

Master's thesis

Robustness of Haptic Shared Control Against Model Inaccuracies During Telemanipulation

J. van Oosterhout

Delft, 19th of September, 2012

Report Number: 1254
Student Number: 1268597

Delft University of Technology
Faculty of 3mE
Department of BioMechanical Engineering

Robustness of Haptic Shared Control Against Model Inaccuracies During Telemanipulation

Jeroen van Oosterhout

Committee:

Prof. Dr. F.C.T. van der Helm	Department of BioMechanical Engineering, 3mE
Dr. Ir. D.A. Abbink	Department of BioMechanical Engineering, 3mE
Dr. ir. Herman Damveld	Department of Control & Operations, Aerospace Engineering
Dr. Ir. C.J.M. Heemskerk	Heemskerk Innovative Technology B.V.
Ir. J.G.W. Wildenbeest	Department of BioMechanical Engineering, 3mE
H. Boessenkool Msc	Department of BioMechanical Engineering, 3mE

Coaches:

Dr. Ir. D.A. Abbink
Dr. Ir. C.J.M. Heemskerk
Ir. J.G.W. Wildenbeest
H. Boessenkool Msc

Preface

During my master's study I found many interesting challenges that ranged from optimizing machine to identifying human performance. Haptic shared control in teleoperation combines these aspects. It joins advantages and strengths of manual control and automation. Yet, this relatively young and multidisciplinary field yields many challenges, focused around the understanding of human, machine, and, moreover, their synergy.

This master's thesis aims to answer the question: "How guiding errors affect task performance and control effort during teleoperation with haptic shared control."

There were various challenges involved to answer the main question. One part was to find and prepare a suitable test-setup. Also the extension of haptic shared control to the third dimension was not trivial. Then there was a human factor experiment involved, which included the testing of subjects. Finally the measured data had to be processed, analysed and reported in this thesis.

The focus of this study is a human factors experiment, which is described in detail in a research paper. The appendices provide background information and give a broad overview of the accomplished work. These appendices allow future students and researchers to gain detailed insight in the challenges faced during this study.

The experimental setup is described in Appendix A. Appendix B shows the work done on modelling and implementation of haptic shared control. The shared control experiment is described in Appendix C, including an extensive overview of the results.

A USB-disk, containing all raw measurement data, all code, all pictures, all literature, etc., that might be of interest has been submitted to the Biomechanical Engineering depository, all of which available on request.

I want to thank all people who contributed to this study. Special thanks to my coaches; David, Cock, Jeroen and Henri thanks for all inspiring discussions and your efforts commenting and reviewing my work. Furthermore I thank Marco for the facilities provided to perform my work at the RHSC at FOM DIFFER. Finally I want to thank my girlfriend Daniëlle; thanks for your support especially during the last straws.

Robustness of Haptic Shared Control Against Model Inaccuracies During Telemanipulation

Jeroen van Oosterhout¹, David A. Abbink¹, Jeroen G.W. Wildenbeest^{1,3}, Henri Boessenkool^{1,2}, Cock J.M. Heemskerk³ and Frans C.T. van der Helm¹

Abstract—To improve teleoperated task performance and control effort, haptic shared control can assist the human operator along a safe and optimal path with continuous guiding forces from an automated controller. But previous research tested such shared control on tasks that were accurately defined in the controller (giving flawless guiding forces), while in a practical implementation guidance will be flawed due to inaccurate model or sensor information. This research investigated the effect of zero, small (7.5 [mm]) and large (17.5 [mm]) guiding errors on task performance and control effort. Included in the experiment were high and low transparency (quality of reflected forces from the environment) to test if that affects the ability to deal with wrong guiding. In a human factors experiment subjects were provided with high and low transparency and inaccurate haptic shared control. The subjects performed a three dimensional virtual reality peg-in-hole type task (30 [mm] diameter; 0.1 [mm] clearance). The results showed that small guiding errors improved task performance and control effort with respect to unguided operations for both transparency levels. Thus despite small guiding inaccuracies haptic shared control still aides operators during telemanipulation while transparency does not affect that aid.

Index Terms—Teleoperation, Remote handling, Haptics, Haptic share control.

1 INTRODUCTION

Telemanipulators make it possible for humans to perform tasks in environments that normally preclude hands-on manipulation. Examples of such environments are space, deep sea, nuclear, micro assembly or minimal invasive surgery. The telemanipulator transfers movements and forces imposed by the human operator on a robotic master device via a controller to a robotic slave in the remote environment. It also transmits back (contact) forces from the environment on the slave to the master device. Figure 1 shows a schematic representation of a telemanipulator with the transmitted (bilateral) information flow between the human and the remote environment. This is also referred to as the Connected Telemanipulator System [1]. The bilateral information flow allows humans to make use of their

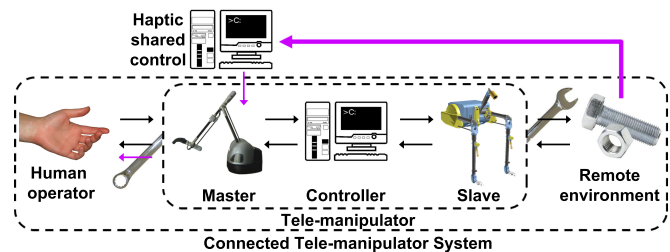


Fig. 1. Elements of the Connected Telemanipulator System combined with haptic shared control.

unique problem solving and manipulative skills in remote environments [2], [3].

The telemanipulator filters and degrades the (force) information that passes through [4] (often referred to as limited transparency) causing decreases performance and increases control effort. One of the techniques to improve the performance and effort is to provide reflected force from the environment [5]. But further improvements in transparency yields only marginal improvements in performance and control effort [6].

Another more promising approach to improve task performance and control effort in teleoperation is haptic shared control: An automated system continuously assisting the operator by controlling the same input device. Practically, haptic shared control keeps the operator in direct control over the task while it provides the operator with continuous support for optimal control actions based on information from the task and environment [7] (see Figure 1). This approach has many implementations (some with a variant definition) in telemanipulation, like guidance to a certain reference position or trajectory (e.g. [8]–[12]), or to shield areas from entering (e.g. [13], [14]).

The research above report positively on the effect of haptic shared control on teleoperated task performance and control effort. Even without transparency, haptic shared control assists the operator to work as fast as in manual operation while control effort improves [12].

¹ Delft Haptics lab, Department of BioMechanical Engineering, Delft University of Technology, Mekelweg 2 2628 CD Delft, The Netherlands

² Remote Handling Study Centre FOM institute DIFFER, PO Box 1207, 3430 BE Nieuwegein, the Netherlands

³ Heemskerk Innovative Technology B.V., Jonckerweg 12 2201 DZ Noordwijk, The Netherlands

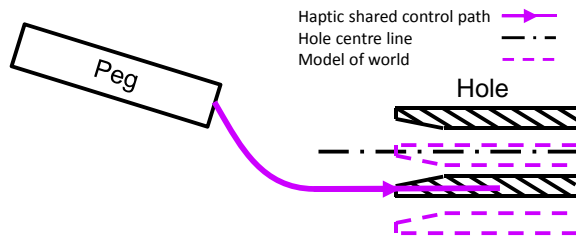


Fig. 2. Schematic representation of the peg-in-hole task, showing guidance based on an inaccurate model (dotted lines) of the environment (solid lines).

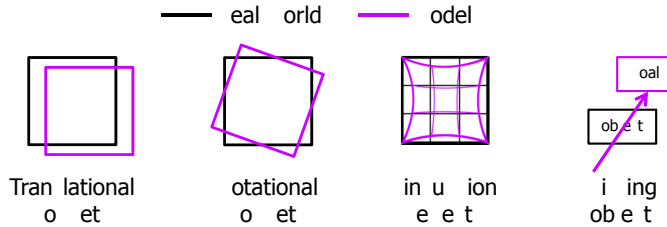


Fig. 3. Four possible mismatches between the real world and a virtual model on which haptic shared control is based. The current research considers the translational error.

However, all these studies assumed that the autonomous controller has perfect knowledge of the environment, meaning that the guidance is flawless, in which case full automation would actually be a far better solution.

In a practical implementation of haptic shared control, the controller will need some sort of (partly known) task model to guide the operator. In the process of making such a model random distortion, with respect to the real world (e.g. by unexpected objects, object deformations or sensory inaccuracies), will find their way in. When the model is wrong, the intelligent automated guiding system takes over the error and guides the operator incorrectly as shown in Figure 2.

Offsets can roughly be categorized into four main effects they have on the internal controller model: translational offsets; rotational offsets; pincushion effects; and missing object information (see Figure 3). To get a first indication of the effect of errors this research examines only translational offsets, but note that a rotational offset and the pincushion effect result in local translational errors.

The effect of small guiding errors on teleoperated task performance is not yet investigated. However, literature shows an example where small guidance errors -unintentionally introduced during calibration- were notable to the operators while unlatching a door, especially in fine movements [15]. Further literature shows that very large intentionally introduced guidance errors in car driving during obstacle avoidance reduce performance to manual control [16].

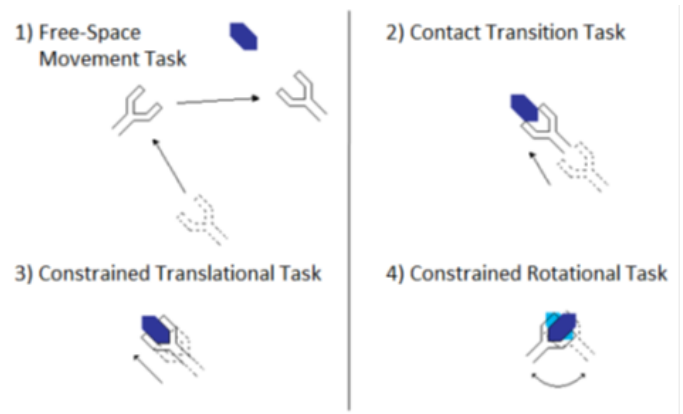


Fig. 4. Adopted from [6]. Four fundamental motion types, for hard-hard contact environments, illustrated by a bolt-and-spanner task.

The main objective of this research is to identify how guiding error affect teleoperated task performance and control effort. To meet this objective for teleoperated tasks in general, the experiment should cover a wide variety of maintenance task. For hard-to-hard contact environments Wildenbeest [6] identified four fundamental motion types that span a wide task space (see Figure 4):

- Free Space Movement. The slave device has no interaction with the environment.
- Contact Transition. The slave device moves in a stage in between Free Space Movement and environmental interaction.
- Constrained Translational Movement. One or more degrees of freedom of the slave device are constrained (e.g. coaxial sliding of pipes).
- Constrained Rotational Movement. The slave device is constrained to rotate along a precise trajectory around a pivot point (e.g. turning a door handle).

Furthermore it is expected that low transparency makes it more difficult to detect and react on inaccurate guiding.

It is hypothesized that:

- Guidance without errors will improve task performance and control effort compared to unguided operations.
- When small errors are present, task performance and control effort are still improved compared to unguided operations.
- Larger errors in the guidance will decrease task performance and control effort to a similar level as during unguided operations.
- Reduced transparency will aggravate task performance and control effort during inaccurate guiding to at best the unguided operations level.

The above effects are visualized in Figure (5). They will also be observed in all subtasks except for Free Space Movement as a small offset in free space cannot affect performance. The effects will be observed most in the

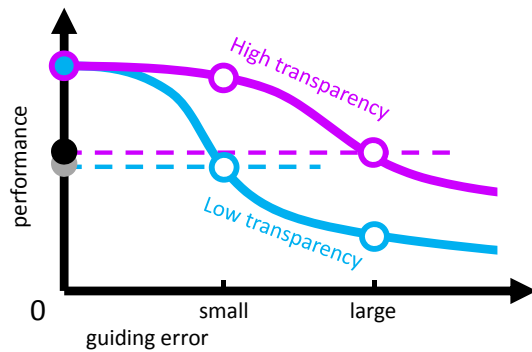


Fig. 5. Visual representation of the hypothesis on performance and control effort. The gray and black dots represent performance for normal (unguided) operation in respectively high and low transparency (as found by [6]). The overlapping purple and blue dots represent performance for flawless assistance in respectively high and low transparency (as found by [12]). The purple and blue lines represent the respective performance drop as a result of a guidance errors.

constrained movement tasks due to jamming.

2 METHODS

2.1 Subjects

For this study 14 participants were tested. All participants are right handed and in the age of 18 to 40 (mean age: 27.07 [years], standard deviation: 5.86 [years]). From this group two subjects are female.

2.2 Experimental set-up

2.2.1 Slave and environment

The slave and environment are combined in a Virtual Reality simulation. The simulation is done with the so called Virtual Slave Simulator [17] where the slave and task are modelled with NVIDIA PhysX rigid bodies. The slave consists of a two finger gripper readily attached to a tool. The rendering of this simulation is done by VR4MAX. The Virtual Slave Simulator updates at 1 [kHz]. Figure 6 shows an impression of the environment and slave.

2.2.2 Task

The experimental task is an the placement of an inner-tube welding tool as shown in Figure 6. The tool has the dimensions of 440x70x125 [mm] (LxWxH), a tip diameter of 30 [mm], a mass of 1.8 [kg]. The tool should be inserted for 140 [mm] in a 30 [mm] diameter tube with a clearance of 0.1 [mm]. The tool is held by the slave gripper at the yellow "gripping feature". The master-slave connection is located just behind the gripping feature and between the two fingers of the gripper as indicated in Figure 6 with point "A".

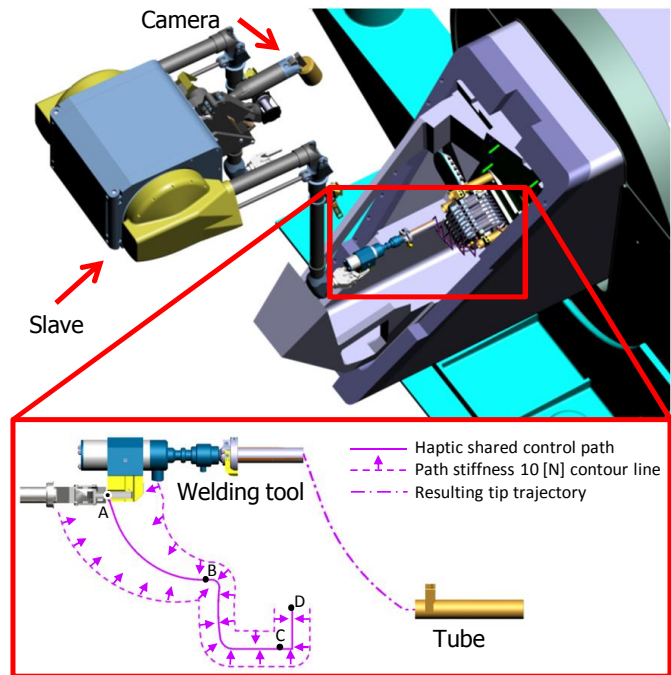


Fig. 6. The top part shows the (virtual) slave in the (virtual) environment. The red bordered part shows a side-view of the welding tool placement task. The procedure is to follow the purple path which is divided in sections A-B, B-C and C-D.

The welding tool placement task includes the four fundamental subtasks and proceeds as follows:

- 1) Free Space Movement: Simultaneously move and turn (20 degrees) the welding tool towards the tube (This subtask ends when the tool is one diameter away from the tube, seen from above).
- 2) Contact Transition part 1: Move the tool against the tube.
- 3) Contact Transition part 2: Rotated the tool horizontally to align it with the tube.
- 4) Constrained Translational Movement: Slide the tool in the tube until it is stopped by the ring on the tool.
- 5) Constrained Rotational Movement: Rotate the tool 90 degrees counter clockwise.

A tilted insertion is used to ease the process compared to a straight insertion. By approaching the tube at an angle the round shapes act a funnel and the degrees of freedom will be constrained step by step. This tilted approach has proven to be successful in peg-in-hole insertions [18], [19].

2.2.3 Master and controller

The master device used is a Haption Virtuose 6D35-45 as shown in Figure 7. This device is capable of representing force feedback in 6 degree of freedom and has a cubic workspace of 450 [mm] and a rotational workspace of 145° - 115° - 148°. It is able to exert a maximum force and torque of 35 [N] and 3.1 [Nm] (respectively

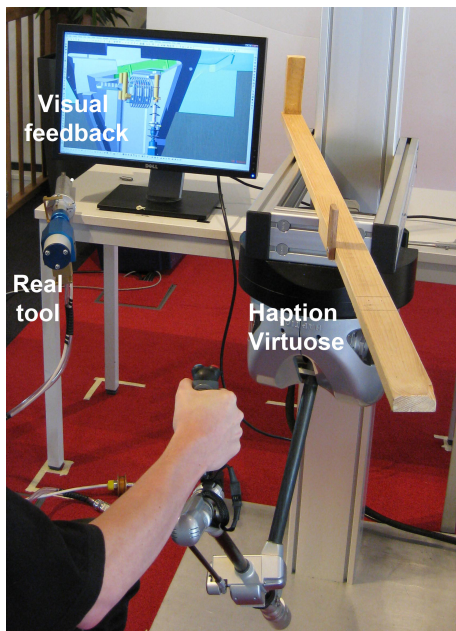


Fig. 7. Impression of the experimental set-up with the Haption Virtuose 6D35-45. And visual feedback form VR4MAX. Also shown are a make-up of the real welding tool and a wooden bar used to define the start position of the master device.

10 [N] and 1 [Nm] continuous). These force are transmitted with a maximum controller stiffness of 2000 [N/m] (translational) and 30 [Nm/rad] (rotation). The apparent inertia is 1 [kg]. The controller is implemented on a real-time Linux system and updates at 1 [kHz] [20].

The Virtuose 6D35-45 comes with an Application Programming Interface (API) which includes a position error controller (also known as PERR-control). This controller calculates proper forces and torques to apply on the master and slave device based on their position, orientation and velocities. More specifications on the controller (e.g. stiffness and damping parameters) follow in the experimental design section.

2.2.4 Haptic shared control

This research uses the haptic shared control implementation of Boessenkool [12] as baseline. His implementation is as follows:

- Free Space Movement consists of a smooth path to which the operator is guided with forces based on the 'look ahead' path error. The look ahead path error is defined as the error to the path at an estimated position at 0.1 [s] in future based on the velocity vector. Guiding forces are applied orthogonal to the path.
- Contact Transition holds a linearly increasing stiffness in orientation guiding and a linearly increasing artificial damping to prevent hard collisions.
- Constrained Translational Movement guides to the right orientation and a snap-feature attracted the

tool (like a magnet) if it is close to the target.

- Constrained Rotational Movement guides only perpendicular to the movement with the snap-feature to the centre of rotation to ensure that the tool stays on the centre of compliance.

Implementation

The guiding consists of three path segments along which the operator is guided during the four fundamental motion types (see Figure 6). The segments consist of 23 to 43 discrete points with a defined orientation. The guiding forces and torques can only build up till 10 [N] and 1 [Nm]. The motion is critically damped for each degree of freedom. In more detail:

- A-B: Free Space Movement with a linear increasing translational stiffness from 100 to 300 [N/m] and a linear increasing rotational stiffness from 2 to 8 [Nm/rad].
- B-C: Contact Transition and Constrained Translational Movement with a translational and rotational stiffness of 300 [N/m] and 8 [Nm/rad] respectively.
- C-D: Constrained Rotational Movement with only a translational stiffness of 300 [N/m]. The path is circular around the tubes centreline and lays 4 [mm] to close to the tube to facilitate a snapping force of 1.2 [N] during rotation.

Modifications

Artificial damping near contact rather decreased the ability to detect potential errors than saving the welding tool from hard shocks during the pilot study. Therefore it was chosen to omit it in the final design.

The look-ahead time showed no benefit in a pilot, while it had the potential to destabilize the controller in low transparency. Therefore it was chosen to omit it in the final design.

Extensions

Parameterizing the rotations for intuitive feedback must be done carefully. A common problem in rotations is gimbal lock (loss of rotational degrees of freedom) due to singularities. Then there is the challenge to find the difference between to rotation in such a way that rotations can be constraint by independent stiffness's. For example the tip of the welding tool suffices no rotational feedback along the rotational axis. In that case there should be no orientation of the tool in which the tip rotation receives force feedback.

Intuitively, the orientation of an object can be composed of a swing component that controls the direction at which the object points and a twist component that lets the object rotate about the pointing axis [21]. This swing twist decomposition can be computed via angle axis or quaternions (also by Euler angle, however they hold a singularity at the reference orientation [22]). This research adopted the method of quaternions to calculate the twist angle (see e.g. [23]).

The haptic shared control is extended with gravity compensation for the weight of the welding tool and

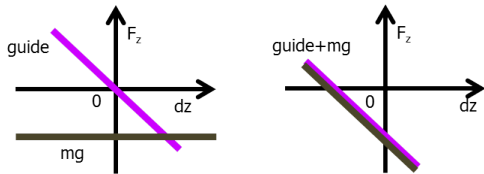


Fig. 8. Combining gravity and tool weight results in a shifted zero force reference force for the operator. Forces are shown in z (up/down) direction as function of position error in z direction.

the operator arm. These weights cause interfering translational errors in the model by shifting the zero force reference level. Normally this level is the path as haptic shared control is designed as a spring perpendicular to this path. But the tool and arm weight stretch this spring by their weight without the operator noticing it (see Figure 8). To remove this interfering translational error, the haptic shared control should have knowledge about the weight of the welding tool and the arm of the operator.

Tool weights are compensated partially in teleoperated maintenance [24], [25]. However, changing weight-inertia properties of an object might hold potential problems such as misinterpreting inertia based on the mass [26]. A pilot study gave evidence that tool weight compensation has no substantial effect on the task performance in this study. Tool weight was compensated in both guided and unguided tasks to make results comparable.

Arm weight compensation is applied as an upward force of 9 [N] (about half the arm weight) if, and only if, the slave sinks below the path. The upward force cannot be applied all the time as that might shift the zero force reference level up, because the amount of weight that rests on the guidance cannot be predicted (each operator has its own intentions). To prevent the system from instability, the force is gradually increased over 2 [mm], equalling a 4500 [N/m] stiff spring.

2.3 Experimental design

2.3.1 Experimental conditions

The experiment consists of two main factors: transparency (F1) and the operation mode (F2). These factors are combined in four experimental conditions as shown in Table 1.

The transparency is divided in low and high quality. High quality is defined as the best performance that the telemanipulator can handle. Low quality represents an -manually tuned- inferior device in the sense of controller stiffness, controller damping, device inertia and maximum force/torque representation. An indication of the device performance in both transparencies is obtained with the HapticAnalysis package by [1] for the force bandwidth in translational direction. This bandwidth is 15 [Hz] and 3 [Hz] respectively for high and low

TABLE 1
The four experimental conditions

Operation mode F2		Transparency F1	
		High	Low
Normal operation		No_HT	No_LT
Haptic Shared Control	perfect	SC_HT_0.0	SC_LT_0.0
	small neg.	SC_HT_-7.5	SC_LT_-7.5
	small pos.	SC_HT_7.5	SC_LT_7.5
	large neg.	SC_HT_-17.5	SC_LT_-17.5
	large pos.	SC_HT_17.5	SC_LT_17.5

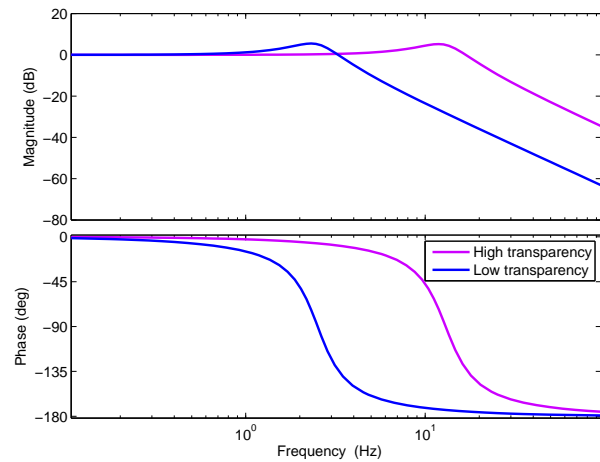


Fig. 9. Force bandwidth performance of the telemanipulator in high and low transparency.

transparency (see Figure 9). Table 2 shows properties of the telemanipulator settings in both transparencies.

The guiding inaccuracies have three levels -zero, small and large- of which the actual magnitudes were identified in a pilot study. From that study, the magnitude of 7.5 (half tube inner radius) and 17.5 [mm] (at the tubes edge) were selected as shown in figure 10. These inaccuracies were applied randomly and in a plus and minus direction in the haptic shared control condition. This randomization is done to minimize learning effects, and because a practical implementation would also hold random positive and negative errors; as systematic errors can be tuned away.

The experiment contains 7 repetitions for the normal operation and 35 repetitions for the assisted operations

TABLE 2
properties of the telemanipulator with high and low transparency

property	High quality	low quality
K_{trans} [N/m]	2000	300
B_{trans} [Ns/m]	14	10.5
K_{rot} [Nm/rad]	16.88	3
B_{rot} [Nms/rad]	0.12	0.105
Slave mass [kg]	0.3	1.2
Slave inertia [kgm^2]	675	2700
Max force [N]	35	12
Max torque [Nm]	3.3	1.2

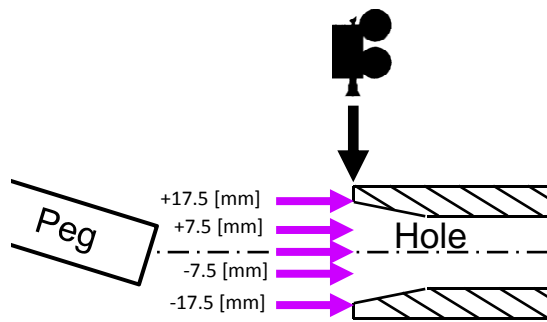


Fig. 10. Magnitude of the translational errors and the relation to the provided visual feedback.

(5 errors times 7 repetitions) in 2 transparencies. That makes the four conditions (Table 1) which were applied randomly to the participants to minimize the influence of learning effects.

2.3.2 Controlled variables

Task instruction and training

Before the experiment subjects were explained about the task and device. They were instructed to:

- Place the tool as specified in Section 2.2.2.
- Perform the task as accurate as possible because of the delicate equipment and high cost on failure.
- Work fast enough or else they would be too expensive as operator.

After the instructions, participants were trained with the task and device (in high transparency without haptic shared control) until they performed the task in about 15 [s]. To indicate tool damage and prevent reckless operations an annoying buzzer sounded and simulation stopped at a too high impact energy. A second training was used to acquaint the participants with flawless haptic shared control (in high transparency). In addition subjects had practice before the start of each new condition during the actual experiment. During the practice with (inaccurate) haptic shared control, the participants were told that the the system could have some sort of inaccuracies.

Visual feedback and direction of guidance error

During the training participants were provided with 3 video camera streams for the top, back and side of the task. In the actual experiment only the top view was available. This top view was positioned above the tube's entrance and placed in line with the translational error (see Figure 10). This occludes the error to eliminate visual feedback from interfering in the effect of guiding errors on task performance and control effort.

2.4 Data acquisition & metrics

Data was collected at 1 [kHz] for: Time; Forces/Torques; Positions/orientation; and Velocities. Figure 11 shows an example typical performed task from a side view

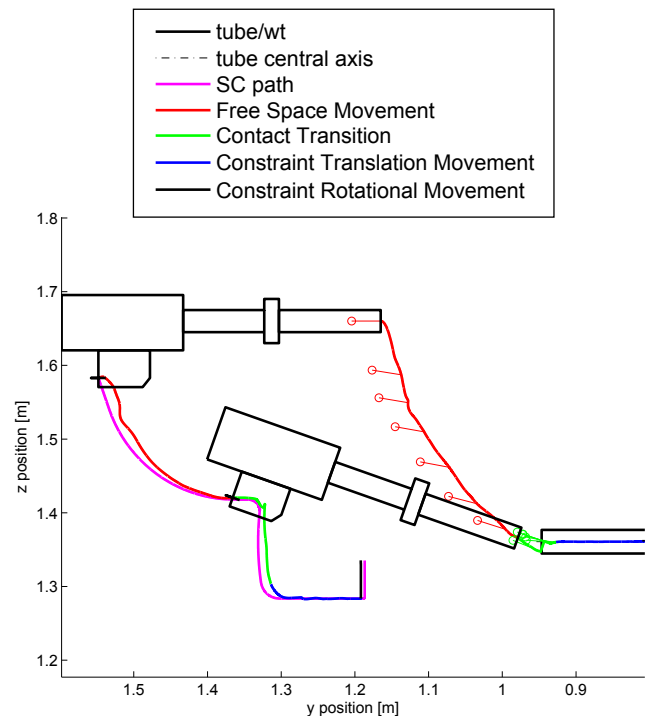


Fig. 11. Typical results from a subject completing the welding tool placement task with perfect haptic shared control and high quality feedback.

perspective. The gathered data serves to evaluate task performance and control effort. The task performance is evaluated in:

t_{tc} time to complete [s], the time in seconds required to complete the task.

Control effort is evaluated in:

t_{tsa} Total steering angle [°], the total amount of rotation that the operator made with his hand, which can be seen as a measure of control effort for the subject to control the system. It is measured as the summed (integrated) difference between two subsequent orientations of the master device.

Furthermore, subjective cognitive workload was measured with the NASA Task Load Index (NASA-TLX) [27]. It rates subjective workload on a scale of 0-100 where a lower score represents a lower workload.

2.5 Data Analysis

The results for each of the 7 repetitions per subjects were averaged and assumed to have a normal distribution. A paired t-test was used to analyse the difference in the experimental conditions. Results were considered to be significant below $p=0.05$.

3 RESULTS

The results of the metrics are presented in this section, starting with the results for the entire task, then the subtasks will be analysed.

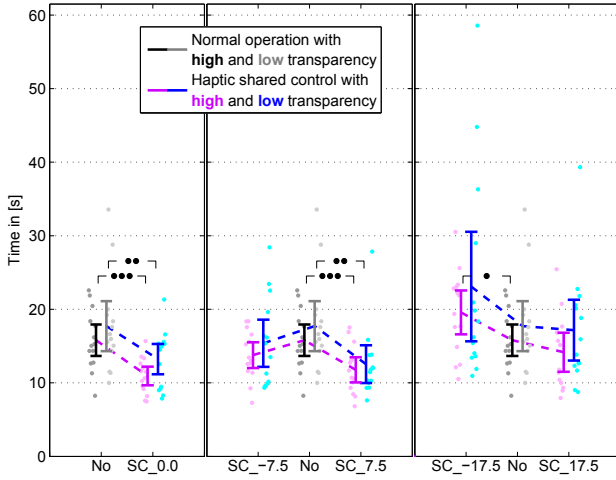


Fig. 12. Task performance - Time to complete for the entire task (14 subjects, 7 repetitions), showing the mean and 95% CI. Haptic shared control allows a shorter or equal time to complete compared to normal operation despite small positive errors. '•••', '••' and '•' respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$

3.1 Effect of transparency and inaccurate Haptic Shared Control on the entire task

Figure 12 and table 3 show the task performance - in terms of time to complete- for the entire task. For assistance without errors, the results show guidance improves task performance compared to normal operations. The improvement is from 15.79 [s] to 10.93 [s] ($p < 0.001$, $t = 5.7$) with high transparency and from 17.71 [s] to 13.23 [s] ($p = 0.005$, $t = 3.4$) with low transparency.

The inaccurate assistance shows an asymmetric pattern. Small positive errors reduce the required time compared to normal operations. The improvement in execution time is from 15.79 [s] to 11.77 [s] ($p < 0.001$, $t = 5.6$) with high transparency and from 17.71 [s] to 12.54 [s] ($p = 0.002$, $t = 4.0$) with low transparency. This while small negative errors do not change performance compared to normal operations ($p > 0.059$).

For a large negative error the required time increases compared to normal operations from 15.79 [s] to 19.57 [s] ($p < 0.013$, $t = -2.9$) for high transparency, while positive large errors do not change performance significantly.

Figure 13 and table 3 show the control effort -in terms of total steering angle- for the entire task. For flawless assistance, this metric shows improvement in control effort compared to normal operations from 272.6 [°] to 185.73 [°] ($p < 0.001$, $t = 5.7$) with high transparency and from 302.7 [°] to 213.4 [°] ($p = 0.001$, $t = 4.1$) with low transparency.

Except for the small positive error in low transparency, all small errors reduce the total steering angle compared to normal operations ($p < 0.004$).

For large errors the total steering angle reduce only

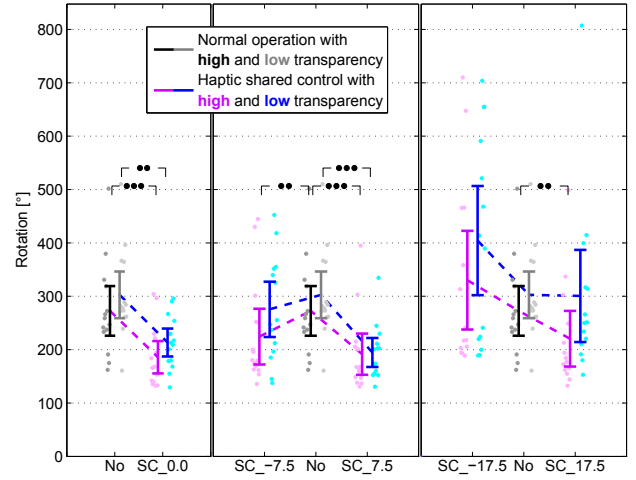


Fig. 13. Control effort - Total steering angle for the entire task (14 subjects, 7 repetitions), showing the mean and 95% CI. Haptic shared control improves or equals control effort compared to normal operation for small errors. '•••', '••' and '•' respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$

TABLE 4

Results for the Self-reported cognitive workload (14 subjects) for normal operations and haptic shared control in both transparencies. The table shows the mean and the 95% confidence interval. Also shown are the differences between them by the t-values and the significance (p-value).

	NASA TLX [-]	
	Mean (95% CI)	t p diff
No_HT to SC_HT	39.19(6.97) 34.14(8.40)	- 2.1 0.060
No_LT to SC_LT	40.45(7.08) 41.02(9.41)	- -0.1 0.889

for the positive error compared to normal operations from 272.6 [°] to 220.5 [°] ($p = 0.008$, $t = 3.1$) provided high transparency.

Table 4 shows the NASA TLX scores. The TLX test has only four measurements because the inaccuracies in haptic shared control were provided randomly in one condition for each transparency mode. The results indicate no change in cognitive workload between normal operations and inaccurate haptic shared control ($p > 0.06$). The test reveals that four out of fourteen participants experienced an increased workload for haptic shared control -compared to normal operations- in both high and low transparency (not shown). Another four experienced a higher workload for haptic shared control -compared to normal operations- in only low transparency (not shown).

TABLE 3

Results for the Time-to-complete in the entire task and Free Space Movement and the total steering angle in the entire task. The results are given per experimental condition and show the mean and the 95% confidence interval. Also shown are the differences between conditions by the t-values and the significance (p-value).

	ttc [s]			tsa [°]			ttc Free Space [s]		
	Mean (95% CI)	t	p diff.	Mean (95% CI)	t	p diff.	Mean (95% CI)	t	p diff.
No in HT to	15.79 (13.65;17.93)	-	-	272.6 (226.1;319.1)	-	-	4.25 (3.33;5.17)	-	-
SC_0.0	10.93 (9.66;12.20)	5.7	<0.001	185.7 (155.4;216.0)	5.7	<0.001	3.73 (3.36;4.10)	1.4	0.180
SC_-7.5	13.77 (12.01;15.52)	2.1	0.059	224.4 (172.2;276.6)	3.8	0.002	4.06 (3.46;4.66)	0.8	0.452
SC_7.5	11.77 (10.06;13.48)	5.6	<0.001	191.5 (152.8;230.2)	8.1	<0.001	3.80 (3.31;4.28)	1.1	0.274
SC_-17.5	19.57 (16.59;22.56)	-2.9	0.013	330.2 (237.7;422.6)	-1.7	0.108	4.08 (3.63;4.53)	0.5	0.636
SC_17.5	14.16 (11.50;16.81)	1.6	0.140	220.5 (168.3;272.6)	3.1	0.008	3.94 (3.22;4.66)	1.5	0.169
No in LT to	17.71 (14.31;21.11)	-	-	302.7 (258.9;346.4)	-	-	4.23 (3.52;4.94)	-	-
SC_0.0	13.23 (11.17;15.29)	3.4	0.005	213.4 (187.2;239.6)	4.1	0.001	3.92 (3.49;4.35)	1.5	0.158
SC_-7.5	15.37 (12.16;18.58)	1.4	0.182	275.6 (223.7;327.4)	0.9	0.395	3.99 (3.47;4.50)	1.0	0.338
SC_7.5	12.54 (9.97;15.10)	4.0	0.002	194.6 (167.4;221.8)	5.3	<0.001	4.14 (3.56;4.73)	0.4	0.716
SC_-17.5	23.09 (15.65;30.53)	-1.6	0.131	404.3 (302.1;506.5)	-2.0	0.062	4.22 (3.69;4.76)	0.0	0.969
SC_17.5	17.16 (13.03;21.30)	0.3	0.744	300.6 (214.2;387.0)	0.1	0.947	4.40 (3.63;5.17)	-0.5	0.592

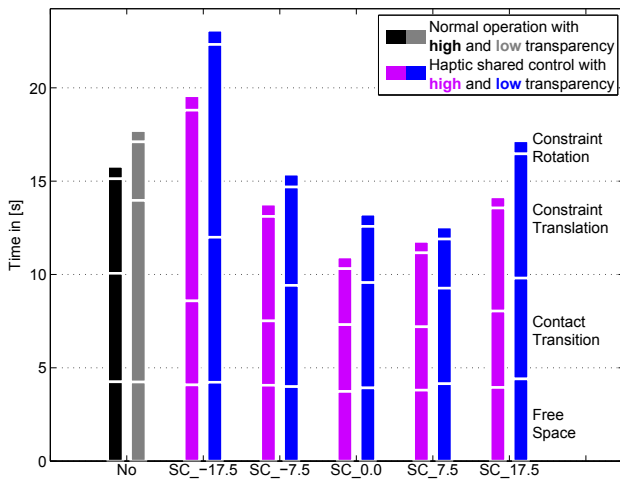


Fig. 14. Task Performance - Time to complete for the entire task (14 subjects, 7 repetitions), separated for the four fundamental subtasks.

3.2 Effect of transparency and inaccurate Haptic Shared Control on the fundamental subtasks

The results above showed how transparency and inaccurate Haptic Shared Control affected the overall task, this section investigates how these effects relate to the four fundamental subtasks. Figure 14 shows a bar chart of the time-to-complete, per fundamental subtask. Table 3 and 5 show the descriptive results and differences.

Figure 15 shows a bar chart of the total steering angle, per fundamental subtask. Table 6 and 7 show the descriptive results and differences.

3.2.1 Free Space Movement

Free space movement shows no change in time-to-complete for haptic shared control compared to normal

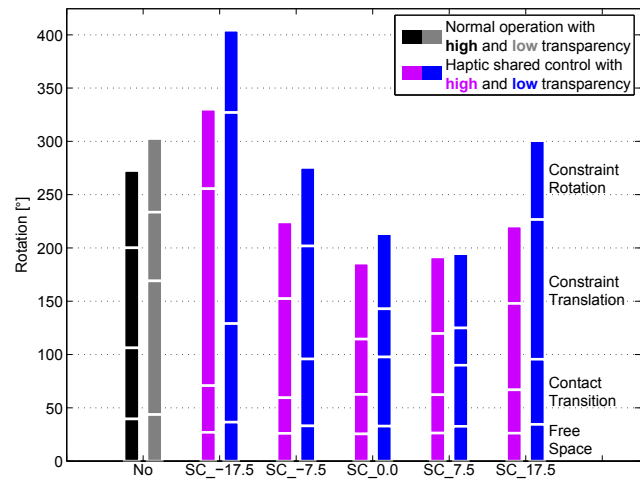


Fig. 15. Control effort - Total steering angle for the entire task (14 subjects, 7 repetitions), separated for the four fundamental subtasks.

operation ($p > 0.158$), see table 3. The total steering angle is not reduced for a large negative error with low transparency compared to normal operation ($p = 0.052$). Other conditions reduce the total steering angle ($p > 0.021$).

3.2.2 Contact Transition

Contact Transition improves time-to-complete for zero error haptic shared control compared to normal operation. This is from 5.81 [s] to 3.58 [s] ($p < 0.001$, $t = 4.9$) with high transparency and from 9.74 [s] to 5.64 [s] ($p = 0.005$, $t = 3.4$) with low transparency.

Large negative error in haptic shared control do not change task performance compared to normal operation provided low transparency ($p = 0.251$). Other inaccuracies improve task performance compared to normal operation ($p < 0.004$) for both transparencies.

TABLE 5

Results for the Time-to-complete in Contact Transition, Constrained Translational Movement and the Constrained Rotational Movement per experimental conditions. The table shows the mean and the 95% confidence interval. Also shown are the differences between conditions by the t-values and the significance (p-value).

	ttc Contact Transition [s]			ttc Constrained Translation [s]			ttc Constrained Rotation [s]		
	Mean (95% CI)	t	p diff.	Mean (95% CI)	t	p diff.	Mean (95% CI)	t	p diff.
No in HT to	5.81 (4.99;6.64)	-	-	5.06 (3.71;6.42)	-	-	0.66 (0.59;0.74)	-	-
SC_0.0	3.58 (2.68;4.48)	4.91	<0.001	3.00 (2.45;3.54)	3.34	0.005	0.62 (0.54;0.71)	1.17	0.263
SC_-7.5	3.45 (2.51;4.39)	6.76	<0.001	5.59 (4.59;6.60)	-0.71	0.489	0.66 (0.58;0.74)	0.12	0.905
SC_7.5	3.40 (2.62;4.17)	9.96	<0.001	3.97 (2.95;4.98)	1.46	0.167	0.61 (0.53;0.69)	1.78	0.098
SC_-17.5	4.50 (3.41;5.59)	3.79	0.002	10.22 (7.83;12.61)	-4.21	0.001	0.77 (0.63;0.92)	-1.75	0.103
SC_17.5	4.10 (3.23;4.97)	4.19	0.001	5.52 (4.05;6.99)	-0.49	0.632	0.60 (0.53;0.66)	2.31	0.038
No in LT to	9.74 (7.02;12.45)	-	-	3.14 (2.38;3.89)	-	-	0.60 (0.53;0.67)	-	-
SC_0.0	5.64 (4.38;6.91)	3.36	0.005	3.01 (2.06;3.96)	0.21	0.837	0.65 (0.58;0.73)	-1.45	0.172
SC_-7.5	5.43 (3.25;7.61)	3.49	0.004	5.27 (3.55;6.99)	-2.40	0.032	0.69 (0.59;0.79)	-1.70	0.113
SC_7.5	5.12 (3.53;6.71)	4.42	<0.001	2.63 (1.76;3.50)	0.86	0.406	0.64 (0.55;0.74)	-0.84	0.418
SC_-17.5	7.77 (3.38;12.16)	1.20	0.251	10.33 (6.12;14.54)	-3.37	0.005	0.76 (0.65;0.87)	-3.49	0.004
SC_17.5	5.41 (4.32;6.49)	3.36	0.005	6.66 (3.63;9.68)	-2.15	0.051	0.70 (0.60;0.80)	-2.14	0.052

TABLE 7

Results for the total steering angle in Contact Transition, Constrained Translational Movement and the Constrained Rotational Movement per experimental condition. The table shows the mean and the 95% confidence interval. Also shown are the differences between conditions by the t-values and the significance (p-value).

	tsa Contact Transition [°]			tsa Constrained Translation [°]			tsa Constrained Rotation [°]		
	Mean (95% CI)	t	p diff.	Mean (95% CI)	t	p diff.	Mean (95% CI)	t	p diff.
No in HT to	66.7 (55.4;78.0)	-	-	93.9 (63.1;124.7)	-	-	72.5 (68.8;76.3)	-	-
SC_0.0	37.1 (29.1;45.0)	4.4	<0.001	51.9 (25.5;78.4)	4.8	<0.001	71.3 (68.4;74.3)	0.9	0.401
SC_-7.5	33.5 (26.0;41.0)	10.5	<0.001	93.0 (46.9;139.1)	0.1	0.944	72.0 (69.7;74.3)	0.4	0.686
SC_7.5	36.0 (28.4;43.5)	7.8	<0.001	57.4 (22.9;91.9)	4.2	0.001	71.9 (69.7;74.1)	0.5	0.627
SC_-17.5	43.7 (34.3;53.2)	4.1	0.001	184.8 (98.2;271.5)	-2.6	0.021	74.6 (70.0;79.2)	-0.7	0.484
SC_17.5	40.7 (34.6;46.7)	4.3	<0.001	81.0 (33.5;128.6)	0.9	0.407	72.6 (69.7;75.5)	-0.0	0.973
No in LT to	125.4 (93.0;157.8)	-	-	64.5 (44.0;85.0)	-	-	69.1 (65.2;73.1)	-	-
SC_0.0	64.9 (46.4;83.3)	3.9	0.002	45.3 (29.9;60.6)	1.5	0.163	70.5 (68.1;72.8)	-0.8	0.439
SC_-7.5	62.9 (34.5;91.2)	4.2	0.001	106.0 (62.8;149.3)	-2.0	0.065	73.7 (70.2;77.2)	-2.1	0.057
SC_7.5	57.4 (39.5;75.2)	5.5	<0.001	34.9 (21.0;48.7)	2.5	0.027	69.8 (68.4;71.1)	-0.3	0.760
SC_-17.5	92.5 (35.4;149.6)	1.7	0.122	198.0 (120.8;275.2)	-3.5	0.004	77.3 (73.0;81.6)	-3.5	0.004
SC_17.5	61.2 (48.5;73.8)	4.3	<0.001	131.0 (53.0;209.1)	-2.0	0.066	74.0 (70.3;77.7)	-5.8	<0.001

TABLE 6

Results for the total steering angle in Free Space Movement per experimental condition. The table shows the mean and the 95% confidence interval. Also shown are the differences between conditions by the t-values and the significance (p-value).

	tsa Free Space [°]		
	Mean (95% CI)	t	p diff.
No in HT to	39.5 (33.1;45.9)	-	-
SC_0.0	25.4 (23.2;27.7)	4.7	<0.001
SC_-7.5	26.0 (23.3;28.6)	4.9	<0.001
SC_7.5	26.3 (23.3;29.3)	4.7	<0.001
SC_-17.5	27.0 (23.9;30.1)	4.2	0.001
SC_17.5	26.2 (23.2;29.2)	4.7	<0.001
No in LT to	43.7 (38.4;48.9)	-	-
SC_0.0	32.8 (28.9;36.7)	3.2	0.007
SC_-7.5	33.0 (29.1;36.8)	3.0	0.009
SC_7.5	32.6 (29.1;36.1)	3.3	0.006
SC_-17.5	36.5 (32.4;40.6)	2.1	0.052
SC_17.5	34.4 (30.2;38.7)	2.6	0.021

Control effort is not changed for a large negative error with low transparency compared to normal operation ($p=0.122$). Other conditions reduce the total steering angle ($p>0.002$).

3.2.3 Constrained Translational Movement

This subtask improve time-to-complete with zero error assistance compared to normal operation from 5.06 [s] to 3.00 [s] ($p=0.005, t=-3.3$) provided high transparency.

Small negative errors degrade task performance compared to normal operation from 3.14 [s] to 5.27 [s] ($p=0.032, t=-2.4$) provided low transparency.

Further this subtask increases time-to-complete for large negative errors compared to normal operation from 5.06 [s] to 10.22 [s] ($p=0.001, t=-4.2$) with high transparency and from 3.14 [s] to 10.33 [s] ($p=0.005, t=-3.4$) with low transparency.

With zero error assistance the total steering angle decreases compared to normal operation from 93.9 [°] to 51.9 [°] ($p<0.001, t=4.8$) with high transparency.

Small positive errors decrease control effort compared to normal operation from 93.9 [°] to 57.4 [°] ($p=0.001, t=4.2$) with high transparency and from 64.5 [°] to 34.9 [°] ($p=0.027, t=2.5$) with low transparency.

Control effort increases for large negative errors compared to normal operation from 93.9 [°] to 184.8 [°] ($p<0.021, t=-2.6$) with high transparency and from 64.5 [°] to 198.0 [°] ($p<0.004, t=-3.5$) with low transparency.

3.2.4 Constrained Rotational Movement

The required time in the rotational movement increases for assistance with large negative errors compared to normal operation from 0.60 [s] to 0.76 [s] ($p=0.004, t=-3.5$) with low transparency. This while it decreases for positive errors with high transparency from 0.66 [s] to 0.60 [s] ($p=0.038, t=2.3$).

The total steering angle increases for large errors compared to normal operation provided low transparency. The increase is from 96.1 [°] to 77.3 [°] ($p=0.004, t=-3.5$) for negative errors and from 74.0 [°] 96.1 [°] ($p<0.001, t=-5.8$) for positive errors.

4 DISCUSSION

This section discusses the effect of inaccurate haptic shared control with respect to normal operation (unguided with tool weight compensation). In the discussion the effects of positive and negative errors are combined. This is done as the intended sensory or model errors would -in real life- also occur in pairs in a normal distribution. Constant offset errors during the execution of many tasks can be compensated for.

The experimental results showed that the welding tool placement task benefits from perfect (zero offset error) haptic shared control -provided high transparency- with respect to normal operation. That is in line with previous research like e.g. [8]–[11], [13], [14]. The results showed similar results for low transparency which is in-line with the findings from Boessenkool [12]. For guidance with small errors (+7.5;-7.5 [mm]) the results were similar to the results for flawless guiding, while large errors (+17.5;-17.5 [mm]) did at best not degrade performance beyond normal operation. In essence, the performance level improved the required time (by 20-30%) and control effort (by 22-31%) when small or zero guiding errors occurred. The cognitive workload showed no change. But the TLX test was measured for the combined zero, small and large errors per transparency level. This was because the errors were mixed (per transparency) level to prevent learning effects.

The experimental results for the entire task, are quite close to the hypothesis that guiding errors affect performance depending on error size and transparency level. Deviation from the expectation is found in the

results for low transparency. It was expected that small and large errors would respectively not change and decrease performance with respect to normal operation. But the results showed that small and large guiding errors respectively improved and did not differ in performance compared to normal operation, just like for high transparency. Such an indifference was previously identified for flawless haptic shared control [12]. The indifference in transparency for flawed guiding indicates that improving transparency does not aid in the detectability of the error.

Examining the subtasks in detail showed that Free Space Movement did benefit from haptic shared control, as the total steering angle reduced compared to normal operations in both transparency condition. However, this subtask is not affected in time-to-complete for inaccurate guiding forces in neither transparency condition. This while Boessenkool [12] showed an improvement 25-30% in his bolt-and-spanner task in Free Space Movement. The results might differ because the heavy welding tool could only be accelerated for 1 or 2 seconds after which participants needed to decelerate it to prepare for the Contact Transition while the bolt-and-spanner task had a light -easy to accelerate- tool. This makes the task different although a similar free space path length (about 20-30 [cm]) is travelled.

In Contact Transition, the experimental results indicated that haptic shared control reduced the required time and total steering angle compared to normal operation, independently of errors and independently of transparency. The reduction by haptic shared control is about 45% and 50% for respectively the required time and total steering angle.

The experimental results for the Constrained Translational Movement showed an asymmetric pattern for positive and negative errors. Still there seems to be marginal benefit of haptic shared control with small guiding compared to normal operations, as the results mainly show equal or improved task performance and control effort. In the same way large guiding errors degrade task performance and control effort compared to unguided operations.

A possible explanation for the -unexpected- asymmetric trend is that the welding tool is held below the centre-line of the "peg". This enhances jamming with negative errors -and reduces jamming with positive errors- if the operator tries to push the tool in the tube. Such jamming was also observed during the test provided assistance with negative errors.

The asymmetry affects only the Constrained Translational Movements as the other subtasks induce no jamming during movement. Practically this makes the results only valid for pegs with a similar configuration (like e.g. drills). For tools with a gripper supports in line with the peg, results would probably be improved and

slightly degraded compared to respectively negative and positive guiding errors.

The Constrained Rotational Movement showed an increased control effort compared to normal operation for large inaccuracies in the guiding provided low transparency. This was expected for large errors but at least perfect assistance should have improved performance according to the hypothesis. The absence of difference in this subtask was also found by Boessenkool [12], but in his experiment participants mentioned the difficulty of the Constrained Rotational Movement and helpfulness of haptic shared control. The later was not the case in the current experiment which is probably a result of the level of difficulty of this subtask. It had for example no potential release the welding tool from the rotational compliance centre. A redesign, such as a gripper that could slide of a door handle during unlatching a door [15] or polishing a curved surface, could reveal a larger benefit from haptic shared control and a larger impact from model errors in a Constrained Rotational Movement.

Limitations

Operator arm weight was compensated by a constant force of 9 [N] if the welding tool sank below the path. This counteracted the -arm weight induced- vertical translational error, which might have affected the test results. First of all the weight compensation is a non-linearity which can be used by the operator as an extra cue to perform the task well. However subjectively, none of the participants consciously noted the difference between up and downward assistance.

Secondly the support force introduced an upward force of 14.25 [N] ($= \max(10 [N]; 300 [N/m] * 17.5 [mm]) + 9 [N]$) for +17.5 [mm] guiding errors while the the -17.5 [mm] error caused an 5.25 [N] force downward. By adding the participants arm weight, the total vertical force floats (depending on the intention of the operator) somewhere between 5.25 [N] and 14.25 [N] up or down. This suggests a changing asymmetry which is sometimes in equal to and sometimes in contrasts with the asymmetry found in the Constrained Translational Movement.

That the experiment showed no differed trend for inaccurate guiding between high and low transparency might indicate that the high and low (respectively 15 [Hz] and 3 [Hz] slave-master force bandwidth) level were not distinct enough. However, a clear trend shows between high and low transparency in Contact Transition in normal operations. For time-to-complete the trend is a from 5.81 [s] (95% CI [4.99;6.64]) to 9.74 [s] (95% CI [7.02;12.45]) in respectively high and low transparency. For total steering angle the trend is is from 66.7 [°] (95% CI [55.4;78.0]) to 125.4 [°] (95% CI [93.0;157.8]) respectively high and low transparency.

Wildenbeest [6] mentioned to expect similar trends

as the above in his three degree of freedom bolt-and-spanner task. However they did not find any. Still they expected that a six degree of freedom task would be fundamentally more difficult due to the decreased of visual feedback quality by multiple views. This expectation are now provided with some evidence.

Also the comparison of (inaccurate) haptic shared control between high and low transparency show a clear trend. For flawless assistance the time-to-complete showed a trend from 3.58 [s] (95% CI [2.68;4.48]) to 5.64 [s] (95% CI [4.38;6.91]) in respectively high and low transparency. For flawless assistance the total steering angle show a trend from 37.1 [°] (95% CI [29.1;45.0]) to 64.9 [°] (95% CI [46.4;83.3]) in respectively high and low transparency. This is in contrast to what Boessenkool [12] noted as that teleoperated tasks may benefit more from focussing on haptic shared control than focussing on improvement of transparency.

Altogether there is strong evidence that transparency levels were distinct enough. Furthermore the results indicated that -in Contact Transition for this task- both transparency and haptic shared control are beneficial for task performance and control effort.

Future work

The effect of model errors relate closely to the stiffness of the haptic shared control (also referred to as the level of haptic authority [7], [28]). Using a low stiffness haptic shared control lessens the effect of an error (operators can easily overrule the guiding) but also provides less clear guiding forces. Increasing the stiffness makes haptic shared control look more like automation in which case small errors will causes and increases execution time [9]. A promising idea to deal with control authority is to make this level shift gradually between the human and haptic shared control depending the task, the operator's intention and the criticality of the task [28].

Another topics to improve haptic shared control is the choice for the "ideal" path used in Free Space Movement and Contact Transition. Currently the path is optimized to insert the tool under an angle to facilitate the peg-in-hole task [18], [19]. It is also optimized to smoothly get to the insertion in one stroke of the device. In real telemanipulation situations an optimized path, completely matching with the human intention, is often not available (or hard to derive). In such cases on-line adaptation shows to be a promising solution for repetitive tasks, even if a sharp cornered path is provided at first [29].

Although haptic shared control in telemanipulation is far from a final design the experimental results imply that for an ideal telemanipulator with haptic shared control: Transparency should be high when the task mainly revolves around Contact Transition; Transparency does not have to be very high when the task mainly in-

volves subtasks other than Contact Transition; And small translational errors in haptic shared control still have a positive effect with respect to unguided operations (especially in Contact Transition).

5 CONCLUSION

This research investigated the influence of inaccuracies in haptic shared control on the task performance and control effort of operators. During a peg-in-hole type task, the operators were provided with: high and low transparency; and guidance with offset errors in vertical translational direction. For the experimental conditions studied it can be concluded that overall:

- Guidance without errors improved task performance and control effort compared to unguided operations.
- Despite small errors, task performance and control effort are still improved compared to unguided operations.
- Larger errors in the guidance decreased task performance and control effort to a similar or worse level as during unguided operations.
- Transparency did not affect the operator's ability to deal with wrong guiding.

REFERENCES

- [1] G. Christiansson, "Hard master, soft slave haptic teleoperation," Ph.D. dissertation, Delft, 2007.
- [2] C. Johnson and A. B. Koku, "Enhancing a human-robot interface using Sensory EgoSphere," Ph.D. dissertation, 2003.
- [3] M. K. OMalley, A. Gupta, M. Gen, and Y. Li, "Shared Control in Haptic Systems for Performance Enhancement and Training," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 1, p. 75, 2006.
- [4] Y. Yokokohji and Y. Iida, "Toy Problem as the Benchmark Test for Teleoperation Systems," pp. 996–1001, 2000.
- [5] B. Hannaford and L. Wood, "Performance evaluation of a six-axis generalized force-reflecting teleoperator," *Systems, Man and ...*, vol. 21, no. 3, pp. 620–633, 1991.
- [6] J. G. W. Wildenbeest, D. Abbink, C. J. Heemskerk, F. C. van der Helm, and H. Boessenkool, "The Impact of Haptic Feedback Quality on the Performance of Teleoperated Assembly Tasks," *IEEE TRANSACTIONS ON HAPTICS*, no. 1939-1412, pp. 1–12, 2012.
- [7] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?" *Cognition, Technology & Work*, pp. 19–28, 2012.
- [8] L. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *Virtual Reality Annual International Symposium, 1993, 1993 IEEE*. IEEE, 1993, pp. 76–82.
- [9] P. Marayong, A. Bettini, and A. Okamura, "Effect of virtual fixture compliance on human-machine cooperative manipulation," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2, no. October. IEEE, 2002, pp. 1089–1095.
- [10] A. Bettini, P. Marayong, S. Lang, A. Okamura, and G. Hager, "Vision-assisted control for manipulation using virtual fixtures," *Robotics, IEEE Transactions on*, vol. 20, no. 6, pp. 953–966, 2004.
- [11] C. Basdogan, A. Kiraz, I. Bukusoglu, A. Varol, and S. Doanay, "Haptic guidance for improved task performance in steering microparticles with optical tweezers." *Optics express*, vol. 15, no. 18, pp. 11 616–11 621, Sep. 2007.
- [12] H. Boessenkool, D. Abbink, C. J. Heemskerk, F. C. van der Helm, and J. G. W. Wildenbeest, "A Task-Specific Analysis of the Benefit of Haptic Shared Control During Tele-Manipulation," *IEEE Transactions on Haptics*, pp. 1–1, 2012.
- [13] S. Park, R. Howe, and D. Torchiana, "Virtual fixtures for robotic cardiac surgery," in *Medical Image Computing and Computer-Assisted InterventionMICCAI 2001*. Springer, 2001, pp. 1419–1420.
- [14] J. Abbott and A. Okamura, "Virtual fixture architectures for telemanipulation," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 2798–2805.
- [15] A. Sverrisson, "Realization and Evaluation of a Remotely Controlled Mobile Robot with Shared Control and Underactuated Hand to Improve the Unlatching of Doors," Master's thesis, Delft, 2012.
- [16] M. Mulder and D. Abbink, "Correct and faulty driver support from shared haptic control during evasive maneuvers," *Systems, Man, and Cybernetics (SMC)*, pp. 1057–1062, 2011.
- [17] C. J. Heemskerk, M. de Baar, H. Boessenkool, B. Graafland, M. Haye, J. Koning, M. Vahedi, and M. Visser, "Extending Virtual Reality simulation of ITER maintenance operations with dynamic effects," *Fusion Engineering and Design*, vol. 86, no. 9-11, pp. 2082–2086, Oct. 2011.
- [18] M. Caine, T. Lozano-Perez, and W. Seering, "Assembly strategies for chamferless parts," *Proceedings, 1989 International Conference on Robotics and Automation*, pp. 472–477, 1989.
- [19] D. Strip, "A passive mechanism for insertion of convex pegs," *Robotics and Automation, 1989. Proceedings., 1989 ...*, pp. 242–248, 1989.
- [20] "Virtose 6D35-45 data sheet," HAPTION S.A., Atelier relais ZA Route de Laval, 53210 SOULGE SUR OUETTE, France. [Online]. Available: http://www.est-kl.com/fileadmin/media/pdf/Haption/Virtuose_6D35-45_en_rev_1.pdf
- [21] F. Grassia, "Practical parameterization of rotations using the exponential map," *Journal of graphics tools*, vol. 3, pp. 1–13, 1998.
- [22] P. Baerlocher and R. Boulic, "Parametrization and range of motion of the ball-and-socket joint," *Workshop on Deformable Avatars*, pp. 1–15, 2001.
- [23] M. Kallmann, "Analytical inverse kinematics with body posture control," *Computer Animation and Virtual Worlds*, vol. 19, pp. 79–91, 2008.
- [24] S. Sanders, "Remote operations for fusion using teleoperation," *Industrial Robot: An International Journal*, vol. 33, no. 3, pp. 174–177, 2006.
- [25] L. Galbiati, T. Raimondi, P. Garetto, and G. Costi, "Control and Operational Aspects of the Mascot 4 force feedback servomanipulator of JET," in *Fusion Engineering, 1991. Proceedings., 14th IEEE/NPSS Symposium on*. IEEE, 1991, pp. 563–566.
- [26] J. McIntyre, A. Berthoz, and F. Lacquaniti, "Reference frames and internal models for visuo-manual coordination: what can we learn from microgravity experiments?" *Brain research. Brain research reviews*, vol. 28, no. 1-2, pp. 143–54, Nov. 1998.
- [27] S. Hart and L. Staveland, "No TitleDevelopment of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," *P. Hancock and N. Meshkati (Eds.)*, pp. 139–183, 1988.
- [28] M. Mulder, D. Abbink, and E. Boer, "Sharing Control With Haptics: Seamless Driver Support From Manual to Automatic Control," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, May 2012.
- [29] A. de Jonge, "The Effect of Online Adaptation on Conflicts in Haptic Shared Control for Free-Air Teleoperation Tasks," Master's thesis, Delft, 2012.

Appendices belonging to the Master's thesis:

Robustness of Haptic Shared Control Against Model Inaccuracies During Telemanipulation

J. van Oosterhout

Content

A.	Experimental setup	3
A.1.	Software	3
A.2.	Telemanipulator	3
A.3.	Task and environment	6
A.3.1.	ITER	6
A.3.2.	Selection of the task	7
A.3.3.	Modelling the task	9
A.4.	Short guideline to run the setup	10
A.5.	Short guide to setup a configuration file	11
A.6.	Short guide to setup a shared control path	12
B.	Shared control designs & implementation	13
B.1.	Design of haptic shared control	13
B.1.1.	Baseline design	13
B.1.2.	Assistance in 3 independent rotations (swing twist decomposition)	14
B.1.3.	Weight compensation welding tool	16
B.1.4.	Weight of operator arm	20
B.2.	Implementation of shared control	23
C.	Haptic shared control experiment	29
C.1.	Task instructions	29
C.2.	Subdivision in subtasks	32
C.3.	Description of evaluation metrics	34
C.4.	Data management	38
C.5.	Results	39
C.5.1.	Raw data	39
C.5.2.	Free Space movement	42
C.5.3.	Contact Transition	44
C.5.4.	Constrained Translation Movement	47
C.5.5.	Constrain Rotation Movement	49
C.5.6.	Subjective	51
C.6.	Ethics	53
	References	55

A. Experimental setup

A.1. Software

The experimental task is performed in a virtual reality environment for which, several software modules are used. To provide a better insight in how the various software modules are related to each other, a flow diagram is shown in Figure 1.

The environment is based on CAT models which are present at the Remote Handling Study Centre. The models are opened in 3DS MAX and provided with physical properties with the Nvidia PhysX plugin. Also cameras, hud views (PIP) and additional objects for the haptic shared control path are implemented here.

The model is exported to VR4MAX (.VMX) and PhysX (.XML) if the modelling is done. Then the Virtual Slave Simulator can load these files provided a proper instruction (defined in a .cfg file). The Virtual Slave Simulator starts both the render and the PhysX simulation. The Virtual Slave Simulator also allows to link a robotic master device (e.g. Virtuose 6D35-45) to the scene.

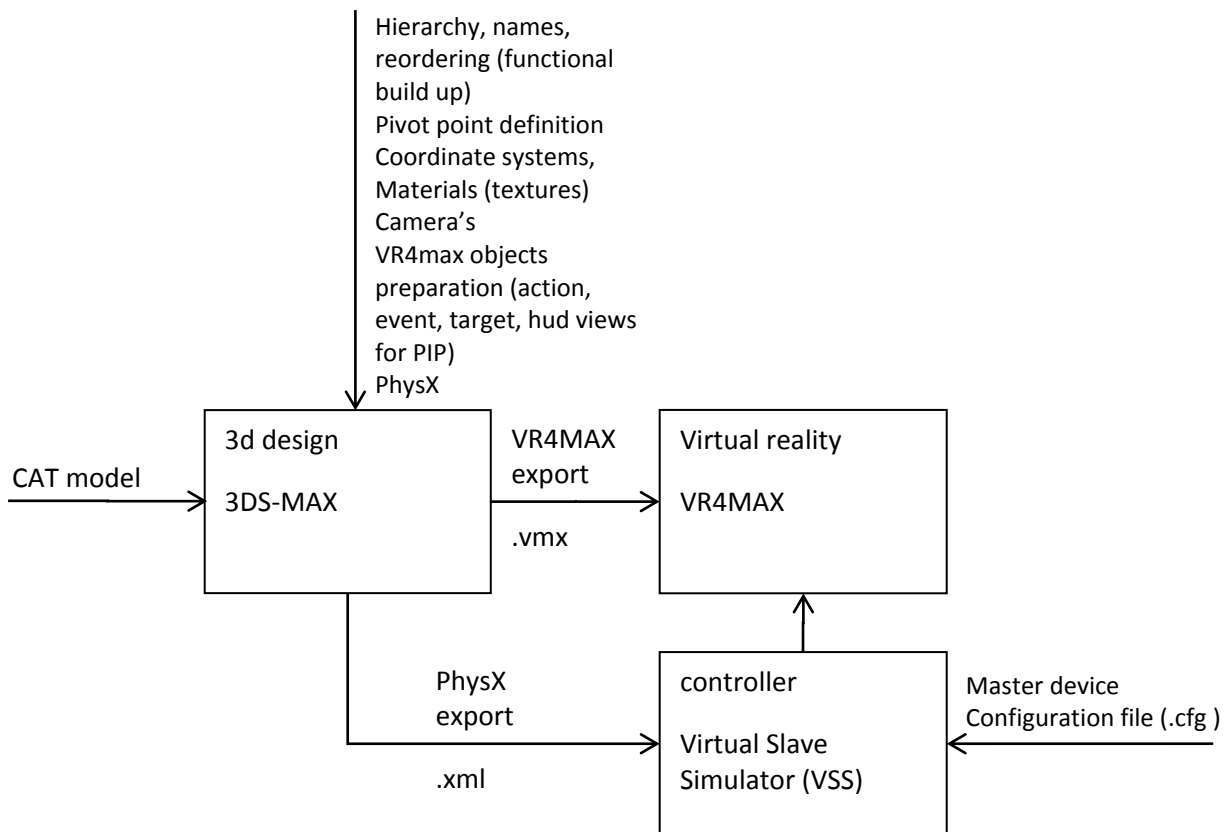


Figure 1 Flow diagram of the used software

A.2. Telemanipulator

The telemanipulator in this experiment consists of a Haption Virtuose 6D35-45 master device. The Slave is implemented a virtual reality simulation (the Virtual Slave Simulator) as shown in Figure 2. The controller between them is a position error controller (also known as PERR-control) which is

implemented in the Application Programming Interface (API) of the master device. The PERR controller obtains positions and velocity signals from the master and slave devices and calculated applicable forces for both devices. The visual representation of the environment is done via the renderer (VR4MAX) which obtains updated positions from the Virtual Slave Simulator. Note that haptic shared control is implemented within the Virtual Slave Simulator to easily access object data (e.g. the path) from the virtual environment.

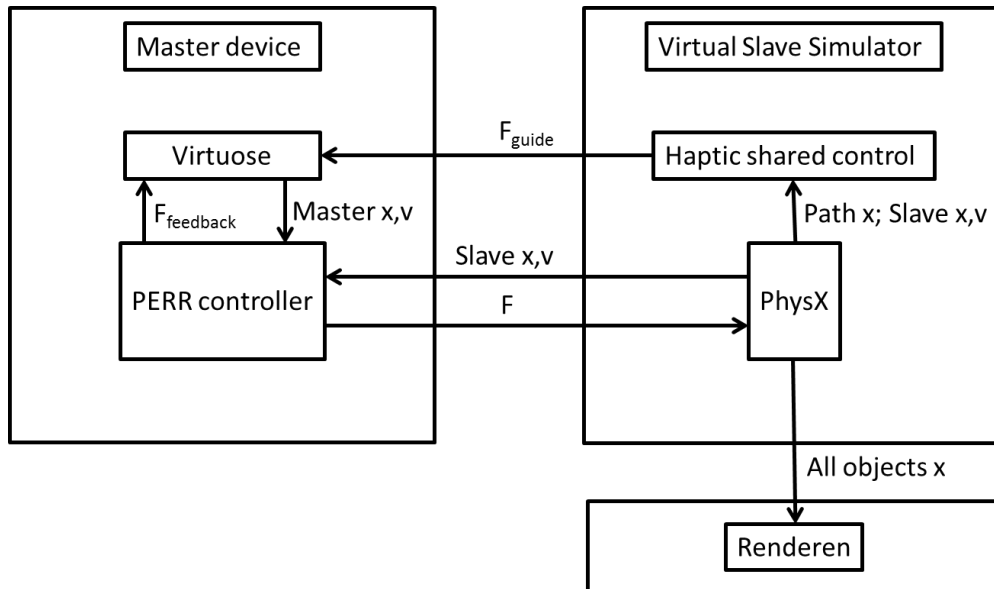


Figure 2: Overview of communication between hard- and software modules in the experimental setup

The PERR controller assigns automatically proper stiffness and damping parameters to provide high reflected force and stable behaviour. The controller parameters are calculated based on the simulation update frequency and the mass/inertia of the slave. These properties and the resulting controller parameters are shown in Table 1. The formula for the controller parameters is however not to be published.

The degradation of the transparency of the telemanipulator was mainly done in the controller (which allowed custom assignment of controller parameters), but also by increasing the slave's mass. Here transparency is measured as the master-to-slave force bandwidth (approximated with a one degree of freedom equivalent device) with the HapticAnalysis package by [8]. The results for force bandwidth are shown in Figure 3 (bandwidth is 15 [Hz] and 3 [Hz] respectively for high and low transparency). Secondly, the transparency was decreased by lowering the maximum output force and torque within the PERR controller. All parameters are shown in Table 1.

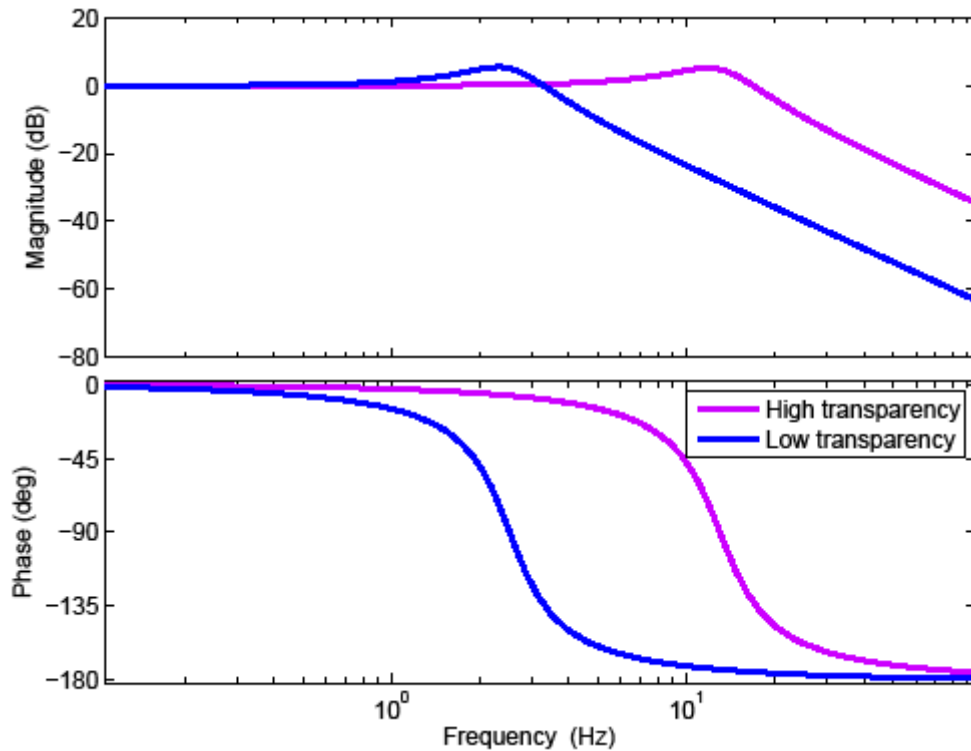


Figure 3: Force bandwidth performance of the telemanipulator in high and low transparency.

Table 1: Properties of the telemanipulator with high and low transparency

property	High quality	low quality
K_trans	2000 [N/m]	300 [N/m]
B_trans	14 [Ns/m]	10.5 [Ns/m]
K_rot	16.88 [Nm/rad]	3 [Nm/rad]
B_rot	0.12 [Nms/rad]	0.105 [Nms/rad]
Slave mass	0.3 [kg]	1.2 [kg]
Slave inertia	675[kgm ²]	2700 [kgm ²]
Max F	35 [N]	12 [N]
Max T	3.3 [Nm]	1.2 [Nm]

The Virtual Slave Simulator is a program in development. During this master's thesis I contributed several upgrades and bug fixes to this packages. Among them a fix that double the PhysX simulation time because it calculated the each simulation step in two sub steps. The most challenging fix was about a bug that generated haptic shocks in the output like shown in Figure 4. During an extensive search for this bug, the entire Virtual Slave simulator was cut down in pieces to divide the modules and conquer the bug. In the end the bug was a result of the accessing the Virtuose API from two different loops in the Virtual Slave Simulator. Probably the bug occurred due to simultaneous access of this API.

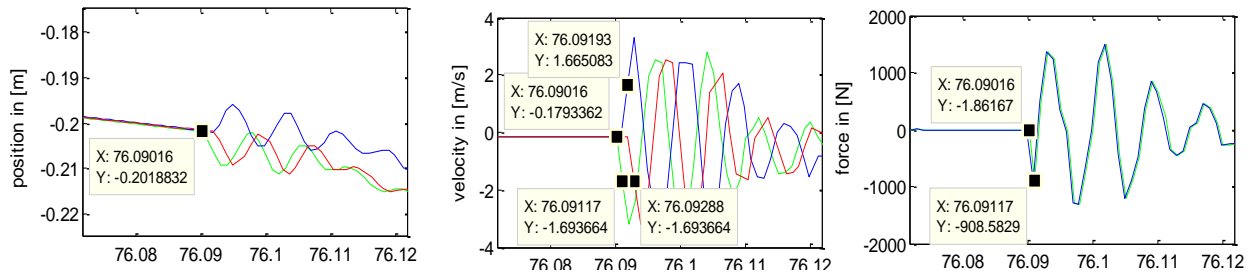


Figure 4: Shock that appeared during simulation as a result of a bug. The green lines represent information measured or set at the PhysX side of the Visual Slave Simulator. The red lines represent information send to the master device. The blue lines represent information obtained from master device.

A.3. Task and environment

The used task and environment for this experiment are related to ITER. The next sections: give background information on ITER; explain how the task was selected; and show how it is modelled.

A.3.1. ITER

This master's thesis project is performed in close cooperation with the Remote Handling Study Centre (RHSC) at FOM DIFFER. In that cooperation it is decided to perform this research in the content of ITER.

ITER is an experimental fusion reactor (see Figure 5) which mission is to prove that fusion can be a durable source of energy. Part of the proof is to show that ITER is maintainable in an effective and safe way. The maintenance is challenging because ITER components have a too high level of nuclear radiation and toxicity to be directly manipulated by hand. Full robotic automation is impossible due to the unpredictable nature of tasks [14] and issues like safety and responsibility. Therefore ITER maintenance will be done via human-in-the-loop teleoperation (In ITER terminology: remote handling maintenance). A part of the maintenance will be done in a Hot Cell facility.

The RHSC aims to provide effective Remote Handling maintenance scenarios for ITER. One of their activities is on the upper port plug: the Electron Cyclotron Heating Upper Port Launcher (see Figure 5). This plug holds the task with which the experiment was performed as will be explained in the next section.

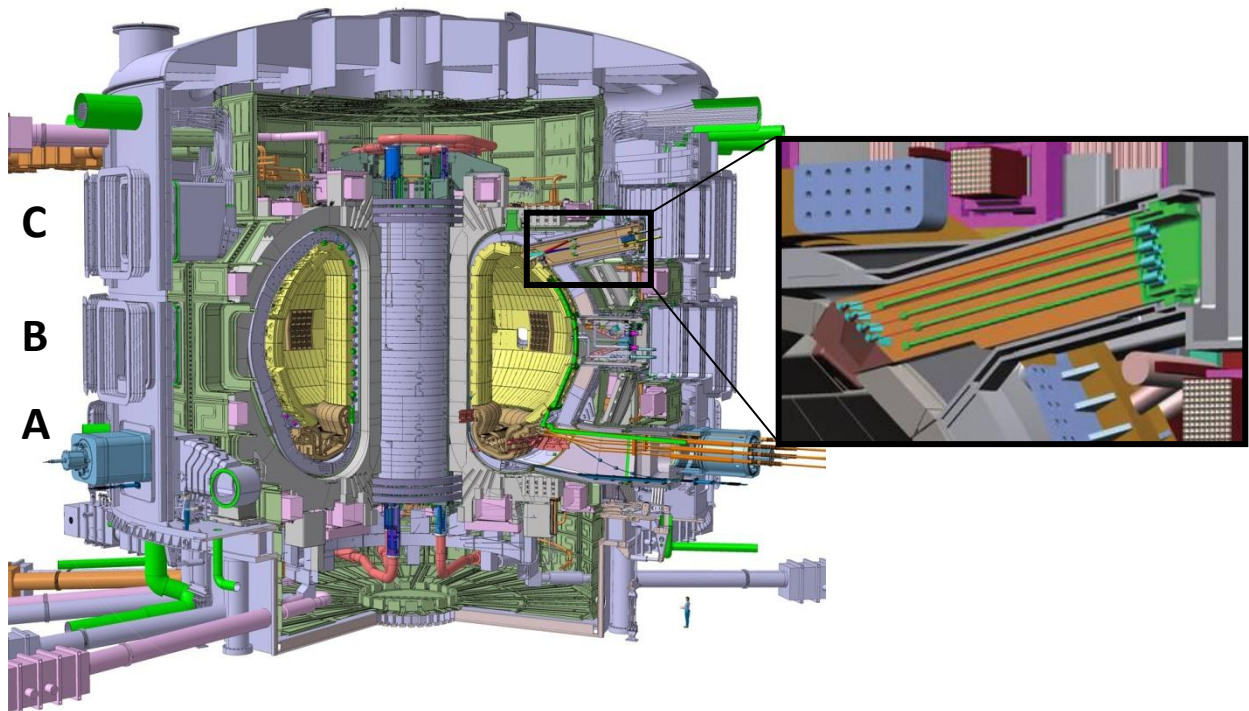


Figure 5: The experimental fusion reactor ITER with the location of the 'diverter ports' (A), the 'equatorial ports' (B), the 'upper ports' (C). Also shown is the Upper Port Launcher on the right. Note the blue man on the bottom showing the scale

A.3.2. Selection of the task

Not every task suits to find the effect of inaccurate haptic shared control and so this section analyses what task does the job. First of all the task must be related to ITER hot cell remote handling maintenance as this research is performed in the context of ITER. Secondly, human beings apply different control strategies for different types of motions, therefore the task should contain enough different types of motion. Types of motion for hard-to-hard contact environments can be classified in four types as identified by Wildenbeest [7]

- Free Space Movement (FSM)
- Contact Transition (CT)
- Constraint Translation Movement (CTM)
- Constraint Rotation Movement (CRM)

Thirdly, inaccuracies in the model have no effect on task performance and control effort if the task does not require to be performed accurately. Therefore the task should contain accurate movement. Finally the task should preferably be a nominal planar task to facilitate the analysis of the 3D 6 degree of freedom data.

An analysis of ITER maintenance task is made to find task that fulfil the demands from above. ITER remote handling tasks can be categorized in five groups as shown in Table 2. Of these groups it is expected that only assembly/disassembly tasks can fulfil all demands. Within this group four task, see Table 3, have been identified that will occur during maintenance.

Table 2: Five groups in which remote handling in which remote handling can roughly be divided.

Hot cell maintenance	FSM	CT	CTM	CRM	Accurate movement	Total
turn / relocate object	+	+	-	-	0	0
inspection	+	-	-	-	-	-
Take samples	+	0	0	0	-	0
Cleaning	+	0	0	0	-	0
Assembly / disassembly	+	+	+	+	+	+

Table 3 Four frequent occurring potential tasks in Hot Cell maintenance

Assembly \ Disassembly	FSM	CT	CTM	CRM	Accurate movement	Total
Pick gripping feature	+	+	-	-	0	0
Peg-in-hole (tool placement)	+	+	+	-	+	+
Place / remove object with help of aligning features	+	+	+	+	+	+
Bolt / unbolt objects	+	+	+	+	0	0/+

Tasks that fulfil the above demands are peg-in-hole tasks and placement of objects with aligning features as shown in Figure 6. The current research chose to use the peg-in-hole type task to applied inaccurate haptic shared control on. It is chosen as it also represents the placement and bolting task to some extent. A bolt runner needs to peg in the bolt and the aligning features need to peg in their holes.

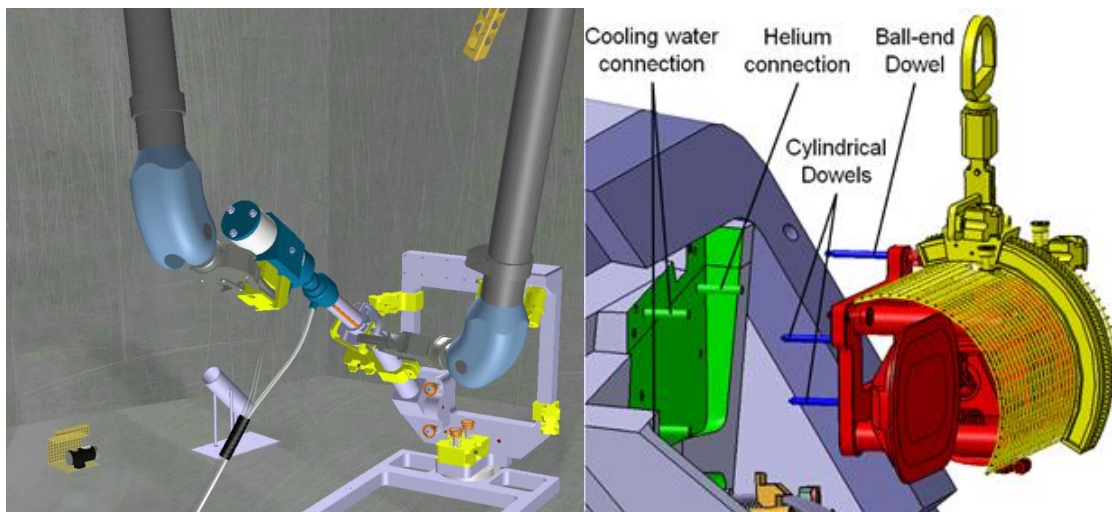


Figure 6 Left shows a peg-in-hole task with an inner tube welding tool during a Virtual reality simulation at the Remote Handling Study Centre. Right shows the placement of a subassembly with help of aligning features with Ball-end Dowel and Cylindrical Dowels.

A.3.3. Modelling the task

The welding tool is modelled and produced (as mock-up) at the Remote Handling Study Centre. The tool has the dimensions of 440x70x125 [mm] (LxWxH), a tip diameter of 30 [mm] and a mass of 1.8 [kg]. The design (shown in Figure 7) is based on the guidelines in the ITER Remote Handling Code of Practice. The CAT model is extended in 3DS MAX with 37 convex meshes that form the concave mesh which is used by PhysX for the dynamics simulation (lower right in Figure 7). This “decomposition” of concave shapes into convex meshes is a common approach for which well written algorithms exists like the Hierarchical Approximate Convex Decomposition (HACD) [9]. However; the welding tool consists of relative simple shapes which were best modelled by hand.

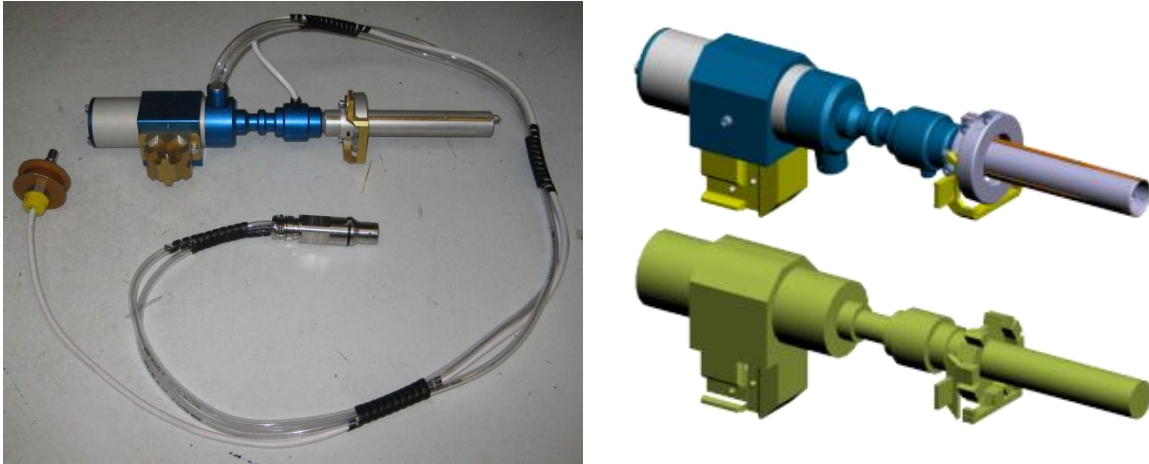


Figure 7 Left hardware mock-up of the welding tool. Upper right: the CAT model of the welding tool. Lower right: PhysX model of the welding tool.

The hole (or tube as in Figure 8) has a diameter of 30.06 [mm]. It contains 16 segments for the outside to approximate its circular shape. The inside of the tube contains 8 segments: Four in the front and four in the back of the tube. The segments have a rotational separation of 90 degree. The inside of the tube is modelled this way to measure “contact forces” inside the tube in up/down and left/right direction. This contact force is the force that PhysX applied on the tool and tube to prevent them from violating the so-called non penetrating constraint.

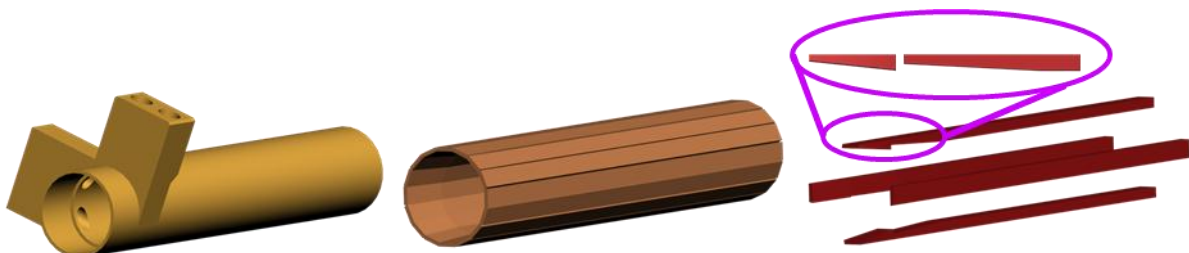


Figure 8 The tube (left), its meshes for the outside of the tube (middle) and its meshes for the inside of the tube (right)

An important environmental property in a physical rigid body simulation is the skin width. This skin width is the overlap that objects may have before the simulation starts to counteract the object penetration. A certain overlap is needed to detect collisions. This skin width is set to 0.02 [mm]. This means that the welding tool has a total clearance of $0.06+2*0.02 = 0.1$ [mm].

Further the environment of the task is model with a selected number of objects as shown in Figure 9.

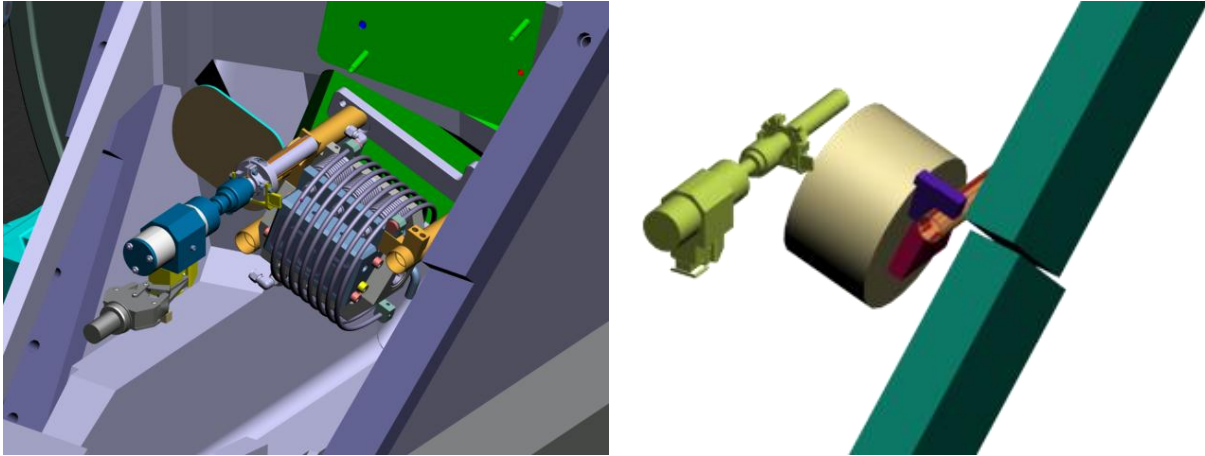


Figure 9 Overview of the model for the entire task.

A.4. Short guideline to run the setup

The setup consisted of two PCs. The first PC, the 'Virtuose PC', runs on a Linux real-time operating system and is connected to the virtuose. The second PC, the "work station" is a Windows 7 based PC on which the virtual models and Virtual Slave Simulator are build and run and from which the Virtuose pc is operated.

To start, turn on both PCs. After some time the Virtuose PC will initialize a calibration of the Virtuose device at which point this device should be free to move to prevent incorrect calibration. The Virtuose PC will then wait for instructions.

On the work station drag the appropriate configuration file (e.g. _SC_GT__000.cfg) onto the Virtual Slave Simulator executable. That will start the simulation/renderer/master automatically while the start-up progress is shown in a command interface as shown in Figure 10.

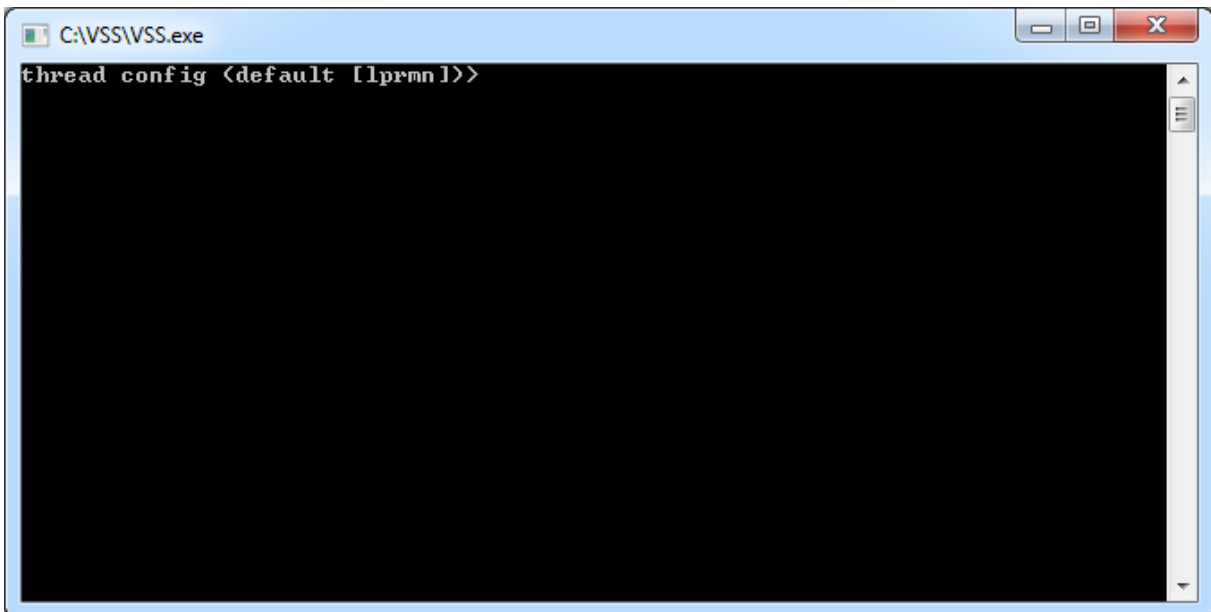


Figure 10: start-up interface of the Virtual Slave Simulator interface.

Finally data will be saved automatically if the welding tool is inside the tube and rotated by 90 degree counter clockwise. The file name is assigned by the Virtual Slave Simulator with a prefix “DL_”, a specified name in the configuration file (e.g. SC_GT__000) and the current date and time (e.g. DL_SC_GT__000_2012-07-30_17-01-36). By default the data is in a binary file (.bin) accompanied an index of saved variables in a comma separated value file (.csv) file.

A.5. Short guide to setup a configuration file

The Virtual Slave Simulator (VSS) should be instructed with commands to start-up the PhysX simulation, the renderer and the master device. These commands can be typed into the command line window, but it easier to write a configuration file that contains the instructions. Basic commands are explained below here and can be used to run a basic simulation with haptic shared control:

- lprv : Tells the VSS to create a “Logger”, “Physics instance”, “Renderer instance” and “Virtuose (Master) instance”. This sequence can be changed to for example “pr” to only work with the PhysX scene and renderer. Note that the Logger is the part of the programme that reports actions from the VSS to the user.
- l start : Tells the VSS to start the specified l, p, r or v. Usually the only the Logger is started at the beginning of the file for debugging purposes. Other instances are started at the end for stability reasons.
- r load : Loads a .VMX file for the Renderer instance that is specified after this command (e.g. c:\scenes\SC.VMX)
- p load : Loads a .XML file for the Physics instance that is specified after this command (e.g. c:\scenes\SC.XML)
- p config : Specifies configuration settings for the Physics instance:
 stepms=1 : Sets the update frequency in milliseconds
- m config : Specifies configuration settings for the Master instance:
 view=Rtu : Sets the conversion of the orientation difference between the master device and the virtual environment
 stepms=1 : Sets the update frequency in in milliseconds
 actor=Hand : Defines the actor name from the Physics simulation that should be actuated by the master device.
- sc config : Specifies configuration settings for haptic shared control
 object_name=Hand Defines the actor name from the Physics simulation that should be guided.
 on_master=true : Tells the controller that forces should be applied on the master. (by default guidance is applied on the slave, this was more stable and easier during debugging)
- sc path : Specifies configuration settings
 for haptic shared control path
 path_name=SC_path_0_ Defines a string with which the controller can look in the Physics simulation to find path elements.

The file should end with sc start, r start, m start, m starts and p start. There are other parameters like stiffness and damping of the assistance and master-slave connection, but these have reasonable default values.

A.6. Short guide to setup a shared control path

Shared control paths can most easily be modelled with static PhysX objects positioned along splines. The spline (a Bezier curve) is used to define a smooth trajectory for the path. With a script the spline is automatically divided in equal segments of which the coordinates are retrieved. At those coordinates –equally space along the spline- the path point objects will be located. The script provides several options to orientate the objects (e.g. point in the direction of: the path, the x-axis, a point in space or another spline). Figure 11 shows how such a path looks like in 3DS MAX. The script can be found at the USB-stick delivered at the BioMechanical Engineering depository.

The path objects are given PhysX properties so that their location and orientation can be retrieved in the Virtual Slave Simulator. As that is the only purpose of the objects, they are made static and non-colliding with other objects.

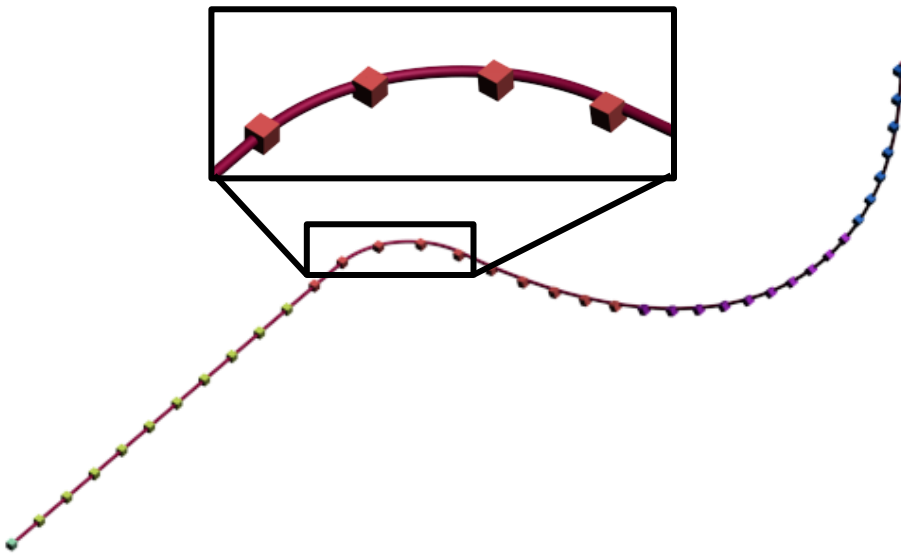


Figure 11: Impression of a path of equally spaced objects (boxes) build from a spline. This path segment is used for the Constraint Rotational Movement during the actual experiment.

B. Shared control designs & implementation

B.1. Design of haptic shared control

Haptic shared control has many different designs to support the operator for optimal control actions. Examples are the virtual fixtures proposed by Rosenberg [10], which work as a virtual ruler assisting a tele-manipulated peg-in-hole task. Another example is the continuous haptic guiding during car following [11] and curve negotiation [12] proposed by Abbink and Mulder. The current study focuses on the implementation as used by Boessenkool [6]

The implementation of Boessenkool is promising over a wide range of tasks in hard-to-hard contact environment, but is limited to a 2 dimensional 3 degrees of freedom set-up. This research proposes an extension to the implementation of Boessenkool in 3 dimensional space with 6 degree of freedom. The first section explains the baseline design (as background information). The second section gives detailed information on the rotational feedback of haptic shared control in 3 dimensions. The third section shows the weight compensation pilot results. And the final section gives the finally design and implemented code.

B.1.1. Baseline design

Haptic shared control is an intelligent system which calculates ideal control actions based on sensor and model information about the slave and environment. The ideal control actions are presented as a force on the master device to guide the operator to the optimal path, but the operator can always resist the guiding forces if he/she does not agree with the system. A general high-level scheme of the proposed shared control is illustrated in Figure 12.

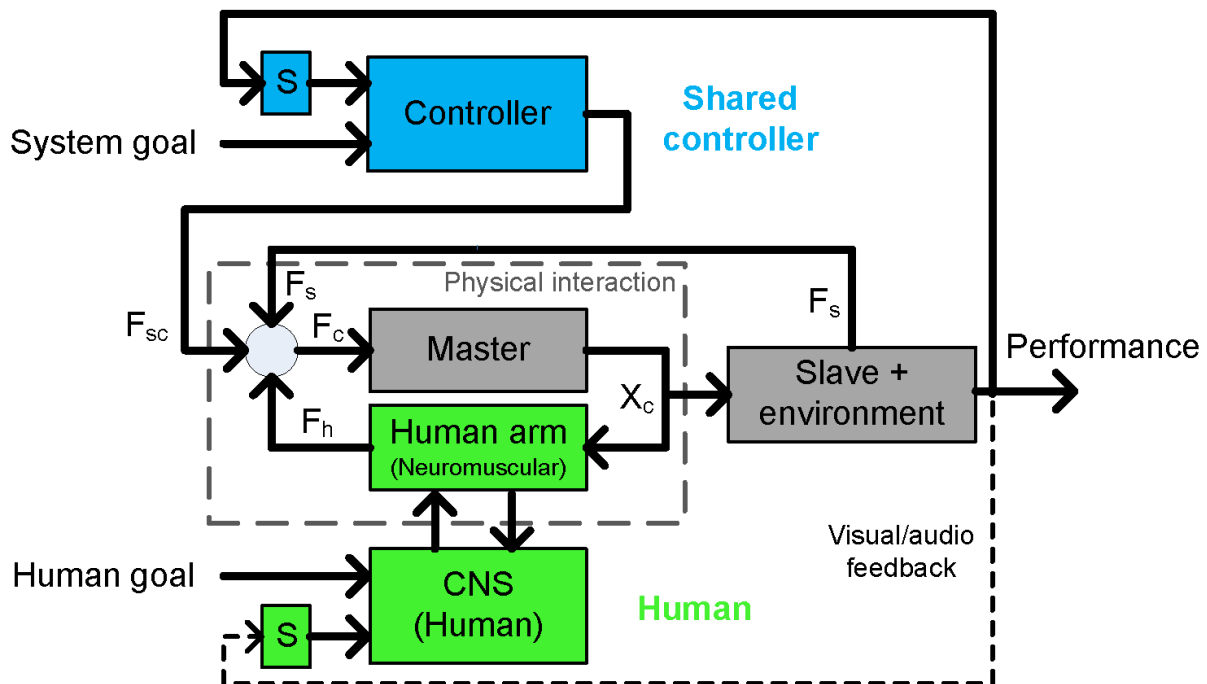


Figure 12 A schematic representation of tele-manipulation with haptic shared control. The input to the master (F_c) is influenced by the feedback forces from the slave (F_s) and the force from the human (F_h) and the shared control system (F_{sc}). The guiding forces are continuously updated based upon sensory feedback (S) from the environment. [from Boessenkool [Boessenkool 2012]]

Boessenkool [6] proposed the following implementation for four fundamental subtasks [7] (see Figure 13):

Free Space Movement.

The position guiding force is based on the look ahead path error with a stiffness of 120 [N/m]. The look ahead path error is defined as the error to the path at an estimated position at 0.1 [s] in the future based on the velocity vector. Guiding forces are not applied along the path. The guidance of the orientation increases linearly to a stiffness of 0.5 [Nm/rad].

Contact Transition.

The position is guided with a stiffness of 120 [N/m]. An artificial damping prevents hard collisions. The tool orientation is guided with a linearly increased stiffness of 0.5 [Nm/rad].

Constrained Translation Movement.

The tool is guided to the right orientation with a stiffness of 0.5 [Nm/rad]. A snap-feature attracted the tool (like a magnet) if it is close to the target.

Constrained Rotation Movement.

The guiding forces act only perpendicular to the direction in which force is applied. The snap-feature is active to ensure that the spanner stays on the bolt head.

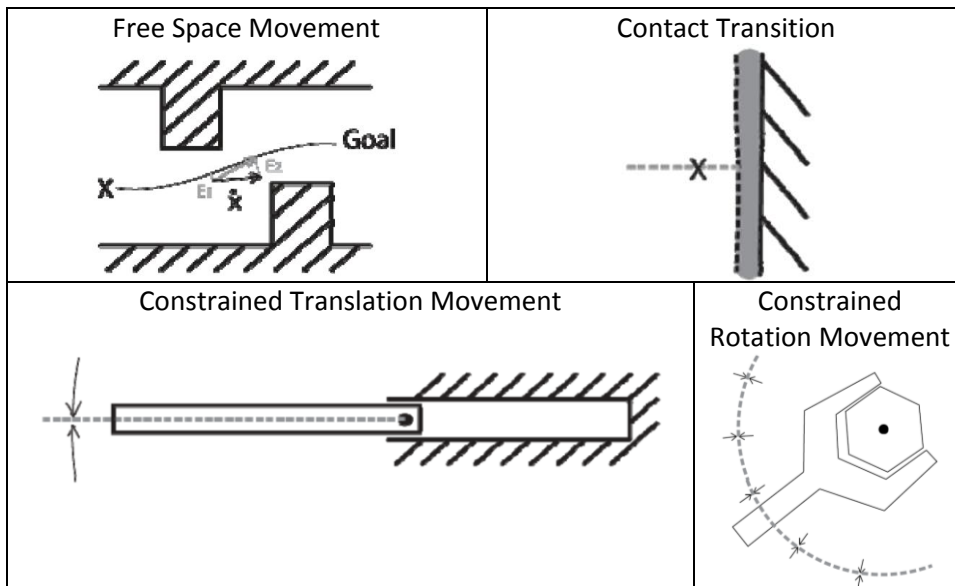


Figure 13: The essence of the implementation in every subtask.

B.1.2. Assistance in 3 independent rotations (swing twist decomposition)

Parameterizing rotations for intuitive feedback must be done carefully. A common problem in rotations is gimbal lock (loss of rotational degrees of freedom) due to singularities. Then there is the challenge to find the difference between two rotations in such a way that rotations can be constrained by independent stiffness's. For example the tip of the welding tool suffices no rotational feedback along the rotational axis. In that case there should be no orientation of the tool in which the tip rotation receives force feedback.

Intuitively, the orientation (R) of the welding tool can be composed of a swing component that controls the direction at which the tip points and a twist component that lets the tool rotate about the tip's rotational axis. [1]. This may be written as:

$$R = R^{Twist} R^{Swing}$$

The twist is easily parameterized by one angle around an axis with respect to a reference orientation. The reference orientation in the current study is defined as the orientation of the haptic shared control path. The x-axis of this path points in the tool tips direction and the z-axis in the upward direction as shown in Figure 14. The twist can be computed via angle axis or quaternions (also by Euler angle, however they hold a singularity at the reference orientation [2]). This research adopted the method of quaternions to calculate the twist angle τ [3]:

$$\tau = 2 \cdot \text{atan2}(q_x, q_w)$$

Where the q (q_x, q_y, q_z, q_w) is the relative quaternion between the tool and desired orientation.

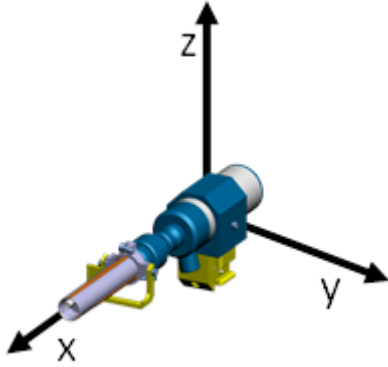


Figure 14: Tool and path orientation for swing twist decomposition

Swing can be explained most easily as an angle axis that lays in the y-z plane. The individual swing_y and swing_z can be calculated with quaternions as follows [3]:

$$\beta = \text{atan2}\left(\sqrt{q_y^2 + q_x^2}, \sqrt{q_x^2 + q_w^2}\right)$$

$$\text{swing}_y = \frac{2 \cdot \beta}{\sin \beta} \cdot \left(\cos\left(\frac{\tau}{2}\right) \cdot q_y - \sin\left(\frac{\tau}{2}\right) \cdot q_z\right)$$

$$\text{swing}_z = \frac{2 \cdot \beta}{\sin \beta} \cdot \left(\sin\left(\frac{\tau}{2}\right) \cdot q_y + \cos\left(\frac{\tau}{2}\right) \cdot q_z\right)$$

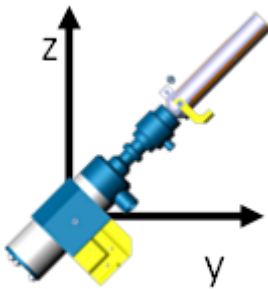


Figure 15: Example in which the tool is located at an 45 degree angle between the y and z axis while it's desired orientation is the x axis (which points outside the paper). Here the twist is 45° (around tooltip), the swing_y is 63.64° (around path axis) and swing_z is -63.64° (around path axis).

As an example imagine the tooltip to be pointing 45° upward like in Figure 15. The expected total swing is 90°. The swing_y and swing_z angle are 63.64° and -63.64°. These values add up to 90° ($\sqrt{63.64^2+63.64^2}$) as expected. Note that the twist (which is a bit harder to imagine) is 45°.

The above formula works only as long as both the y and z torsion stiffness are equal. Reconsider the above example and that now the tip should be free to rotate around the z axis. Then its stiffness is zero and the tool should only rotate by 45° around the paths x axis, while it now rotates by 63.64° around the paths y axis. To compensate for this the author derived the following method:

- Check which swing axis is more stiff (y or z)
- If e.g. y, then project the tooltip vector on the xy plane
- Calculate the angle α with the x-axis
- Fictively rotated the tool by $\alpha \cdot (K_y - K_z) / K_y$ around the z axis towards the x axis. Here K_y and K_z are the torsion stiffness of respectively the y and z axis.
- Calculate the swing components with the fictive tool orientation.
- Rotate the swing component back to original tool orientation.

B.1.3. Weight compensation welding tool

Gravity causes interfering translational errors in the model due to the weight of the welding tool and the operator arm. This is because the guidance acts as a spring perpendicular to the path to indicate a zero force reference level. The additional weight causes the zero reference line to shift down the real path as the “path spring” is stretched by the weight of the tool and the arm of the operator. Therefore the weight of the tool should be compensated for. Distorting the weight-inertia relation causes misinterpretation of the inertia [4]. A pilot test is done to get an indication of the effect of zero weight on performance.

The effect zero weight was tested for vertical translation as the welding tool would be moved in that direction. The misinterpretation of inertia might result in an excessive overshoot or settling time and thus negatively influence the performance. The task was to repeatedly translate a pack of milk (1 [kg]) over a random distance (10 to 200 [mm]) in vertical direction. A weight of 1 [kg] was taken as it is in the same order of size as the welding tool (1.8 [kg]) and is the maximum continuous force that the Virtuoso can represent. A visualization (and dimensional size and inertia) of a milk pack is chosen to aid the participants with the step size of the task.

It was expected that people would obtain a certain baseline performance after a couple of trials for normal weight. For zero weight objects, this baseline will also be obtained although it takes longer. (see Figure 16)

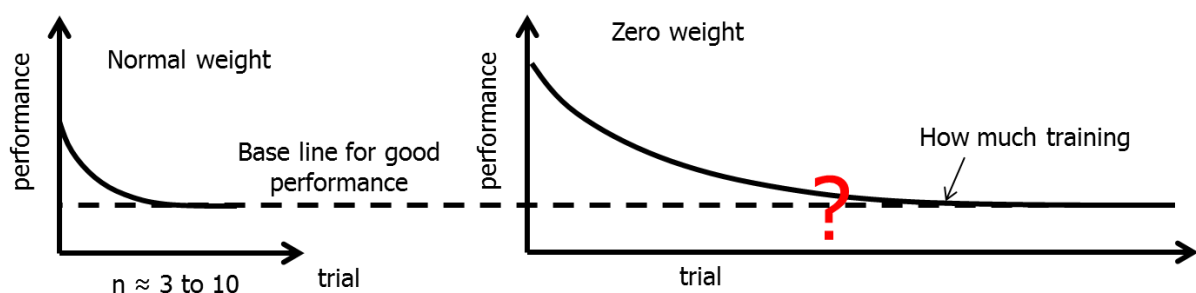


Figure 16: Expected results from the zero weight tests. The performance level obtained in the normal condition will eventually also be obtained in the zero weight condition.

Experimental setup

The task is performed on the same setup as the main experiment (see Figure 17). In the virtual reality the physical representation of the milk pack was free to move anywhere. The graphical representation was constraint at a fixed distance from the camera and had a vertical reference line (green) attached to it. At the same distance from the camera a red line jumped every 2.5 seconds to another random vertical position (minimally to 10 [mm] up or down, maximally till the border of a 300 [mm]range). The red line jumped 35 times per session.

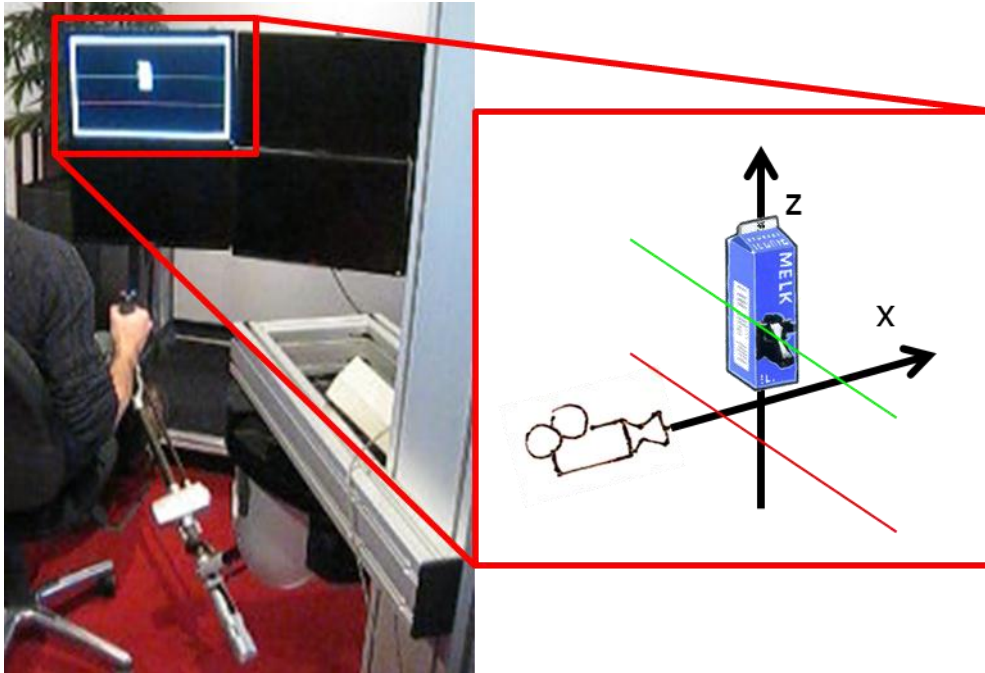


Figure 17: Impression of the experimental set-up with: the Haption Virtuose 6D35-45 in upside-down configuration; visual feedback from VR4MAX; and an impression of the task in the red boundary.

Experimental design

The conditions for the task were normal (1 [kg]) and zero weight (0 [kg]). The normal weight condition was performed for four sessions in a row (to ensure that participants learned the device and object dynamics) and the zero weight condition for three sessions in a row. The task instructions were to follow the red reference line as accurate as possible.

Subjects data acquisition and metric

Four subjects performed the test. The measured data was the time, the reference level and vertical position of the milk pack at a sampling rate of 1 [kHz]. The performance was measured as overshoot and settling time (the time required to reach and stay within a range of 5% of the final value).

Data analysis

The gathered steps were normalized to compare them (see in Figure 18 and Figure 19). The normalized steps were evaluated with the matlab command “stepinfo”.

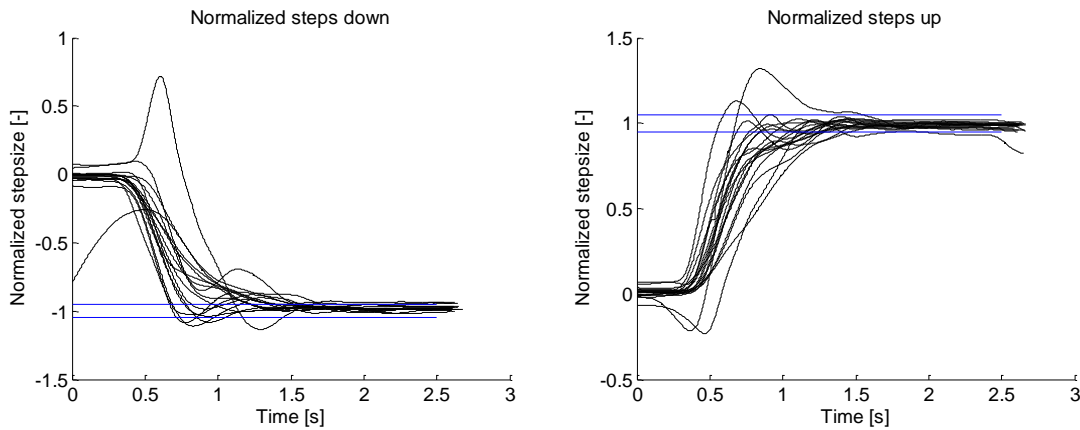


Figure 18: The normalized steps for the last (fourth) session of the normal weight condition of one participant. Steps are divided in downward steps (left) and upward steps (right).

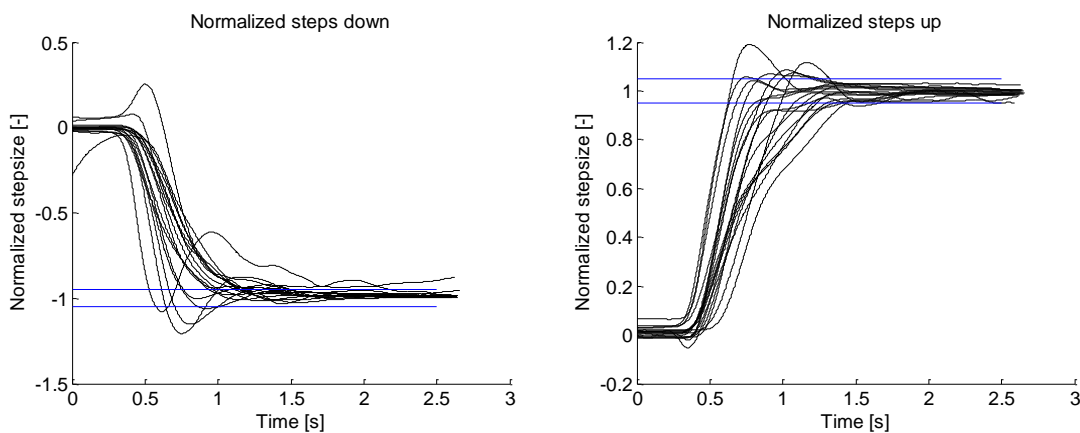


Figure 19: The normalized steps for the last (third) session of the zero weight condition of one participant. Steps are divided in downward steps (left) and upward steps (right).

Results

The settling time and overshoot value for one participant can be found in Figure 20 and Figure 21. The last session of each condition is divided in two sets which are averaged. These eight means (4 participants, 2 per participant) are represented in the boxplots in Figure 22 for the settling time and Figure 23 for the overshoot.

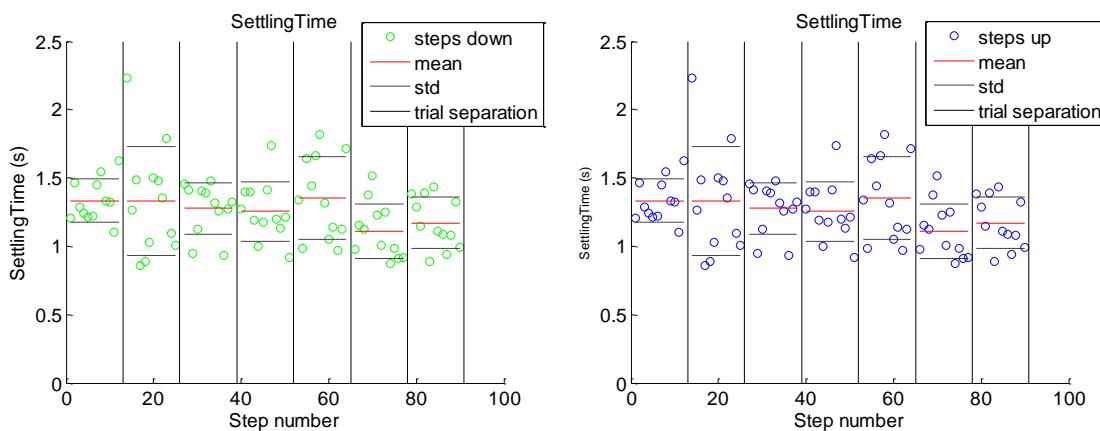


Figure 20: Analysed results on settling time for steps down (left) and up (right) from all seven sessions from one participant. The first four columns represent the data of the normal weight condition and the last three columns represent the data of the zero weight condition. Non-settled steps are excluded.

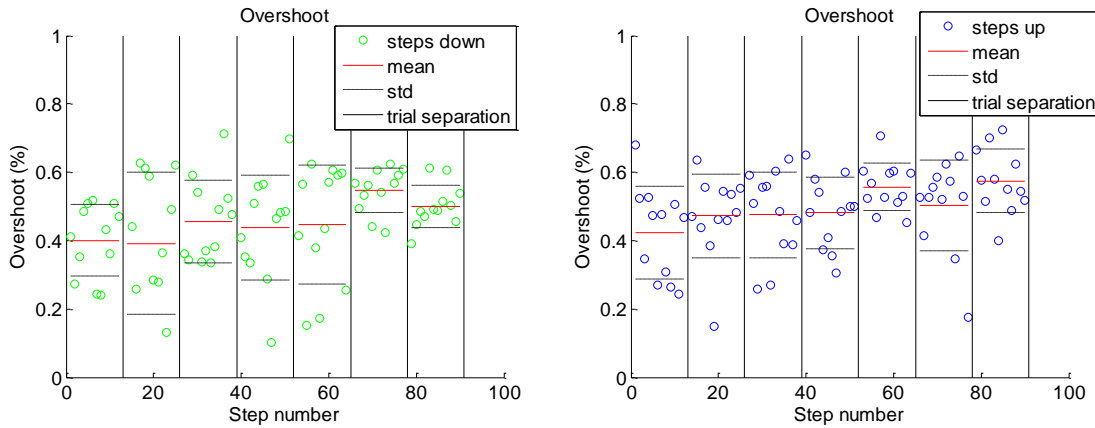


Figure 21: Analysed results on overshoot for steps down (left) and up (right) from all seven sessions from one participant. The first four columns represent the data of the normal weight condition and the last three columns represent the data of the zero weight condition. Non-settled steps are excluded.

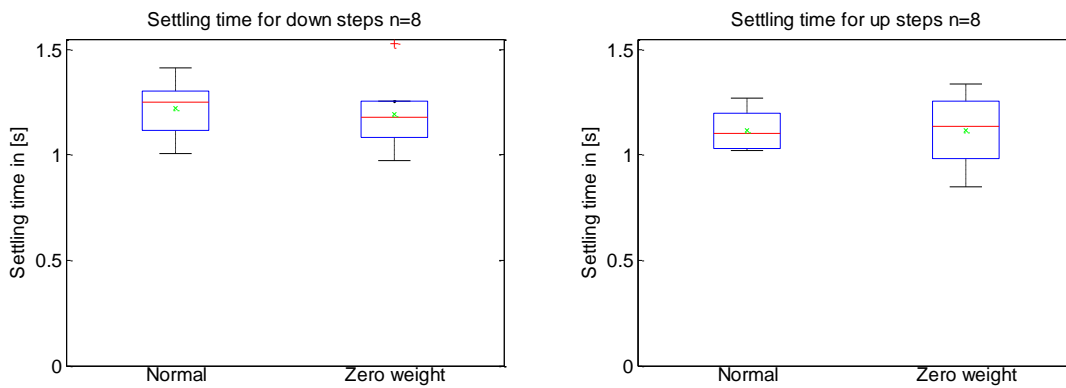


Figure 22: Results for the settling time for the last (fourth) session with normal weight and the last (third) session with zero weight. Data is presented in boxplots (n=8) which shows the median (red), the 25th and 75th percentiles (blue), the most extreme data points (black) not considered outliers (red +). Also shown is the mean (green x)

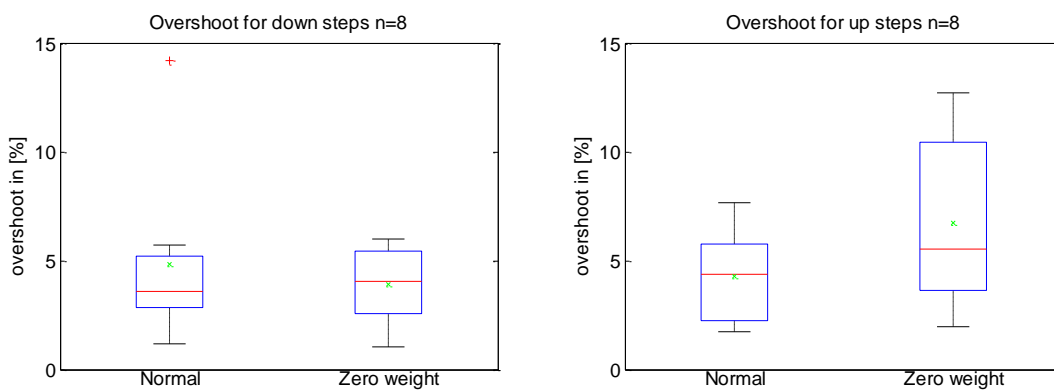


Figure 23: Results for the overshoot for the last (fourth) session with normal weight and the last (third) session with zero weight. Data is presented in boxplots (n=8) which shows the median (red), the 25th and 75th percentiles (blue), the most extreme data points (black) not considered outliers (red +). Also shown is the mean (green x)

Discussion

The results over the several sessions provide no evidence of a clear learning effect. The settling time results provide no evidence of a trend between normal and zero weight for both steps up and down.

The overshoot results provided evidence there is a trend between normal and zero weight for moving up. But the welding tool is supposed to move down to the tube, for which there is no evidence that zero tool weight affects performance.

B.1.4. Weight of operator arm

In a preliminary study it was found that operators rest their arm on haptic shared control. The arm weight caused translational offset up to 1 [cm] just before the tube. With the offset the “composition” guiding force can be estimated via the path stiffness. The path stiffness is 500 [N/m], meaning that the upward force is about 5 [N]. 5 [N] is about one third of the weight of the combined hand and forearm (16 [kg] [13]). The other two thirds is carried by the upper arm.

To show the above findings, the tooltip and path positions are plotted in Figure 24, Figure 25 and Figure 26. Note that for this preliminary study: the path was not optimized; the tip was guided instead of the gripper; and the guidance was also applied on the slave, not on the master. The path was far from optimal as it had a corner between a vertical path and a more horizontal path. The corner was not treated well by the algorithm that fades over between paths. This caused the participant to sink far below the path after the fade over. But the main point is that the participant gradually sinks deeper into the path after that he recovered quite well from the fade over.

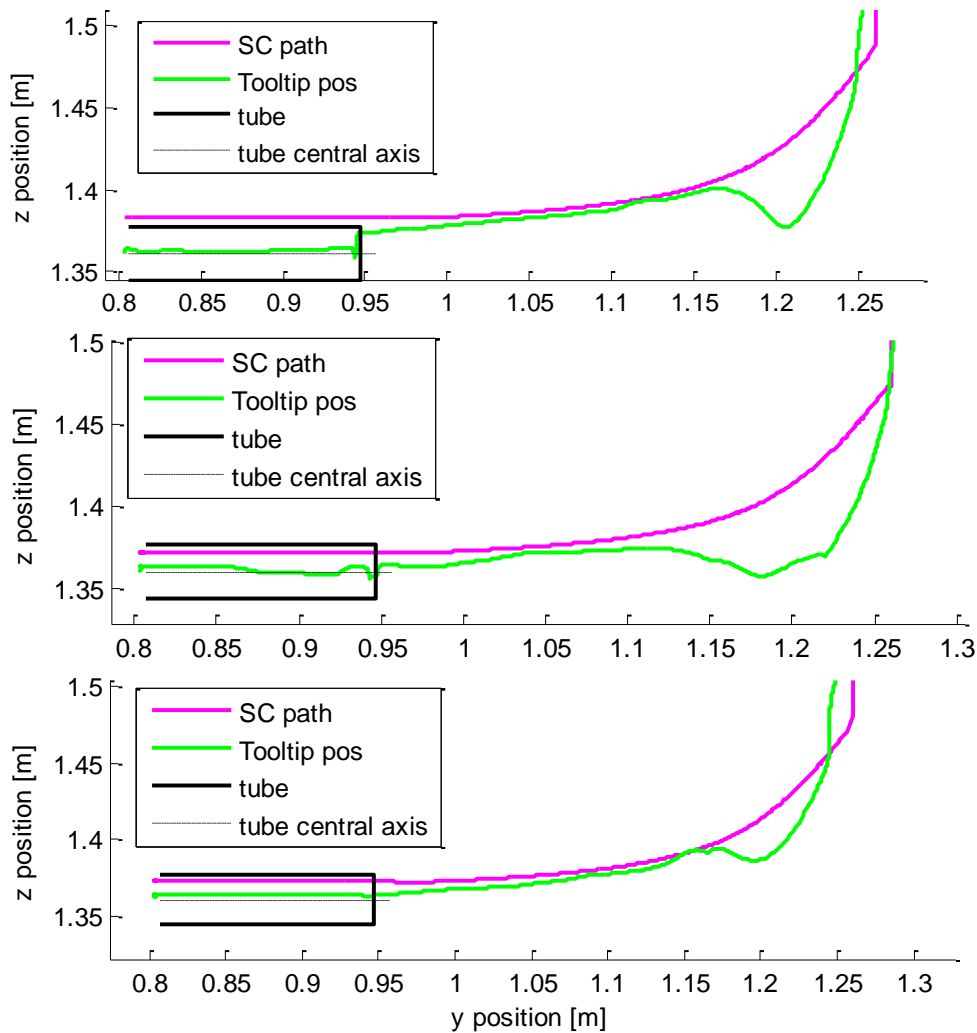


Figure 24: Tooltip position (green) and path (purple) with positive errors during the pilot test. Here the tool moves from the right to the left.

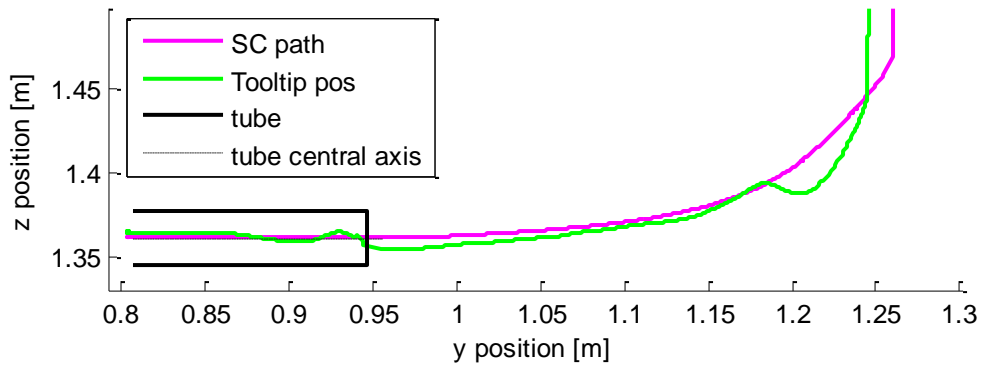


Figure 25: Tooltip position (green) and path (purple) with zero errors during the pilot test. Here the tool moves from the right to the left.

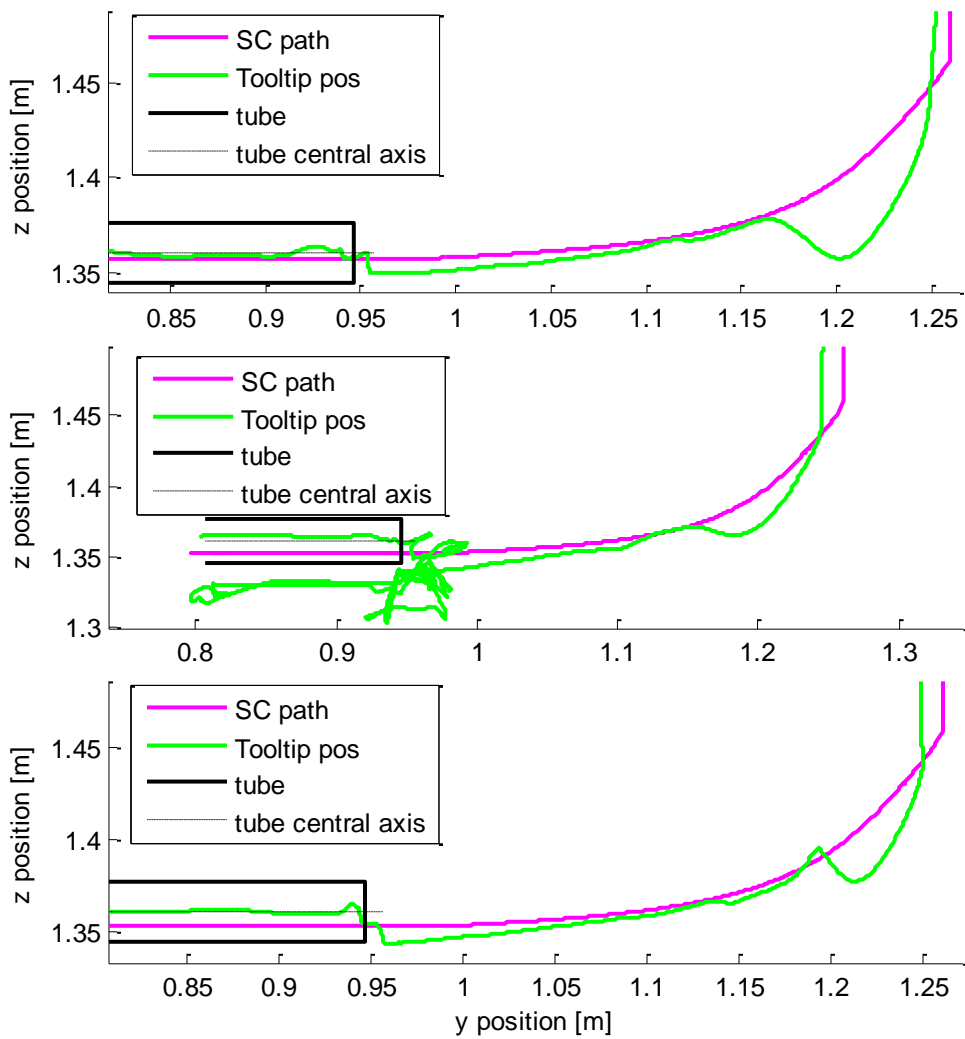


Figure 26: Tooltip position (green) and path (purple) with negative errors during the pilot test. Here the tool moves from the right to the left.

To prevent participants to interfere in the guiding error with their arm weight, it had to be compensated for. This compensation is a 9 [N] upward force (about half the arm weight [13]) on top of the normal guiding forces. This force is only applied if the tool sinks below the path.

The arm weight compensation force could not be applied continuously as the intentions of the operator are unknown. He/she might strive to minimal effort (rest on the guiding) or feel carefully what the guiding suggests to do (carry their own weight). Knowing that, it can be seen that continuous weight compensation can result in positive errors or counter intuitive guiding:

- Positive errors arise with continuous weight compensation on horizontal paths when the operators carry their own weight. In that case the guidance must counteract itself.
- Counter intuitive guiding arise with continuous weight compensation on vertical paths when operators carry their own weight. In that case operators get the sensation of being pushed up.
- Counter intuitive guiding arise even more with continuous weight compensation in a path that bends from horizontal to vertical. When own operators carry their weight they feel normal guiding for starters but are pushed up when the path starts to tilted. At that moment the path guiding forces (that act only orthogonal to the path) do not compensated the entire weight compensation anymore.

To show that the arm compensation worked, and might have affected the results, Figure 27 and Figure 28 give some shared control force data of the vertical detection during Constraint Translational Movement. Figure 27 shows the time traces for guidance force while the model had a +17.5 [mm] error. The traces show that both the guiding ($300 \text{ [N/m]} * 17.5 \text{ [mm]} = 5.25 \text{ [N]}$) and the arm weight compensation (9 [N]) contribute to the total guiding force (14.25 [N]). Figure 28 shows the time traces for guidance force while the model had a -17.5 [mm] error. The traces show that only the guiding contributed to the total guiding force (5.25 [N]). Although this asymmetry did not reflect in the final results of the test, it might have affected them. Still note that Figure 27 and Figure 28 do not show how much arm weight rested on the master device.

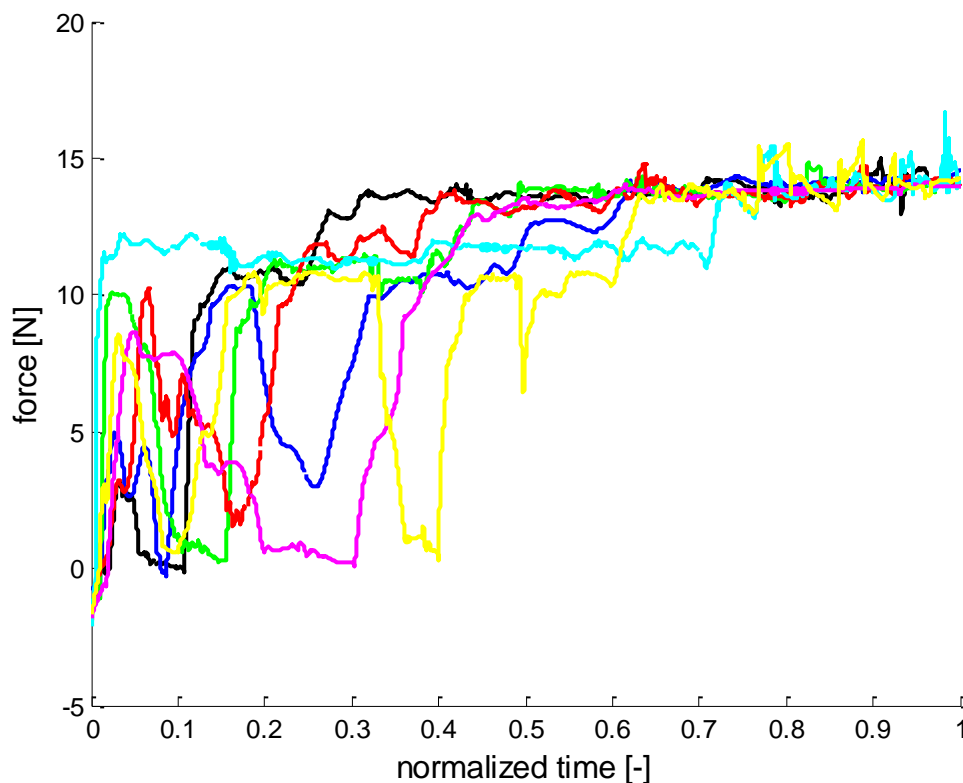


Figure 27: Normalized time trace of the guiding force in the z (vertical) direction during guidance with a +17.5 [mm] error. The data represent the 7 repetitions of one participant.

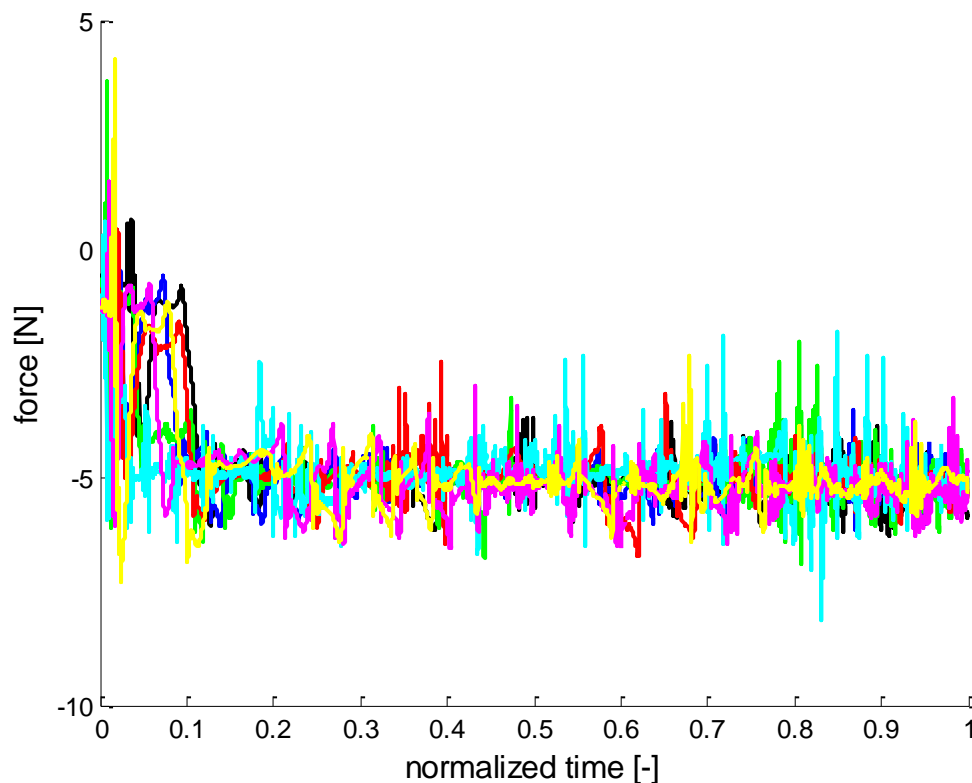


Figure 28: Normalized time trace of the guiding force in the z (vertical) direction during guidance with a -17.5 [mm] error. The data represent the 7 repetitions of one participant.

B.2. Implementation of shared control

The implementation of the shared control system will be discussed in more detail in this section (the full c++ code can be found at the USB-stick delivered at the BioMechanical Engineering depository). This implementation is added as a class in the Virtual Slave Simulator (the Virtual Slave Simulator code will not be shown or discussed).

The Shared_control class holds two subclasses which are presented below:

```
class Shared_control {
class Path{          // holds all information about paths
public:
    Path()          // constructor with default values
    : name(""), path_points(0) {
    trans_k[0] = 10; trans_k[1] = 10; trans_k[2] = 10;
    trans_k_end[0] = 10; trans_k_end[1] = 10; trans_k_end[2] = 10;
    trans_zeta[0] = 0.5,trans_zeta[1] = 0.5,trans_zeta[2] = 0.5;
    rot_k[0] = 10; rot_k[1] = 10; rot_k[2] = 10;
    rot_k_end[0] = 10; rot_k_end[1] = 10; rot_k_end[2] = 10;
    rot_zeta[0] = 0.5,rot_zeta[1] = 0.5,rot_zeta[2] = 0.5;
    path_force = 0.0;}

    std::string name; // name of path
    NxCVec3 trans_k; // stiffness at start of path
    NxCVec3 trans_k_end; // stiffness at end of path
    NxCVec3 trans_zeta; // damping of path
    NxCVec3 rot_k; // stiffness at start of path
```

```

    NVec3 rot_k_end;           // stiffness at end of path
    NVec3 rot_zeta;           // damping of path
    float path_force;         // force along direction of path

    Actor_vector path_points; // vector per path with all path points
};
typedef std::vector<Path> Paths_vector; // contains all paths

class Protect{              // holds all information about artificial damping in Contact
                            Transition
public:
    Protect()               // constructor with default values
    : p_protect(0), name(""){
    trans_c = 200.0f; rot_c = 0.2f ;}

    NActor      *p_protect;
    std::string name;
    float trans_c;
    float rot_c;

}_protect;
}

```

The class path is used to save information for individual paths. In this way paths can easily be made and adjusted for different subtasks. A path is made when the user sends a name to the shared controller to identify paths with. For this name (e.g. "sc_path_1_") the controller will look in the current PhysX scene for objects (NActors) with a similar name (e.g. "sc_path_1_001", "sc_path_1_002", ...). The pointers of such actors are saved in alphabetic order in the path_points vector for that instance of the class Path.

The class protect is used to hold information for the artificial damping in contact transition. The approach is to model a larger object (the protector) around the welding tool which does not collide with but reports on collision with other objects. The controller knows that on contact with the protector it should initiate the artificial damping. Note that artificial damping was removed from the test setup before this code was perfected.

The Shared_control class holds several functions of which the most important is handle_shared_control(). It is called after a PhysX update to calculate new guiding forces with the new positions. It does the following:

- As long as the object to guide (p_object) is too far from the path, it loops over all paths and path points to find the nearest point and checks if it is in "snapping" range.
-

```

// for all paths
for (int i = 0 ; i<nr_paths; i++)
{
    // for all points in path
    for (int j = 0 ; j<(int)paths[i].path_points.size() ; j++)
    {
        // get positions
        NActor* a = paths[i].path_points[j];
        p_pos = wGetGlobalPosition(a);
        p_obj = wGetGlobalPosition(p_object);
        // calculate error
        er_glo = p_pos-p_obj;
        float dist = er_glo.magnitude();
    }
}

```

```

        if (dist < dist_shortest) {
            dist_shortest = dist;
            closest_path = i;
            closest_point = j;
        }
    }
}
if (closest_path == -1) return -1;
snapped_to = closest_path; // in the next loop this path is used for shared
                           control

```

- If snapped it gets the current position of the object to guide (p_object) and can apply the look ahead time.

```

•
NxVec3 o1 = GetGlobalPosition(p_object);
if (look_ahead_time != 0){
    NxVec3 vtg = GetLinearVelocity(p_object);
    o1 = o1 + vtg*look_ahead_time;
}

```

- It looks for the two closest path points (p1 and p2) in the path currently snapped_to. It gets their orientations (p1_quat and p2_quat) and finds the closest point (o2) on the line between p1 and p2 (interpolation). Then calculate the corresponding desired orientation of the path by spherical linear interpolation (SLERP).

```

//get the interpolation point between p1 and p2
float i_p = ((o1[0] - p1[0])*(p2[0]-p1[0]) + (o1[1] - p1[1])*(p2[1]-
    p1[1])+(o1[2] - p1[2])*(p2[2]-p1[2])) / (pow(p2[0]-
    p1[0],2)+pow(p2[1]-p1[1],2)+pow(p2[2]-p1[2],2));
o2 = p1+ i_p*(p2-p1); //get the truly closeted point
o2_quat.slerp(i_p,p1_quat,p2_quat); //get desired orientation

```

- It calculates the rotational error from the relative quaternion (q_rel) between the path and guided object.

```

float tau = 2*atan2(q_rel.x,q_rel.w); //get twist angle

//check for unequal stiffness's
float k_y_p = 0; float k_z_p = 0;
if (stiffness_y>=stiffness_z){
    k_y_p = stiffness_y/stiffness_y; k_z_p = stiffness_z /stiffness_y;
}
if (stiffness_z >stiffness_y){
    k_y_p = stiffness_y /stiffness_z; k_z_p = stiffness_z /stiffness_z;
}

```

```

// vector that is used for a fictive rotation (if stiffness are unequal)
NxVec3 u0(1.0f,0.0f,0.0f); // vector at desired orientation (path)
NxVec3 ut=u0;
q_rel.rotate(ut); // transform to guided object orientation

```

```

NxQuat q_z;
if (k_z_p < 1.0f) // more stiff around y
{
    NxVec3 ut_xy(ut[0],ut[1],0.0f); // project guided object on xy plane
    if (ut_xy.magnitude())>pow(10.0, -5))
    {

```

```

        // calculate the angle
        float angle = acos(u0.dot(ut_xy/ut_xy.magnitude()));
        if (ut_xy[1]<0.0f) // guided object in -y
            q_z.setWXYZ(cos(-angle/2*(1.0f-k_z_p)),0,0,sin(-
                angle/2*(1.0f-k_z_p))); // rotate positive around z axis
        else // guided object in +y
            q_z.setWXYZ(cos(angle/2*(1.0f-k_z_p)),0,0,sin(
                angle/2*(1.0f-k_z_p))); // rotate negative around z axis
        q_z.conjugate();
        q_rel_k = q_z*q_rel;
    }
}
else if (k_y_p < 1.0f) // rotation torque is more stiff around z
{ ... similar ... }

// calculate swingy (Sy) and swingz (Sz)
float gama = tau/2;
float beta = atan2(sqrt(pow(q_rel_k.y,2)+pow(q_rel_k.z,2)), q
    sqrt(pow(q_rel_k.x,2)+pow(q_rel_k.w,2)));
float sinc_beta = 1.0f;
if (beta != 0) sinc_beta = sin(beta)/beta;
float Sy = 2/sinc_beta*( cos(gama)*q_rel_k.y - sin(gama)*q_rel_k.z );
float Sz = 2/sinc_beta*( sin(gama)*q_rel_k.y + cos(gama)*q_rel_k.z );

// correct tau for large angles
if (tau > (4.0f*atan(1.0f)) ) tau = -2*(4.0f*atan(1.0f))+tau;
if (tau < -(4.0f*atan(1.0f)) ) tau = 2*(4.0f*atan(1.0f))+tau;

erl[0] = -tau;
erl[1] = -Sy;
erl[2] = -Sz;

```

- It calculates local stiffness and damping depending on the location on the path (assuming equally spaced path points).

```

local_k = stiffness_start + ((stiffness_end-stiffness_start) /
    (float)nr_points)*((float)current_point);
local_c = NxVec3(sqrt(stiffness[0]*mass_inertia[0])*zeta[0]*2 ,
    sqrt(stiffness[1]*mass_inertia[1])*zeta[1]*2 ,
    sqrt(stiffness[2]*mass_inertia[2])*zeta[2]*2);

```

- It calculates the guiding force and torques.

```

// calculate forces and add damping in path frame
sc_path_force[0] = etl[0]*tk[0] -vtl[0]*tc[0];
sc_path_force[1] = etl[1]*tk[1] -vtl[1]*tc[1];
sc_path_force[2] = etl[2]*tk[2] -vtl[2]*tc[2];

// calculate torques and add damping in frame of guided object
sc_path_torque[0] = erl[0] -vr1[0]*rc[0];
sc_path_torque[1] = erp[1] -vr1[1]*rc[1];
sc_path_torque[2] = erp[2] -vr1[2]*rc[2];

// convert forces from path frame to world frame
o2_quat.rotate(sc_path_force);

// convert torques from guided object frame to world frame

```



```

o1_quat.rotate(sc_path_torque);

// remove forces that point in the direction of the path
NxVec3 dir_vect = p2-p1;
sc_path_force = sc_path_force-dir_vect* (dir_vect.dot(sc_path_force) /
    dir_vect.dot(dir_vect));

// but if specified there can be a force in the path direction
if (p.path_force !=0 ) {
sc_path_force = sc_path_force+dir_vect*p.path_force/dir_vect.magnitude();}

// check that forces and torques do not exceed their limit
if (sc_path_force.magnitude() > F_max) sc_path_force *=
    F_max/sc_path_force.magnitude(); // if force is larger than F_max: then
    scale it down
if (sc_path_torque.magnitude() > T_max) sc_path_torque *=
    T_max/sc_path_torque.magnitude(); // if torque is larger than T_max:
    then scale it down

```

Calculate the arm weight compensation

```

// apply arm weight compensation is master is below the path
if (o1[2] < o2[2]) {
    if (snaped_to != -1) { // but only if on a path
        op_comp_force[2] = (o2 [2]-o1[2])*operator_hand_k-10*vt1[2];
        if (op_comp_force[2] > max_op_comp_force) op_comp_force[2] =
            max_op_comp_force; // limit at max force
        // remove components along the path
        op_comp_force = op_comp_force -dir_vect*
            (dir_vect.dot(op_comp_force) / dir_vect.dot(dir_vect));
    }
}

```

The shared_control class has much more than is presented here. Like for example smoothly fading in and out of paths and fading between two paths. But that can be found at the USB-stick delivered at the BioMechanical Engineering depository.

C. Haptic shared control experiment

This appendix describes the executed haptic shared control experiment in more detail. Appendix C.1 contains the tasks instruction. Appendix C.2 describes the subdivision of the task into subtasks. The evaluation metrics and subjective measures are described in appendix C.3. Appendix C.4 shows an overview of the data management. Finally, appendix C.4 shows an overview of the experimental results.

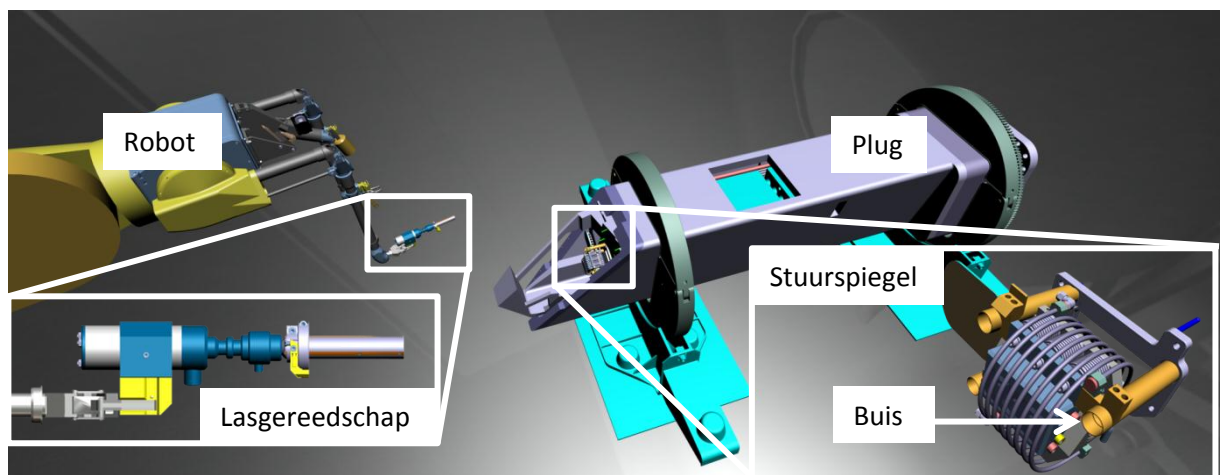
C.1.Task instructions

Before the start of the experiment, the following task instruction was handed out to the test subjects (in Dutch or English).

Task instruction [Dutch]:

Inleiding

Vandaag ben je een proef monteur voor de fusie centrale ITER. Als monteur gebruik je niet je handen, maar bedien je een robot in de "hotcell" (zie figuur 1). Je gebruikt een robot omdat de te repareren componenten radioactief en besmet zijn met giftige stoffen.



Figuur 1: overzicht van de hotcell met robot, plug, stuurspiegel, buis en lasgereedschap.

In de hotcell staat momenteel een "plug" uit de reactor. Deze "plug" wordt gebruikt om het plasma te verwarmen en stabiliseren met een soort magnetronstraling. Deze stralen worden gestuurd door de stuurspiegel. De spiegel is vanwege slijtage vervangen maar nog niet vast gelast aan het koelsysteem. Daarvoor ga jij het lasgereedschap in de buis plaatsen.

Hoe dit plaatsen in zijn werk gaat lees je zo meteen bij de training. De training is er om gewend te raken aan de situatie. Na deze training voor je de taak +-84 uit. Dit verdeelt over vier verschillende condities. Een conditie kan de taak vergemakkelijken of vermoeilijken. Daarom krijg je voor elke conditie ook +-2 oefeningen.

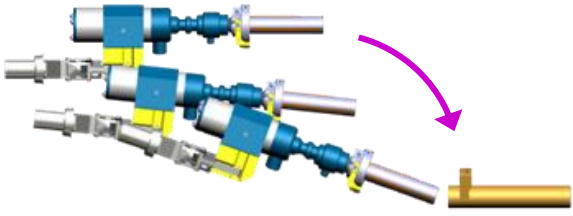
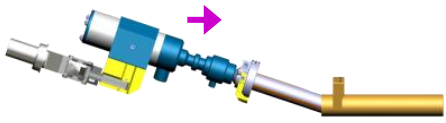
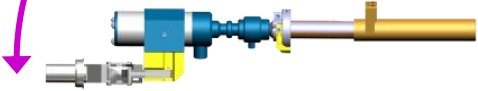
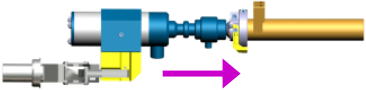

Merk op dat je tijdens de training 3 camerabeelden krijgt (boven-, zij-, en achteraanzicht). Zo heb je optimaal beeld om de taak te oefenen. Tijdens het definitieve werk heb je slechts één camera beeld (bovenaanzicht).

Instructie

Voer de taak zo nauwkeurig mogelijk uit, want als het lasgereedschap te hard stoot beschadig je stuurspiegel en moet het gereedschap opnieuw gekalibreerd worden en stop de simulatie. Dat is een grote materiele en financiële schadepost. Werk anderzijds niet te langzaam, want als je niet snel genoeg werkt ben je als operator te duur ;-)

Training

Om het plaatsen van de tool te vergemakkelijken moet je de volgende procedure aanhouden:

<p>Draai het lasgereedschap onder een hoek van 20 graden en beweeg deze tegelijkertijd naar beneden zodat de tip voor de opening ligt.</p>	
<p>Duw de tool tegen de buis aan. Door de ronding van het gereedschap centreert de tool automatisch naar het midden van de buis.</p>	
<p>Draaide tool horizontaal zodat deze in lijn ligt met de buis.</p>	
<p>Schuif de tool in de buis tot deze niet verder kan.</p>	
<p>Draai de tool 90 graden tegen de klok in.</p>	

Task instruction [English]:

Introduction

Today you are a test operator for the fusion reactor ITER. As an operator you cannot use your hands, but you control a robot in the "hotcell" (see Figure 1). You use a robot to repair components because the components are radioactive and contaminated with toxic substances.

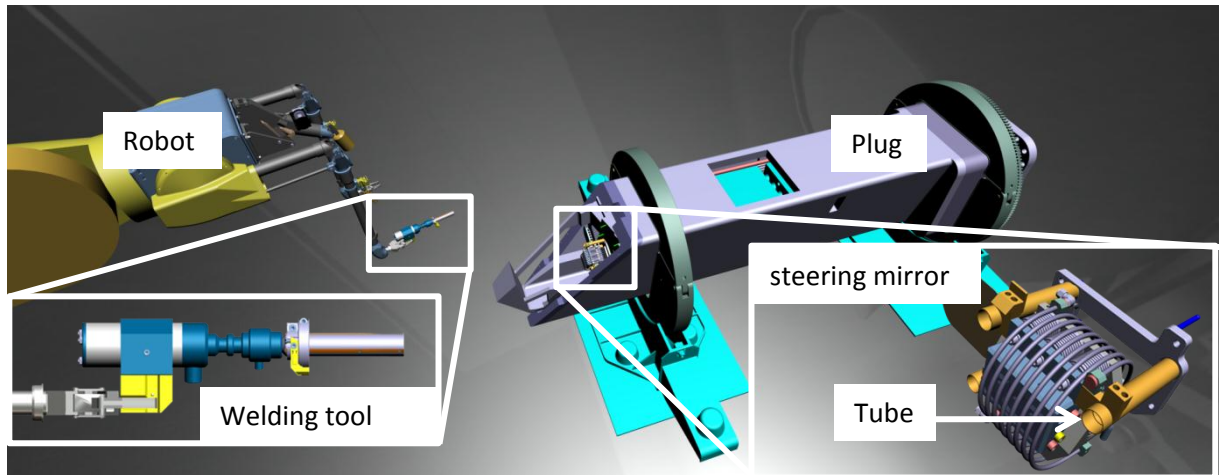


Figure 1: overview of the hotcell with robot, plug, steering mirror, tube and welding tool.

Currently there is a "plug" from the reactor the hotcell. This "plug" is used to heat and stabilize the plasma with a kind of microwave radiation. These rays are steered by the Steering mirror. The mirror is replaced due to wear but not yet welded to the cooling system. Therefore you will place the welding tool in the tube.

How this procedure go will be explained in the training hereafter. The training is there to become familiar with the situation. After this training you will execute the task +-84. The tasks are divided over four different conditions. A condition can make the task easier or more difficult. Therefore you get +-2 training tasks before each condition.

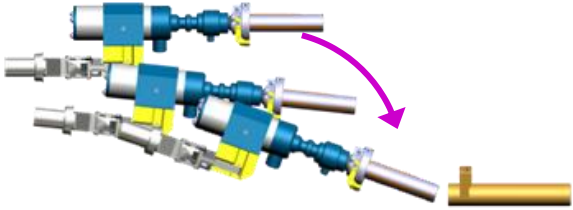
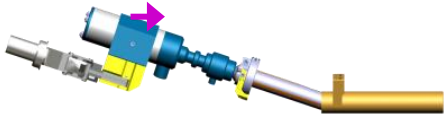
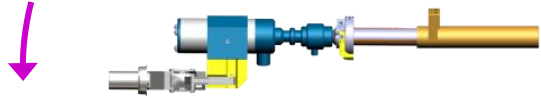
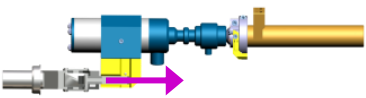
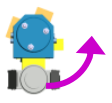
Note that you get 3 camera images during the training (top, side, and rear view). So you have optimal visual feedback to train the task. During the real work you have only one camera image (top view).

Instruction

Execute the task as accurate as possible, because if work to rough you can damage the steering mirror and the welding tool must be recalibrated and the simulation stops. It results in big material and financial costs. On the other hand work do not too slow, because if you're not fast enough, you are too expensive as an operator;-)

Training

The procedure to place the tool is as follows:

Turn the welding tool at an angle of 20 degrees and move it simultaneously down so that the tip is in front of the opening.	
Push the tool against the tube. By the roundness of the tool it automatically centres toward the middle of the tube.	
Rotated the tool horizontally to align it with the tube.	
Slide the tool in the tube until it stops.	
Turn the tool 90 degrees counter clockwise.	

Further the participants got the following introduction at the start of the shared control training:

In this condition you get help of haptic shared control. Shared control means that the computer lets you feel with forces how you can perform the task best. The computer guides you on a path in space.

In this exercise, the shared control is perfectly calibrated for the task. The forces that you feel are therefore 100% correct to accomplish the task successful. Later in the real test, it could be that the shared control contains errors.

Do not think the controller will do the work for you. You are still responsible for the safe and successful completion of the task. For example, you should verify that the tool actually sits in the tube.

So let it guide you but stay alert!

C.2.Subdivision in subtasks

To analyse the recorded data per subtask the data needs to be divided for the four fundamental subtasks. This division was automated using m-file "hsc_analysis.m" and based mainly upon the

position of the tip of the welding tool with some use of the position of the gripper. The tooltip is used as main criteria as that is the point that operators try to control. (Multiple operators were aided by visual guiding of the gripper position in a demonstration on a conference. Most of them where confused about this seemingly “wrongly located” guiding)

The tip position is not measured and therefore calculated via the slave position and orientation.

```
% set hand to tip distance
hand_to_tip_pos = [0,-383.506,77.1506]/1000;
hand_to_tip_pos = quatrotate([1/sqrt(2),0,0,-1/sqrt(2)],hand_to_tip_pos);

% gather rotation data (Virtual slave uses [qx,qy,qz,qw] while matlab uses
[qw,qx,qy,qz]). And calculate the tip position
wt_quat = quatconj([d.slave_rot(:,4) d.slave_rot(:,1:3)]);
tip_pos_rel = quatrotate(wt_quat,hand_to_tip_pos);
tip_pos = d.slave_pos + tip_pos_rel;
```

In a for-loop the data was tested on criteria of the subtasks. If the criteria were fulfilled, then the data index and time were recorded. The task started (Free Space Movement) when the user faded into the first path:

```
% find start time (the time we fade to path 1)
if (start_time==-1 && d.subtask(i)== 1)
    start_time = d.time(i);
    start_index = i;
end
```

The Contact Transition started when the tip was one diameter away from the tube seen from above:

```
% find transition between FSM and CT (at specific radius from tube)
dist_vect = tube_entrence(2)-tip_pos(i,2);
if (FSM_to_CT_time==-1 && abs(dist_vect) < 0.030)
    FSM_to_CT_time= d.time(i);
    FSM_to_CT_index = i;
end
```

The Constraint Translational Movement started when the tip was 10 [mm] passed the tubes entrance AND not above the tube AND not below the tube. On top of that the slave position was used to check if the tool was 20 [mm] (about 3 degree) from horizontal:

```
% find transition between CT and CTM (at specific depth of tip in tube and
z position of slave)
if (CT_to_CTM_time==-1 && tip_pos(i,2)<=(tube_entrence(2)-0.010) &&
tip_pos(i,3)<=(tube_entrence(3)+0.010) && tip_pos(i,3)>=(tube_entrence(3)-
0.010) && d.slave_pos(i,3)<(tube_entrence(3)-user_to_tip_pos(3)+0.02))
    CT_to_CTM_time= d.time(i);
    CT_to_CTM_index = i;
end
```

The Constraint Rotational Movement started when the tip was at a specific y location in space AND the tool was rotational velocity exceeded 0.5 [rad/s]:

```
% find transition between CTM and CRM (at specific depth of tip in tube and
a specific rotational velocity)
if (CT_to_CTM_time~-=-1 && CTM_to_CRM_time==-1 && tip_pos(i,2)<=0.813 &&
d.slave_vel_rot(i,2) >0.5)
    CTM_to_CRM_time= d.time(i);
```

```
CTM_to_CRM_index = i;  
end
```

The task ended at the end of the data set. The data set was terminated by the Virtual Slave simulator when the tool rotated about 75 degree from the vertical plane:

```
end_time = d.time(end);  
end_index = length(d.time);
```

C.3. Description of evaluation metrics

The performance of the tele-operated task can be quantified by a broad variety of metrics. Task performance metrics like time-to-complete and exerted contact forces are commonly used. To measure the response of the human operator to the inaccurate guiding, also a number of control effort and mental load metrics is included.

The analysed metrics can be divided into three categories:

- Task performance (TP) metrics:
 - tcc = Time-to-complete
 - dtt = Distance to tube
 - $F_{e,max}$ = Maximal force exerted on environment
 - FR = Fault rate (“insertion” below the tube for more than 1 tube diameter).

- Control effort (CE) metrics:
 - ttp = Total travelled path
 - tsa = Total steering angle
 - ia = Insertion attempts
 - wc = wiggle count

- Mental load (ML):
 - Self-reported workload (NASA TLX)

In fact the Control Effort metrics ‘physical workload’ (W_{phys}), ‘maximal operator force’ ($F_{op,max}$) and ‘average operator force’ ($F_{op,av}$) should also be analysed. These metrics give valuable information about how humans deal with the shared control forces. Unfortunately a suitable force sensor was not available, making it not possible to measure the human input forces.

Not all metrics were available or relevant for all motion types. Table 4 shows an overview of the analysed metrics per motion type.

Table 4: Analysed metrics, divided into Task Performance metrics (TP), Control Effort metrics (CE) and Mental load metrics (ML).

Matric id.	Metric description	Abbrev.	Total task	FSM	CT	CTM	CRM
TP1	Time-to-complete	tcc	x	x	x	x	x
TP2	Distance to tube	dt		x			
TP3	Maximal force exerted on environment	$F_{e,max}$			x	x	x
TP4	Fault rate	FR			x		
CE1	Total steering angle	t _{sa}	x	x	x	x	x
CE2	Insertion attempts	ia			x		
CE3	wiggle count	wc				x	
ML1	NASA TLX workload	TLX	x				
ML2	Subjective: How fast?	-	x				
ML3	Subjective: How accurate?	-	x				
ML4*	Subjective: Did the guiding help?	-	x				

*Only available

Calculation of the metrics

Time-to-complete:

- TP1 - Time-to-complete
The time-to-complete a (sub) task was calculated by subtracting the proper time points (e.g. $CT_time_to_complete = CT_to_CTM_time - FSM_to_CT_time$)
- TP2 - Distance to tube
Vertical distance to the tube centreline measured at the end of the Free Space Movement.
- TP3 - Maximal force exerted on environment
The maximum measured contact force between the tool and tube. Technically this is a semi contact force. This is the force that PhysX applied on the tool and tube to prevent them from violating the so-called nonpenetrating constraint. But it will be suitable for comparison of trials.
- TP4 - Fault rate.
The number of trials in which the welding tool was placed more than 30 [mm] underneath the tube as shown in Figure 29.

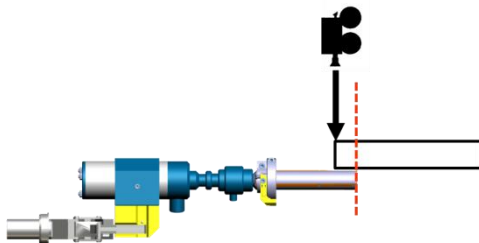


Figure 29: A fault insertion occurs when the tool is pushed more than one tool diameter (30 [mm]) under the tube.

- CE1 - Total steering angle
The total amount of rotation that the operator made with his hand, which can be seen as a measure of control effort for the subject to control the system. It is measured as the summed (integrated) difference between two subsequent orientations of the master device.
- CE2 - Insertion attempts
The number of attempts made to insert the tool in the tube during Contact Transition. An attempt is made when (seen from participant's perspective) the tool crosses the tubes entrance line in forward direction.
- CE3 - wiggle count
The number of up/down reversals made during Constraint Translational Movement. A reversal is counted when the contact point between the tool and the tube changed from up to down or reversed. This metric was possible as the tube had two "sensors" modelled in it.
- ML1 NASA TLX workload
Self-reported workload was obtained using the standard NASA TLX questionnaires. The NASA TLX weights and ratings were obtained using hard-copy questionnaires.
- ML2, ML3, ML4 - Subjective: How fast/accurate? and: Did the guiding help?
The subjective measures were obtained using questionnaires (see questionnaires on following pages).

Experiment 1 – Questionnaire

Date: -07-2012

Questions before experiment:

-Subject name:

-Age:

-Left-handed or right-handed: [LEFT / RIGHT]

-Background (study):

-Hobbies (related to the task/locomotion, for example: sport/repair of bicycles/..):

.....

-Computer gaming (hours per week):

-Problems with locomotion/motor system ('motoriek'): NO / YES,

Questions during experiment:

1. Conditions 1 – No HSC / GT

-How **accurate** did you perform the task? [1 (totally not accurate) ---- 8 (very accurate)]

1	2	3	4	5	6	7	8

-How **fast** did you perform the task? [1 (totally not fast) ---- 8 (very fast)]

1	2	3	4	5	6	7	8

2. Conditions 2 – No HSC / BT

-How **accurate** did you perform the task? [1 (totally not accurate) ---- 8 (very accurate)]

1	2	3	4	5	6	7	8

-How **fast** did you perform the task? [1 (totally not fast) ---- 8 (very fast)]

1	2	3	4	5	6	7	8

3. Conditions 3 –HSC / GT

-How **accurate** did you perform the task? [1 (totally not accurate) ---- 8 (very accurate)]

1	2	3	4	5	6	7	8

-How **fast** did you perform the task? [1 (totally not fast) ---- 8 (very fast)]

1	2	3	4	5	6	7	8

-To what extent **did the guiding help** you with the task? [1 (totally not helpful) ---- 8 (very helpful)]

1	2	3	4	5	6	7	8

4. Conditions 4 –HSC / BT

-How **accurate** did you perform the task? [1 (totally not accurate) ---- 8 (very accurate)]

1	2	3	4	5	6	7	8

-How **fast** did you perform the task? [1 (totally not fast) ---- 8 (very fast)]

1	2	3	4	5	6	7	8

-To what extent **did the guiding help** you with the task? [1 (totally not helpful) ---- 8 (very helpful)]

1	2	3	4	5	6	7	8

C.4. Data management

A vast amount of data is gathered during a single trial. But accurate timing –and thus measuring- on Microsoft Windows is not trivial. The High Precision Event Timer (HPET) seems the best clock device for accurate timing according to Smedinga [5]. He tested and implemented this system in the Virtual Slave Simulator to ensure real time performance. To measure the performance he stored the timing data on predefined memory location of the RAM modules during the measurement. Afterward the data was stored on the hard drive. In this way write actions have minimal effect on the simulation and measurements.

The logging implementation of Smedinga is used and extended to measure master, slave and environmental information. The information is measured in 89 variables at 1 [kHz] (the same frequency as the simulation). A raw data vector has the following content.

- 1: time [s]
- 2: button / dead man switch [-]
- 3-5: input force [N]
- 6-8: input torque[Nm]
- 9-11: master position [mm]
- 12-15: master orientation (quaternion in [qx qy qz qw]) [-]
- 16-18: master linear velocity [mm/s]
- 19-21: master rotational velocity [rad/s]
- 22-24: slave position [mm]
- 25-28: slave orientation (quaternion in [qx qy qz qw]) [-]
- 29-31: slave linear velocity [mm/s]
- 32-34: slave rotational velocity [rad/s]
- 35-37: path force [N]
- 38-40: path torque [[Nm]
- 41-43: nearest path position [mm]
- 44-48: path orientation (quaternion in [qx qy qz qw]) [-]
- 49: path index number[-]
- 50-52: contact force with tube frontal section [N]
- 53-55: contact position with tube frontal section [m]
- 56-58: contact force with tube frontal section [N]
- 59-62: contact position with tube frontal section [m]
- 63-65: contact force with tube frontal up/down section [N]
- 66-69: contact position with tube frontal up/down section [m]
- 70-72: contact force with tube rear up/down section [N]
- 73-75: contact position with tube rear up/down section [m]
- 76-79: contact force with tube frontal left/right section [N]
- 80-82: contact position with tube frontal left/right section [m]
- 83-85: contact force with tube rear left/right section [N]
- 86-89: contact position with tube rear left/right section [m]

This data is restructured, analysed and visualized in matlab as presented in the flowchart in Figure 30.

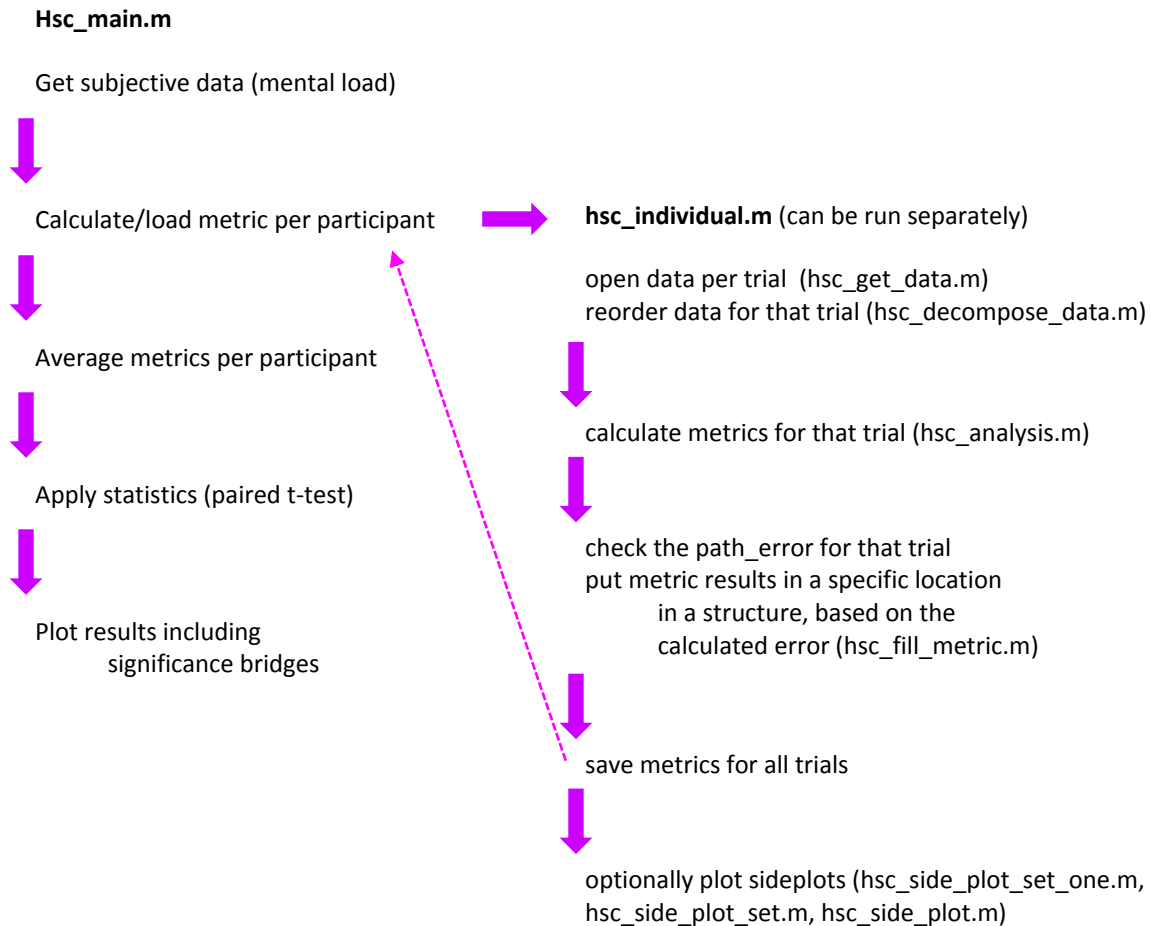


Figure 30: The flowchart of the data analysis algorithms.

C.5.Results

This appendix presents the raw experimental data and the experimental results for the metrics explained in Appendix C.3. The raw data is shown in Appendix C.5.1. The subtask results are presented in appendices C.5.2 to C.5.5. Finally appendix C.5.6 gives the subjective results for the entire welding tool placement task.

All presented results contain the measurement data of 14 subjects and 7 repetitions per condition. To quantify the differences between the experimental conditions, paired t-tests were done. In every paired t-test the data for normal (unguided) operations was compared to each of the (inaccurately) guided conditions. This is done separately for high and low transparency. Results were tested with a significance level of $p = 0.05$. When results were significant a significance bridge is plotted between the data sets with dots representing the significance level. Here ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p \leq 0.001$, $p \leq 0.01$ and $p \leq 0.05$.

C.5.1. Raw data

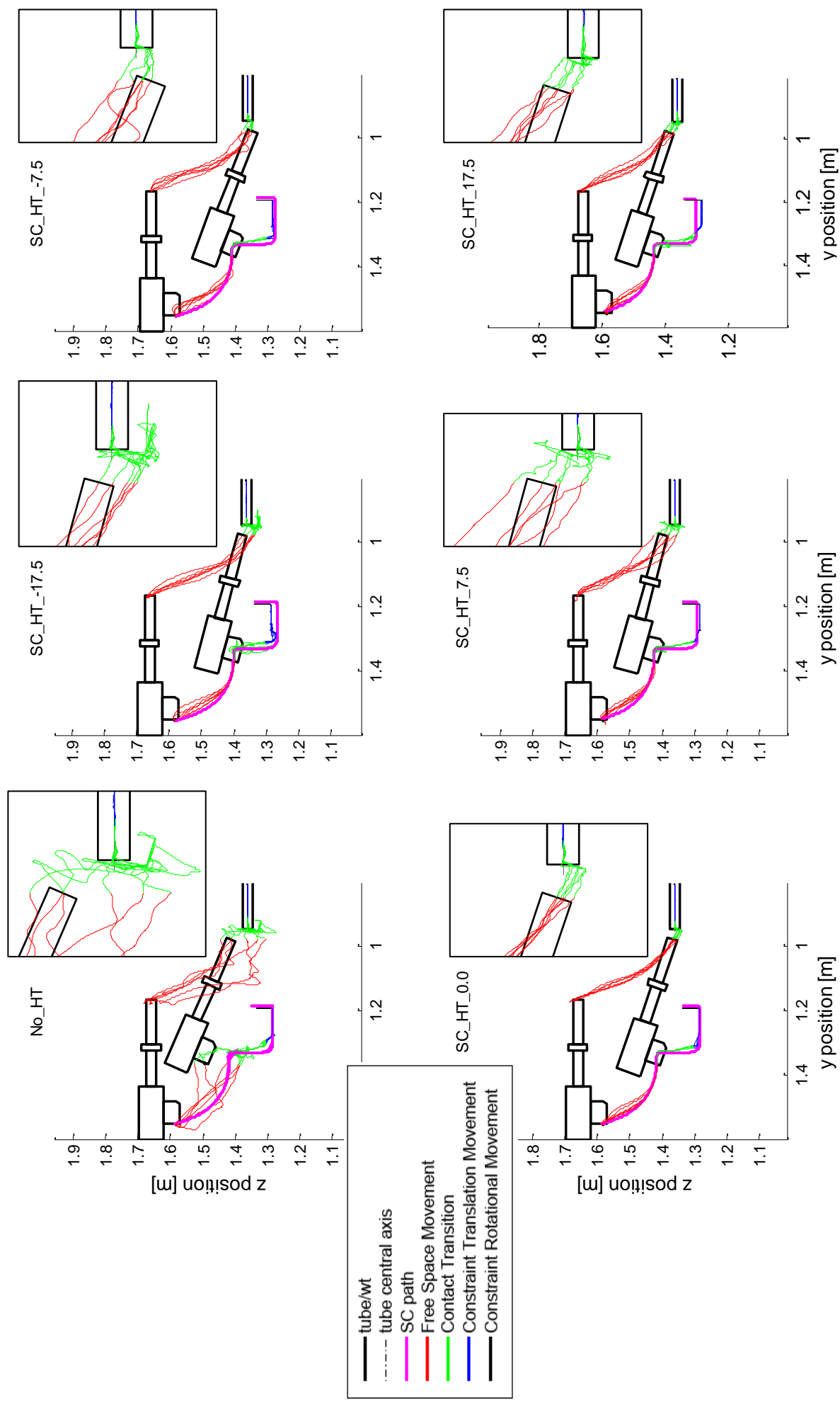


Figure 31: Raw data from one participant for high transparency (HT) in combination with normal operation (No) and (inaccurate) haptic shared control (SC_-17.5; SC_-7.5; SC_0.0; SC_7.5; and SC_17.5). Each plot shows 7 repetitions seen from the side with the tube on the right and the welding tool on the left. The travelled paths for both the slave and the tooltip are plotted with similar colour coding.

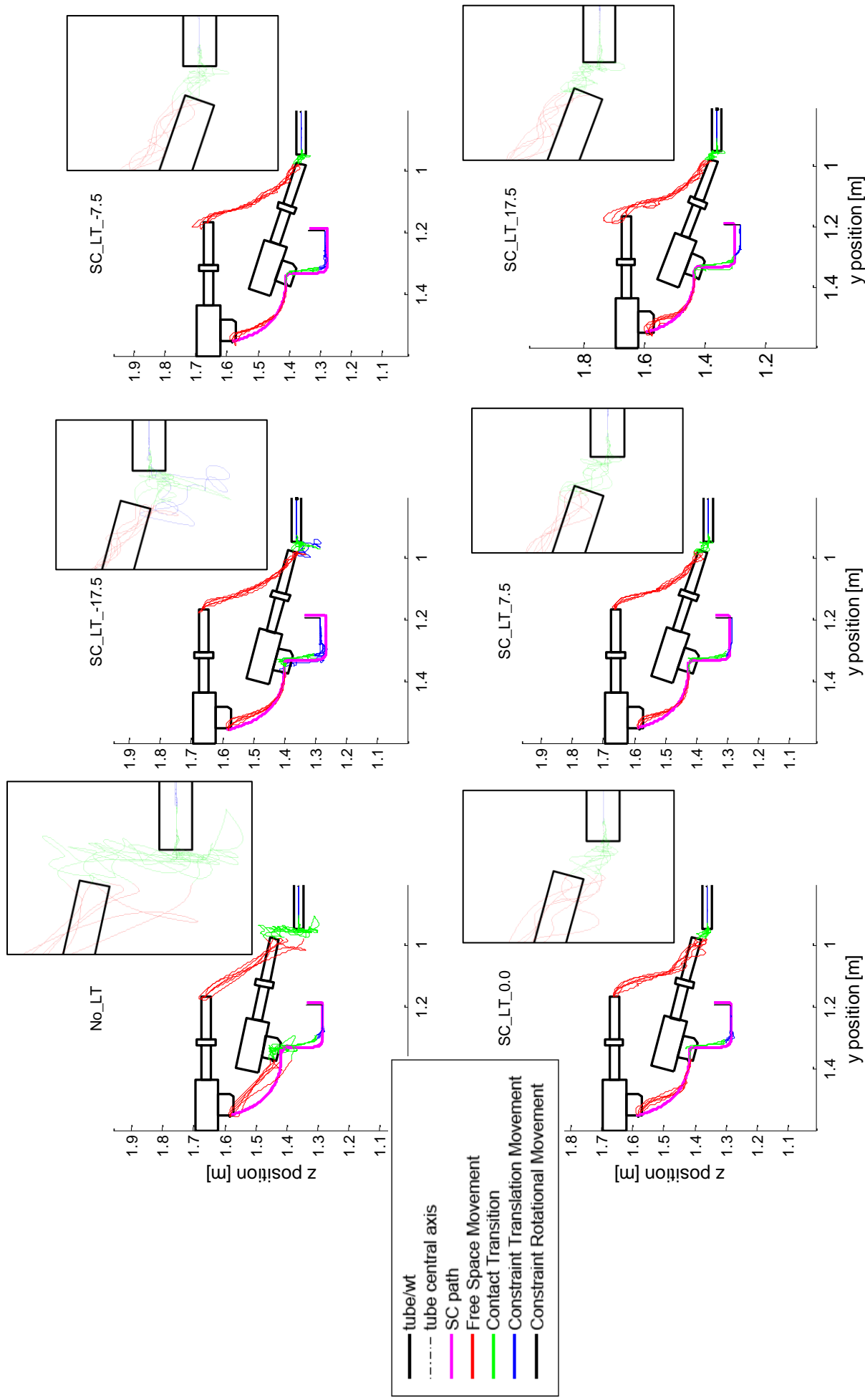


Figure 32: Raw data from one participant for low transparency (LT) in combination with normal operation (No) and (inaccurate) haptic shared control (SC__{-17.5}; SC__{-7.5}; SC__{0.0}; SC__{7.5}; and SC__{17.5}). Each plot shows 7 repetitions seen from the side with the tube on the right and the welding tool on the left. The travelled paths for both the slave and the tooltip are plotted with similar colour coding.

Figure 31 and Figure 32 show the raw (position) data as obtained for one participant. The figures nicely illustrate the benefit of shared control during Free Space Movement as the red lines lay close to each other and on the right course for the tube entrance. The swarm of green lines show the struggle that participants had to find the tubes entrance. The blue lines -of the master during inaccurate guiding- indicate that there really is an offset.

C.5.2. Free Space movement

Results for the Free Space Movement are shown in Figure 33, Figure 34 and Figure 35 for respectively time-to-complete, distance to tube and total steering angle. A first observation shows that required time is hardly reduced by perfect haptic shared control compared to normal operations. On the other hand the results show that the distance to the tube centre line is much shorter with guidance. Similar results are seen in comparing guidance with small and large errors to normal operations.

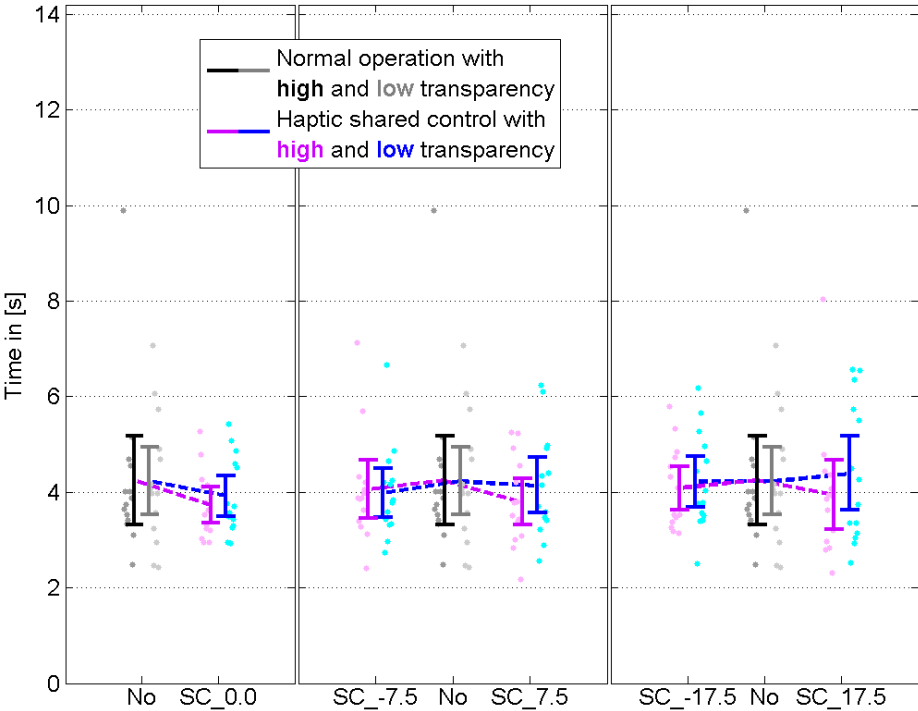


Figure 33: Task Performance – Time-to-complete for Free Space Movement (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

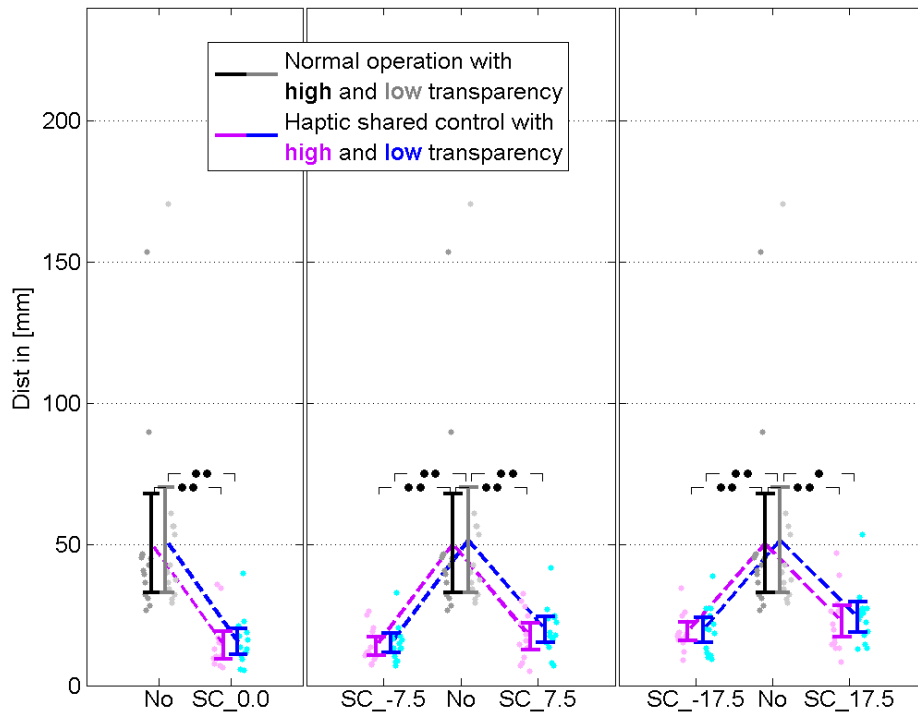


Figure 34: Task Performance – distance to tube for Free Space Movement (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

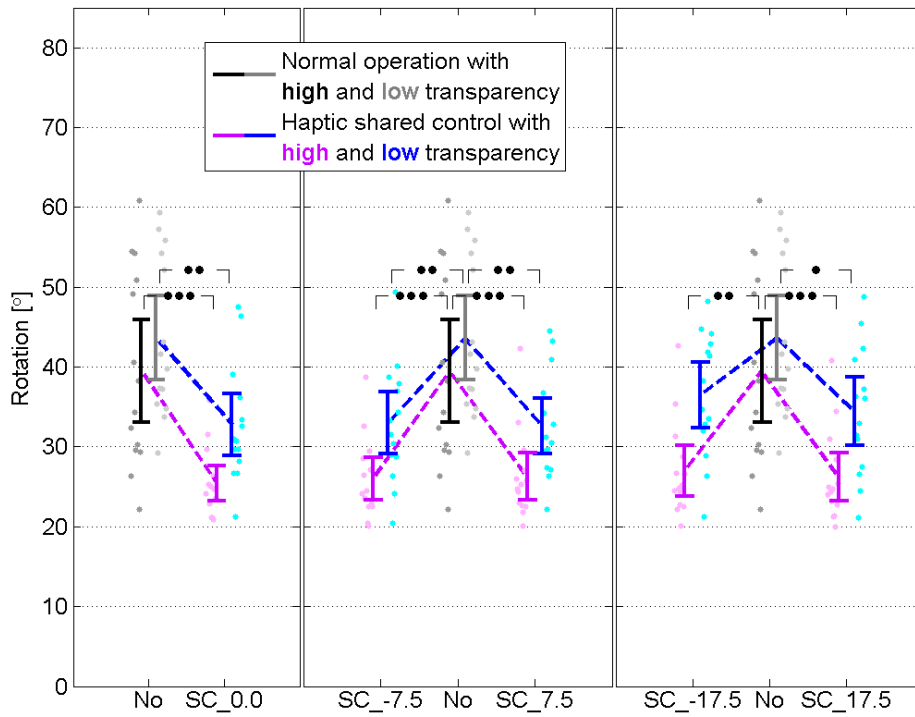


Figure 35: Control effort – Total steering angle for Free Space Movement (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

C.5.3. Contact Transition

The Contact Transition results for task performance are shown in Figure 36, Figure 37 and Figure 38 for respectively time-to-complete and maximal force exerted on environment (and failure rate for which no analysis is done) . For perfect shared control they show that the required time decreases while contact forces do differ compared to normal operations.

For small and large guiding errors a similar pattern is observed. However, positive guiding errors facilitate lower contact forces compared to unguided operations for high transparency. Further the required time for large negative errors do not improve compared to normal operation in low transparency. In that condition a few participants had mayor trouble to insert the tool.

Figure 39 and Figure 40 show results for respectively total steering angle and insertion attempts. They show that control effort decreases by comparing perfect guidance to no guidance. Similar results are found for the small and large guiding errors.

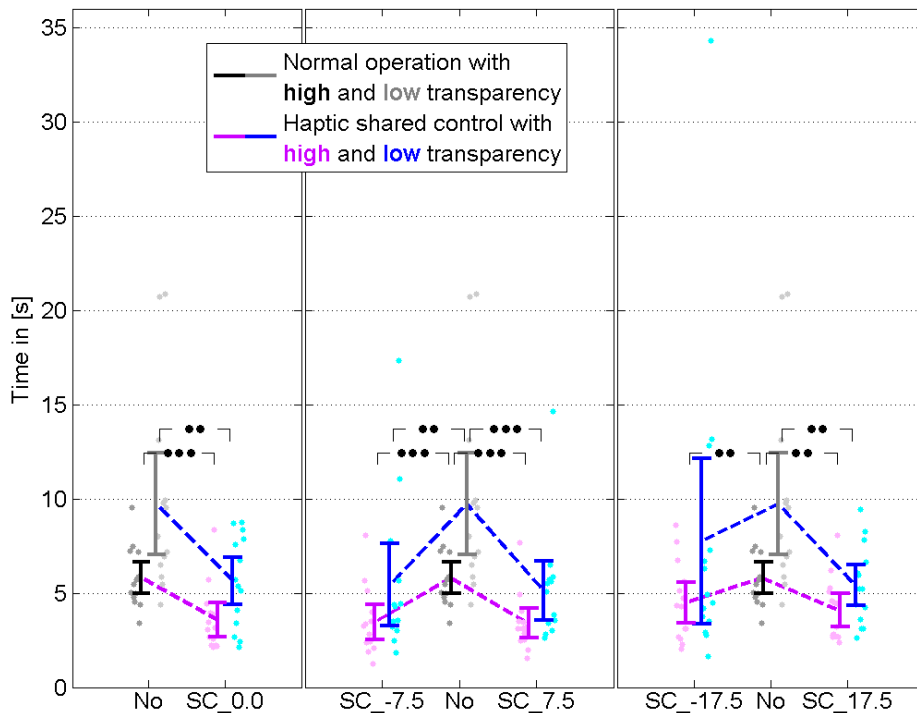


Figure 36: Task Performance – Time-to-complete for Contact Transition (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

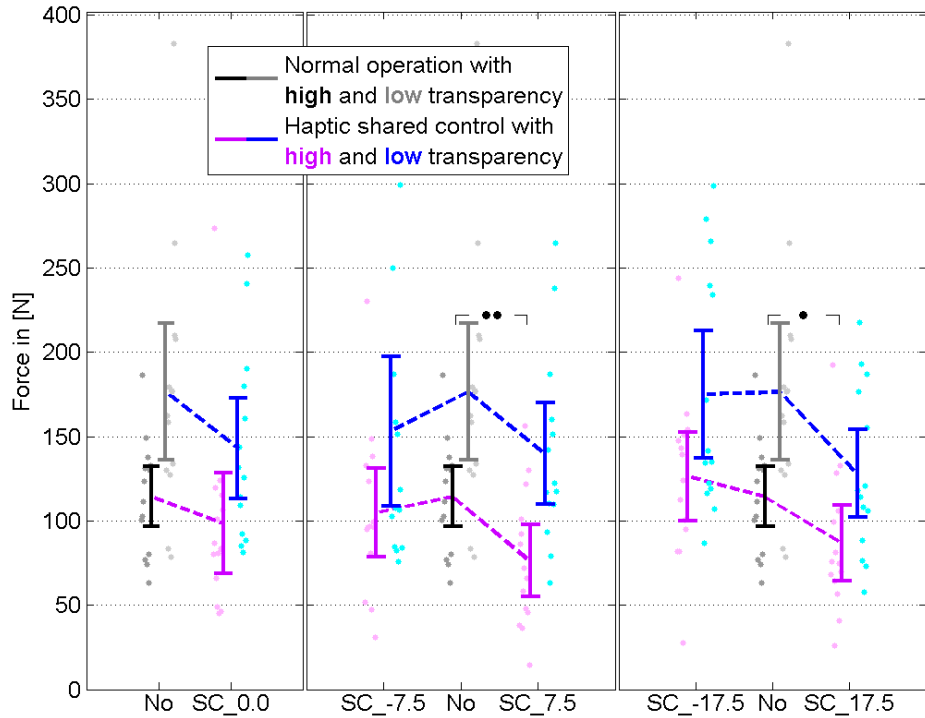


Figure 37: Task Performance – Maximal force exerted on environment for Contact Transition (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

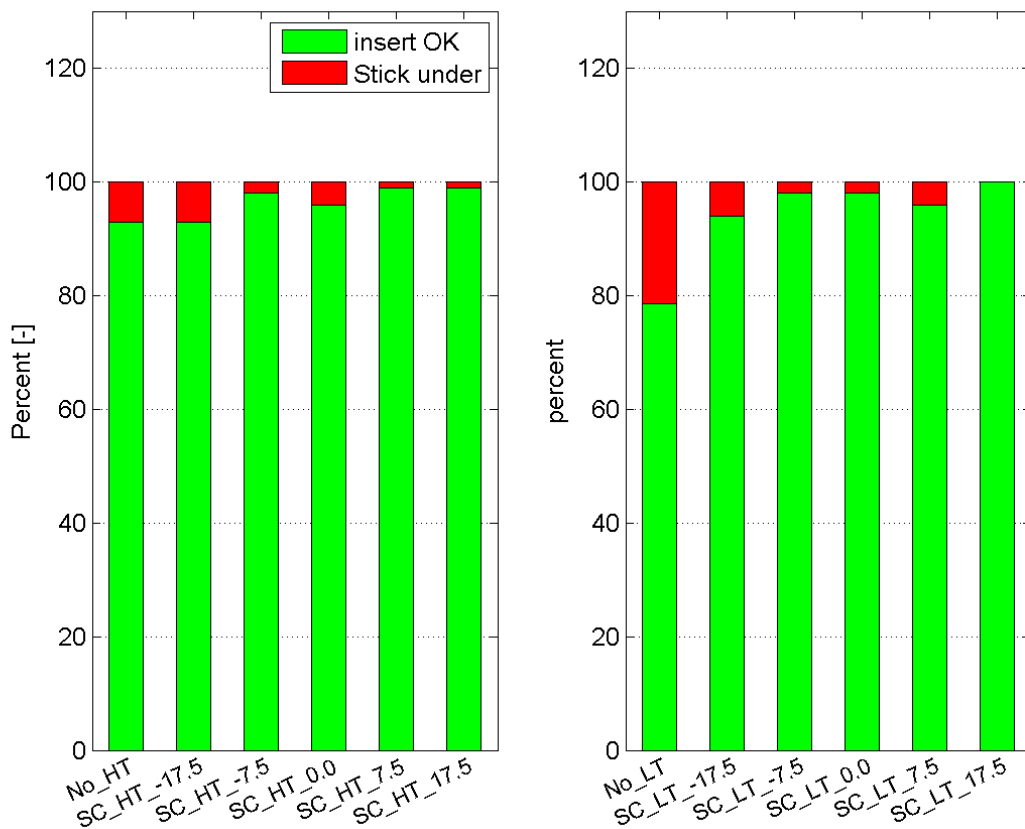


Figure 38: Task Performance – Failure rate for Contact Transition (14 subjects, 7 repetitions). The bars (n=98) represent the successful (green) and unsuccessful (red) trials in percentages for the unguided operations and (inaccurately) guided condition in high and low transparency.

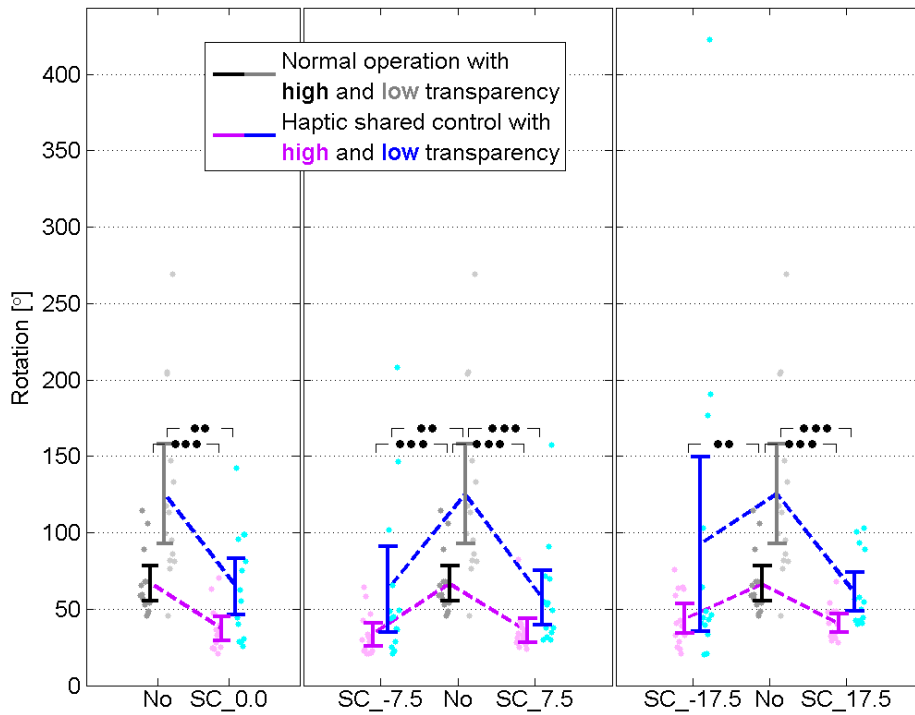


Figure 39: Control effort – Total steering angle for Contact Transition (14 subjects, 7 repetitions), showing the mean and 95% CI. '•••', '••' and '•' respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

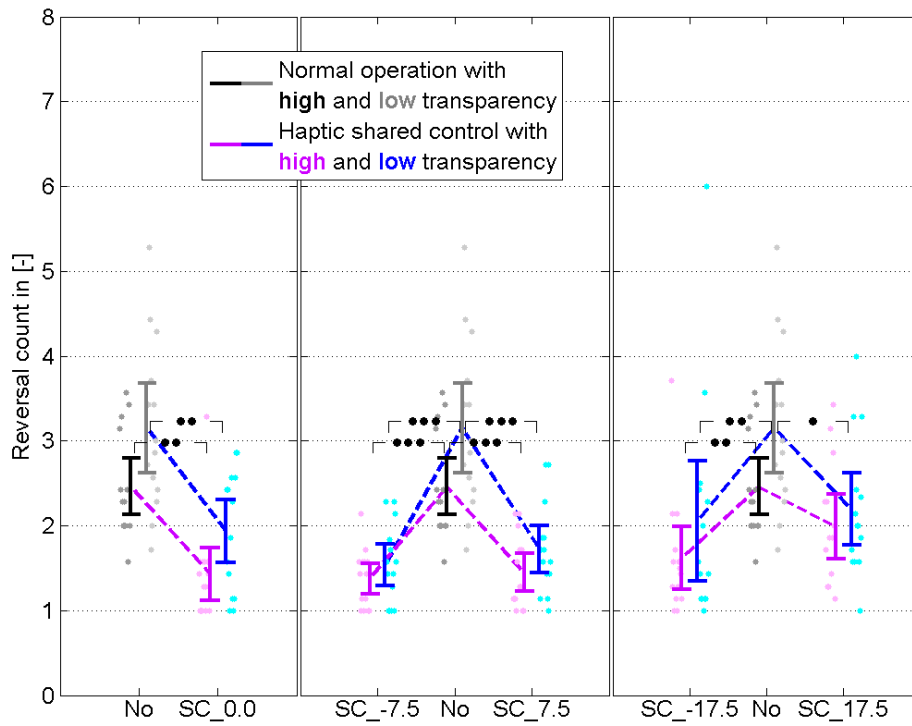


Figure 40: Control effort – Insertion attempts for Contact Transition (14 subjects, 7 repetitions), showing the mean and 95% CI. '•••', '••' and '•' respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

C.5.4. Constrained Translation Movement

Figure 41, Figure 42, Figure 43 and Figure 44 show respectively the results for time-to-complete, maximal force exerted on the environment, total steering angle and wiggle count for Constrained Translation Movement. A first observation is that all results have an asymmetric trend as explained in the discussion of the paper. Especially the wiggle count shows this trend clearly. That metric truly represents the difficulty that people had as a result of jamming during the Constraint Translational Movement.

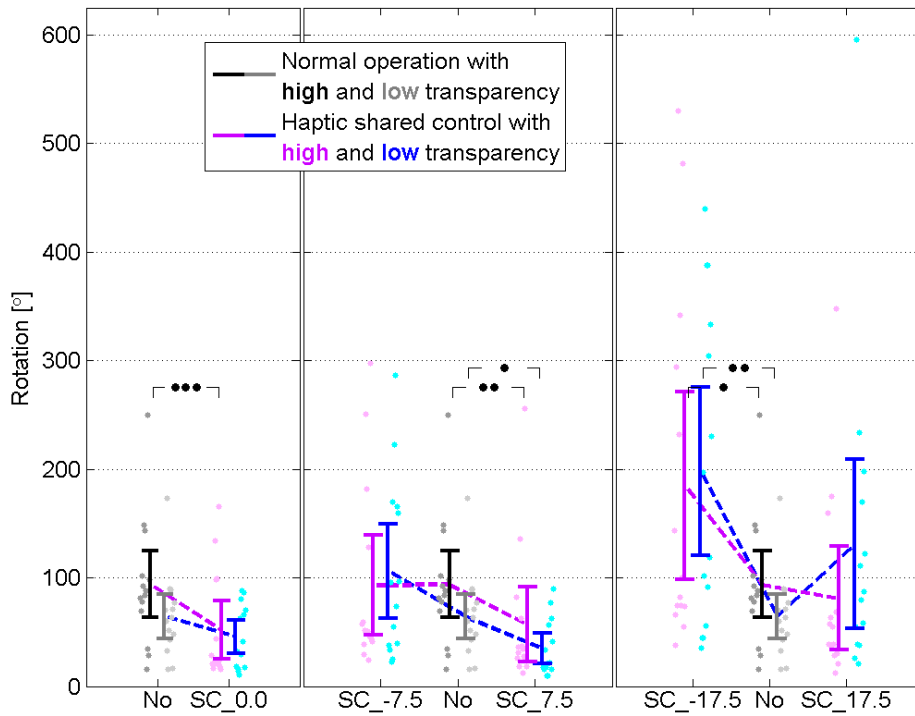


Figure 41: Task Performance – Time-to-complete for Constraint Translation Movement (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

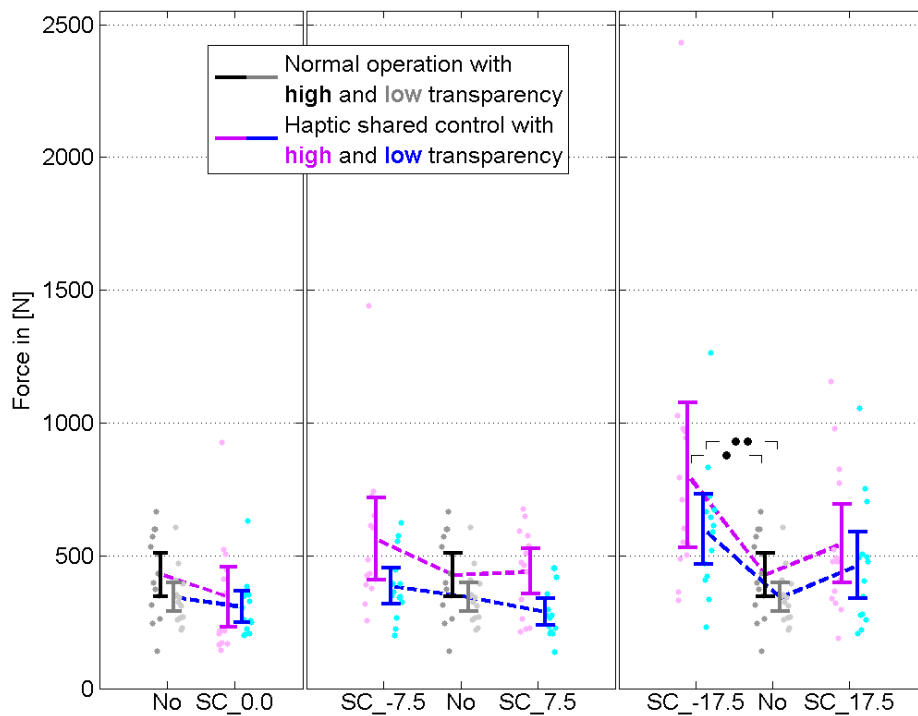


Figure 42: Task Performance – Maximal force exerted on environment for Constrained Translation Movement (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

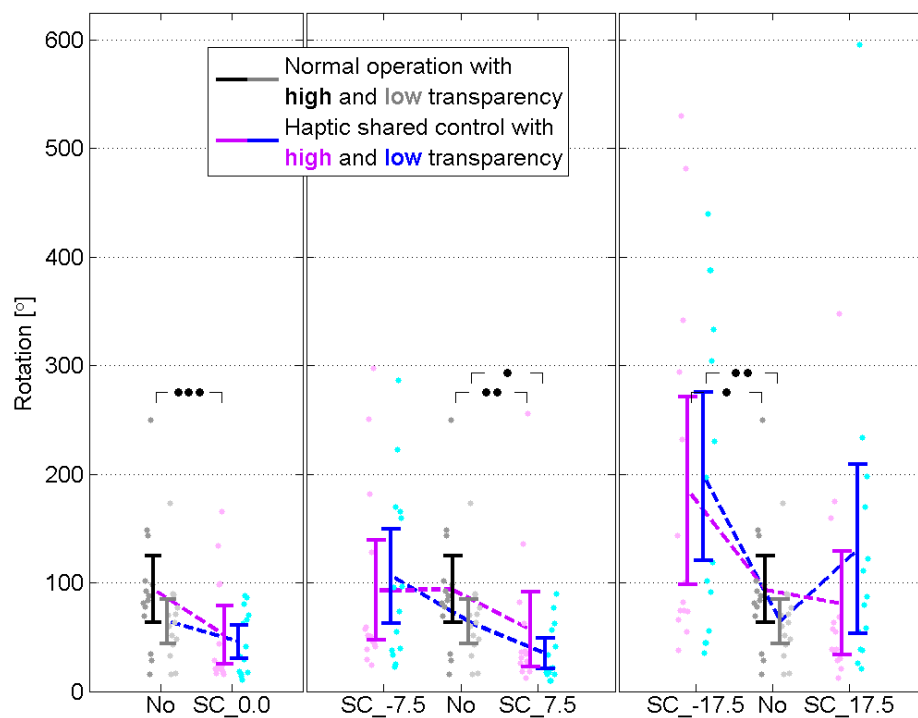


Figure 43: Control effort – Total steering angle for Constrained Translation Movement (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

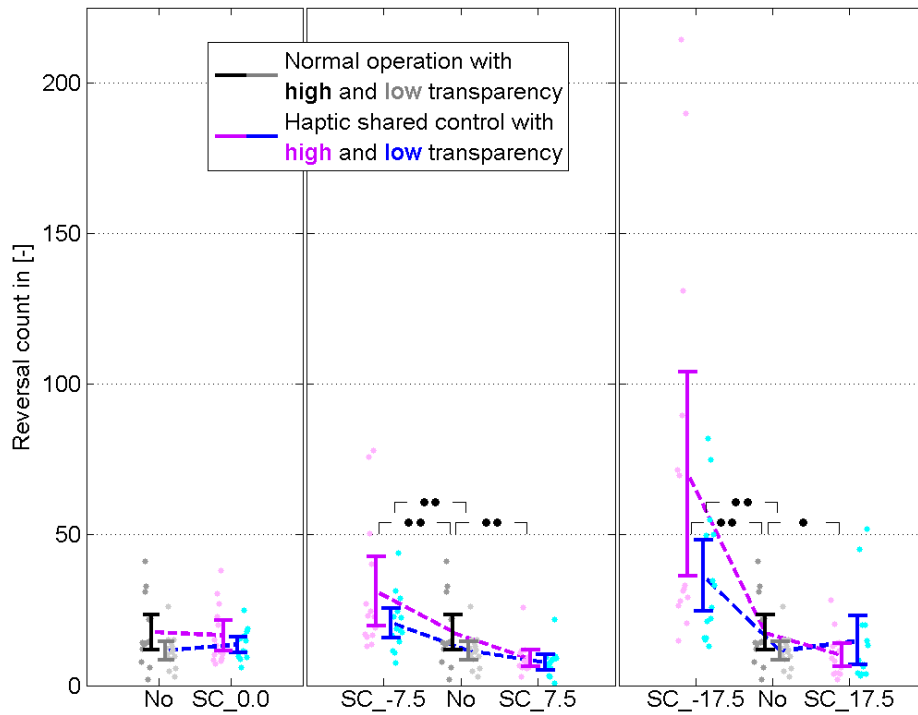


Figure 44: Control effort – Wiggle count for Constrained Translation Movement (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

C.5.5. Constrain Rotation Movement

Figure 45, Figure 46 and Figure 47 show respectively the results for time-to-complete, maximal force exerted on the environment and total steering angle for Constrained Rotational Movement. This task shows to be hardly affected by both perfect and inaccurate haptic shared control compared to normal operations. Differences are only found between normal operations and guidance with large errors.

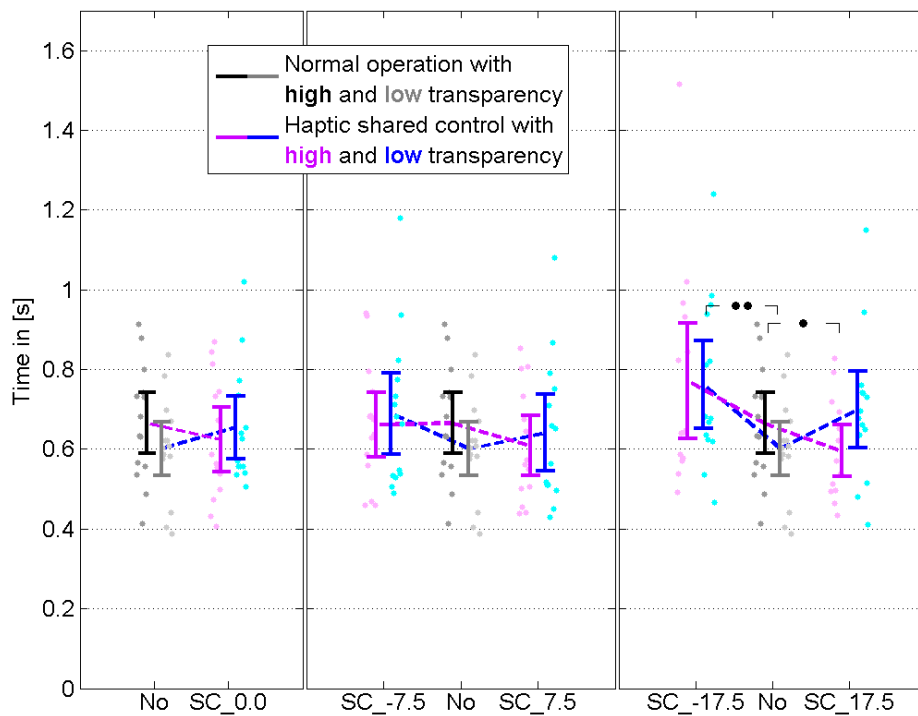


Figure 45: Task Performance – Time-to-complete for Constrained Rotation Movement (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

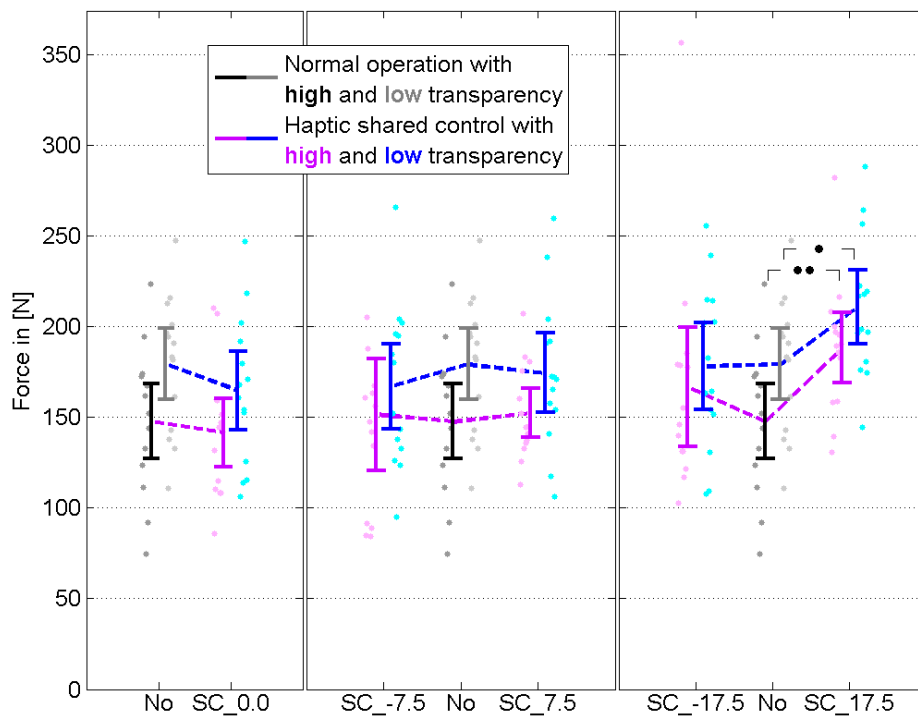


Figure 46: Task Performance – Maximal force exerted on environment for Constrained Rotation Movement (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

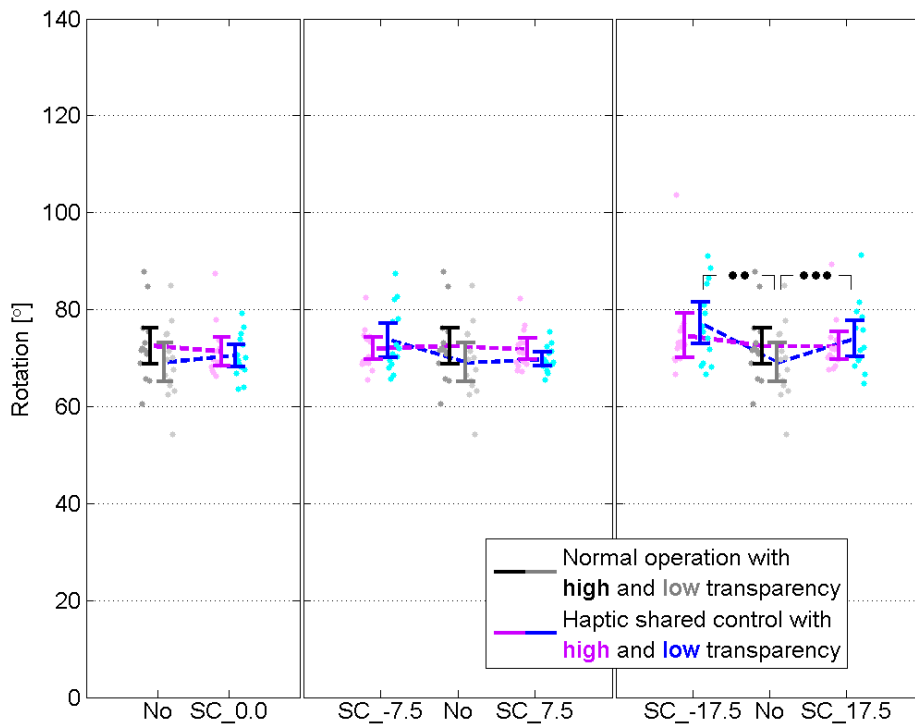


Figure 47: Control effort – Total steering angle for Constrained Rotational Movement (14 subjects, 7 repetitions), showing the mean and 95% CI. ‘•••’, ‘••’ and ‘•’ respectively denote the significance of $p < 0.001$, $p < 0.01$ and $p < 0.05$.

C.5.6. Subjective

Figure 48 and Figure 49 show the subjective measures for the entire task. The tests have only four measurements because the inaccuracies in haptic shared control were provided randomly in one condition for each transparency mode. The TLX test results reveal that four out of fourteen participants experienced an increased workload for haptic shared control -compared to normal operations- in both high and low transparency. Another four experienced a higher workload for haptic shared control -compared to normal operations- in only low transparency. Further the subjective measures provide evidence that subjects felt to perform the task faster (9 out of 14) and more accurate (10 out of 14) with the mixed guidance than without, provided high transparency.

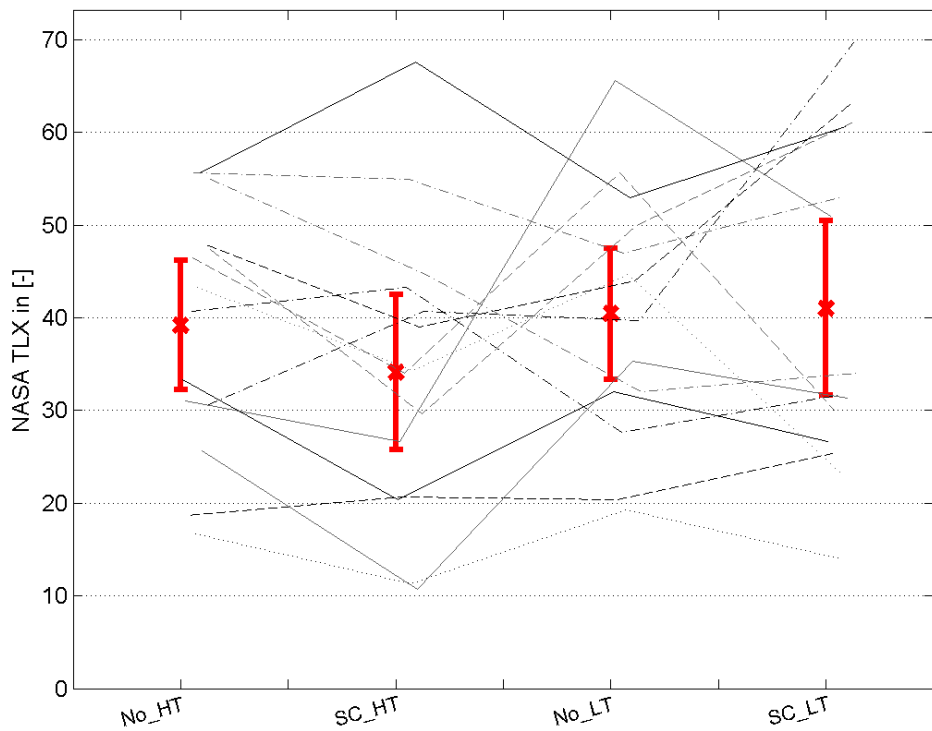


Figure 48: Mental load – NASA TLX workload for the entire task (14 subjects), showing the mean, 95% CI and trend per participant. Note that (inaccurate) haptic shared control could only be measured for the total condition.

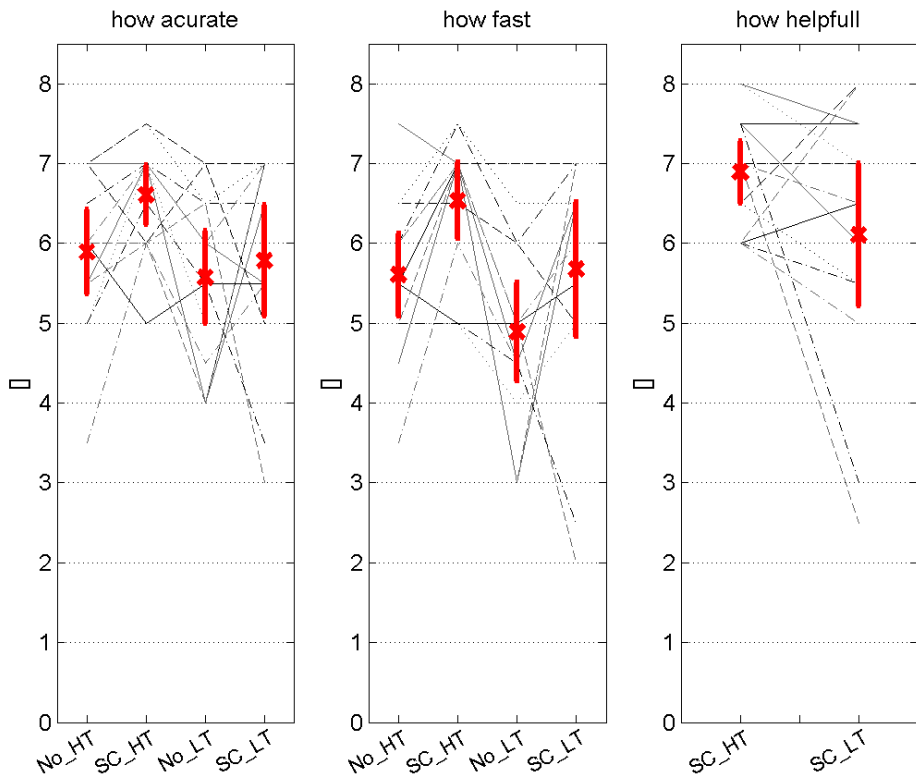


Figure 49: Mental load – Subjective measure for the entire task: how accurate; how fast and how helpful. (14 subjects) showing the mean, 95% CI and trend per participant. Note that (inaccurate) haptic shared control could only be measured for the total condition.

C.6.Ethics

Before potential participants were approached to take part the experiment, it was tested if the experiment was ethical to be performed by humans. This is done by completing the checklist provided by the Central Committee on Research Involving Human Subjects (CCMO) (<http://www.ccmo-online.nl/main.asp?pid=1&taal=1>). This filed in (unsigned) checklist is presented below

Project title:

Robustness of Haptic Shared Control Against Model Inaccuracies

Name(s) of researcher(s):

Jeroen van Oosterhout

Name of supervisor (if applicable):

David Abbink

	Yes	No
1. Does the study involve participants who are particularly vulnerable or unable to give informed consent? (e.g., children, people with learning difficulties, patients, people receiving counselling, people living in care or nursing homes, people recruited through self-help groups)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2. Will it be necessary for participants to take part in the study without their knowledge and consent at the time? (e.g., covert observation of people in non-public places)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3. Will the study involve actively deceiving the participants? (e.g., will participants be deliberately falsely informed, will information be withheld from them or will they be misled in such a way that they are likely to object or show unease when debriefed about the study)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4. Will the study involve discussion or collection of information on sensitive topics? (e.g., sexual activity, drug use, mental health)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5. Will drugs, placebos, or other substances (e.g., drinks, foods, food or drink constituents, dietary supplements) be administered to the study participants?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6. Will blood or tissue samples be obtained from participants?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
7. Is pain or more than mild discomfort likely to result from the study?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
8. Does the study risk causing psychological stress or anxiety or other harm or negative consequences beyond that normally encountered by the participants in their life outside research?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
9. Will financial inducement (other than reasonable expenses and compensation for time) be offered to participants?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
10. Will the study involve recruitment of patients or staff through the TU Delft, or working at a TU Delft site?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
11. Has or will this research be submitted to a research ethics committee other than this one? (if so, please provide details)	<input type="checkbox"/>	<input type="checkbox"/>



Name of Committee:

Date of submission:

Submission or approval number (if known):

12. If you have answered NO to questions 1-10 above (i.e., a more detailed submission to an ethics committee is not required), please very briefly (100-200 words) summarise your research, stating the question for the research, who will participate, the number of participants to be tested and the methods to be used.

Task performance in tele-operation is not optimal due to limited visual feedback and transparency. A promising method to improve tele-operation is haptic shared control. Yet shared control is tested only in (perfect) laboratory conditions, while real application will suffer from model and sensory inaccuracies. The effect of inaccuracies on the efficacy of shared control is unknown. This study investigates the influence of a translation error between haptic shared control and the environment on performance.

For this purpose 15-20 students/friends/family/colleges will be recruited for a two hour test. They will be asked to perform a peg-in-hole type task with 84 repetitions taking 5 to 15 seconds each. The repetitions are divided in 4 conditions. With each condition the participant is explained about possible mismatches. After the test he/she will be explained the purpose of the experiment and the true mismatch.

References

- [1] Grassia, F. (1998). Practical parameterization of rotations using the exponential map. *Journal of graphics tools*, 3, 1–13.
- [2] Baerlocher, P., & Boulic, R. (2001). Parametrization and range of motion of the ball-and-socket joint. *Workshop on Deformable Avatars*, 1–15.
- [3] Kallmann, M. (2008). Analytical inverse kinematics with body posture control. *Computer Animation and Virtual Worlds*, 19, 79–91. doi:10.1002/cav.v19:2
- [4] McIntyre, J., Berthoz, A., & Lacquaniti, F. (1998). Reference frames and internal models for visuo-manual coordination: what can we learn from microgravity experiments? *Brain research. Brain research reviews*, 28(1-2), 143–54.
- [5] Smedinga, D (2012). Real-time master/slave control loops. Internship report, InHolland
- [6] Boessenkool, H., Abbink, D. A., Heemskerk, C. J. M., van der Helm, F. C. T., & Wildenbeest, J. G. W. (2012). A Task-Specific Analysis of the Benefit of Haptic Shared Control During Tele-Manipulation. *IEEE Transactions on Haptics*.
- [7] Wildenbeest, J. G. W., Abbink, D. A., Heemskerk, C. J. M., van der Helm, F. C. T., & Boessenkool, H. (2012). The Impact of Haptic Feedback Quality on the Performance of Teleoperated Assembly Tasks. *IEEE TRANSACTIONS ON HAPTICS*.
- [8] Christiansson, G. A. V. (2007). *Hard master, soft slave haptic teleoperation*. Ph.D. dissertation, Delft, 2007.
- [9] Mamou, K., & Ghorbel, F. (2009). A simple and efficient approach for 3D mesh approximate convex decomposition. *Image Processing (ICIP), 2009 16th ...*, 3501–3504.
- [10] Rosenberg, L. B. (1993). Virtual fixtures: Perceptual tools for telerobotic manipulation. *Virtual Reality Annual International Symposium, 1993., 1993 IEEE* (pp. 76–82). IEEE.
- [11] Abbink, D. A., Boer, E. R., & Mulder, M. (2008). Motivation for continuous haptic gas pedal feedback to support car following. *Intelligent Vehicles Symposium, 2008 IEEE* (pp. 283–290). IEEE.
- [12] Mulder, M., Abbink, D. A., & Boer, E. R. (2008). The effect of haptic guidance on curve negotiation behavior of young, experienced drivers. *2008 IEEE International Conference on Systems, Man and Cybernetics* (pp. 804–809). IEEE. doi:10.1109/ICSMC.2008.4811377
- [13] R.F. Chandler, C.E. Clauser, J.T. McConville, H.M. Reynolds, J.W. Young (1975). Investigation of Inertial Properties of the Human Body. Wright-Patterson Air Force Base, OH (1975) DOT HS-801 430
- [14] Rolfe, a. (2007). A perspective on fusion relevant remote handling techniques. *Fusion Engineering and Design*, 82(15-24), 1917–1923. doi:10.1016/j.fusengdes.2007.04.049