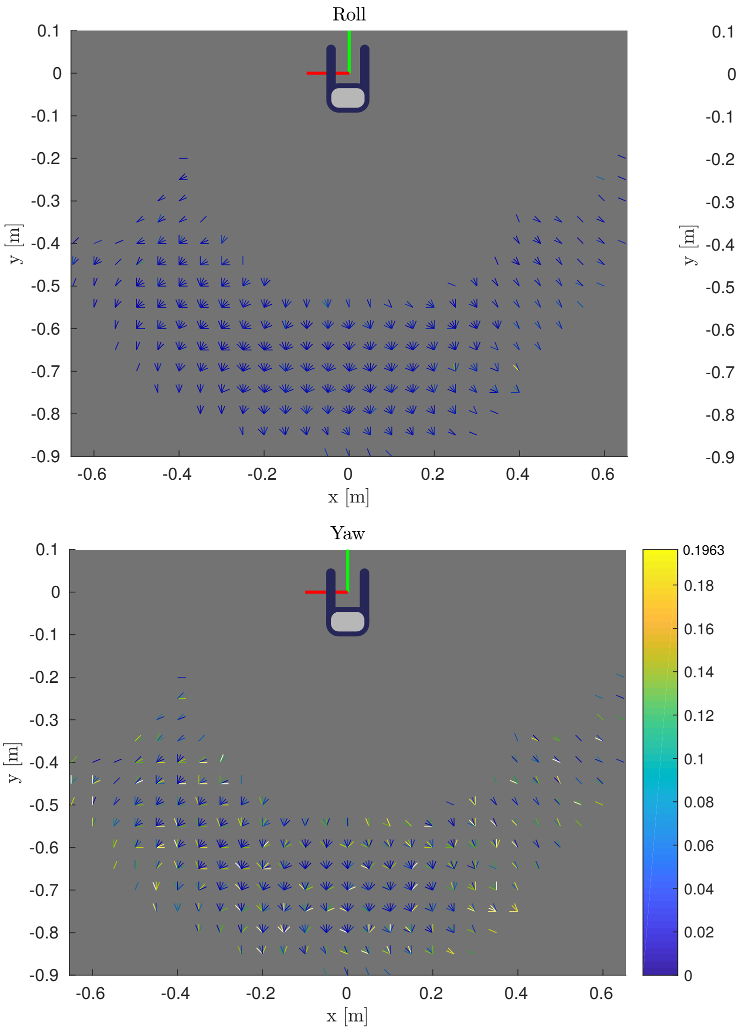


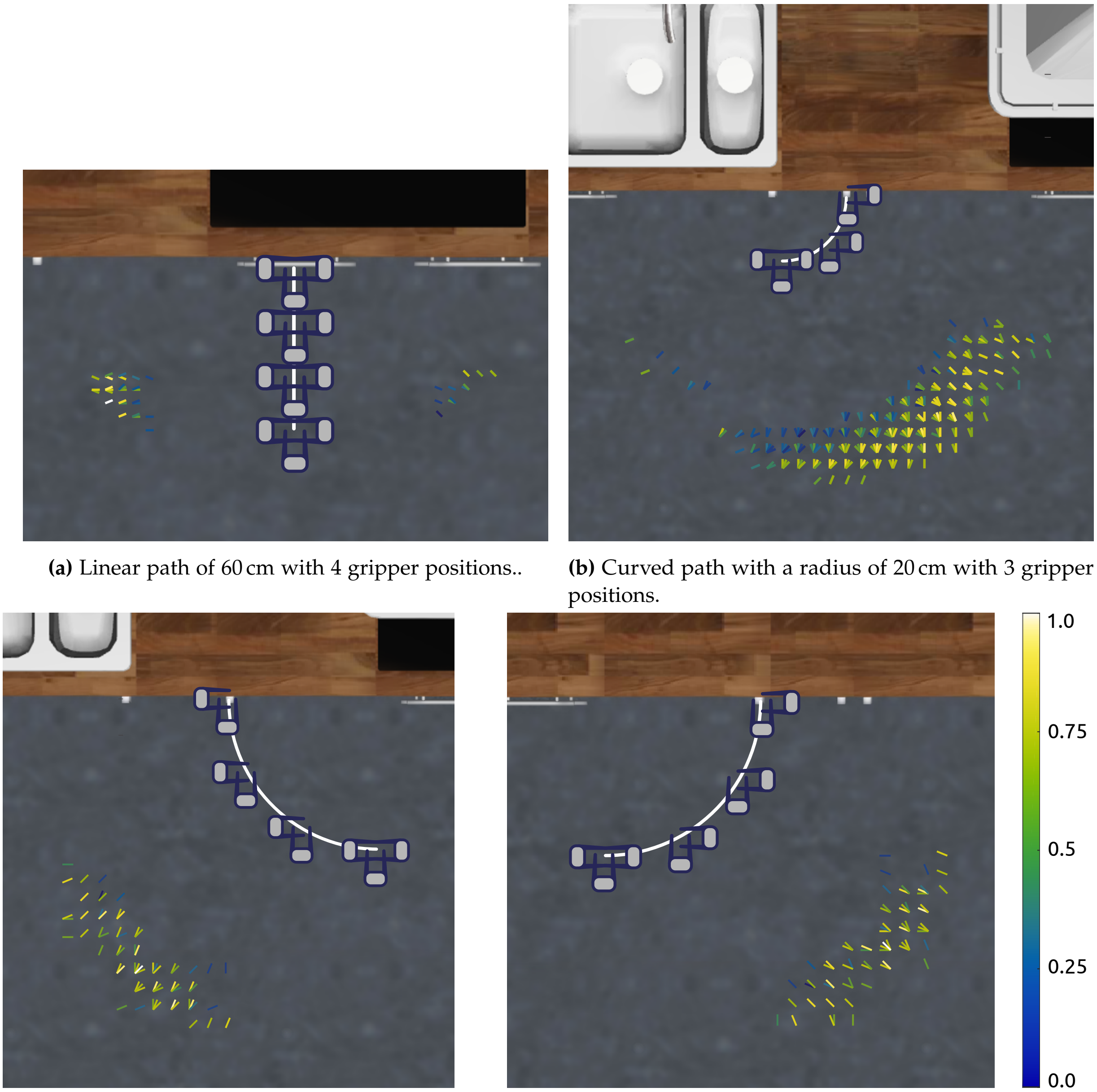
Figure 36: Difference between the grid and used sampled pose in radians. Gripper added for illustrative purposes.

accurately. The simulation setup could be an advantage if the real setup has limited availability.

- Master simulation
  Instead of editing the EE goal as is being send by the master, as has been done for this project, the EE goals can also be generated. To successfully do this, kinematics and self-collisions of the master device should be known accurately, as well as the resolution.
- Master actuation
  In the setup used for this project, the master device was capable of exerting force in translational directions. To make sure that all positions are sampled adequately these forces could be controlled to steer the master device to certain positions, maybe even corresponding to the eventual discretisation grid. The operator than still needs to operate the master in such a way that the relevant orientations are sufficiently sampled.

Besides the sampling, the ITCM method could benefit from taking the manipulability of the master devices configuration into account. Due to the fact that the Geomagic Touch Omni has six degrees of freedom and the space in which it operates has six degrees of freedom as well, the master device is non redundant. This is because it makes it easier to see what the operator is doing, what commands it gives. This ensures that for a given master pose, the masters configuration and thus the manipulability is set.

The development of this robot placement method has been influenced by the robot Marco. For example, if the lift is not generalisable as a prismatic but as a revolute joint, the possible base orientations would only coincide in the same position for a specific lift height. Because the transform between the `shoulder` and the `base_footprint` frame would than have a variable distance. This distance is taken into account while discretising the recorded poses and is thus done off-line, on-line adaptation to varying distance in the horizontal plane could thus not be taken into account. Continuing on the lift, the footprint of Marco's base is rather



(a) Linear path of 60 cm with 4 gripper positions..

(b) Curved path with a radius of 20 cm with 3 gripper positions.

(c) Curved path with a radius of 40 cm with 4 gripper positions.

(d) Curved path with a radius of 40 cm with 4 gripper positions.

Figure 37: The linear path has the same gripper orientation for each gripper position, and the curved path has different gripper orientations per position.

small. A larger footprint could make it harder to use a single (or two) ITCMs to describe the workspace for all lift configurations. Also for the visibility score, it is assumed that the camera is always at the same position, otherwise the displacements due to varying *yaw* and *pitch* angles would be taken into account. For Marco these displacements are in the order of centimetres and can thus be neglected for the visibility score. For other robots this might not be the case.

Although the ITCM is based on recorded configurations, the eventual map contains poses. Because mobile manipulators generally have more degrees of freedom in their arms than six, the arm can reach most poses in multiple ways. It is thus not necessarily so that the recorded configuration is actually the configuration that will be used to do the task after the robot has been positioned. Also the grippers are placed within the restrictions due to the merge method of the maps, such that they do not necessarily describe the task at the centimetre precision or with not precisely the correct orientations. And what is the manipulability and visibility score of the path in between a pick and a place position? On the other hand, the execution of the task is not set in stone. The operator is free to do pick the object with an top-grasp when it is positioned for a side grasp. Likewise, it is hard to imagine the workspace of a robot where two good reachable positions are not reachable from each other. Thus a good reachable path is likely to exist between both positions.

In Figure 34 the ITCM can be seen for one gripper. It is notable that the map is larger on the left side than on the right. This might be due to the manual sampling of the workspace. Although a sampling time of around 1 h is quite extensive for an area of 1.2 m long and 40 cm wide. A more logical reason seems to be that the arm is on the left side of the robot's centre. This can be seen in Figure 47 (Appendix E, p. 43).

Although the task generation is restricted by the map discretisation, a wide range of tasks can be generated, as has been shown in this appendix.

In practice, the robot will most likely not be positioned at the precise pose as selected by the TS-ITCM. This maybe due to errors in sensors that detect the start pose of the task or sensors that localise and thus verify the robot's pose or both. Precise positioning might not be necessary, the highest scoring pose of the TS-ITCM in the paper is not far away from the best pose of an expert operator. Moreover, the experiment was conducted with real hardware thus the sensor errors discussed above were also present. This did not, however, cause the performance to collapse nor the operator effort to increase. Abolghasemi et al. [7] developed a method to estimate a reachability score with Gaussians. This would add another layer between the original and map data, which increases the chance of base positions being labelled as possible while they are not. On the other hand base poses which might result in a do-able tasks probably are not the best base positions. Estimating the TS-ITCM with Gaussians could increase task generation flexibility because the discretisation grid would not be so important.

Strengths of the ITCM methods are that it is based on data of the real world setup which wildly increases the change of successful task execution and that the base poses are scored based on manipulability and visibility. Additionally, the ITCM is usable to generate task specific maps for a wide range of tasks.

References

[1] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," *Proceedings - IEEE International Conference on Robotics and Automation*, no. 2, pp. 1970–1975, 2013.

[2] G. Touch, "Haptic force feedback device," *Geomagic: http://geomagic. com/en/products/phantom-omni/overview*.

[3] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[4] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, "Manipulability analysis," *IEEE-RAS International Conference on Humanoid Robots*, no. 3, pp. 568–573, 2012.

[5] M.-J. Tsai, *Workspace geometric characterization and manipulability of industrial robots*. PhD thesis, The Ohio State University, 1986.

[6] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised*, vol. 118. Springer, 2017.

[7] P. Abolghasemi, R. Rahmatizadeh, A. Behal, and B. Ladislau, "A real-time technique for positioning a wheelchair-mounted robotic arm for household manipulation tasks," *Workshop on Artificial Intelligence Applied to Assistive Technologies and Smart Environments (ATSE-16) at AAAI-2016*, pp. 2–7, 2016.

D. Action Related Place (ARPlace)

Stulp et al. [1] proposed the Action-Related Place (ARPlace) method, in which a generalised success model (GSM) is learned for a task. In [2] they expanded the ARPlace method to pick advantageous objects if multiple are available or multiple objects from the same place if they are sufficiently close. And they showed how the ARPlace method could be extended with various cost functions, for example distance to robot (when one of multiple equal objects must be picked). Experience based learning is used to differentiate between reachable and successful positions. Furthermore, instead of selecting a best position, the output of ARPlace is a distribution representing the probability that a grasp from that position will be successful. This distribution is based on the learned model as well as the state estimation of the robot and target object. Because the position is represented as a probability distribution, a robot can navigate with 'least commitment'. This enables a robot to cope with changing task parameters and changing parameter knowledge of a robot.

ARPlace uses the reachability map as presented by Zacharias et al. [3] to distinguish between theoretically reachable and unreachable positions. Thereafter it evaluates the success of theoretically reachable positions through experience based learning. This learning is first done off-line in simulation and continues on-line in the real world, the results are stored with respect to Task Relevant Parameters (TRP).

In this Appendix the implementation is briefly elaborated and the it's shortcomings discussed.

1. Generate test configurations

Similar to the process in Appendix B, a file is generated with poses to be reached. In this case however the robot would be 'moved' and the end effector (EE) goal would stay stationary. This is however compensated for in the generation of EE goals for the inverse kinematics program.

The test results of the inverse kinematics (IK) are written to a file that can then be interpreted by the MATLAB program. The MATLAB script takes as input the variation of the base position in x and y coordinates with respect to the origin and the variation of the EE goal pose. The EE pose can be varied in distance from the table edge and orientation around the goal object, see Figure 38. Rotation of the EE has not been investigated further. For simplicity only a top grasp has been investigated and due to the Degrees of Freedom (DOF) in the wrist of the tiago arm, no relevant results were expected. For a side grasp, the rotation does become relevant however. In this implementation of the ARPlace method, not the goal object but the goal pose of the EE is varied. Because I did not want to move the robot or re-initialize the simulation with different robot start positions, one single start position is used. The robot is stationed at 'world' origin. To evaluate the EE poses from the varying base positions, the pose needs to be transformed per varying base position to account for the non varying base.

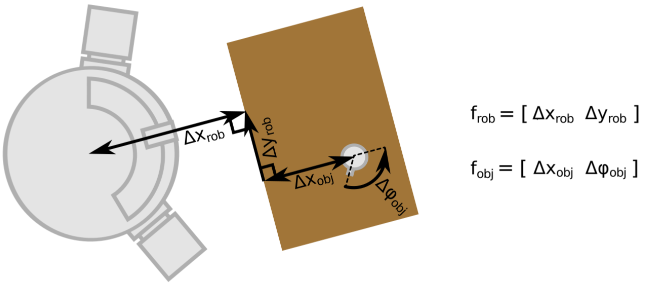
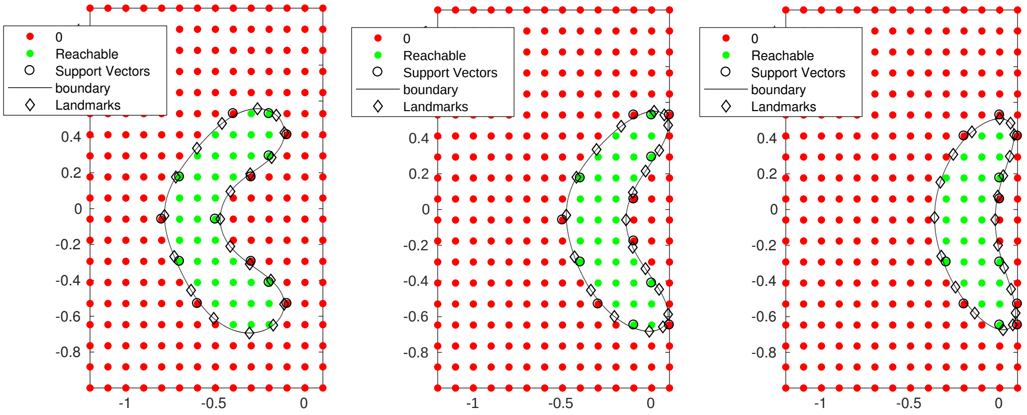

Δx_{rob}
Δy_{rob}
Δx_{obj}
$\Delta \varphi_{obj}$
$f_{rob} = [\ \Delta x_{rob}\ \ \Delta y_{rob}\ ]$
$f_{obj} = [\ \Delta x_{obj}\ \ \Delta \varphi_{obj}\ ]$

Figure 38: Feature space of the robot and the goal object. [2]

2. Inverse kinematics

MoveIt is used to evaluate the theoretic reachability of EE locations from a slew of base positions. In MATLAB a selection of base positions is generated for a selection of EE goals. The distance and rotation to these goals from the base position are written to a `.txt` file. This file is read by a ROS node I wrote which uses the `setFromIK` function from `MoveIt!` to evaluate if it is reachable. The result is written to a `txt` file which is interpreted by a MATLAB program.



(a) x_{ee} is 0 cm **(b)** x_{ee} is 30 cm **(c)** x_{ee} is 40 cm

Figure 39: Base positions relative to world origin (0,0), the EE goal is at x = 0.30 m. The base position from which the EE goal is reachable are marked green, the boundary of the classification SVM is shown and the landmarks chosen from that boundary.

The IK check can be extended by a simulation of the situation. Then an extra classifier SVM could be added to destinghuish between robot positions from which simulation resulted in success and positions from which the inverse kinematic check resulted in success. Another option is to use the manipulability measure [4], then a regression SVM should be implemented to represent that information. Due to time constraints these options were not implemented.

3. From reachability data to ARPlace

1) Support Vector Machine

Per EE goal the difference between robot poses from where it can reach the EE and robot poses from which it cannot reach the EE goal can be described by an classification Support Vector Machine.

The data file from the MoveIt node is loaded in to MATLAB and per EE goal a classifier SVM is learned. The matlab function `fitcsvm` has been used with an infinite `BoxConstraint` and `KernelScale` of 1. The `BoxConstraint` represents the inverse of misclassified admissibility, I have chosen the largest possible value because I want to avoid that base positions are included from which the EE goal could not have been reached. The `KernelScale` represents σ. The data in the datafile (except the labels) would be divided by σ, but I have chosen $\sigma = 1$.

In addition to the results of the MoveIt test, an outer layer of positions with failed IK attempts is added to ensure closed SVM's.

Because we are going to use the SVM to generate a Point Distribution Model as opposed to using the SVM as a classifier, all the data in the data-file is used to train the SVM. The resulting boundaries can be seen in 39 as the black lines. The support vectors are denoted by the black rounds just a little bit bigger then the green and red points.

2) Point Distribution Model

It can be seen in figure 39 that the boundaries shape is dependent on the distance of the EE goal to the tables edge. In order to predict the shape and position of a boundary based on a new EE goal, a shape description is needed. The shape will be described by landmarks on the boundary. Because the position of the boundary is relevant is well, the boundaries will not be aligned before landmark selection.

3) Landmark selection

To select the landmarks these options have been identified:

- Manual selection by a person
- Automated selection based on properties of the shape
- Automated equidistant selection.

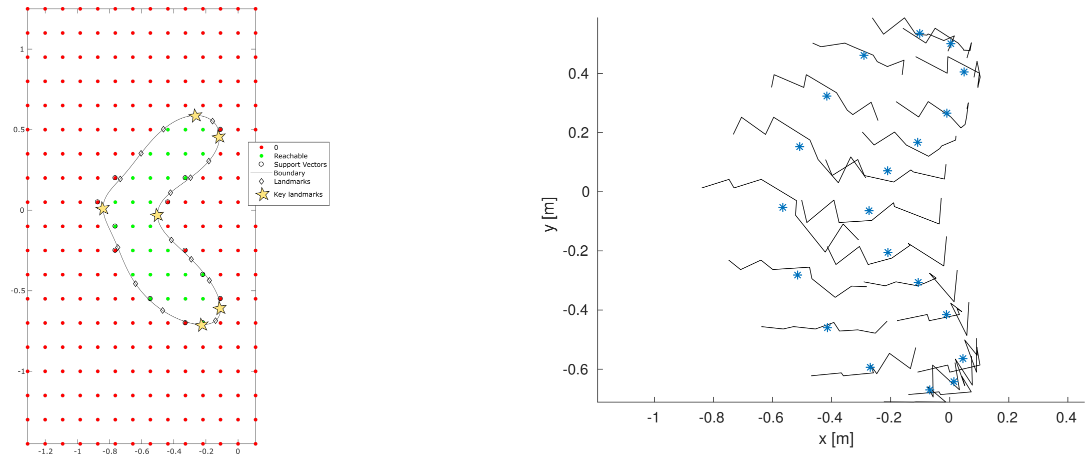
- Automated selection based on minimization of a Point Distribution Model related cost function

In the image recognition domain, usually the landmarks are selected by people. To prevent variance due to imprecise selection by humans, and to enable fast and low effort processing of the boundaries produced by the SVM, I wanted automated selection. In the paper of [2] they used an optimization algorithm to minimize the error between original and reconstructed contours and to maximize the amount of energy in the least amount of degrees of freedom of the Point Distribution Model. One landmark at a time, the position is varied and evaluated. They noted however that their algorithm is "the most computationally intensive step in the off-line learning phase".

Because neither the manual selection, nor the automated cost minimization seemed appealing, and because the boundaries I got had a distinct 'bean-like' shape, I wrote some code to select landmarks based on the shape. Points with the smallest x and y and points with the largest x and y were of interest because these are easily selected with the help of a computer, see Figure 40a. Around the same height of the x_{min} landmark, there can be seen a local x_{min} landmark, above and below a global and local x_{max} landmark can be seen.

Between these 6 points, 14 additional points are split between the segments of the boundary defined by the first 6 points. On those segments, the points are assigned equidistantly. Therefore the assumption is made that the shape of the SVM's boundary is constant enough to enable this method for landmark selection.

The selection of landmarks enables processing of the raw data by the point distribution model approach. The resulting landmark distribution for all the boundaries currently analysed can be checked with the figure shown in Figure 40b.



(a) Key landmarks used for landmark selection

(b) The lines are the landmarks of all the boundaries and the $*$ is the position of the same landmark number over all boundaries.

Figure 40: Selection of landmarks and resulting landmarks for a number of different boundaries.

4) Calculating the Point Distribution Model

For each sample M in the task space (varying object location) a column of length equal to the amount of landmarks (k) multiplied by the amount of dimensions the landmarks are defined in (in this case two), is added to the matrix **H**.

$$\mathbf{H} = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^M \\ y_1^1 & y_1^2 & \dots & y_1^M \\ x_2^1 & x_2^2 & \dots & x_2^M \\ y_2^1 & y_2^2 & \dots & y_2^M \\ \vdots & \vdots & & \vdots \\ x_k^1 & x_k^2 & \dots & x_k^M \\ y_k^1 & y_k^2 & \dots & y_k^M \end{bmatrix}$$

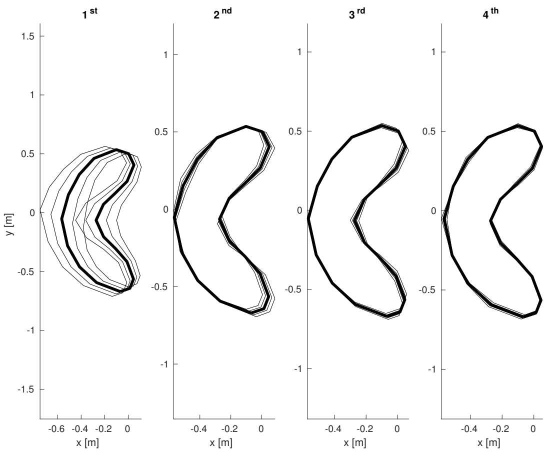


Figure 41: The first four deformation modes (thin black lines) and the mean boundary ($\mathbf{H}$) as a thick black line.

The mean of coordinates 1 till M is called $\bar{\mathbf{H}}$ and represents the average shape and location of the success boundary. The point distribution model is defined as $h_{new} = \bar{\mathbf{H}} + \mathbf{P}\, b_{new}$. This means that we can generate new boundaries based on the mean boundary ($\bar{\mathbf{H}}$), the deformation of the boundary per deformation mode ($\mathbf{P}$) and the objects location (b_{new}).

The $\mathbf{P}$ matrix is composed of the eigen vectors corresponding of the covariance matrix of $\mathbf{H}$. To reduce computational cost, the matrix $\mathbf{P}$ maybe composed only of the eigen vectors corresponding to the largest eigenvalues of the covariance matrix. The sum of these largest eigenvalues should be more then the sum of all the other eigen values.

Getting a feel for what the eigenvectors represent, we can isolate the effect of each eigenvector by forcing the rest in the b_{new} vector to zero. Then we can multiply that eigenvector with it's positive and negative eigen value. For the first deformation mode that would be done like this:

$$h_{\text{1st deformation mode}} = \bar{\mathbf{H}} + \mathbf{P} \begin{bmatrix} 0 \\ \vdots \\ \sqrt{\sigma_1} \end{bmatrix}$$

How b_{new} is generated based on a goal object position, is shown in the generalized success model section.

5) Generalized Success Model

To generalize the knowledge in the PDM, we need to relate the change of the EE goal to the change of the boundary relative to the mean boundary. The change of the boundary can be expressed by calculating for each EE goal position, the linear regression between the difference between the boundary (H_m) for the EE goal position m and the mean boundary $\bar{\mathbf{H}}$ and the matrix of eigen vectors $\mathbf{P}$. We can calculate B as follows, and $\mathbf{B}$ will be a matrix with as much rows as landmarks, multiplied by the amount of properties per landmark (for 20 landmarks and x and y as properties, this would be $20 \times 2 = 40$) and $\mathbf{B}$ will have as much columns as M boundaries.

$$\mathbf{B} = \begin{bmatrix} B_1 = \mathbf{P}^T\left(H_1 - \bar{\mathbf{H}}\right) & B_2 = \mathbf{P}^T\left(H_2 - \bar{\mathbf{H}}\right) & \dots & B_M = \mathbf{P}^T\left(H_M - \bar{\mathbf{H}}\right) \end{bmatrix}$$

If we introduce a vector T in which the EE goal positions are stored: $T = \begin{bmatrix} 0 & 0.05 & 0.10 & \dots & 0.45 \end{bmatrix}^T$.

The $\mathbf{W}$ matrix is calculated as the linear relation between the EE goal poses (T) and the boundary difference matrix ($\mathbf{B}$). Thus the $\mathbf{W}$ matrix describes the amount of change in the boundary as result of a change in EE goal pose.

$$\mathbf{W} = \left(\begin{bmatrix} \mathbf{1} & T \end{bmatrix}\right) \backslash \mathbf{B}^T$$

With the $\mathbf{W}$ matrix, we can construct new deformation vectors (b_{new}) based on new EE goal positions. Because $\mathbf{W}$ is the linear relation between T and $\mathbf{B}$, rewriting the previous equation to insert a new EE goal position, gives

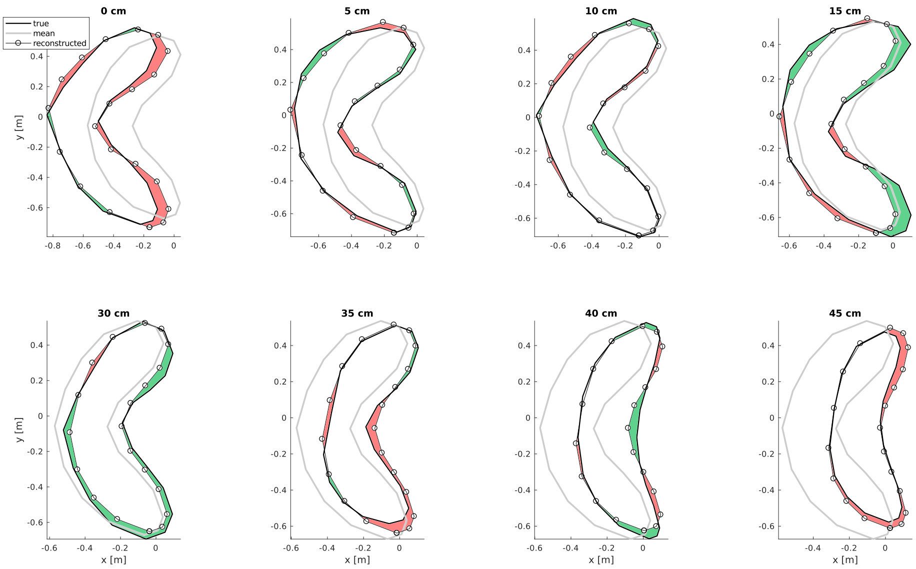


Figure 42: Mean, original and reconstructed boundaries for a number of object distances. The green areas denotes an incorrect exclusion by the reconstructed boundary, the red denotes an incorrect inclusion by the reconstructed boundary.

us the new deformation vector b_{new}. To check the power in each deformation mode the plot in Figure 41 is made. It can be seen that the most power is present in the first deformation mode. This determines the amount of eigenvectors there should be in the **P** matrix.

$$b_{new} = \left(\begin{bmatrix} \mathbf{1} & t_{new} \end{bmatrix} \mathbf{W} \right)^T$$

In Figure 42 a reconstruction error can be seen between the original boundaries and reconstructed boundaries (the mean of all the original boundaries is shown for context). For this reconstruction, the first two deformation modes of the PDM are used. It can be seen in figure 41 that the first two would represent enough of the deformation energy to reconstruct the shape accurately. The reconstruction errors seen in Figure 42 vary between a half centimeter and a couple of centimeters in each boundary.

6) Merge ARPlace

How the probabilistic map should be merged, is depended on if the maps concern the same object or not. If the map concerns the same object from multiple directions, a union of the maps should be made. To make a union of the map for each pixel in the map, the largest value is selected. Because the change of a successful grasp is not increased if the robot is able to grasp it in multiple orientations.

If it concerns two (or more) objects from the same direction, the probability of two successful grasps from the same position should be evaluated. Therefore, an element-wise multiplication of the two (or more) maps is done (this is easy because they share the origin of the table as centre).

4. Generate ARPlace on-line

To evaluate the ARPlace method in simulation, ground truth data is used from the /gazebo/model_states topic. The node `pub_map_merged` creates two `goalobject_sub` objects, which subscribe and save the ModelStates of objects whose first part of their name correspond to ‘table’ (the surface) or ’coke’ (the object). See Figure 43.

For each ‘coke’ object found in the simulation an `arplace_obj` is generated in which an ARPlace distribution is stored. If the probability of reaching all the ‘coke’ objects is larger then a value hard-coded in `pub_map_merged`, that distribution is published, if not the distribution corresponding to the first ‘coke’ object is published. How these maps are combined is explained in the previous section (6), p. 39).

To refrain the robot from navigating into the table, I wanted to subtract those disadvantageous positions from the ARPlace distribution using a inflated obstacle costmap. To explore this I overlayed the ARPlace distribution over the costmap, the result is shown in Figure 44. Considering we want to use the inflation layer to subtract from the ARPlace distribution, would the yellow square (denoted by the black arrow) be the best position or should the value of the inflation layer be subtracted first? This question has been circumvented by excluding a fixed distance from the tables edges in the ARPlace distribution.

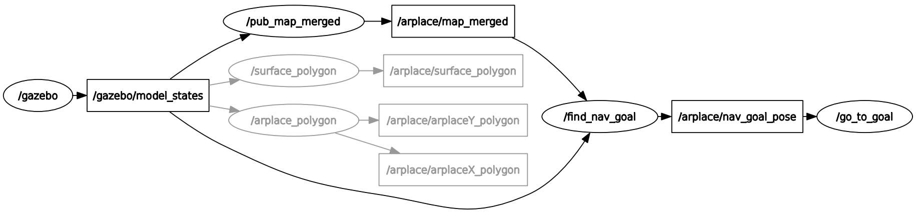


Figure 43: Overview of the ARPlace method's on-line part. The black parts are essential, the gray parts are for the polygon visualisation in rviz.

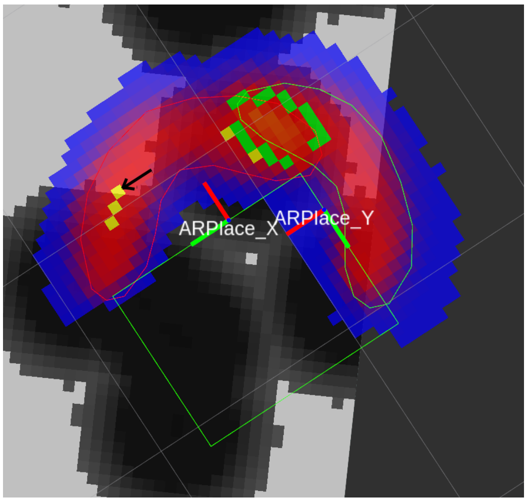


Figure 44: ARPlace distribution overlayed on the inflated obstacle costmap. Black arrow added for illustrative purposes.

The height (z) and the distance to the edge (x) are varied in Figure 45. For two power-content percentages a Generalized Success Model is made, for 90 % (3 DOF in GSM) and 99 % (7 DOF in GSM). For reconstruction a polynomial fitting tool has been used instead of the linear method for the 1DOF case as has been discussed in the rest of the report. A second degree polynomial has been used, following the method as used by [2].

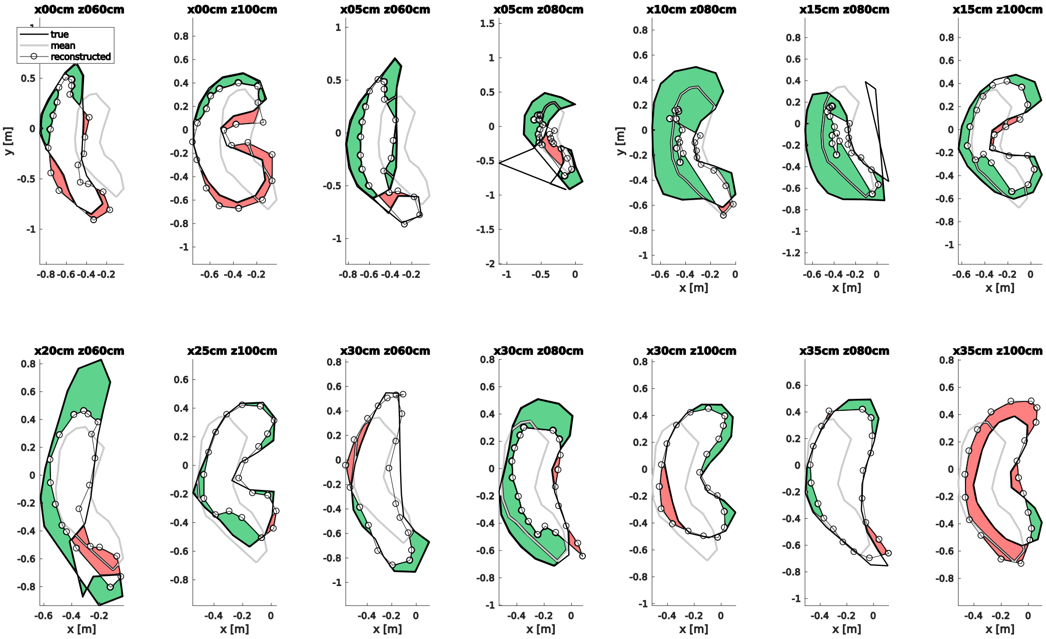


Figure 45: Height and distance to the edge is varied.

5. Evaluation

The ARPlace method of [2] has been implemented because it matched the most requirements from the literature study. The goal of the literature study was to find an navigational goal selection method that could best be used to support or replace positioning the robot preceding a telemanipulation task.

As a result of hands-on experience, the ARPlace method compliance to these requirements can be better assessed. Although imprecise goal object locations could be handled by the other methods reviewed in the literature study as well by doing a Monte-Carlo simulation, the ARPlace method has an advantage due to the computationally-light matrix multiplication needed to find boundaries wherein a successful grasp is possible. Evaluating which points are within how many boundaries is, however, computationally more expensive. ARPlace keeps it's advantage here.

The projected advantage of learning the limitations of the control algorithms of the robot by simulations is not applicable to the telemanipulation case. Although ARPlace could learn from experience of grasps done with telemanipulation to incorporate the new experiences the off-line part would have to be run again.

When the task description was extended from one to two parameters ($x_{\text{table edge}}$ and γ_{cup}) in order to enable side grasps, a large part of the generated boundaries were not usable because there were multiple boundaries instead of one. This was not compatible with the Point-Distribution-Model, see Figure 45 (p. 40).

Stulp et al.'s ARPlace method uses autonomous grasping in simulation to find out from which positions grasps are possible, however from these positions human telemanipulation strategies are not necessarily possible as well. To account for these human strategies, splines can be added to the path in a similar way as extra objects have been added.

References

[1] F. Stulp, A. Fedrizzi, F. Zacharia, M. Tenorth, J. Bandouch, and M. Beetz, "Combining analysis, imitation, and experience-based learning to acquire a concept of reachability in robot mobile manipulation," *9th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS09*, pp. 161–167, 2009.

[2] F. Stulp, A. Fedrizzi, L. Mösenlechner, and M. Beetz, "Learning and reasoning with action-related places for robust mobile manipulation," *Journal of Artificial Intelligence Research*, vol. 43, pp. 1–42, 2012.

[3] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: Representing robot capabilities," *IEEE International Conference on Intelligent Robots and Systems*, pp. 3229–3236, 2007.

[4] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

E. Telemanipulation system

The telemanipulation setup consisted of laptop with a Geomagic Touch [1] as the master and a protype Tiago robot (Marco) as the slave. With both controllers connected over WiFi. The operator gives position and orientation commands via the master and receives forces (no torque) as haptic feedback. The operator also received visual feedback from the 'cockpit' software (see Figure 48) from the camera in the head and the camera on the wrist of the robot. As the task was carried out behind the operator (see Figure 46), he or she could hear when the cup touched the table.

The slave, Marco, is shown in Figure 47. The slave receives position and orientation goals and tries to reach them with its `arm_7_link` frame. Commands are interpreted with respect to the `base_footprint` frame. If the goal is out of the slave's workspace, the controller mostly tries to obey the position command opposed to the rotational command.

The master side is depicted in Figure 48 consists of a laptop running Ubuntu 14.04. In Figure 49 it is shown how the commands from the omni are changed in between trials of the participant. Just as during the recording of the ITCM, see Appendix C, the data (or messages) from the Geomagic Touch are passed through a new node. But instead of always changing the messages, the messages are only edited after the pick-and-place task has been completed. This is detected by keeping track of the condition of the experiment, and thus the location of the task. The condition is fed in the software by the researcher. If the `gripper_grasp_link` has been near the place location, but the master is near the start position, the the inserted node takes over control from the geomagic touch. It sends commands to go to the original pose and to open the gripper. In this way the position of the arm's end effector and the state of gripper is the same before each start of the experiment.

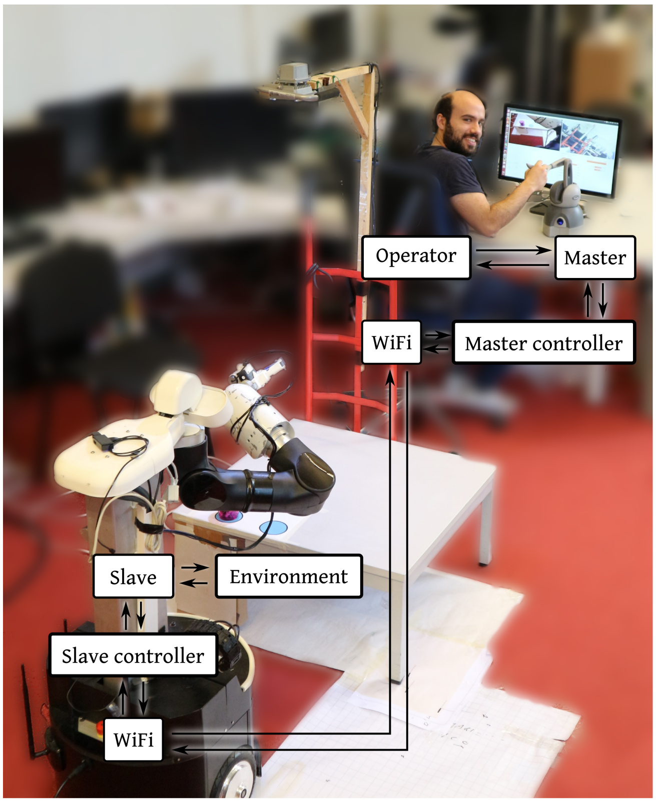


Figure 46: Setup of experiment with an overlay of the different components of the telemanipulation setup.

References

[1] G. Touch, "Haptic force feedback device," *Geomagic: http://geomagic. com/en/products/phantom-omni/overview*.

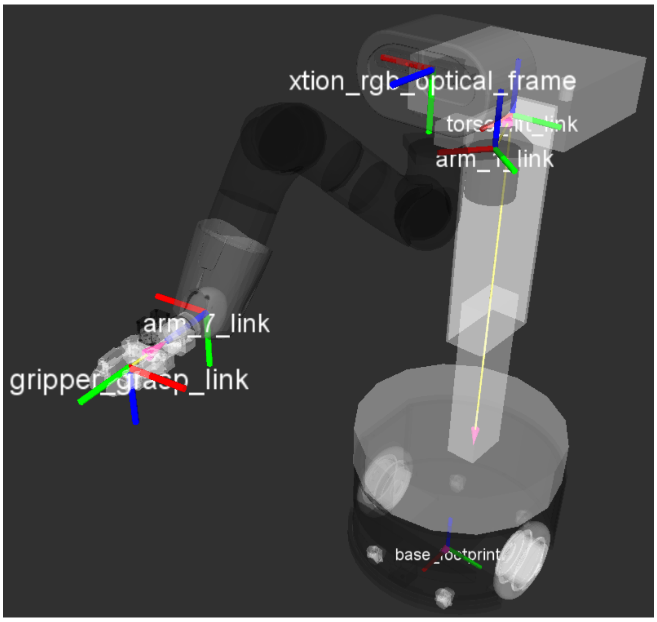


Figure 47: Robot model of Marco with relevant frames (`base_footprint`, `torso_lift_link`, `xtion_rgb_optical_frame`, `arm_1_link`, `arm_7_link`, `gripper_grasp_link`). The axes are labelled with colours in the RGB, xyz standard. Thus Red is the x axis, Green the y and Blue the z.

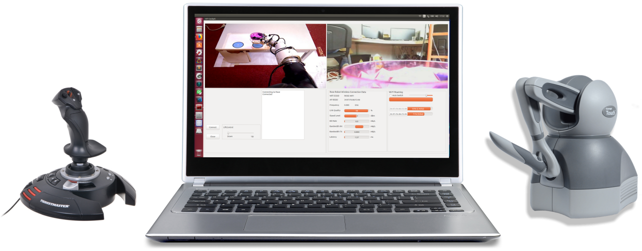


Figure 48: Overview of cockpit setup with Thrustmaster joystick, a generic laptop running ubuntu 14.04 with the cockpit software developed by Heemskerk Innovative Technologies and Geomagic Touch Omni.

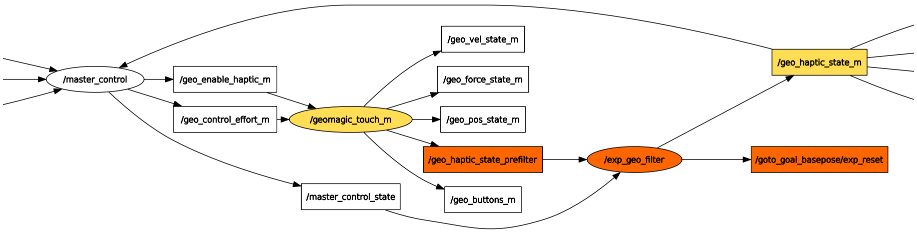


Figure 49: ROS nodes (elipses) and topics (rectangles) relevant to the experiment. Similar to Figure 31 (p. 27), the orange nodes are added to take over control from the participant after and reset the configuration of the arm.

F. Results of the experiment

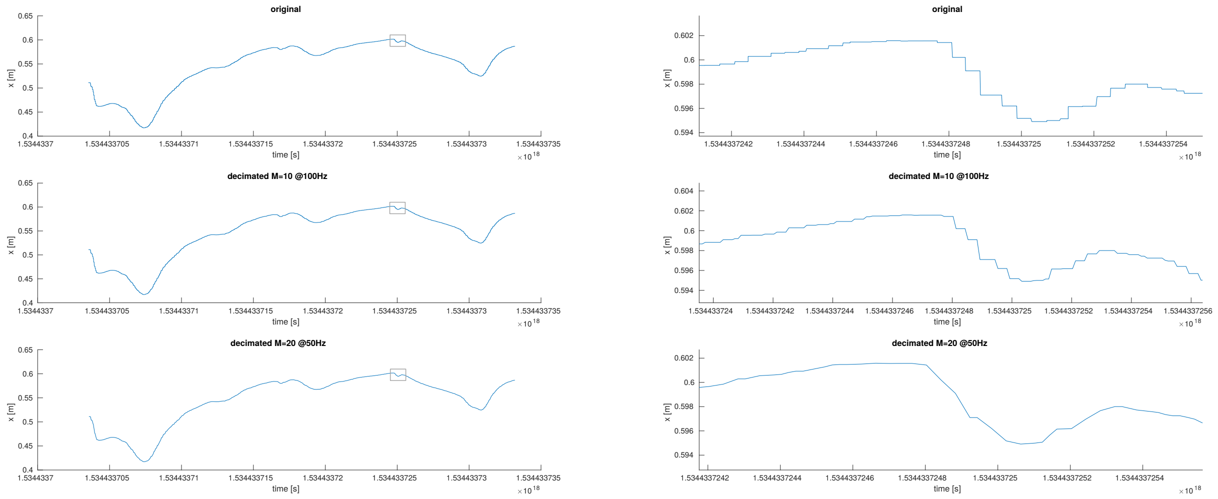
The same control strategy was used for all conditions People do the same, it only takes longer.

1. Data acquisition

Data was recorded with the rosbag [1] system. In the ROS framework, programs (nodes) communicate via topics. A node can publish on a topic and another can subscribe to it. With the rosbag program all messages on the relevant topics are recorded, some topics are recorded would anything have gone wrong it could be used to deduce what the problem is.

Data of the master and slave device was recorded at 1 kHz. It was filtered with a second order butterworth filter with a cut-off frequency of 50 Hz in both the backwards and forwards direction (to prevent phase shift and doubling the filter's order). A cut-off frequency of 50 Hz was chosen because it is double the highest frequency humans could produce (25 Hz [2]). And therefore nothing of the original signal would be filtered out, the result can be seen in Figure 50. After it was filtered, the data was interpolated linearly to 50 Hz.

All other data was not at such high frequencies that down-sampling was necessary.



(a) Entire trial

(b) Zoomed in part of the trial

Figure 50: The x position of one participants trial is shown. On the left the entire trial and on the right a zoomed in part of the trial. The top panels show the original data, the middle panels the filtered and interpolated data at 100 Hz and in the bottom panels the data is once again filtered and interpolated to 50 Hz.

2. Raw data

In Figure 51 (p. 45) the travelled path of the slave can be seen in $x\,y$ coordinates. It has been chosen to depict te slave's path instead of the masters, because it can be linked to the task and is therefore easier to interpret. From the figure it is apparent that the table is not as easy to reach from the base pose belonging to the ITCM-low condition as from either the base pose of the ITCM or Expert condition. The distance in $x\,y$ is farther for the ITCM-low than both the Expert and ITCM conditions. When comparing the ITCM with the Expert condition, it is conspicuous that the gripper starts almost above the table in between the pick and place markers. Whereas the gripper of the base position of the Expert condition starts in front the place marker around 10 cm away from the table.

The distance to the place marker over time shows also a larger distance to be covered from the base pose belonging to the ITCM-low condition (Figure 52 (p. 46)). Because the task is in six dimensions (translational

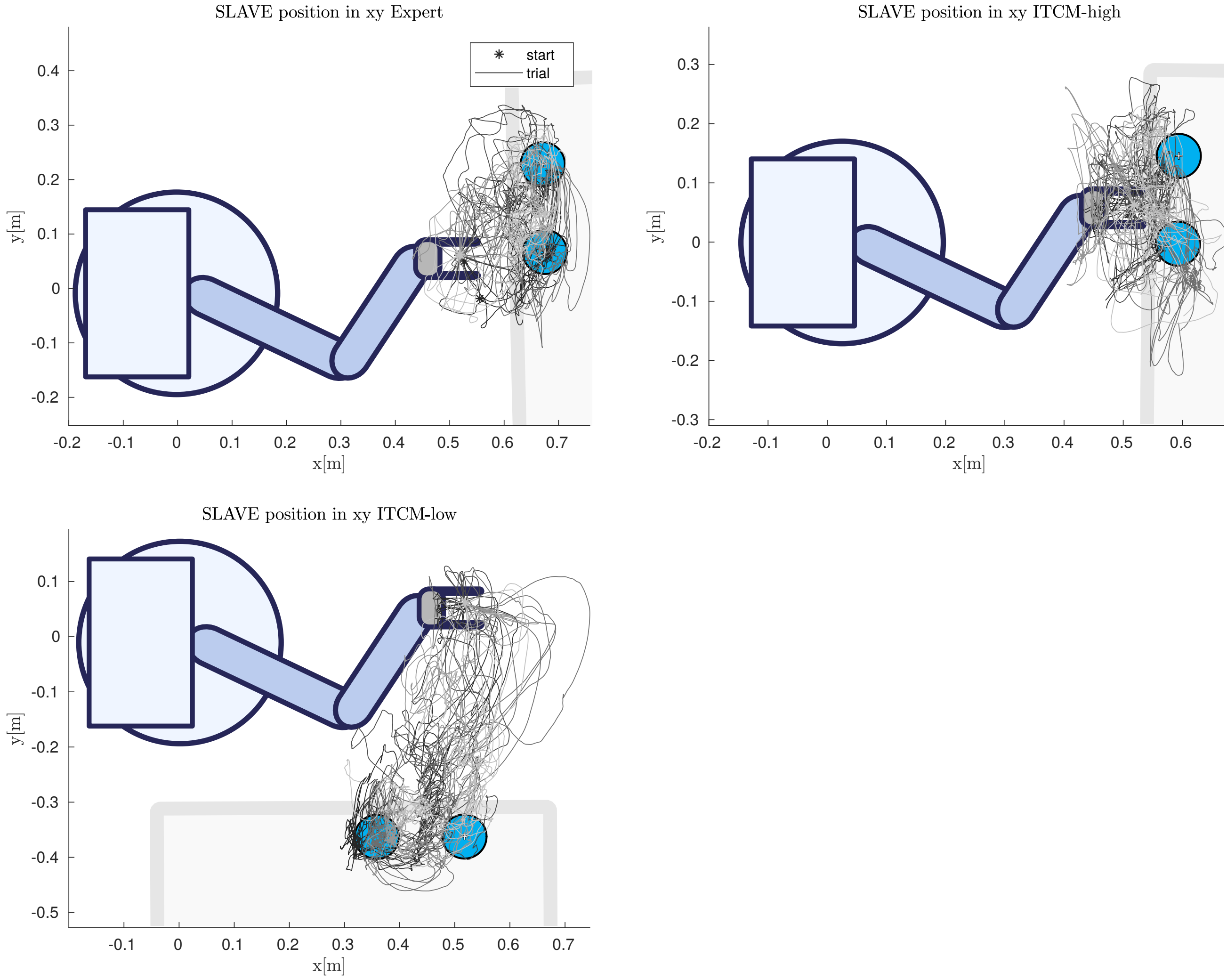


Figure 51: Position of the `gripper_grasp_link` frame relative to the `base_footprint` for all repetitions part of the experiment in the three different conditions. Each shade of grey are seven repetitions of one participant. A top view of the Marco robot and the task in the different positions is added for context. The start of each repetition is denoted with a '*'.

and rotational) and the participants were mostly inexperienced, participants needed to learn how to do the task. The mean of all participants did not show a clear learning effect during the experiment as can be seen in Figure 53 (p. 46). Although some effect still can be seen (-3 s on average per repetition) for the ITCM-low condition, this is not bad for the experiment. Because it is mostly relevant how good somebody can do the task instantly from the position selected by a certain method and not how good people can do the task if they are really trained to do it from a certain position. The mean is not plotted for the training because the amount of samples is not equal per trial/repetition. The learning curve is fitted as a power law function ($y = a \cdot x^b + c$), with the following result:

Condition	a	b	c
Familiarisation	4779.81	-0.00291	-4694.52
Expert	43.80	-0.0824	$-21.5 \cdot 10^{-6}$
ITCM-high	37.79	$-2.4 \cdot 10^{-11}$	-0.0437
ITCM-low	7323.21	-0.000978	-7249.85

These lines are also shown in Figure 53 (p. 46).

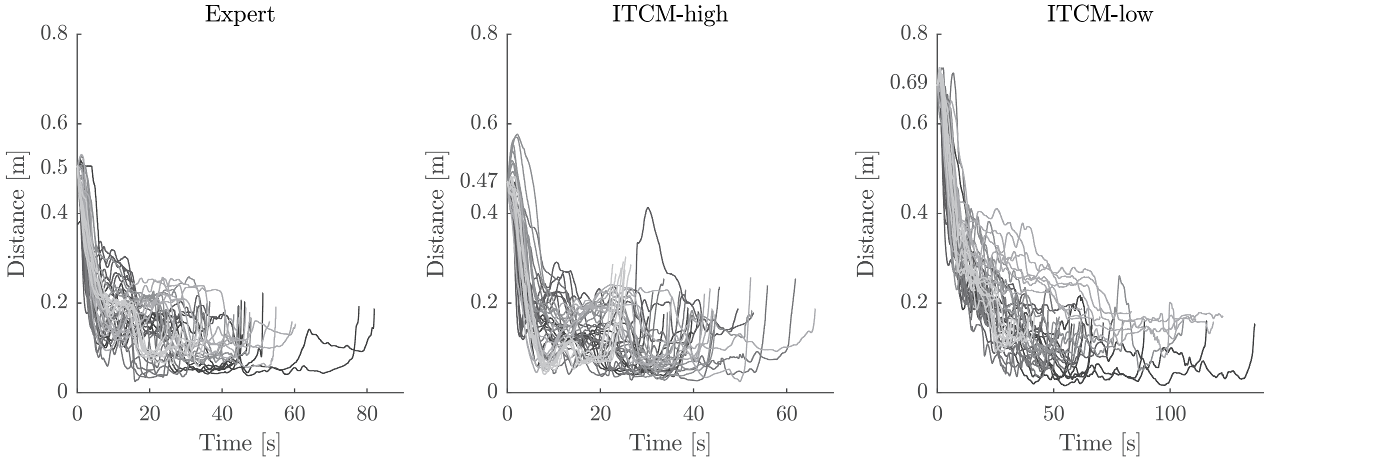


Figure 52: The distance between the `gripper_grasp_link` and the place marker of the task. Note the larger initial distance for the ITCM-low condition.

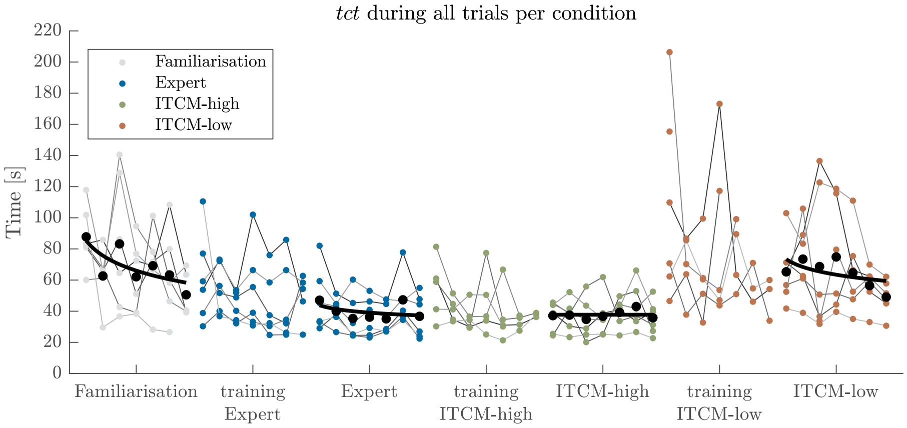


Figure 53: The thick black line is the mean *tct* per repetition. The repetitions of one participant are connected with a thin line a shade of grey. The colours of the dots correspond with the condition.

3. Definition and computation of metrics

The self reported difficulty score is not really a metric, it is just a away of asking people the same question and comparing the answers.

The weighted workload (*wwl*), is defined by [3] as the NASA-TLX. Scores are rated (r) on a template with 20 options for each of the six dimensions (m). The weight per dimension ($w(m)$) is used to multiply the rated value and the total sum is divided by 15 to result in a value between zero and hundred. The score is calculated as

$$wwl = \frac{\sum_{m=1}^{6} r(m) * 5 * w(m)}{15} \tag{27}$$

The cumulative steering angle (*csa*)is defined in [4] as the total amount of rotation that the operator exerted on the master device. It is computed by summing all differences in orientation in consecutive time steps (s).

$$rpy = [roll\ pitch\ yaw]^T \tag{28}$$

$$csa = \sum_{k=2}^{s} ((roll_k - roll_{k-1}) + (pitch_k - pitch_{k-1}) + (yaw_k - yaw_{k-1})) \tag{29}$$

The cumulative steering translation (*cst*) is defined as the total amount of translational movement that has been made by the input-point of the master device. It is computed by summing all differences in position in consecutive time steps (s).

$$cst = \sum_{k=2}^{s} \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2 + (z_k - z_{k-1})^2} \tag{30}$$

The spectral arc length (*sal*) is defined by Balasubramanian et al. [5] as the length of the curve that describes the (amplitude and frequency normalized) Fourier magnitude spectrum up to a certain frequency. Because *sal* is a metric for the smoothness of a speed profile, the direction of the speed is not used, only the absolute speed over time. The computation is described in [5].

4. Results metrics

In addition to the results presented in the paper, in this section graphs of metrics are shown which did not cary the main story line presented in the paper. Also, the steering effort by task time and the amount of gripper actions per repetition are plotted. Per participant the mean of seven succesful repetitions is used for the statistical analysis. The distribution of these means does not suggest that they are not normally distributed. This has been tested with a Kolmogorov-Smirnov test see Table 5.

The top two graphs of Figure 54 show a significant increase in steering effort, as was reported in the paper. As was also suggested in the paper, the amount of increase in travelled path is in the same ballpark as the increase in task completion time (*tct*). Calculating the amount of rotation and translation that has been carried out in the total task and divide it by the time of that instance of the task, gives us the average rotational and translational velocity as depicted in the two bottom graphs of Figure 54. In Table 4 the mean and standard deviation of the means per participant are shown and the result of the one way repeated measures ANOVA on those means per participant. Together with the spectral arc length, see Figure 55, this suggests that participants did not alter their steering effort per time, but merely took longer in the ITCM-low condition and therefore tasks from the base pose belonging to that condition took more effort to complete.

The amount of gripper actions per repetition (Figure 58) does not show something interesting in the difference between conditions.

Participants were instructed to correct the position of the cup if it was positioned not entirely within the circle of the place marker. To investigate whether these corrections had influence on the difference found between distributions of the condition's *tct*, the task was segmented. The first segment is approach, this is from coupling the master and slave until the gripper is first closed (the gripper starts open). The second segment is transfer, this is from the end of approach until the gripper is first opened. The last segment is retreat, and starts at the end of transfer and ends when the gripper is more than 30 cm away from the place task.

The approach segment is thus entirely correction free. In the transfer segment, there could be corrections in grasping the cup and in the retreat segment there could be corrections in placing the cup. Similar to the *tct*, the time for the approach and transfer is significantly longer in the ITCM-low condition compared to the ITCM condition and no significant difference has been found between the ITCM and Expert condition. It is notable that for the transfer and retreat segment, there are a couple of very fast and a couple of very slow repetitions. This is reflected in the larger confidence interval of time for each segment in the ITCM-low condition than the ITCM and Expert condition. The performance in the ITCM-low condition thus has a lot more variability per participant than the ITCM and Expert conditions. It has been tested that these mean times are normally distributed with a Kolmogorov-Smirnov test see Table 6.

In the retreat segment it can be seen that there is a large part of the repetitions between the 0 and 10 s. That the other repetitions took longer is probably because corrections needed to be done. In those corrections precise movement is a lot more important than in the normal retreat part. In the normal retreat part only when te gripper is near the cup, the movement can still influence the outcome. This thus suggests that the *tct* difference in ITCM and ITCM-low condition is due to the precision of the task. In Table 7 the mean and standard deviation of the time per segement is given per condition, the results of the anova are shown and the post hoc test.

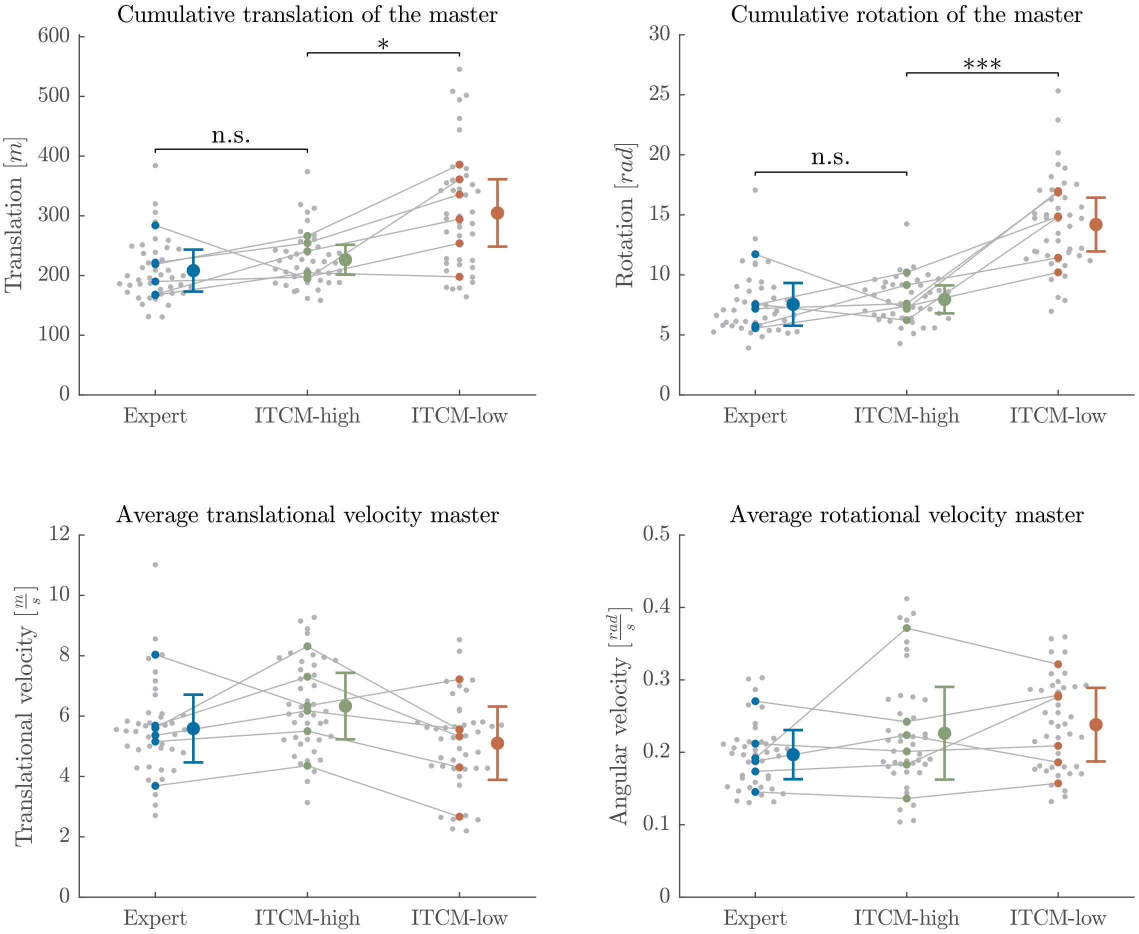


Figure 54: Traveled path effort metrics and the same metric devided by the *tct* of its repetition. The large dot with the whiskers is the mean of the participants means with the 95 % confidence interval. The coloured dots represent the mean per participant and the grey lines link means of one participant and each grey dot represents a repetition. The horizontal bars denote a post-hoc tests, where 'n.s.' stands for not significant, '*' for $p < 0.05$, '**' for $p < 0.01$ and '***' for $p < 0.001$

Table 4: Descriptive and inferential statistics effort per time

Metric	Descriptive				ANOVA	Post-hoc	
	Expert	ITCM-high	ITCM-low			ITCM-high – ITCM-low	Expert – ITCM-high
$\frac{csa}{tct}$	0.20(0.04)	0.23(0.08)	0.24(0.06)	$[\frac{\text{rad}}{\text{s}}]$	$F(15) = 0.6684$, $p = .527$	$p = .39$ $r = .09$	$p = .22$ $r = .24$
$\frac{cst}{tct}$	5.59(1.40)	6.33(1.38)	5.10(1.52)	$[\frac{\text{m}}{\text{s}}]$	$F(15) = 1.117$, $p = .353$	$p = .91$ $r = .42$	$p = .19$ $r = .28$

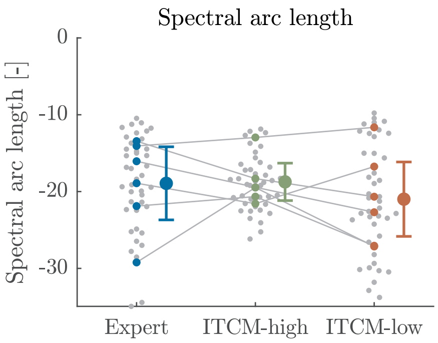


Figure 55: Spectral arc length

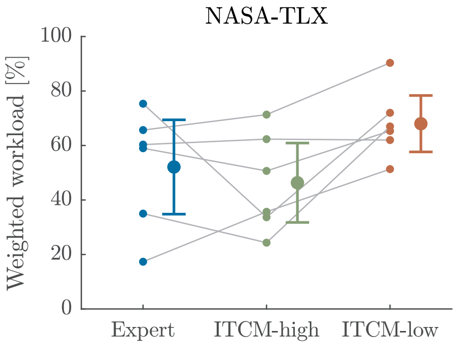


Figure 56: Weighted workload per participant for each condition, as resulted from the NASA-TLX.

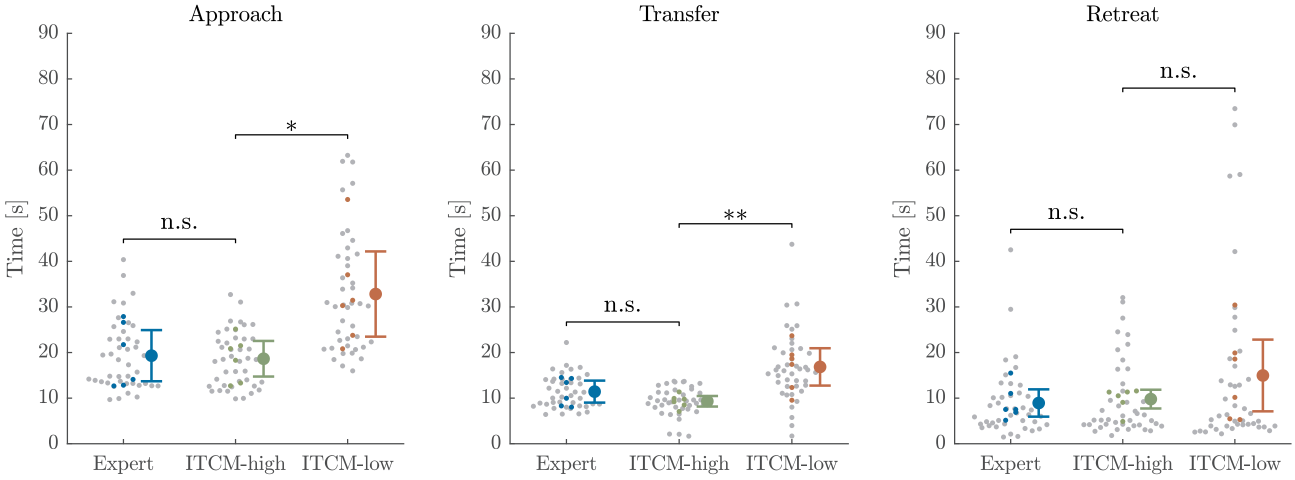


Figure 57: Task completion time divided over segments. A one tailed t-test showed difference between ITCM and ITCM-low condition for the approach and retreat segment. In the retreat segment such a difference is not significant. How to interpret the graph see Figure 54.

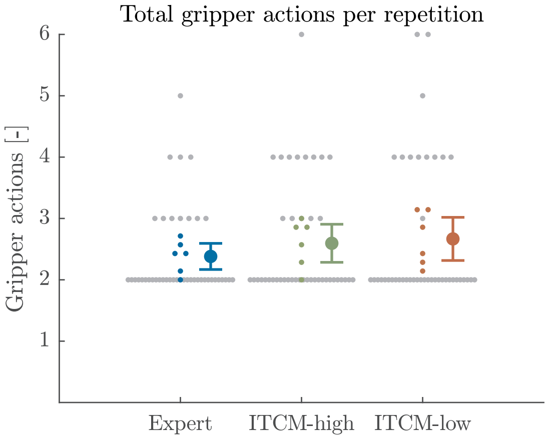


Figure 58: Total gripper actions per repetition for each condition.

Table 5: Probability that the means per participant are normally distributed for all metrics and conditions, as tested with the Kolmogorov-Smirnovtest.

Metric	Expert	ITCM-high	ITCM-low
tct	.32	.51	.47
wwl	.59	.88	.90
csa	.48	.99	.41
cst	.79	.99	.44
sal	.52	.69	.87

Table 6: Probability that the mean time per participant are normally distributed per segment as tested with the Kolmogorov-Smirnovtest.

Metric	Expert	ITCM-high	ITCM-low
t_{approach}	.61	.96	.94
t_{transfer}	.91	.60	.24
t_{retreat}	.30	.27	.40

References

[1] D. Thomas, T. Field, J. Leibs, and J. Bowman, "Rosbag-ros wiki," *URL: http://wiki. ros. org/rosbag*, 2014.

[2] J. McAuley, J. Rothwell, and C. Marsden, "Frequency peaks of tremor, muscle vibration and electromyographic activity at 10 hz, 20 hz and 40 hz during human finger muscle contraction may reflect rhythmicities of central neural firing," *Experimental Brain Research*, vol. 114, no. 3, pp. 525–541, 1997.

[3] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*, vol. 52, pp. 139–183, Elsevier, 1988.

[4] M. Van Osch, D. Bera, K. Van Hee, Y. Koks, and H. Zeegers, "Tele-operated service robots: ROSE," *Automation in Construction*, vol. 39, pp. 152–160, 2014.

[5] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet, "A robust and sensitive metric for quantifying movement smoothness," *IEEE transactions on biomedical engineering*, vol. 59, no. 8, pp. 2126–2136, 2012.

Table 7: Descriptive and inferential statistics *tct* segment. Between ITCM and ITCM-low it is tested whether ITCM-low has significantly higher *tct*, between ITCM and Expert it is tested whether ITCM is higher than Expert.

Metric	Descriptive				ANOVA	Post-hoc	
	Expert	ITCM-high	ITCM-low			ITCM-high – ITCM-low	Expert – ITCM-high
t_{approach}	19.31(7.01)	18.64(4.88)	32.83(11.68)	[s]	$F(15) = 5.509$, $p < .05$	$p < .01$ $r = .66$	$p = .57$ $r = .061$
t_{transfer}	11.43(3.01)	9.31(1.45)	16.85 (5.12)	[s]	$F(15) = 7.271$, $p < .01$	$p < .01$ $r = .74$	$p = .92$ $r = .44$
t_{retreat}	8.93(3.74)	9.79(2.57)	14.97 (9.83)	[s]	$F(15) = 1.64$, $p = .227$	$p = .12$ $r = .37$	$p = .33$ $r = .14$

G. Pilot study

Prior to the experiment, a pilot study was conducted to verify if the procedure was clear, generated usable data and that participants would be able to learn how to handle the telemanipulation setup. The goals of this pilot study are thus to get an indication of the amount of participants needed and to find how the experiment can be improved.

1. Hypotheses

The hypotheses of the pilot study are the same as for the final experiment.

1. Telemanipulation performance and effort will be worse from the ITCM-low position than the ITCM-high position.
2. Telemanipulation performance and effort will be better or equal from the ITCM-high position than the Expert position.

2. Conditions

- Expert
- ITCM-high
- ITCM-low

3. Metrics

The performance of a robot in the care-home environment can be evaluated by the time the operator takes to do the tasks with the robot and the amount of failures needed to reach seven successful trials for a condition. To measure the effort of the operator both the translational and rotational steering input can be used. Besides these metrics, the personal experience of the operator's workload has been measured with a NASA-TLX (task load index) questionnaire and the participants were asked to rate the difficulty of the task without a scale.

- Performance

	abrev.	unit
Number of failed attempts	num$_{fail}$	–
Task completion time	*tct*	s

- Effort

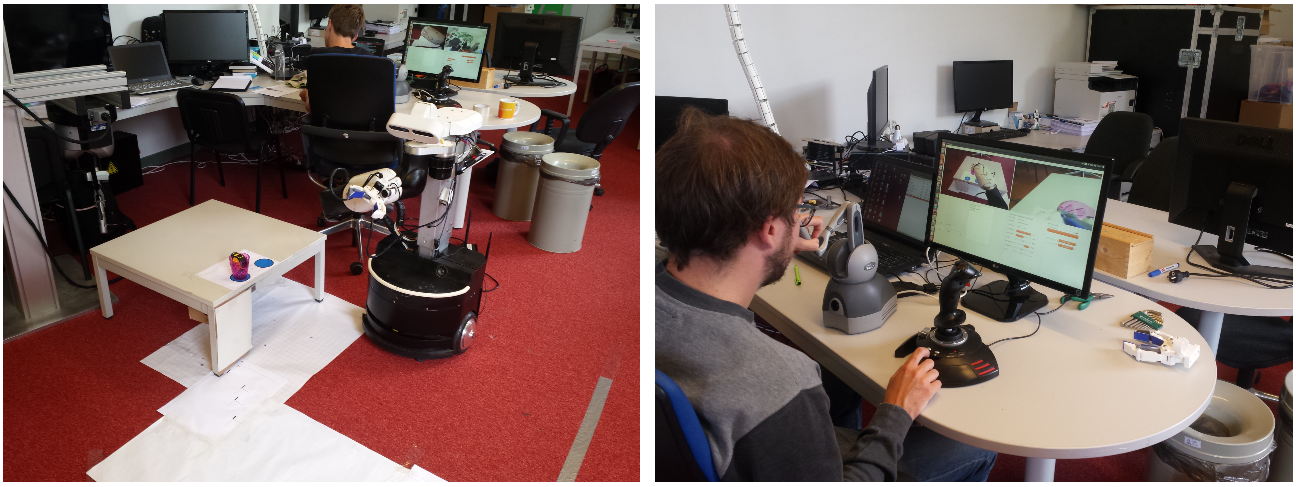
	abrev.	unit
Cumulative steering angle	*csa*	[rad]
Cumulative steering translation	*cst*	[mm]
Weighted workload	*wwl*	[-]
Difficulty	*score*	[-]

4. Experimental protocol

Two left handed participants and myself did the experiment. Data of the first participant was not saved correctly, so my data was combined with the NASA-TLX and difficulty scores of the first participant. Moreover for the second participant, the robot was positioned incorrectly with respect to the table. This caused the data gathered through the pilot study to be not usable for the study.

The participants were asked to do the task as fast as possible while ensuring that: the object ends up within the goal area and no unnecessary contacts are made. If the cup is placed outside of the area of the post-it, the cup should be repositioned. The task fails if:

- a time limit is exceeded of 2 minutes.



(a) Marco with the positions of the table taped to the floor **(b)** Cockpit with left handed participant

Figure 59: Cockpit and setup of the pilot experiment

Table 8: pilot results

Metric	Expert	ITCM-high	ITCM-low	ANOVA
tct [s]	26.91 (15.34)	21.28 (11.52)	46.93 (21.99)	F(3) = 1.28, p = .40
cst [mm]	162.48 (27.41)	178.82 (41.80)	284.13 (42.53)	F(3) = 6.07, p = .088
csa [rad]	6.24 (2.05)	5.82 (2.52)	15.93 (4.29)	F(3) = 6.78, p = .077
sal [-]	-16.28 (5.51)	-13.22 (4.48)	-20.62 (4.28)	F(3) = 1.21, p = .41
wwl [-]	42.17 (4.95)	42.17 (26.63)	66.33 (6.60)	F(3) = 1.50, p = .35

- the cup becomes unreachable from the robot's current base pose (because it is swept of the table or dropped)

The participant first does a familiarization trial session consisting of:

- (1 min) free air movement (get a feel for the response speed)
- (2 min) workspace exploration (take on different end-effector poses)
- practice telemanipulation strategies
  These were told to the participants. (use shadows and approach diagonally for depth information, check if the cup is between the fingers by occlusion or movement)
- (7 successful executions) practice the task

Thereafter he/she fills in the NASA-TLX questionnaire. Then the participant will do in random order all conditions the:

- (7 executions) training phase
- (7 successful executions) experiment

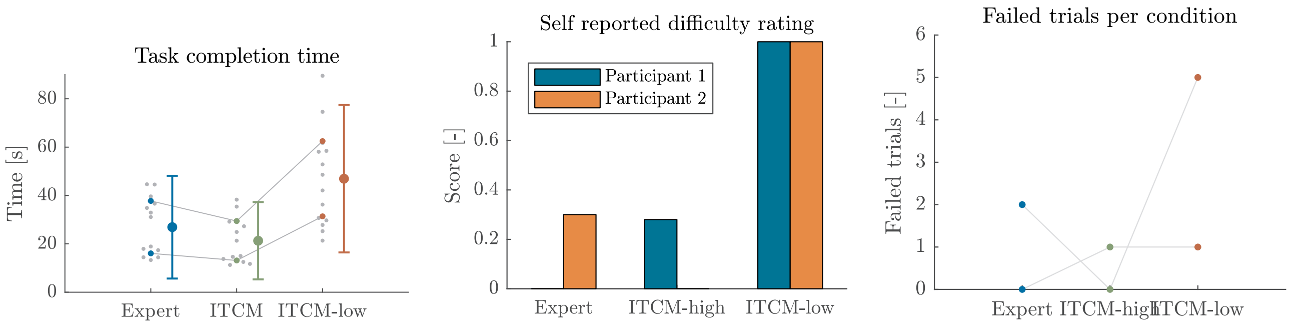
And for each condition the NASA-TLX and the difficulty score were administered.

5. Results

Figure 60 (p. 53), shows the main results of the two participants. Figures of additional metrics can be seen in the figures in Figure 61 (p. 54).

No significant effects are seen between the means of all participants per conditions for all metrics. This has been tested per metric with a one-way repeated measures ANOVA, the results are shown in Table 8 (p. 52).

Data from this pilot study indicates that a sample size of three participants would result in significant differences between the ITCM-high and ITCM-low condition results.



(a) Task completion time

(b) Difficulty score

(c) Fails

Figure 60: Main results of the pilot experiment. The Large dot with the whiskers is the mean of the participants means with the 95 % confidence interval. The coloured dots represent the means per participant and the grey lines link means of one participant. Each grey dot represents the result of a repetition, sorted per condition.

6. Consequences for final experiment

Changes to the procedure:

- Less participants
  Originally it was conceived to have twelve participants (more than ten and dividable by the amount of permutations of three conditions). But because the experiment takes a long time per participant (2 - 3 hours), this is not feasible.
- Task instruction
  The task instruction will be limited to constrain only the final position of the cup, because spurious contacts are not measured and do not lead to longer execution times.
- Fails not as metric
  During the pilot it became apparent that whether a trial resulted in a fail was often coincidental. Most of the times the participant would judge the tables edge to be closer, releasing the cup just before the tables edge. Or the cup was pushed outside the arms reach while attempting to grasp it. Moreover due to the incremental nature of the metric, it is chosen to leave it out of the final experiment.
- Time limit
  No time limit will be used to cut off longer trials. This would otherwise bias the *tct* results.
- Save guard data
  Data was lost while moving from cockpit laptop to usb-drive. To circumvent this, copy to usb-drive and copy to another usb-drive as well.
- Force-feedback toggle
  Toggling the forcefeedback by the participant was confusing for participant (to many buttons), not necessary and caused the researcher to easily lose track of the force-feedback state.

Changes to the familiarisation and instruction:

- Task strategy
  It will be mandatory to use the prescribed task strategy. This is to reduce the risk of participants adopting a new strategy during the experiment increasing the within subject variance not caused by the conditions.
- Giving an example
  The researcher will show how the task strategy is employed for the next condition by doing the task himself and reciting the different task strategy stages.
- Removing free air movement
  During familiarisation stage, the participant will not do a free-air movement because it was not useful for the participant to get familiar with the telemanipulation setup.

Changes to the setup:

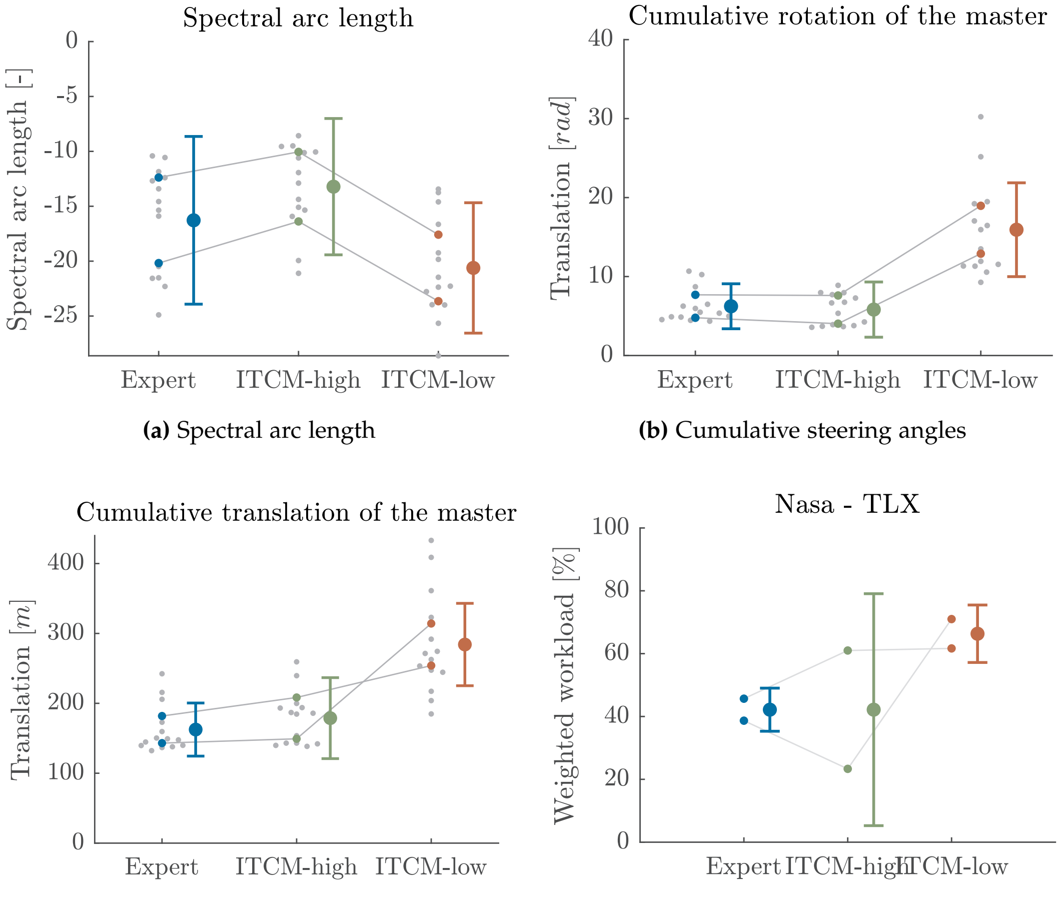
- Overhead lighting
  This will enable the participants to better see the distance between the gripper and the table by using shadows on the table.
- Rotate 180°

Because the researcher is than able to sit near the participant and reset the position of the cup.

- Filter force feedback
  The force feedback in the direction the gripper is pointing (y axes gripper, x axis sensor) had increased noise during the pilot experiment. To prevent this during the experiment, the cut-off frequency of the filter has been lowered from 100 Hz to 10 Hz.

7. Conclusions

The pilot experiment brought to light a lot of little improvements that can be made to the final experiment. It has also come to light that the experiment takes a lot of time per participant. To keep the time needed to do experiments manageable, not a lot of participants can be used. Fortunately the amount of participants needed to get a statistical significant difference, with a large effect, between the ITCM-high and ITCM-low condition would be around four. To get all possible order permutations of the conditions, the amount of participant needs to be six. Thus it is decided to use six participants in the final experiment.



(a) Spectral arc length

(b) Cumulative steering angles

(c) Cumulative steering translations

(d) NASA-TLX

Figure 61: Additional results. See Figure 60 (p. 53) for general representation of metrics.

H. Remarks participants

The following remarks are the results from the open questions "What was hard?", "What was holding you back from being better at the task" and spontaneous utterances of the participants. Remarks of which at least one similar remarks have been said are aggregated in the Table 9.

Table 9: Overview of participants' responses on the question what made the task hard. Only factors that have been mentioned more than once have been included in this table.

Factors that made the task hard	Expert	ITCM	ITCM-low
Transform between camera view and telemanipulation input	0	0	4
Cup easily pushed outside workspace	2	0	0
Easier	0	2	0
Grasping	1	0	1
Limited depth information (arm in front of shadow)	1	0	1
Retreat	1	0	1

1. Participant 1

1) Expert positioned

- Retreat was hard, grasping as well because it was far away from the robot.
- Thumb was acing (it was the last condition)

2) ITCM-high positioned

- Transferring the cup along the table was hard.
- A lot easier than ITCM-low.

3) ITCM-low positioned

- Retreat is the hardest part, it is really hard to grasp the object.
- It seems easy, but it is hard to tell why it is difficult

2. Participant 2

1) General remarks

- The participant found it hard to go straight with the gripper.
- Grasping the object is hard, the cup is more easily pushed than grasped.
- It is hard to judge depth of with the 2D screen.

2) Expert positioned

- Because the goal is far away, the arm needs to be stretched to reach the goal. This causes the shadow of the gripper on the table to be behind the arm. Further limiting the depth information.
- Accidentally pushing the cup instead of grasping, quickly causes the cup to be unreachable because the table is quite far away.

3) ITCM-high positioned

- The arm obstructs the view less compared to the expert condition.

4) ITCM-low positioned

- Moving the gripper is hard because the view from the overhead camera is in a different frame than the telemanipulation input.
- It is hard to find a place in the workspace of the robot where the gripper can be oriented in a way that a grasp of the cup can be attempted.
- At the start of the movement the gripper is not visible in the overhead view. This part should thus be done without visual feedback but from memory.

3. Participant 3

Participant had about 10 hours of telemanipulation experience, of which none with this setup.

1) General remarks

- The control loop and camera images of the robot are sometimes delayed due to instability of the WiFi signal, therefore sometimes feedback should be anticipated instead of received.
- Touching the table feels not like a hard object where you cannot push through but more like a jumpy pudding, this feels unnatural.
- The button to toggle the gripper state (closed/open) malfunctioned sometimes.

2) Expert positioned

- Grasping and placing the cup is very near the edge of the workspace of the master. So if you accidently push the cup instead of grasp it in the first go, the cup is becomes unreachable.

3) ITCM-low positioned

- Because the camera is looks all the way to the right but the telemanipulation is still with respect to the `base_footprint` frame of the robot, the control is unintuitive.

4. Participant 4

1) General remarks

- Moving the gripper slowly is a lot harder than keeping it steady when moving.
- In some orientations, the gripper suddenly takes on a different orientation. [7]
- The master device, a Geomagic Touch Omni was not found to be ergonomic. This manifested itself in confusing the couple toggle button and the gripper toggle button.
- The gripper toggle button was not always responsive.

2) ITCM-low positioned

- The participants reported that his hand got cramped because of the needed orientation of the pen.
- Transforming between the view of the head camera to the frame of the telemanipulation input was taxing mentally.
- It is hard to get depth information out of the image because the arm is obstructing its shade on the table.

5. Participant 5

1) General remarks

- Holding the pen is physically taxing and uncomfortable; the hand is to high.
- Pressing the correct button on the pen was unintuitive.

[7] This is because the whole-body kinematic controller of the robot control the wrist of the arm, and not the gripper. Therefore, the robot thinks it does a small correction on the orientation of the wrist. But this actually causes a larger displacement of the gripper.

6. Participant 6

Participant had reported to have around 10 weeks of telemanipulation experience, of which more than half with this setup.

1) General remarks

- Time pressure was holding him back from being better at the task.

2) Expert positioned

- An offset in the neutral bias of the force-torque sensor made holding the pen cost physical effort.

3) ITCM-high positioned

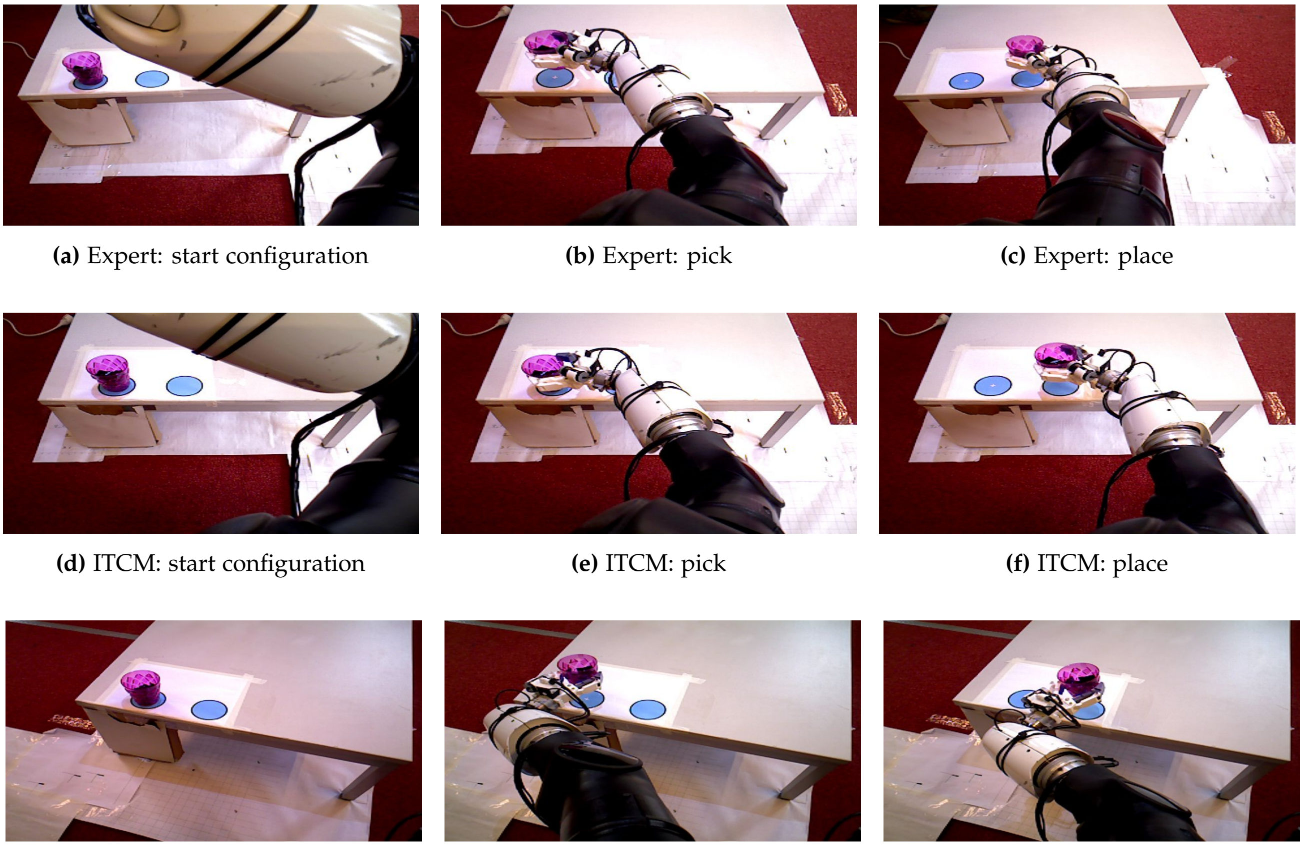
- This position was not hard, it was easy.

4) ITCM-low positioned

- The orientation of the table with respect to the robot made the task hard.
- Also the orientation of the robot with respect to the task view.
- However the view is clearer than the expert positioned

I. Head camera views

The view the participants had from the cockpit is depicted per condition and for three different moments depicted in Figure 62. The start moment, when the EE is in the start position before the participant couples the master and slave. The pick moment, when the cup is picked from the table. The place moment, when the cup is placed on the table. These images are cropped out of screen shots of the cockpit interface. The resolution and compression is thus equal to what the operators could use.



(a) Expert: start configuration

(b) Expert: pick

(c) Expert: place

(d) ITCM: start configuration

(e) ITCM: pick

(f) ITCM: place

(g) ITCM-low: start configuration

(h) ITCM-low: pick

(i) ITCM-low: place

Figure 62: Different view points and arm configuration per condition (Expert, ITCM, ITCM-low) for the start of the task, during pick and during placement of the cup. Shadows can be seen to help judging the height difference between the gripper and the table. Also note that the gripper cannot be seen in the start position from the ITCM-low condition.

J. Participant Information Sheet

Participant information

Researcher:	Karel Sirks
Supervisors:	ir. J.G.W Wildenbeest and dr.ir. D.A. Abbink
Title:	Comparing autonomous and manual navigational goal selection
Date:	11th September 2018

Purpose

The purpose of this study is to make autonomous navigation more useful for semi-autonomous mobile robots. This could be applied to homecare-robots in domestic environments where they are able to navigate autonomously but manipulation would need to be controlled manually from a distance due to the higher variety in tasks.
The result of this research will help these robots to select a base position from where an operator can successfully execute a task using telemanipulation.

Task

Your task is to pick up the purple cup and place it within the designated area.
This should be done **as fast as possible**, while ensuring that the object ends **up within the goal area**.
Time is measured from the beginning of manipulation until the robot's gripper is outside of a sphere of 30cm near the placed cup.

The researcher will say that the cup is not placed precise enough <u>when the cup is no longer between the gripper's fingers</u>. No comment will be made if the cup is placed correctly. Misplaced cups should be corrected until the cup is in the right place.
A trial is marked failed if the cup has become unreachable (e.g. because it has fallen or was pushed of the table).

As participant you are asked to execute the pick-and-place task 7 times per condition for training and 7 times successfully per condition for the experiment. That means that failed executions do not count for the 7 times limit during the experiment. Between the training and each condition, there is time to relax and stretch. After each condition, the questionnaire will be administered, and a score of the difficulty will be asked. See Fig. 1 for the experimental overview.

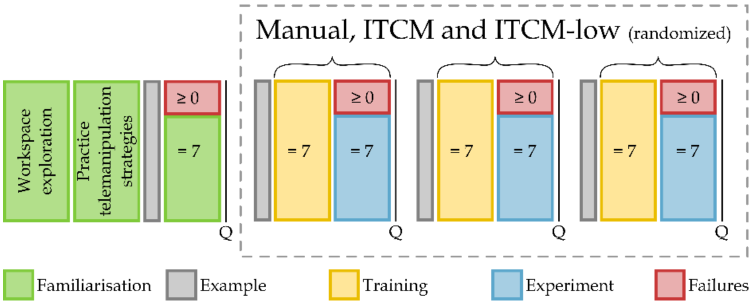


Fig. 2 Experimental protocol overview

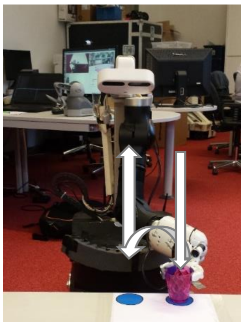


Fig. 1 Marco ready for the experiment.

No scale will be used for the difficulty rating, so give the first condition any number that feels good. Thereafter assign numbers for the other conditions in such a way that they match your subjective feeling about the difficulty. No zero or negative numbers may be used, otherwise there is no limit to the numbers you may use. Aim to let each number represent the difficulty you experienced.

The entire experiment will take approximately two and a half hours.

First a familiarization phase is done to get you familiar with the telemanipulation setup. This consist of four subphases:

- Workspace exploration: explore the limits of the robots and the input devices workspace
  - Contact with table
- Practice telemanipulation strategies: do the pick and place task and focus on these strategies:
  - shadows to estimate height/ distance to table
  - side of gripper to push object
  - finger of gripper to push object
  - a path diagonal to the robot to get depth information
  - occlusion of the fingers to infer that the object is within the gripper
  - movement of the object to infer that the object is within the gripper

For performing the tasks the same strategy must be used by all participants.

- Couple the input-device to the robot arm (dark grey button)
- Lower the gripper to the height of the cup (such that cup is visible on gripper cam)
- Orient the gripper 45° to the table if possible
- Start approach of the cup
- Check if gripper encloses the cup (occlusion fingers, how does the cup move if touched)
- Grasp it with the finger tips (to make the retreat easier)
- Transfer cup (do not go to far up, finding the height is difficult)
- Release cup (light grey button)
- Retreat from cup
- If the cup is not within the black circle push it more towards the centre. Only re-grasp if it is not possible to push the cup more towards the centre.
- Return the pen to its starting pose
- Uncouple (dark grey button)

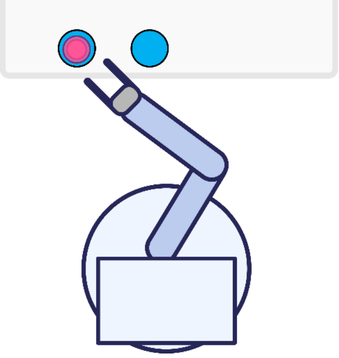
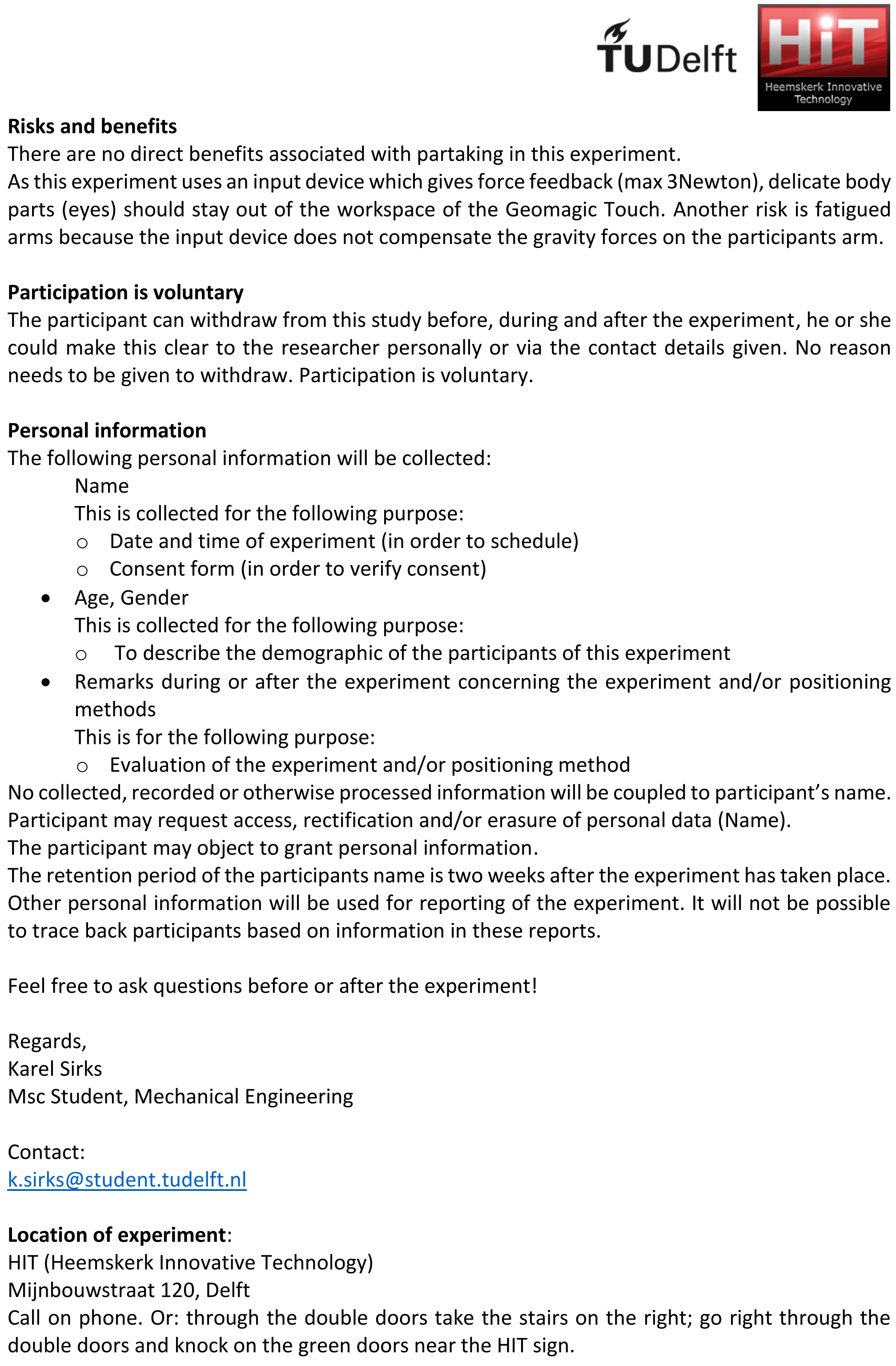


Fig. 3 45° approach to cup

Important:

- Hold the pen firmly
- Couple the pen and the robot arm if the position and orientation correspond
- If gripper does not react. Just press the button another time.



Risks and benefits

There are no direct benefits associated with partaking in this experiment.
As this experiment uses an input device which gives force feedback (max 3Newton), delicate body parts (eyes) should stay out of the workspace of the Geomagic Touch. Another risk is fatigued arms because the input device does not compensate the gravity forces on the participants arm.

Participation is voluntary

The participant can withdraw from this study before, during and after the experiment, he or she could make this clear to the researcher personally or via the contact details given. No reason needs to be given to withdraw. Participation is voluntary.

Personal information

The following personal information will be collected:

Name
This is collected for the following purpose:
  - Date and time of experiment (in order to schedule)
  - Consent form (in order to verify consent)

- Age, Gender
  This is collected for the following purpose:
  - To describe the demographic of the participants of this experiment
- Remarks during or after the experiment concerning the experiment and/or positioning methods
  This is for the following purpose:
  - Evaluation of the experiment and/or positioning method

No collected, recorded or otherwise processed information will be coupled to participant's name.
Participant may request access, rectification and/or erasure of personal data (Name).
The participant may object to grant personal information.
The retention period of the participants name is two weeks after the experiment has taken place.
Other personal information will be used for reporting of the experiment. It will not be possible to trace back participants based on information in these reports.

Feel free to ask questions before or after the experiment!

Regards,
Karel Sirks
Msc Student, Mechanical Engineering

Contact:
k.sirks@student.tudelft.nl

Location of experiment:
HIT (Heemskerk Innovative Technology)
Mijnbouwstraat 120, Delft
Call on phone. Or: through the double doors take the stairs on the right; go right through the double doors and knock on the green doors near the HIT sign.

TU Delft