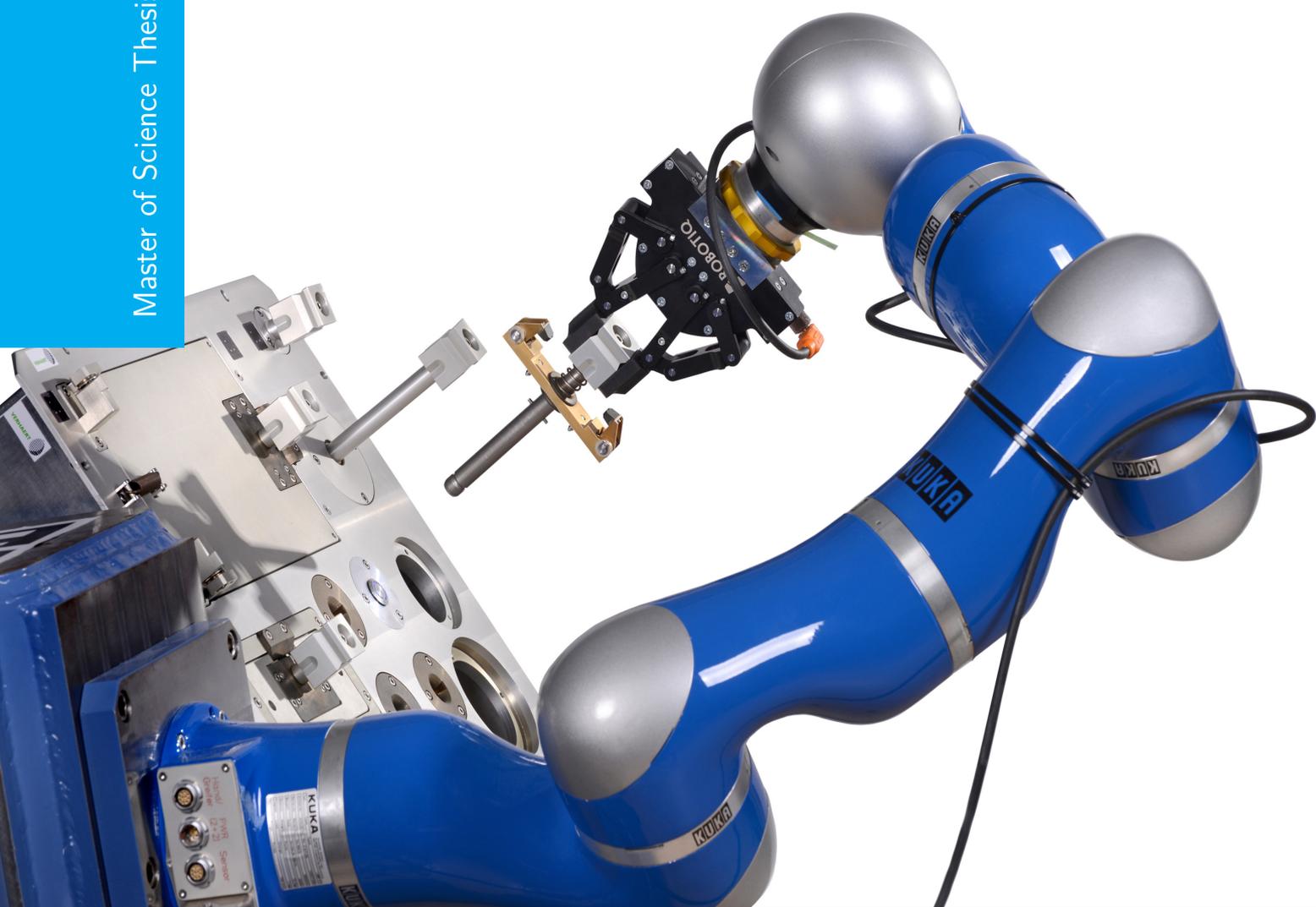


Teleoperation Support System Robust to Uncertainties in Haptic Guidance

Nicky Mol

Master of Science Thesis



Teleoperation Support System Robust to Uncertainties in Haptic Guidance

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft
University of Technology

Nicky Mol

February 15, 2016

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of
Technology



The work in this thesis was conducted on behalf of, and carried out in cooperation with ESA ESTEC Telerobotics & Haptics Laboratory.



Copyright © Delft Center for Systems and Control (DCSC)
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
DELFT CENTER FOR SYSTEMS AND CONTROL (DCSC)

The undersigned hereby certify that they have read and recommend to the Faculty of
Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis
entitled

TELEOPERATION SUPPORT SYSTEM ROBUST TO UNCERTAINTIES IN HAPTIC
GUIDANCE

by

NICKY MOL

in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE SYSTEMS AND CONTROL

Dated: February 15, 2016

Supervisor(s):

prof.dr.ir. R. Babuška

Dr.Ing. A. Schiele

Ing. J. Smíšek

Reader(s):

Dr.Ing. J. Kober

Abstract

When tasks need to be performed in remote, potentially inaccessible and/or hostile environments for humans, it is often more convenient to send a robot. Such environments are often only partially known, and therefore not suitable for fully autonomous robots to operate in. In these situations teleoperation can be used, which combines the problem solving capabilities of the human with the precision and durability of the robot. Robots are extremely good at performing accurate precision motions, whilst humans are best at making complex planning and decisions.

Having even partial knowledge of the environment allows the provision of guidance to the operator via haptic forces. For example, this haptic guidance can assist an operator with accurate alignment of a connector for a complex connection mating task. It has been shown that haptic guidance, which is based on models of the environment, can improve task performance. However, the majority of these studies have been conducted assuming that the intended goal of the operator is known to the support system a priori and that the environment is perfectly known. In the context of space robotics, there are often multiple tasks available in the environment and the task the operator intends to execute is not known a priori. Without a goal, the support system is not able to provide the operator with haptic guidance. Moreover, due to the unstructured nature of the environment, uncertainties can result in a mismatch of the models with respect to the environment. This results in an error in the haptic guidance that is provided to the operator. The goal of this thesis is twofold:

Firstly, when multiple possible goals are accessible in the environment, assistance can only be provided when the intended goal of the operator is known to the support system. Since the intention of the operator is not a physical signal that can be interpreted by the support system, it can not be communicated with the support system without performing additional actions. The operator is for example able to specify the intended goal to the support system by using a GUI or through speech. However, by utilising the control inputs of the operator to predict the intended goal, these additional actions can be skipped, which results in a more natural, seamless and faster teleoperation system. In this thesis, several prediction methods are validated and compared in simulation and by using real teleoperation data.

Secondly, depending on the magnitude of uncertainty the task performance can be reduced. In order to robustify the task performance of precise manipulation tasks in the presence of uncertainty, an outer admittance loop has been implemented on an impedance controlled KUKA robot. The outer admittance control loop changes the set point of the impedance controller based on contact forces and torques. This method is implemented and validated

on an experimental setup by doing peg-in-hole insertion experiments. Moreover, from this method information about the environment can be extracted in order to update the models that are used by the support system online, resulting in improved haptic guidance provided to the operator.

It has been concluded that memory-based prediction methods pose as a feasible method to predict the intentions of the operator in a teleoperated reaching task. Moreover, robustness to uncertainty is not increased by making the slave more compliant. However, pseudo-admittance control poses as a feasible method to robustify task performance against uncertainty of the peg-in-hole task and can be used to reduce uncertainty by using estimated information of the environment.

For future research, both methods should be validated in a human-in-the-loop study, on a teleoperation system where the operator is provided with haptic guidance, so the effects on the prediction method and of the force feedback can be studied.

In theory, theory and practice are the same. In practice,
they are not.

– *Albert Einstein*

Table of Contents

Acknowledgements	xiii
1 Introduction	1
1-1 Motivation	1
1-2 Background	3
1-2-1 Robot interaction control	4
1-2-2 Bilateral teleoperation	5
1-2-3 Shared control	8
1-3 Problem description	10
1-4 Goals	13
1-5 Approach	14
2 Macroscale uncertainty reduction by intention prediction	17
2-1 Approach: Intention prediction methods	17
2-1-1 Amnesic intention prediction	17
2-1-2 Memory-based intention prediction	18
2-1-3 Inferred intention prediction	19
2-2 Validation methods	21
2-2-1 Simulation study	21
2-2-2 Experimental study	23
2-3 Results	26
2-4 Discussion	27
3 Robustifying task performance in presence of microscale uncertainty	31
3-1 Problem	31
3-2 Approach: Pseudo-admittance control	33
3-2-1 Theoretical analysis	36

3-2-2	Simulated performance evaluation	38
3-3	Experimental validation	41
3-3-1	Experimental setup	41
3-3-2	Task and metrics	42
3-3-3	Results	46
3-4	Discussion	53
4	Conclusions	57
4-1	Future work	58
A	Other bilateral teleoperation architectures	59
A-1	Position-position	59
A-2	Position-force	60
B	Memory-based prediction derivation	63
	Bibliography	67
	Glossary	73
	List of Acronyms	73
	List of Symbols	73

List of Figures

1-1	The Interact Centaur robot [1]. Left: mock-up. Right: real rover.	1
1-2	Task-board containing a variety of tasks and a 7-DOF KUKA lightweight robotic manipulator.	2
1-3	Frontal view of task-board containing a variety of tasks.	2
1-4	Still image from a video stream taken of the remotely operated Special Purpose Dexterous Manipulator (SPDM) [2], which is able to perform a range of servicing tasks on the RRM module mounted on the ISS (courtesy of National Aeronautics and Space Administration (NASA) and the Canadian Space Agency (CSA)). . . .	3
1-5	Admittance relates a force input to a motion output. It can be thought of as the mobility when being pushed.	4
1-6	Impedance relates a motion input to a force output. It can be thought of as the resistance against motion.	4
1-7	High level impedance control architecture.	5
1-8	High level bilateral teleoperation interconnection of one-port and two-port network system representation.	5
1-9	High level bilateral teleoperation control scheme. The operator acts upon the knowledge of some goal that is to be achieved and audio/visual information, by exerting forces on the master device $F_h(s)$. These forces inflict a motion of the master device $X_m(s)$, and this motion is transferred through the communication channel to the slave robot. The forces upon contact with the environment $F_e(s)$, are transferred through the communication channel to the master device, where the operator is able to feel the feedback force.	6
1-10	General four-channel architecture.	7
1-11	A schematic representation of a input-mixing shared control scheme. The motion from the master device $X_m(s)$, that is inflicted by the operator now enters the support system. The support system uses models of the task and the environment to determine an optimal input. The optimal input will then be 'mixed' with the operator inflicted input, which will determine the input $X_{input}(s)$, that is transferred through the communication channel to the slave robot.	9
1-12	A schematic representation of a haptic shared control scheme. The support system uses models of the task and the environment to determine an optimal input. The optimal input is converted into a guiding force $F_g(s)$, that acts as a physical input on the master device and thus can be felt by the operator.	10

1-13	Schematic two dimensional representation of a peg-in-hole task where X_{G_1} is the intended goal. Macroscale uncertainty due to a mismatch in intentions between the operator and support system is depicted in blue. Microscale uncertainty due to a mismatch of the real environment and the models the support systems uses is depicted in orange.	12
1-14	1) Since the reference position has not yet been met, the support system provides the master device with guidance forces $F_g(s)$. 2) The slave robot will move according to the input on the master device until it makes contact with the environment. 3) Since the inaccurate reference $X_g(s)$ is not met, the master device will continue pushing the operator which shifts $X_m(s)$ into the environment. This will result in a force acting on the environment, which is also reflected to the operator as $F_{fb}(s)$. 4) The increasing force feedback will reduce the effect of the force from the haptic guidance on the master device until the forces are in equilibrium. In steady state, the resulting force on the master device is zero. The uncertainty in the guidance is masked from the operator, while at the slave side the robot can still exert significant forces $F_e(s)$ on the environment (adapted from [3]).	13
2-1	Schematic two dimensional example of amnesic prediction. Based solely on the current slave robot configuration, it can be concluded that the distance between X_{G_1} and the current slave robot configuration, d_1 , is smaller compared to the distance between X_{G_2} and the current slave robot configuration, d_2 , and thus X_{G_1} has the highest probability to be the intended goal (adapted from [4]).	18
2-2	When the operator intends to go to X_{G_2} but has to travel past X_{G_1} , amnesic prediction will predict the X_{G_1} to be the intended goal for the largest part of the trajectory. From point X_p and further, amnesic prediction is able to predict the correct goal.	19
2-3	Schematic two dimensional example of memory-based prediction. $\xi_{X_{init} \rightarrow X_{G_1}}^*$ (orange) and $\xi_{X_{init} \rightarrow X_{G_2}}^*$ (blue) represent the optimal trajectories from the initial position to respectively X_{G_1} and X_{G_2} and $\xi_{X_{init} \rightarrow X_s(t)}$ represents the trajectory that the operator has already travelled. Although X_{G_1} is closer to the current slave robot configuration, the travelled trajectory has been less costly towards X_{G_2} , which is the predicted goal (adapted from [4]).	20
2-4	Schematic two dimensional example of inferred intention prediction. Although the X_{G_1} is closer to the current slave robot configuration, X_{G_2} is predicted to be intended by the operator. This is based on the deviation of the optimal trajectory for each goal and the sequence of past slave robot configurations. The light blue area denotes the accumulated deviation of the optimal trajectory of X_{G_1} from the sequence of past slave robot configurations, while the light orange area denotes the accumulated deviation of the optimal trajectory of X_{G_2} from the sequence of past slave robot configurations. The grey area denotes the shared accumulated deviation of the optimal trajectories of both X_{G_1} and X_{G_2} from the sequence of past slave robot configurations. The light blue covers a greater area then the light orange and is therefore less likely to be the intended goal of the operator (adapted from [5]).	20
2-5	Architecture of the modelled teleoperation system, which is based on the four channel teleoperation architecture [6]. The environment is not considered here.	22
2-6	Minimum jerk motion, where (a) depicts the position trajectory, (b) the velocity trajectory and (c) the acceleration trajectory.	23
2-7	Minimum jerk motion position trajectories, where (a) depicts the reference trajectory, (b) the reference trajectory with added ZMWN and (c) the filtered reference trajectory that acts as an input on the master device.	24

2-8	The minimum jerk reference trajectory is complemented with ZMWN. The master device filters the high frequent behaviour before it is transmitted to the slave robot. The feedback loop makes sure that the reference trajectory $X_r(t)$ is followed. . .	24
2-9	Example of a three dimensional minimum jerk trajectory with simulated tremor. .	25
2-10	Experimental setup, where the operator interacts with the master device to operate the slave robot which can be seen in the display (from [7]).	26
2-11	Percentage of the entire trajectory that the goal is correctly predicted on simulated data.	27
2-12	Percentage of the entire trajectory that the goal is correctly predicted on real teleoperation data.	27
2-13	Exemplary trajectories from real data to show the goal prediction for different methods in time domain. Left in each figure: 3D plot of the trajectory, including the positions and velocities in x , y and z directions. Right in each figure: The predicted goal for different methods over time for the trajectory that is depicted on the left. The striped line denotes the intended goal.	28
3-1	High level impedance control architecture.	32
3-2	Left: Example of a translational offset in $-x$ direction. Right: Example of a rotational offset around the y axis.	32
3-3	High level adaptive impedance control architecture.	34
3-4	High level pseudo-impedance control architecture the admittance controller in the outer loop.	35
3-5	The system architecture overview.	38
3-6	Impedance control architecture with outer admittance control loop with delay. . .	39
3-7	Step responses of the impedance controlled robot with different impedance parameters, with the responses of the estimated models for a step of 5 mm at $t = 0$. .	39
3-8	Step responses of the impedance controlled robot in contact with different environments.	40
3-9	Step responses of the pseudo-admittance controlled robot in contact with different environments.	40
3-10	Step responses of the pseudo-admittance controlled robot in contact with different admittance control parameters, B_y	41
3-11	Step responses of the simulated pseudo-admittance controlled robot in contact with different admittance control parameters, K_y	41
3-12	Step responses of the simulated pseudo-admittance controlled robot with different impedance parameters.	42
3-13	Experimental setup with KUKA LWR 4+ as slave robot, ATI Gamma force-torque sensor with attached peg and the task-board.	43
3-14	Tool frame Ψ_{tool} and f/t frame $\Psi_{\text{f/t}}$ on the robot peg and force-torque sensor. .	43
3-15	Example of a rotational offset illustrated on the INTERACT task-board.	44
3-16	Example of a translational offset illustrated on the INTERACT task-board. . . .	44
3-17	The absolute force exerted on the environment and the maximum value of the absolute force.	45
3-18	Change of $\hat{X}_e(s)$ and $X_c(s)$ when performing a peg-in-hole insertion with a translational offset of 6 mm. Note that $\hat{X}_e(s) \neq 0$ because it represents the estimated value of the position of the wall of the hole, not the centerline.	46

3-19	Boxplot of the maximum insertion depth for rotational offsets versus different impedance control parameters.	47
3-20	Boxplot of the maximum insertion depth for translational offsets versus different impedance control parameters.	47
3-21	Boxplot of the average weighted forces and torques exerted on the environment for rotational and translational offsets versus different impedance control parameters.	48
3-22	Boxplot of the average weighted forces and torques exerted on the environment for translational offsets versus different impedance control parameters.	48
3-23	Boxplot of the maximum forces exerted on the taskboard for rotational offsets versus different impedance control parameters.	49
3-24	Boxplot of the maximum forces exerted on the taskboard for translational offsets versus different impedance control parameters.	49
3-25	Boxplot of the maximum insertion depth for rotational offsets versus different admittance control parameters.	50
3-26	Boxplot of the maximum insertion depth for translational offsets versus different admittance control parameters.	50
3-27	Boxplot of the average weighted forces and torques exerted on the environment for rotational offsets versus different admittance control parameters.	50
3-28	Boxplot of the average weighted forces and torques exerted on the environment for translational offsets versus different admittance control parameters.	50
3-29	Boxplot of the maximum forces exerted on the taskboard for rotational offsets versus different admittance control parameters.	51
3-30	Boxplot of the maximum forces exerted on the taskboard for translational offsets versus different admittance control parameters.	51
3-31	Boxplot of the average weighted forces and torques exerted on the environment for rotational offsets for methods with and without position dependency.	52
3-32	Boxplot of the maximum force exerted on the environment for rotational offsets for methods with and without position dependency.	52
3-33	Pseudo-admittance controlled insertion with a rotational offset of 3° and $K_y = [0.1 \text{ N/m}, 0.05 \text{ Nm/rad}]$	52
A-1	Two-channel position-position architecture.	60
A-2	Two-channel position-force architecture.	61

List of Tables

3-1	Routh-Hurwitz array of $\frac{X_s(s)}{X_m(s)}$	37
3-2	Experiments performed for different control parameters and rotational offsets.	44
3-3	Experiments performed for different control parameters and translational offsets.	45
3-4	Experiments used to show the effects of rotational and translational offsets versus impedance control parameters.	47
3-5	Experiments used to show the effects of rotational and translational offsets versus different admittance control parameters.	48
3-6	Experiments used to show the effects of rotational offsets versus adding position dependency (K_y) in the admittance control loop.	51

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor Jan Smíšek for all his patience, enthusiasm, motivation, and immense theoretical and practical knowledge that helped me to finish this thesis. Without the countless discussions and ‘teaching moments’ we’ve had, the thesis that is lying before you would not have reached the level it has now. I wish him all the best in preparation for the upcoming defence of his Ph.D dissertation.

I also would like to thank André Schiele for giving me the opportunity to do my research at the Telerobotics & Haptics Laboratory. Moreover, I want to thank him for his support and critical feedback during the course of this Master’s thesis. I would like to thank him and the European Space Agency for being able to make use of all the lab equipment that I needed in order to conduct this research. During my stay I’ve had the luck to witness the HAPTICS-1, HAPTICS-2 and INTERACT flight experiments, which all were quite the experience.

I would like to thank Prof. Robert Babuška for all the helpful inputs and thoughts that he provided during our meetings. Moreover, I would like to thank him for supporting me in doing this research at the Telerobotics & Haptics Laboratory.

Furthermore, I would like to thank all people of the Telerobotics & Haptics Laboratory for all the fun, educative and exciting moments. Many of you helped me out at some point in the process and contributed in that way to the final result.

Last but not least, I would like to thank my parents for their support and encouragement and my brother for all of his advice along the way. He likes to think that he is my ‘great example’ and given the choices that I make, I have to admit that it sure starts to look like that. The biggest thanks goes out to my beloved Marije, for her endless support and patience during the entire duration of my studies and especially during this Master’s thesis. Without her always being there for me, none of this would have been possible.

Delft, University of Technology
February 15, 2016

Nicky Mol

Chapter 1

Introduction

1-1 Motivation

There are many situations where it is more convenient to send a robot instead of a human to perform a manipulation task. Examples are situations where such tasks need to be performed in remote, potentially inaccessible and/or hostile environments for humans. Environments like this are often only partially known and unstructured, and therefore not suitable for fully autonomous robots to operate in. In these situations teleoperation can be used, which allows combining the skill sets of the robot and human that is operating it. In this way the planning and decision capabilities of the human operator can be exploited by the remotely located robot, while the human remains located in a safe, accessible environment.



Figure 1-1: The Interact Centaur robot [1]. **Left:** mock-up. **Right:** real rover.

To explore the limits of state-of-the-art teleoperation systems, in September 2015 the European Space Agency (ESA) performed a new space technology demonstration experiment, called INTERACT [1]. This experiment has been performed from on-board the International Space Station (ISS), involving the Interact Centaur robot, which can be seen in Figure 1-1. From the ISS Danish astronaut Andreas Mogensen remote-controlled the Interact Centaur robot, which was located on Earth, by means of teleoperation. The Interact robot consists of an AMBOT mobile platform, two 7-DOF KUKA lightweight robotic arms [8] with Robotiq parallel grippers and a stereo camera mounted on a SCHUNK light weight robotic arm. The

manipulators are equipped with force-torque sensors on each joint and additional ones on their wrists. The operator, on board the ISS, used a force reflective haptic master device and a tablet computer in order to carry out the control tasks. The control tasks involve: moving the platform on rough outdoor terrain from an initial position to a location where a task-board is situated, alignment of the platform with the task-board and performing a predefined constrained peg-in-hole insertion task which can be found on the task-board, as illustrated in Figures 1-2 and 1-3. During the experiment the alignment of the peg with the predefined hole was done with help of a motion capture system. Due to uncertainties, small misalignments caused the peg to get jammed in the hole.

This Master's thesis will focus on two parts. The first part will be on estimation of the intended task in an environment where multiple tasks are accessible, so the operator can be assisted with guidance towards that task. The second part will be on robustifying the performance of the peg-in-hole insertion task in the presence of uncertainty and on reducing the uncertainty. The motion control and alignment of the platform are not part of this Master's thesis.

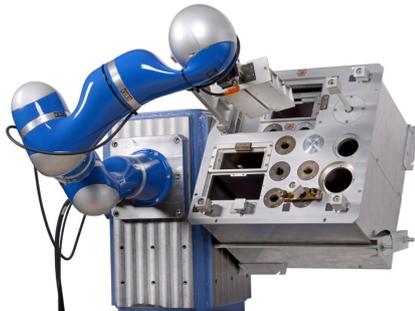


Figure 1-2: Task-board containing a variety of tasks and a 7-DOF KUKA lightweight robotic manipulator.

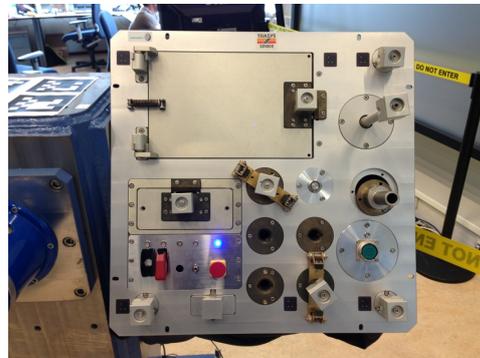


Figure 1-3: Frontal view of task-board containing a variety of tasks.

Possible application beyond INTERACT: Robotic Refuelling Mission (RRM)

In order to illustrate where the contribution of this Master's thesis could possibly be applied apart from INTERACT, a high level example from the space domain is given.

In Figure 1-4 a still image of the video stream from the remotely operated Special Purpose Dexterous Manipulator (SPDM) [2] can be seen, which is mounted on the ISS. This robot is able to perform a range of servicing tasks on the RRM module [9], which is mounted to an external platform on the ISS. The tasks that can be performed are: cutting and manipulation of wires, unscrewing caps, opening and closing valves, transferring fluid and providing inspection capabilities. The goal of this experiment is to eventually repair and refuel satellites in space, specifically satellites that are not designed to be serviced, without the need for an astronaut to do a space-walk. Environments like this are often only partially known and unstructured, and therefore not suitable for fully autonomous robots to operate in.

Teleoperation has been developed for these kind of situations, where an operator remotely operates a robot. The SPDM is remotely controlled by mission operators at NASA's Johnson



Figure 1-4: Still image from a video stream taken of the remotely operated Special Purpose Dexterous Manipulator (SPDM) [2], which is able to perform a range of servicing tasks on the RRM module mounted on the ISS (courtesy of National Aeronautics and Space Administration (NASA) and the Canadian Space Agency (CSA)).

Space Center, which is located in Houston. The operator is able to see what he is doing through the video stream, but the vision is often of poor quality and part might even be blocked by the manipulator or other objects. The visual feedback is complemented with force feedback, so the operator can haptically ‘feel’ what he is doing in the remote environment. By keeping the human-in-the-loop, skills like decision making and adaptability can be utilised in order to execute precise manipulation tasks.

By providing the operator with haptic guidance, that aid him in completing his task, the task performance can be increased. A support system provides haptic guidance by using models of the task and environment. However, in real space scenarios like in this example, the operator has multiple tasks he can choose to execute, so the support system does not know which task the operator is intending to execute a priori. Moreover, due to the unstructured nature of the environment uncertainties can result in a mismatch of the models with respect to the environment. This results in an error in the haptic guidance that is provided to the operator, which can degrade task performance [10].

1-2 Background

In this section, a short background will be provided to the reader on the subjects of robot interaction control (Section 1-2-1), bilateral teleoperation (Section 1-2-2) and shared control (Section 1-2-3).

1-2-1 Robot interaction control

In order to perform remote robotic manipulation tasks that involve physical contact with the environment, pure motion control is not suitable. Especially in contact with rigid environments, an error in position can cause unbounded contact forces and instability. When in contact with such environment, the interaction forces must be accommodated for, so the manipulator is able to comply with the environmental constraints. Over the last decades the importance of compliance has been widely recognized, which resulted in two main approaches for achieving compliant motion. Hybrid force/motion control [11] decomposes the space the end-effector operates in, into a subspace where the end-effector is not able to move in due to environmental constraints and a subspace in which the end-effector is free to move in. The end-effector is position controlled in the unconstrained subspace and force controlled in the constrained subspace resulting in compliant behaviour. Hybrid force/motion control ignores the dynamic relation between the environment and the manipulator resulting in inaccurate control of force and position and robustness problems.

To overcome this problem, impedance control was proposed [12], which maintains a desired dynamic relation between the position of the end-effector and the contact force by regulating the mechanical impedance of the manipulator. Since the magnitude of contact force depends on the reference motion, a poor choice of the reference motion can still result in undesirable excessive contact forces. The inability to track a contact force reference is a disadvantage of impedance control over hybrid position/force control.

Two control methods can be distinguished, which differ in their input and how they react to that. Admittance can be thought of as the mobility of a physical object when it is being pushed while impedance can be thought of as the resistance a physical object gives against motion [13]. Admittance is the causal dual of impedance, $Y(s)=Z(s)^{-1}$. These principles are illustrated in Figures 1-5 and 1-6. Here $F(s)$ represents the forces and $X(s)$ represents the motions. Where not deemed necessary and in figures, the Laplace s was left out for clarity.

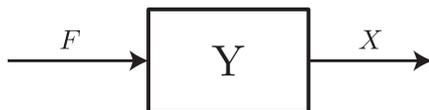


Figure 1-5: Admittance relates a force input to a motion output. It can be thought of as the mobility when being pushed.

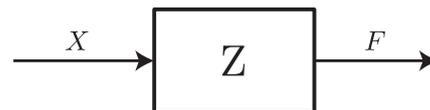


Figure 1-6: Impedance relates a motion input to a force output. It can be thought of as the resistance against motion.

In general, admittance control provides good accuracy in soft environments or free space since deformations of the environment are large and interaction forces are small. In stiff environments admittance control can result in instability. On the other hand, impedance control provides good performance in stiff environments because deformations of the environment are small and the interaction forces are large.

However, because of friction in the joints and other unmodelled dynamics, impedance control has poor accuracy when in soft environments or free space [14]. By using an inner torque control loop, the friction can be compensated for and better performance can be achieved in free space [8]. Because of these characteristics, an impedance controlled robot manipulator will be considered in this Master's thesis for which the high level control diagram is illustrated in

Figure 1-7. Here $X_r(s)$ is the reference motion, $X_s(s)$, the output of the slave robot and $F_e(s)$ are the forces and torques that are inflicted upon interaction with the environment $Z_e(s)$, that can be described as an impedance. $F_c(s)$ represents the outcome of the impedance controller.

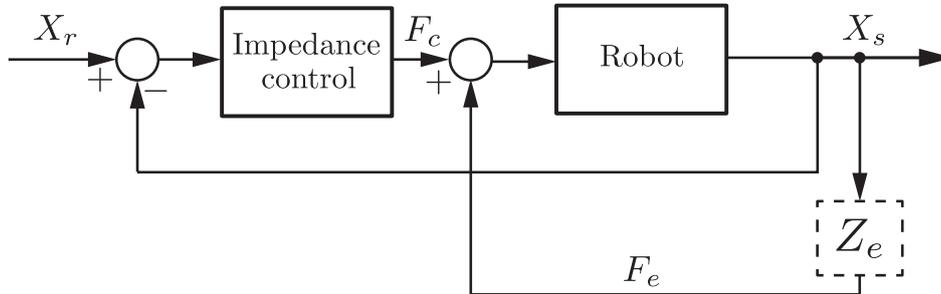


Figure 1-7: High level impedance control architecture.

1-2-2 Bilateral teleoperation

Like illustrated in the example (Section 1-1), robots are well suited to substitute humans in performing tasks in environments that are inaccessible or too dangerous for humans to operate in. Often these environments are only partially known and unstructured and thus robots can in general not be fully automated. In order to perform complex tasks in these environments, human input is still required. This can be achieved by having the operator controlling a remotely located slave robot by interacting with a master device which is directly accessible by the operator. These systems are called teleoperated systems. Teleoperation extends the human capability to perform complex tasks in a remote environment by providing the operator with conditions that are similar to the ones in the remote environment. For this purpose, not only visual but also haptic feedback, in which reaction forces from the remote environment are transmitted through the master device to the operator, is needed. This is achieved by transmitting force, position and velocity data bilaterally, both from the master to the slave and from the slave back to the master. When a teleoperation system provides haptic feedback to the operator it is called a bilateral teleoperation system. As illustrated in Figure 1-8, such a system can be divided into five subsystems, the human operator, the master device, the communication channel, the slave device and the environment [15]. Here the velocity flows through the ports and the force is measured across the ports.

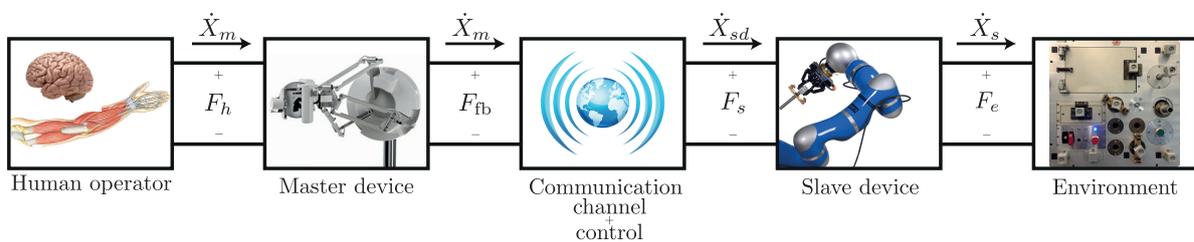


Figure 1-8: High level bilateral teleoperation interconnection of one-port and two-port network system representation.

The system equations are conventionally expressed in terms of velocities [6] but from this point on positions are used instead to facilitate the notion of position based goals. The

human operator commands the master device with a motion $X_m(s)$. The motion that the operator inflicts on the master device is transferred to the communication channel. From the communication channel a set-point for the slave motion $X_{sd}(s)$, is transferred to the slave device and from there the actual slave motion $X_s(s)$, is transferred to the environment. The force that the operator inflicts on the master device is given by $F_h(s)$. The reflected force from the slave robot to the master device is given by $F_{fb}(s)$. The controlled motor force provided to the slave robot is given by $F_s(s)$ and the force resulting from contact with the environment is given by $F_e(s)$. In addition to the network system representation, a high level system diagram of bilateral teleoperation can be seen in Figure 1-9.

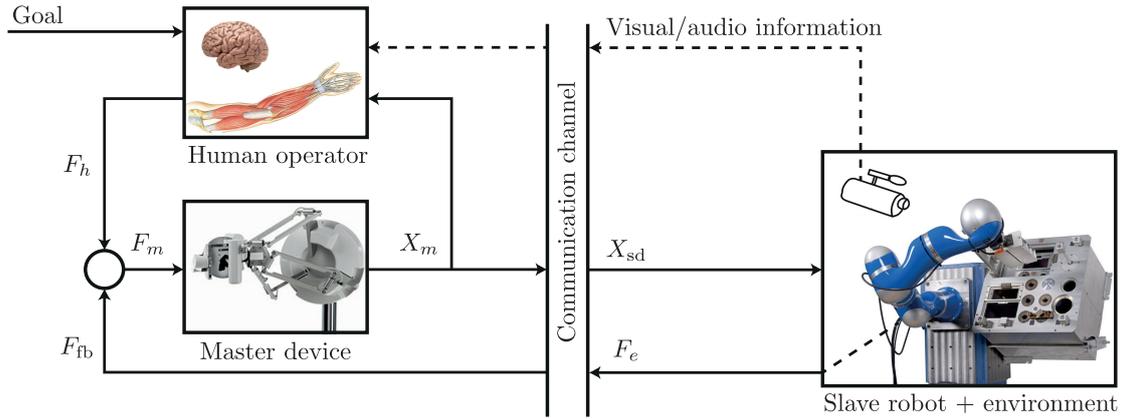


Figure 1-9: High level bilateral teleoperation control scheme. The operator acts upon the knowledge of some goal that is to be achieved and audio/visual information, by exerting forces on the master device $F_h(s)$. These forces inflict a motion of the master device $X_m(s)$, and this motion is transferred through the communication channel to the slave robot. The forces upon contact with the environment $F_e(s)$, are transferred through the communication channel to the master device, where the operator is able to feel the feedback force.

The overall dynamics of the teleoperation system are given by the general hybrid matrix formulation [16], with $H(s)$ the hybrid matrix:

$$\begin{bmatrix} F_h(s) \\ \dot{X}_m(s) \end{bmatrix} = \underbrace{\begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix}}_{H(s)} \begin{bmatrix} \dot{X}_s(s) \\ -F_e(s) \end{bmatrix} \quad (1-1)$$

The transmitted impedance to the operator, $Z_t(s)$ can be found by rewriting the general hybrid matrix formulation and finding the relation between $F_h(s)$ and $\dot{X}_m(s)$:

$$Z_t(s) = \frac{F_h(s)}{\dot{X}_m(s)} = (H_{11}(s) - H_{12}(s)Z_e(s))(H_{21}(s) - H_{22}(s)Z_e(s))^{-1} \quad (1-2)$$

Ideally the operator is not able to distinguish between the feeling of the remote environment and the real world. For the operator to experience interaction with the remote environment as if he is physically present, the impedance rendered to the operator by the master device, $Z_t(s)$, has to be the same as the impedance of the remote environment, $Z_e(s)$. The system is then called transparent and transparency is used as measure of the performance. The notion

of transparency was defined by Lawrence [6] in the early 1990's. Lawrence realized that in order to achieve a transparent system, the motions and forces both at the master and slave side must be equal over all frequencies:

$$\left. \begin{array}{l} \dot{X}_m(s) = \dot{X}_s(s) \\ F_h(s) = F_e(s) \end{array} \right\} \text{for all } s \quad (1-3)$$

This can be achieved for:

$$H(s) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (1-4)$$

Lawrence [6] as well as Yokokohji and Yoshikawa [17] have shown that for theoretical ideal transparency, a general four-channel architecture is required. The four-channel architecture can be seen in Figure 1-10. The hybrid matrix follows from the four-channel architecture:

$$\begin{aligned} H_{11}(s) &= (Z_m + C_m) (C_1 + C_3(Z_m + C_m))^{-1} (Z_s + C_s - C_3C_4) + C_4 \\ H_{12}(s) &= -(Z_m + C_m) (C_1 + C_3(Z_m + C_m))^{-1} (I - C_3C_2) - C_2 \\ H_{21}(s) &= (C_1 + C_3(Z_m + C_m))^{-1} (Z_s + C_s - C_3C_4) \\ H_{22}(s) &= -(C_1 + C_3(Z_m + C_m))^{-1} (I - C_3C_2) \end{aligned} \quad (1-5)$$

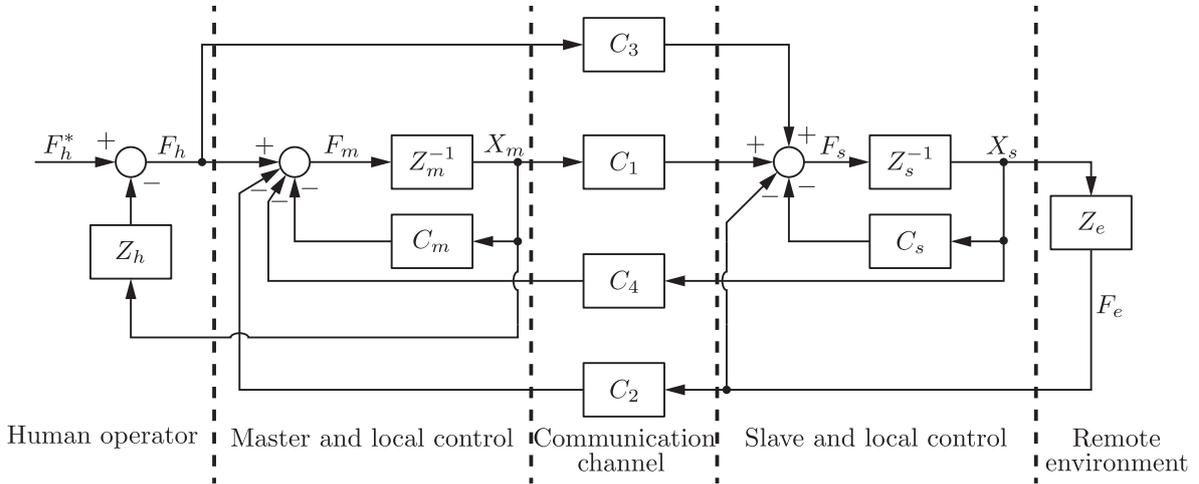


Figure 1-10: General four-channel architecture.

Here, $Z_m(s)$ and $Z_s(s)$ represent the dynamics of the master device and the slave robot respectively and $F_m(s)$ represents the combined force acting on the master device. The local controller on the master side is given by $C_m(s)$ and on the slave side by $C_s(s)$. Furthermore there are four communication channels ($C_1(s)$, $C_2(s)$, $C_3(s)$ and $C_4(s)$) that couple control for respectively motion forward, force backward, force forward and motion backward. The generality of this notation lies in the fact that this structure can also be used for two-channel

architectures. Details on the position-position and the position-force architecture can be found in Appendix A.

Although haptic feedback from this physical interaction with the environment has been shown to improve the task performance [18], it poses limitations on stability of the closed-loop system. Often a trade off has to be made in terms of system performance and task performance. However, another promising approach for improving task performance in teleoperation is discussed in Section 1-2-3.

1-2-3 Shared control

Sheridan was among the first to discuss what role computers would be playing in teleoperation [19, 20, 18]. Based on his work, control architectures are often categorised by the style and coupling between the operator and the slave robot. In bilateral teleoperation, which is discussed in Section 1-2-2, the operator is strongly coupled with the slave robot. This side of the spectrum is called direct control and the coupling is realized by the exchange of force and motion signals between the master device and the slave robot. Direct control requires no level of autonomy so the motion of the slave robot is directly controlled by the operator using the master device. The lack of autonomy and the inadequacies of the system can often make it tedious to perform complex tasks.

On the other side of the spectrum the coupling between the operator and the slave robot is realized by high-level directives that the operator provides to a computer which closes an autonomous control loop through actuators and sensors on the slave side. This is called supervisory control [19]. Environments that are partially known and unstructured are not suited for robots with a high level of autonomy to operate in since autonomous systems are incapable of dealing with large uncertainties [21].

In between these two extremes lies shared control. The idea of shared control is that the operator is kept in the direct loop, but a support system continually shares control over the system to be controlled, with the operator. The support system acts as an intelligent autonomous system and based on models from the task and the environment it is able to derive what the optimal input to the system to be controlled should be.

Shared control architectures

Abbink and Mulder [22] have pointed out a trend in existing shared control literature and defined two main approaches:

1. Input-mixing shared control (Illustrated in Figure 1-11)
2. Haptic shared control (Illustrated in Figure 1-12)

In input-mixing shared control, as illustrated in Figure 1-11, the input to the system that is to be controlled, $X_{\text{input}}(s)$, is a mix of both the desired input of the operator and the optimal input which is calculated by the support system. It must be noted that optimal input in this sense means the optimal input to the best knowledge of the support system, meaning that due to uncertainty it might not be the actual optimal input. In this architecture, the operator

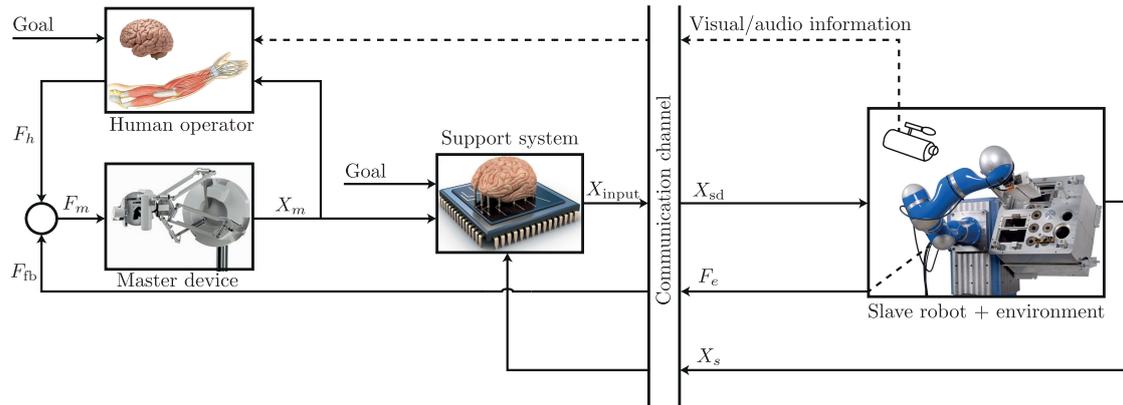


Figure 1-11: A schematic representation of an input-mixing shared control scheme. The motion from the master device $X_m(s)$, that is inflicted by the operator now enters the support system. The support system uses models of the task and the environment to determine an optimal input. The optimal input will then be ‘mixed’ with the operator inflicted input, which will determine the input $X_{input}(s)$, that is transferred through the communication channel to the slave robot.

has no influence over the actual input to the system that is to be controlled, and might not even be aware of what the support system is trying to achieve.

In haptic shared control, as illustrated in Figure 1-12, the optimal output of the support system is converted into guidance forces $F_g(s)$. These guidance forces act as an additional physical input on the master device and cause the system to perform the optimal output of the support system [23]. In this way the operator is aware of what the support system is trying to achieve and is able to choose whether to comply with these intentions (by becoming more compliant) or to stay in control and overrule the actions of the support system (by stiffening up), since the output of the master device still is the only input to the controlled system.

When designing a haptic shared controller, at least the following things should be considered:

1. The conversion of sensor information from the remote environment and slave robot states into an optimal control input.
2. The conversion of these optimal control inputs into guidance forces that act on the master device.
3. Matching the intentions of the support system with the intentions of the operator.

In order to be able to provide appropriate assistance, the intended goal of the operator and the goal that the support system uses, should be the same. If this is not the case, the haptic guidance provided by support system will only hinder the operator when executing the desired task. Furthermore, when the models that the support system uses represent the environment and task perfectly, there is no need for a human-in-the-loop and tasks can be performed autonomously. In practical applications however, this is often not the case. Due to the unstructured nature of the environment, uncertainties can result in a mismatch of the models with respect to the environment. This results in an error in reference of the haptic guidance that is provided to the operator.

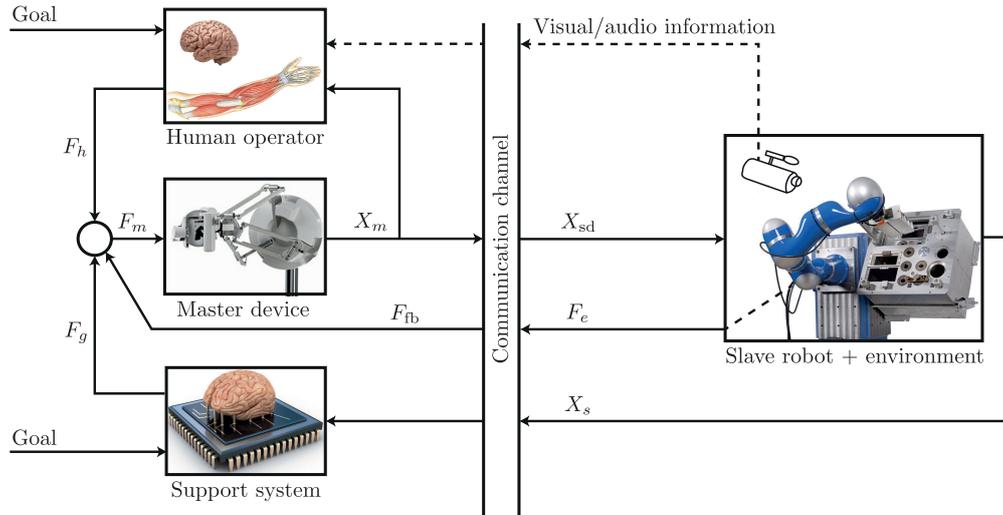


Figure 1-12: A schematic representation of a haptic shared control scheme. The support system uses models of the task and the environment to determine an optimal input. The optimal input is converted into a guiding force $F_g(s)$, that acts as a physical input on the master device and thus can be felt by the operator.

1-3 Problem description

Previous studies have shown that haptic guidance improves task performance in various applications, like tele-surgery [24, 25, 26, 13, 27, 28], remote assembly [29, 30, 31, 32, 7] and vehicle steering [33, 34]. However, the majority of these studies have been conducted under two assumptions. The first assumption being that there is only one task to be executed and this task has been communicated in advance to the support system, so the goals of the operator and support system are identical. In realistic scenarios, there are often multiple possible tasks present in the environment and the intentions of the operator are not known to the support system a priori.

Second, it is assumed that the models that are used by the support system are perfect representations of the environment, which therefore results in accurate haptic guidance. In practical applications, the environments are often partially known and unstructured resulting in a mismatch of the models with respect to the environment. This means that the support system determines an ‘optimal’ reference which is offset from the real goal in the environment. The guidance that is provided will therefore be inaccurate. Recent experimental work done by van Oosterhout et al. [10] found that depending on the magnitude and direction, uncertainty in haptic shared control can degrade task performance. However, no methods to robustify task performance against uncertainty are presented. The problem of this thesis can be formulated as:

Main problem:

Uncertainties can degrade the performance of haptic guidance systems

From the main problem, two categories can be distilled that both contribute to the total uncertainty, but have different sources. The uncertainty in goal reference which is considered

in the first category, contributes to the total uncertainty in the haptic guidance on a large scale and will therefore be referred to as **macroscale uncertainty**. The uncertainty in goal reference that is considered in the second category, contributes to the total uncertainty in the haptic guidance on a smaller scale and will therefore be referred to as **microscale uncertainty**.

Macroscale uncertainty

A key challenge in shared control is that the support system often does not know which task the operator intends to achieve a priori. Since the true intention of the operator is not a physical signal that can be interpreted by the support system, it can not be communicated to the support system without performing some additional actions. The operator is for example able to specify the intended task to the support system using a GUI or through voice commands.

In the case where the goal location of the support system is not the same as the location of intended goal, an error arises resulting in inaccurate guidance that is provided to the operator. This relatively large uncertainty in the haptic guidance, that can be in the order of magnitude of centimeters depending on the predicted goal location, will hinder the operator when reaching for the intended goal location. This sub-problem can be therefore be defined as follows:

Sub-problem 1:

Macroscale uncertainty represents a possible mismatch between the intended goal of the operator and the predicted goal of the support system, which results in a large error in the reference trajectory on which the haptic guidance is based.

This sub-problem is illustrated in Figure 1-13 in blue.

Microscale uncertainty

In practical teleoperation scenarios the support system does not always have perfect knowledge of the environment. Therefore, there will always be some uncertainty present in the models of the task and environment. These uncertainties can originate for example from elastic and/or plastic structural deformations or errors in sensor readings or control. The uncertainties give rise to errors in the models which result in a mismatch between the models and the real world. Since the support system uses these models to determine the reference trajectory on which the haptic guidance is based, the operator will be guided towards an inaccurate location in the real world. The study conducted by van Oosterhout et al. [10] found that the effects on overall task performance of a peg-in-hole assembly task, is dominated by the moment the peg is inserted due to its potential for jamming. Especially rotational offsets can cause the peg to jam in the hole. This sub-problem can be therefore be defined as follows:

Sub-problem 2

Microscale uncertainty represents a possible mismatch between the models that the support system uses and the real environment, which results in a small error in the reference trajectory that can degrade task performance.

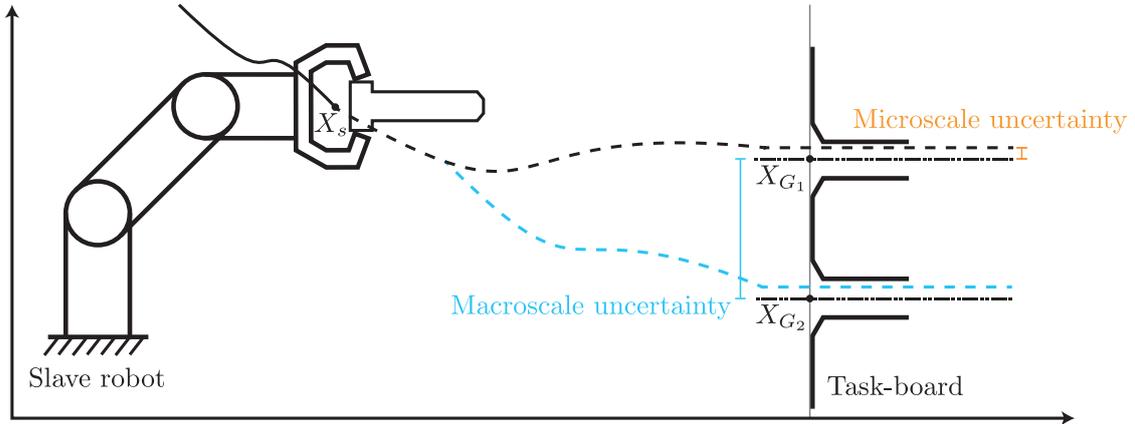


Figure 1-13: Schematic two dimensional representation of a peg-in-hole task where X_{G_1} is the intended goal. Macroscale uncertainty due to a mismatch in intentions between the operator and support system is depicted in blue. Microscale uncertainty due to a mismatch of the real environment and the models the support systems uses is depicted in orange.

This sub-problem is illustrated in Figure 1-13 in orange.

While the study conducted by van Oosterhout et al. [10] has shown that task performance is relatively robust against small uncertainty in haptic guidance, Smisek et al. [3] have identified another sub-problem that arises in the case of inaccurate haptic guidance. This problem is best visualised using a haptically guided peg-in-hole insertion example. Despite the inaccurate haptic guidance, the operator is able to overrule the forces and still complete the insertion, especially when there is a chamfer around the hole, as illustrated in Figure 1-14. The blue line represents the real center-line of the hole X_{hole} , and the red center-line represents the reference position affected by translational geometric modelling errors $X_g(s)$, that the support system uses in order to provide haptic guidance. The green line $X_s(s)$, represents the current position of slave robot and the orange line X_m , represents the input from the master device.

1) Since the reference position has not yet been met, the support system provides the master device with guidance forces: $F_g(s) = C_g(s)E_g(s)$ with $E_g(s) = X_g(s) - X_s(s)$ being the error between the inaccurate reference position and the current slave robot position and $C_g(s)$ the shared control gain. 2) The slave robot will move according to the input on the master device until it makes contact with the environment. 3) Since the inaccurate reference $X_g(s)$ is not met, the master device will continue pushing the operator which shifts $X_m(s)$ into the environment. This will result in a force acting on the environment, which is also reflected to the operator as $F_{fb}(s)$. 4) The increasing force feedback will reduce the effect of the force from the haptic guidance on the master device until the forces are in equilibrium, since they counteract each other. In steady state, the resulting force on the master device is zero. The uncertainty in the guidance is masked from the operator, while at the slave side the robot can still exert significant force F_e on the environment. This sub-problem is defined as:

Sub-problem 3:

Microscale uncertainty represents a possible mismatch between the models that the support system uses and the real environment, which results in a small error in the reference trajectory that masks undesired forces acting on the environment from the operator.

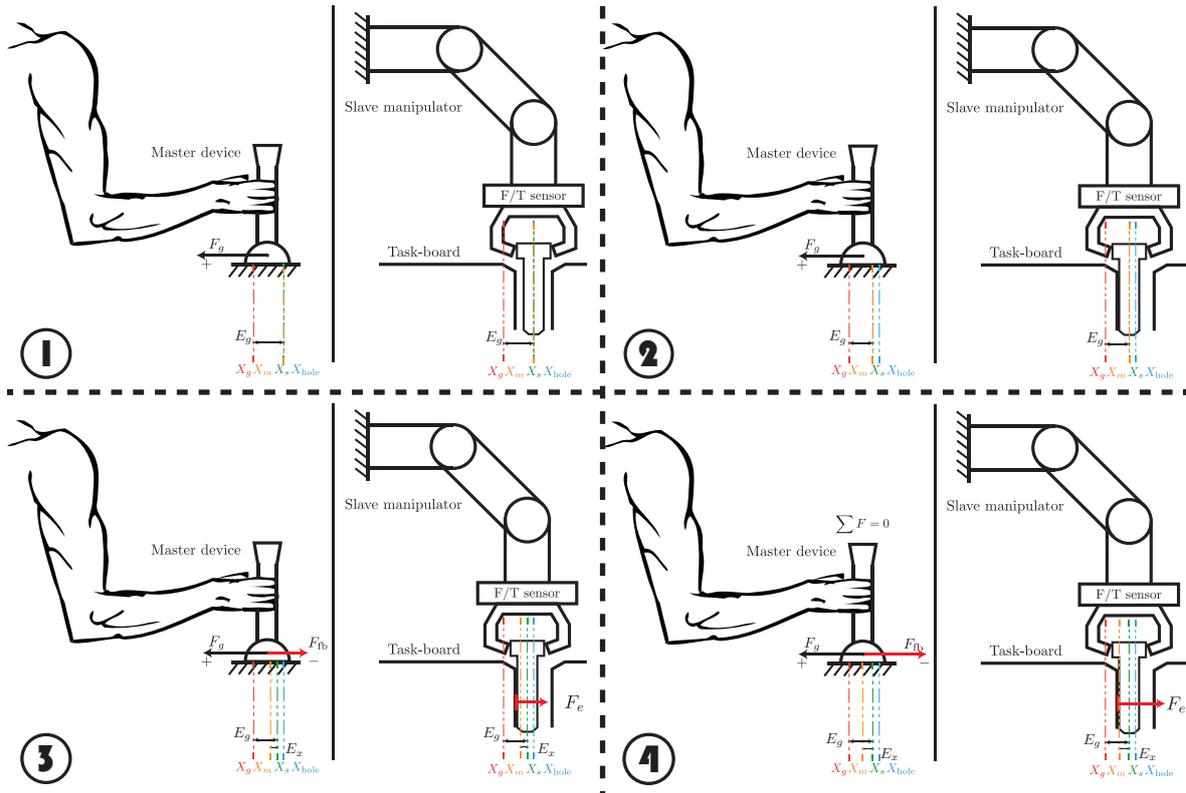


Figure 1-14: **1)** Since the reference position has not yet been met, the support system provides the master device with guidance forces $F_g(s)$. **2)** The slave robot will move according to the input on the master device until it makes contact with the environment. **3)** Since the inaccurate reference $X_g(s)$ is not met, the master device will continue pushing the operator which shifts $X_m(s)$ into the environment. This will result in a force acting on the environment, which is also reflected to the operator as $F_{fb}(s)$. **4)** The increasing force feedback will reduce the effect of the force from the haptic guidance on the master device until the forces are in equilibrium. In steady state, the resulting force on the master device is zero. The uncertainty in the guidance is masked from the operator, while at the slave side the robot can still exert significant forces $F_e(s)$ on the environment (adapted from [3]).

1-4 Goals

There are three goals in this thesis, that aim to solve the three sub-problems that have been defined in Section 1-3. The first goal is related to the first sub-problem:

Goal 1:

Conceive a method that reduces the macroscale uncertainty.

The goal of this method is to reduce the macroscale uncertainty until it is eliminated entirely. The elimination of the uncertainty would mean that macroscale error in the haptic guidance, which can be in the order of magnitude of centimeters, is also eliminated. The operator will be guided towards the goal location with only the small microscale uncertainty in haptic guidance present.

The second goal is related to the second sub-problem:

Goal 2:

Conceive a control method that robustifies task performance in the presence of microscale uncertainty.

The goal of this method is to robustify task performance by conceiving a control method that can cope with microscale uncertainty.

The third goal is related to the third sub-problem:

Goal 3:

Conceive an online method, that is able to update the model of the environment that the support system uses and reduces the microscale uncertainty.

The goal of this method is to reduce the microscale uncertainty more than the uncertainty that typically arises when using a commercial video motion capture system (Vicon [35]), which is in the order of magnitude of millimeters. By elimination of the macroscale uncertainty and reduction of the microscale uncertainty, the models can be updated online so that the magnitude of the incorrect guidance will be reduced and assembly tasks becomes less tedious for the operator to execute.

By achieving these goals, the operator will receive assistance while executing a task in a multi-task environment without having to manually communicate his intentions to the support system. Moreover, the performance of the peg-in-hole task in presence of microscale uncertainty will be improved and any forces acting on the environment will no longer be masked from the operator.

1-5 Approach

In order to accomplish these goals, different approaches are followed which are defined in this section.

Approach for goal 1:

In order to eliminate the macroscale uncertainty and provide the operator with haptic guidance towards the intended goal, the support system needs to be aware of the intended goal. Since the intention of the operator is not a physical signal that can be interpreted by the support system, it can not be communicated with the support system without performing some additional actions. The operator is for example able to specify the intended goal to the support system using a GUI or through speech. However, by utilising the control inputs of the operator to predict the intended goal, these additional actions can be skipped, which results in a more natural, seamless and faster teleoperation system. In this thesis, several prediction methods are validated and compared in simulation and by using real teleoperation data.

Approach for goal 2:

For an impedance controlled robot to still be able to insert the peg in presence

of microscale uncertainty, it is desired to have a low desired stiffness so the robot complies with the environment. However, a low stiffness results in poor position tracking in free air. A high stiffness results in better position tracking in free air. However, in contact with the environment, high stiffness results in high forces and therefore poor compliance. In this thesis, the impedance controlled robot is complemented with an outer admittance control loop. The outer admittance control loop changes the set point of the impedance controller based on the contact forces. This method is implemented and validated on an experimental setup by doing peg-in-hole insertion experiments.

Approach for goal 3:

In order to reduce the microscale uncertainty, the models that are used by the support system have to be updated online with new information from the environment. The admittance control loop changes the set point of the impedance controller by a correction term. This correction term can be extracted and used to update the models that are used by the support system online.

Macroscale uncertainty reduction by intention prediction

This chapter gives an overview and a performance comparison of intention prediction methods that were identified in the literature study [36]. First, Section 2-1 provides the theoretical background of the prediction methods. Section 2-2-1 covers the modelling and simulation of the operator and the implementation of the intention prediction methods. In Section 2-2-2, the experimental setup is described, on which data for the validation of the prediction methods is obtained. The results of the simulation and observational studies are presented in Section 2-3. Finally, the results are discussed in Section 2-4.

2-1 Approach: Intention prediction methods

In a realistic scenario, like the INTERACT task-board example that is presented in Section 1-1, there are often multiple tasks that the operator is able to execute. It is assumed that the model contains several tasks, expressed in Cartesian coordinates, which will be referred to as possible goals. These goals are contained in the set of accessible goals: $\mathcal{G} \in \{X_{G_1}, X_{G_2}, \dots, X_{G_i}\}$, with i the number of goals. One of these goals is the intended goal, $X_{G_{\text{intended}}}$. By looking at current and/or past configurations of the slave robot, the probability of the operator reaching for each of these goals can be calculated. The goal with the highest probability is most likely the goal that the operator is reaching for. Three methods to predict the intended goal of the operator will be treated in the following sections.

2-1-1 Amnesic intention prediction

In amnesic prediction [4], the probability that the operator is reaching for a goal is calculated for all possible goals. The probabilities are determined by only looking at the current configuration of the slave robot, resulting in the predicted goal $X_G^*(t)$:

$$X_{G^*}(t) = \arg \max_{X_{G_i} \in \mathcal{G}} P(X_{G_i} | X_s(t)) \quad (2-1)$$

$P(X_{G_i} | X_s(t))$ can be estimated by granting the highest probability to the goal location that has the smallest distance from the current slave robot configuration $X_s(t)$:

$$X_{G^*}(t) = \arg \min_{X_{G_i} \in \mathcal{G}} \|X_s(t) - X_{G_i}\| \quad (2-2)$$

The goal that is closest to the slave robot is determined to be the intended goal, as illustrated in Figure 2-1. X_{G_1} and X_{G_2} are two goal locations and $X_s(t)$ is the current slave robot configuration. Based on the amnesic prediction, X_{G_1} is the goal with the highest probability to be the intended goal since $d_1 < d_2$.

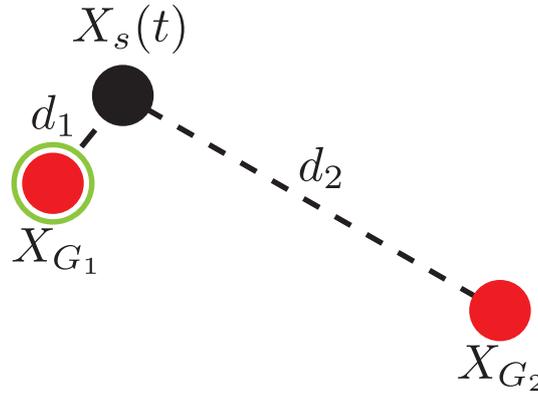


Figure 2-1: Schematic two dimensional example of amnesic prediction. Based solely on the current slave robot configuration, it can be concluded that the distance between X_{G_1} and the current slave robot configuration, d_1 , is smaller compared to the distance between X_{G_2} and the current slave robot configuration, d_2 , and thus X_{G_1} has the highest probability to be the intended goal (adapted from [4]).

2-1-2 Memory-based intention prediction

While amnesic prediction has the advantage of being straightforward to implement, it lacks the ability to predict the correct intentions of the operator in some cases. In the case that the operator intends to go to X_{G_2} but has to travel past X_{G_1} , as illustrated in Figure 2-2, amnesic prediction will predict the X_{G_1} to be the intended goal for the largest part of the trajectory. Only from point X_p and further, amnesic prediction is able to predict the correct goal.

In order to be able to predict the correct intentions in an earlier stage of the motion, amnesic prediction does not suffice. In memory-based prediction [4], besides using the current slave robot configuration also the set of past slave robot configurations ($\xi_{X_{\text{init}} \rightarrow X_s(t)}$) is taken into account:

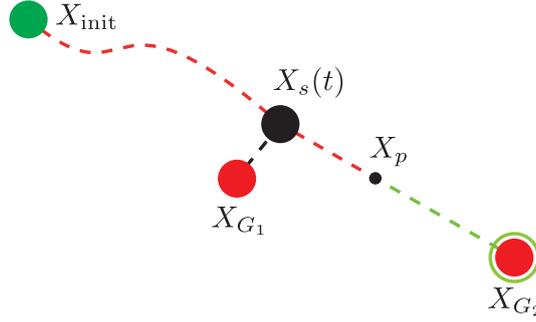


Figure 2-2: When the operator intends to go to X_{G_2} but has to travel past X_{G_1} , amnesic prediction will predict the X_{G_1} to be the intended goal for the largest part of the trajectory. From point X_p and further, amnesic prediction is able to predict the correct goal.

$$X_{G^*}(t) = \arg \max_{X_{G_i} \in \mathcal{G}} P(X_{G_i} | \xi_{X_{\text{init}} \rightarrow X_s(t)}) \quad (2-3)$$

In Appendix B, a derivation of memory-based prediction is presented. This derivation involves Bayes' theorem and Laplace's methods in order to approximate integrals. This derivation simplifies to the memory-based prediction method:

$$X_{G^*}(t) = \arg \max_{X_{G_i} \in \mathcal{G}} e^{-J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_s(t)})} \frac{e^{-J_{X_{G_i}}(\xi_{X_s(t) \rightarrow X_{G_i}}^*)}}{e^{-J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_{G_i}}^*)}} P(X_{G_i}) \quad (2-4)$$

Equation (2-4) evaluates the efficiency when the operator is going towards a goal, through the travelled trajectory $\xi_{X_{\text{init}} \rightarrow X_s}$ and the optimal trajectory that is to be travelled $\xi_{X_s(t) \rightarrow X_{G_i}}^*$, relative to the optimal trajectory towards that goal, $\xi_{X_{\text{init}} \rightarrow X_{G_i}}^*$. When the cost of the travelled trajectory $\xi_{X_{\text{init}} \rightarrow X_s(t)}$ and the optimal trajectory that is to be travelled $\xi_{X_s(t) \rightarrow X_{G_i}}^*$, is much higher compared to the cost of the optimal trajectory towards a certain goal $\xi_{X_{\text{init}} \rightarrow X_{G_i}}^*$, the goal is most likely not the intended goal of the operator. The assumption has been made that the slave robot configurations optimize a goal dependent cost function $J_{X_{G_i}}$. This is illustrated in Figure 2-3, where $\xi_{X_{\text{init}} \rightarrow X_{G_1}}^*$ (orange) and $\xi_{X_{\text{init}} \rightarrow X_{G_2}}^*$ (blue) represent the optimal trajectories from the initial position to respectively X_{G_1} and X_{G_2} and $\xi_{X_{\text{init}} \rightarrow X_s(t)}$ represents the trajectory that the operator has already travelled. Although X_{G_1} is closer to the current slave robot configuration, the travelled trajectory has been less costly towards X_{G_2} , which is the predicted goal.

2-1-3 Inferred intention prediction

The usage of a cost function can be advantageous, since they can be trained on the typical motion patterns of individual operators. However, the training of a cost function is time consuming and therefore not always desired. Inferred intention prediction [5] is a prediction method that eliminates the need for a cost function by looking at the accumulated piecewise

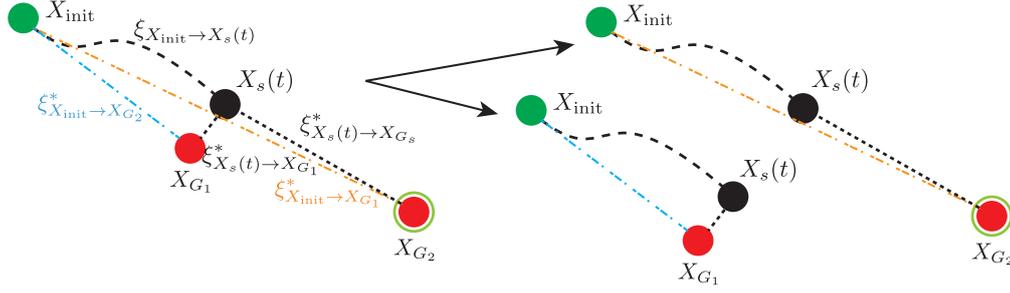


Figure 2-3: Schematic two dimensional example of memory-based prediction. $\xi_{X_{init} \rightarrow X_{G_1}}^*$ (orange) and $\xi_{X_{init} \rightarrow X_{G_2}}^*$ (blue) represent the optimal trajectories from the initial position to respectively X_{G_1} and X_{G_2} and $\xi_{X_{init} \rightarrow X_s(t)}$ represents the trajectory that the operator has already travelled. Although X_{G_1} is closer to the current slave robot configuration, the travelled trajectory has been less costly towards X_{G_2} , which is the predicted goal (adapted from [4]).

orthogonal point-to-line distance between the current operator input $X_s(t)$ and the optimal line from the starting input of the operator to each of the goals $\xi_{X_{init} \rightarrow X_{G_i}}^*$:

$$X_{G^*}(t) = \arg \min_{X_{G_i} \in \mathcal{G}} \sum_{t=0}^T \frac{\|(X_s(t) - X_{init}) \times (X_s(t) - X_{G_i})\|}{\|X_{init} - X_{G_i}\|} \quad (2-5)$$

Equation (2-5) infers that the intended goal is chosen by looking at the deviation of the travelled trajectory from the optimal goal trajectories. The further the deviation for a given goal, the less likely that goal is the intended goal, as illustrated in Figure 2-4.

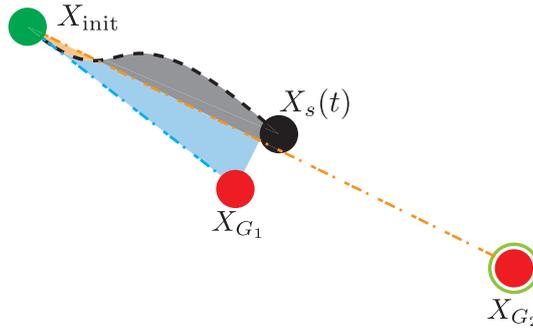


Figure 2-4: Schematic two dimensional example of inferred intention prediction. Although the X_{G_1} is closer to the current slave robot configuration, X_{G_2} is predicted to be intended by the operator. This is based on the deviation of the optimal trajectory for each goal and the sequence of past slave robot configurations. The light blue area denotes the accumulated deviation of the optimal trajectory of X_{G_1} from the sequence of past slave robot configurations, while the light orange area denotes the accumulated deviation of the optimal trajectory of X_{G_2} from the sequence of past slave robot configurations. The grey area denotes the shared accumulated deviation of the optimal trajectories of both X_{G_1} and X_{G_2} from the sequence of past slave robot configurations. The light blue covers a greater area than the light orange and is therefore less likely to be the intended goal of the operator (adapted from [5]).

2-2 Validation methods

In this section, two methods will be used to validate the prediction methods that were explained in the previous section. First, a simulation study is done. Second, the methods are validated off-line on data from a previous human-in-the-loop teleoperation study.

2-2-1 Simulation study

A simulation study was conducted as a first step in validating the intention prediction methods. The teleoperation system and the human that operates the teleoperation system are modelled and used in simulated reaching tasks in order to get first insights.

Assumptions

The models are subject to following assumptions that have been made in order to simplify them for implementation.

1. All controllers and teleoperator components are modelled as linear time-invariant systems.
2. The model of the environment is known beforehand. The model contains the locations of multiple tasks that can be executed by the operator. The operator is reaching for one out of four possible goal locations that are located on the task-board. The environment is free of obstacles, so the slave robot can move freely without chance of any collision.
3. The initial position of the trajectory is randomly chosen in a certain vicinity above the task-board. The operator begins operating the master device at $t = 0$, so the initial velocity and acceleration are assumed to be zero. The final position is randomly chosen from a discrete set containing these four goal locations. Since only a reaching motion is considered here, the final velocity and acceleration of the operator are zero.
4. The operator moves between the initial and final positions in a ‘natural manner’, or employs a minimum jerk motion strategy where the velocity profile of the motion is bell-shaped.

Teleoperation model

The teleoperation architecture is modelled as the four channel architecture as in [6]. Since no contact is considered in this simulation, the environment has been left out, resulting in the architecture that is illustrated in Figure 2-5. All signals and dynamical systems are given in Laplace domain. Where not deemed necessary and in figures, the Laplace s was left out for clarity.

The commanded position and force gains are set to $C_1 = C_s = 1000$ N/m and $C_3 = 1$. The position feedback gains are set to $C_4 = -C_m = 500$ N/m. The operator model ($Z_h(s)$) is discussed in the next section.

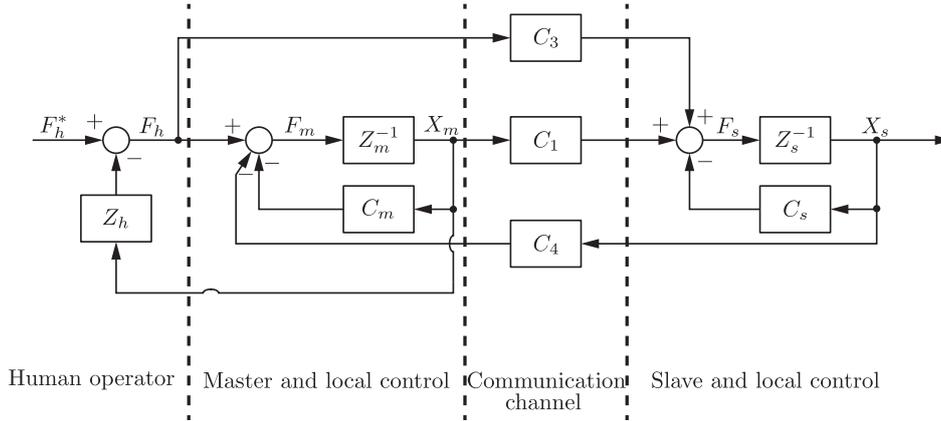


Figure 2-5: Architecture of the modelled teleoperation system, which is based on the four channel teleoperation architecture [6]. The environment is not considered here.

Operator model

The operator is not a part of the teleoperation system, however when the operator supplies input to the master device it mechanically coupled with the teleoperation system and affects the dynamics of the entire system. The inputs of the operator determine the configuration of the slave robot, which is used as input for the different prediction methods. For this model it is assumed that the operator provides human-like motions as input, that employ the minimum jerk strategy [37, 38]. The planning of a reference trajectory takes place in the brains of the operator. The reference trajectory represents a natural motion from initial position (x_0, y_0, z_0) to end position (x_f, y_f, z_f) that starts at time $t=0$ and end at time $t=T$, optimizes the following cost function:

$$C(t) = \frac{1}{2} \int_0^T \left(\left(\frac{d^3 x(t)}{dt^3} \right)^2 + \left(\frac{d^3 y(t)}{dt^3} \right)^2 + \left(\frac{d^3 z(t)}{dt^3} \right)^2 \right) dt \quad (2-6)$$

The trajectory minimizes the integral of the square of jerk magnitude. The initial position is randomly chosen inside an area above the task-board. The end position is randomly chosen to be one out of four possible goals, that are situated on the task-board, as illustrated in Figure 2-9. Furthermore, it is assumed that the initial velocity and acceleration are zero, since this is the starting point of the motion. The final velocity and acceleration are also assumed to be zero. An exemplary minimum jerk motion is illustrated in Figure 2-6, where **(a)** depicts the position trajectory, **(b)** the velocity trajectory and **(c)** the acceleration trajectory.

The minimum jerk motion results in a straight trajectory from the initial position to the final position, however in real teleoperation tasks, it is hard for the operator to perform straight line motion. Therefore a tremor is modelled by adding Zero Mean White Noise (ZMWN) that gradually increases its amplitude with increasing velocity to the reference motion. The noise is filtered to get rid of the high frequent behaviour, this results in a trajectory as illustrated in Figure 2-7. The operator is modelled as a stiffness K_h . The more the operator contracts its muscles, the stiffer he becomes and the better the tracking of the reference trajectory. The block diagram of the model of the operator is illustrated in Figure 2-8.

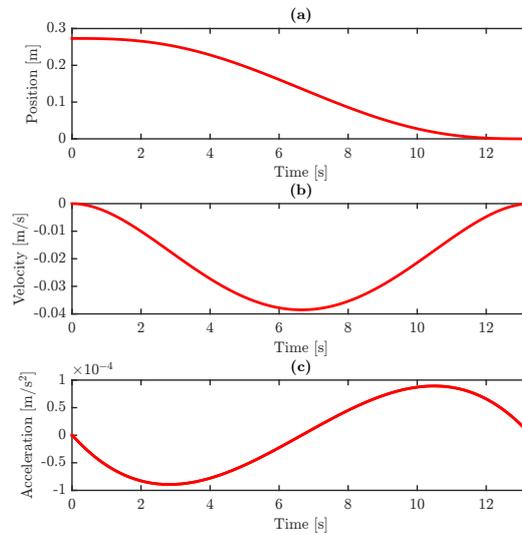


Figure 2-6: Minimum jerk motion, where **(a)** depicts the position trajectory, **(b)** the velocity trajectory and **(c)** the acceleration trajectory.

The resulting trajectory represents a trajectory coming from the operator, an exemplary trajectory is illustrated in Figure 2-9. The trajectory is used as input for the master device.

Task

In simulation, the operator is tasked to perform a reaching motion from a randomly selected starting position to one out of four possible goals. The motion is modelled as a minimum jerk motion with ZMWN added in order to simulate human tremor on the trajectory, as presented above. The simulation stops after the end time of the reference trajectory, T , is reached. A total of 1000 reference trajectories have been simulated.

2-2-2 Experimental study

An experimental study was conducted as a second step in validating the intention prediction algorithms, using data from a previous human-in-the-loop teleoperation study by Kimmer et al. [7]. In the following sections the experimental setup that has been used will be explained and the task of the operator will be explained.

Experimental setup

The experimental setup consists of a haptic master device, a slave robot holding a peg and the task-board which contains the goal locations. The slave robot that is used is the KUKA LWR 4+ robot [8], mounted with a 6-DOF ATI force/torque sensor and a Robotiq parallel gripper. The gripper is holding a 155 mm long peg with a diameter of 14.0 mm. The slave robot is operated in impedance mode with a programmed Cartesian stiffness of $K_s = 500$

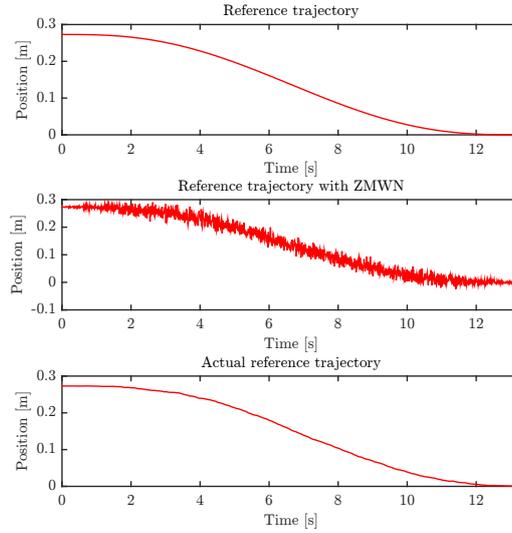


Figure 2-7: Minimum jerk motion position trajectories, where **(a)** depicts the reference trajectory, **(b)** the reference trajectory with added ZMWN and **(c)** the filtered reference trajectory that acts as an input on the master device.

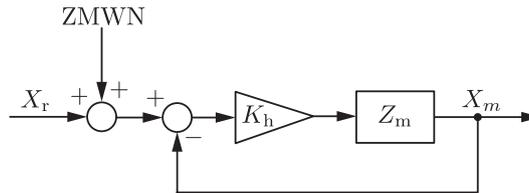


Figure 2-8: The minimum jerk reference trajectory is complemented with ZMWN. The master device filters the high frequent behaviour before it is transmitted to the slave robot. The feedback loop makes sure that the reference trajectory $X_r(t)$ is followed.

N/m for translations and $K_s = 20$ Nm/rad for rotations. The slave robot is controlled with a Force Dimension Sigma.7 master device. The task-board contains four holes for the peg to be inserted in, they correspond to the goal locations. The hole axes are parallel and separated from each other with a distance of 80 mm in both x and y directions. The operator is not able to see what he is doing directly, but visual feedback is provided with a LG liquid-crystal display which has a diameter of 42 inches and has fullHD resolution. The centreline of the camera image is equal to the centre of the four holes, ensuring that the amount of visibility in both x and y directions is equal. The setup can be seen in Figure 2-10.

Task

Eight different operators are tasked with the following instruction: *Using the robot manipulator, insert the peg in the highlighted hole while exerting as little force as possible on the task-board. One insertion must be done within 45 seconds.* The operator controls the master device with his right hand. Before each trial, the operator is instructed to move 20 cm away from the task-board. Once the distance has been reached, a randomly selected hole is selected

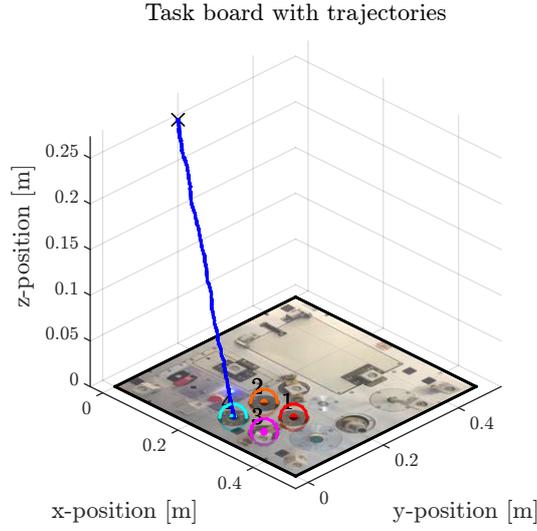


Figure 2-9: Example of a three dimensional minimum jerk trajectory with simulated tremor.

and communicated with the operator and the trial begins. The position furthest away from the task board is chosen to be the initial position of that trial. The time limit is used so the operator uses similar execution strategies throughout the experiment. The trial is ended when the operator makes contact with the task-board, or when the slave reaches a position threshold in z direction of -1 mm.

Independent variables

The independent variables are the different prediction methods described in Section 2-1.

For memory based prediction, two different cost functions have been used. The first one is based on the distance between the optimal trajectory from the current slave position to the goals and the optimal trajectory from the starting position to the goals. The use of this cost function will be referred to as: *Memory - distance*.

$$\begin{aligned}
 J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_s(t)}) &= \int_0^T |\dot{X}_s(t)| dt & (2-7) \\
 J_{X_{G_i}}(\xi_{X_s(t) \rightarrow X_{G_i}}^*) &= |X_{G_i} - X_s(t)| \\
 J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_{G_i}}^*) &= |X_{G_i} - X_s(0)|
 \end{aligned}$$

The second cost function is based on the direction and velocity of the slave with respect to the goals. The use of this cost function will be referred to as: *Memory - velocity*.

$$\begin{aligned}
 J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_s(t)}) &= \int_0^T |\dot{X}_s(t)| dt & (2-8) \\
 J_{X_{G_i}}(\xi_{X_s(t) \rightarrow X_{G_i}}^*) &= \int_0^T \frac{X_{G_i} - X_s(t)}{|X_{G_i} - X_s(t)|} \cdot \dot{X}_s(t) dt \\
 J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_{G_i}}^*) &= \int_0^T \frac{X_{G_i} - X_s(0)}{|X_{G_i} - X_s(0)|} \cdot \dot{X}_s(t) dt
 \end{aligned}$$

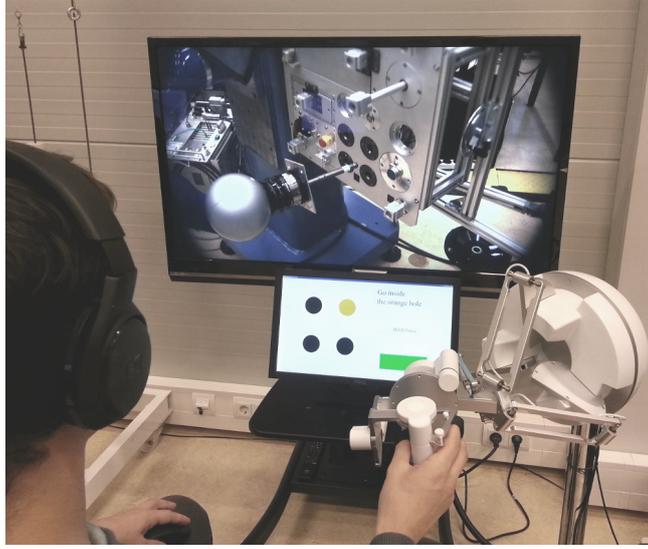


Figure 2-10: Experimental setup, where the operator interacts with the master device to operate the slave robot which can be seen in the display (from [7]).

Dependent variable

The performance of the prediction is measured by the dependent variable. The dependent variable is evaluated by looking at what percentage of the total trajectory, the correct goal has been identified, as illustrated in Figure 2-13 in time domain. This is referred to as the ‘correctness measure’. This metric is calculated as follows:

$$\text{Correctness} = \frac{100}{\xi_{X_{\text{init}} \rightarrow X_s(T)}} \int_0^T \phi(X_G^*, X_{G_{\text{intended}}}) dt \quad (2-9)$$

where:

$$\phi(X_G^*, X_{G_{\text{intended}}}) = \begin{cases} X_s(t) & , \text{ if } X_G^* = X_{G_{\text{intended}}} \\ 0 & , \text{ if } X_G^* \neq X_{G_{\text{intended}}} \end{cases}$$

2-3 Results

The presented results show the percentage of the total trajectory the correct goal has been identified for simulated reaching trajectories in Figure 2-11 and for real teleoperated trajectories in Figure 2-12, in the form of box plots. The box plots show the median, which are connect to each other using a line and the 25th and 75th percentiles, respectively q_{25} and q_{75} . The whiskers extends up to the most extreme data point that is not an outlier. A data point is considered an outlier if $x > q_{75} + w(q_{75} - q_{25})$ or if $x < q_{25} - w(q_{75} - q_{25})$, where $w = 1.5$. Outliers are depicted in the boxplots as circles. Additionally, triangles are plotted where the range between the centers show the 95% confidence interval. Figure 2-13 illustrates the goal prediction in time domain for different methods.

It can be observed when using real data, the performance of amnesic prediction increases, while memory based methods show a slight decrease and inferred prediction shows a large decrease in performance. Moreover, the performance of real data is more spread compared to simulated data. For both simulated data, memory based prediction methods show best performance and between them there is no significant difference. For real data, the memory based prediction methods and the amnesic prediction show no significant difference. Furthermore it can be noted that for all prediction methods using real data, the whiskers of the boxplot reach up to 0%, except for amnesic prediction.

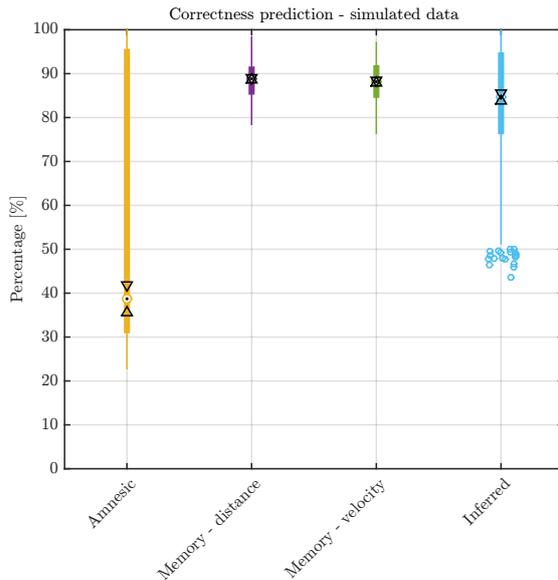


Figure 2-11: Percentage of the entire trajectory that the goal is correctly predicted on simulated data.

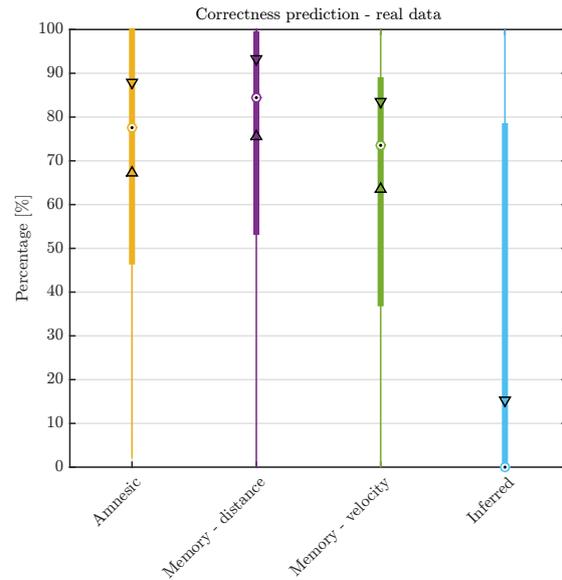


Figure 2-12: Percentage of the entire trajectory that the goal is correctly predicted on real teleoperation data.

2-4 Discussion

Memory based prediction methods show better performance compared to other methods

While amnesic prediction in most cases only predicted the intended goal at the very end, memory based prediction methods were able to predict the intended goal in an earlier stage of the motion. This is however not always the case, when for example the operator first goes for the wrong goal, the cost function is gaining value for that particular goal. If the operator at the last moment decides to go for another goal, the amnesic prediction is able to predict this goal sooner compared to other methods.

Prediction on real teleoperation data shows different performance compared to simulated data

This is the result of assumptions that were made in the modelling phase, that turn out to be different in a real scenario. The motion of the operator does not follow the minimal jerk

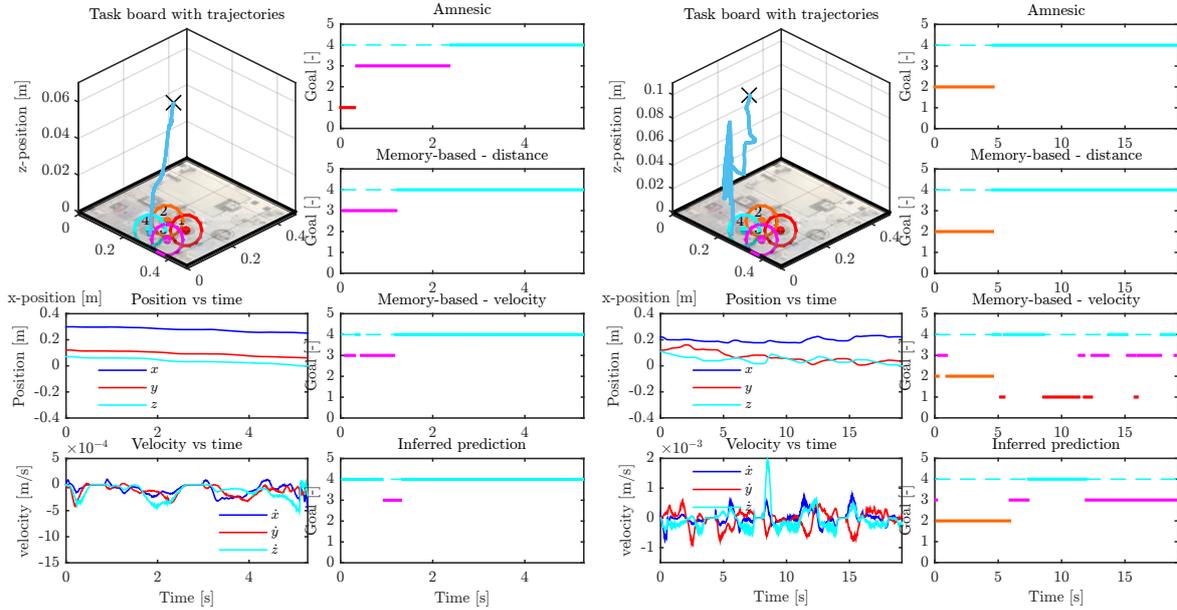


Figure 2-13: Exemplary trajectories from real data to show the goal prediction for different methods in time domain. **Left in each figure:** 3D plot of the trajectory, including the positions and velocities in x , y and z directions. **Right in each figure:** The predicted goal for different methods over time for the trajectory that is depicted on the left. The striped line denotes the intended goal.

reaching trajectory as suspected and illustrated in Figure 2-13. Moreover, the trajectory of a real operator does not describe the modelled straight line with added ZMWN trajectory. Another assumption that does not hold is that the operator does not always reaches a goal location, but often ‘misses’ the goal at first after which a motion away from the task-board is made in order to retry reaching the goal.

The larger spread in real data prediction can also be explained by the fact that some of the assumptions that were made don’t hold in real teleoperation scenarios.

Less space between goals would result in worse prediction

For both real as simulated data prediction performance decreases if the space between goals decreases. For amnesic methods this would mean that a smaller change in distance is needed to predict the wrong goal and for memory-based and inferred methods, the costs for the different goals would show less difference between them so a smaller deviation is needed for the cost of the wrong goal to take over.

Prediction on pre-generated data

In a real scenario the support system would start providing haptic guidance based on the outcome of the prediction. The guidance, for both correct as incorrect predicted goals, should

influence the trajectory of the slave robot. This directly influences the prediction. Depending on how the haptic guidance is provided this could mean that if the intended goal is incorrectly predicted, the guidance pushes the user towards this goal, fortifying the predicted intended goal.

Robustifying task performance in presence of microscale uncertainty

In this chapter, two of the sub-problems related to microscale uncertainty, will be addressed. First, the problem will be explained in more detail, in Section 3-1. Secondly, outer loop admittance control is proposed as an approach for this problem in Section 3-2. In Section 3-2-1, the closed-loop system will be analysed by looking at theoretical stability bounds and steady-state behaviour. This is followed by a simulation study in Section 3-2-2, where the performance of the closed-loop system is evaluated. The method is validated by experiments, which are treated in Section 3-3-1 followed by the results in Section 3-3-3. Finally, the results are discussed in Section 3-4. Also for this chapter the Laplace s was left out for clarity, if not deemed necessary and in figures.

3-1 Problem

In general, assembly tasks require the robot to be in contact with the environment. Impedance control aims at controlling the interaction between the robot and the environment by controlling a desired dynamic relation between them, as highlighted in Section 1-2-1. The impedance controlled KUKA LWR 4+ robot is illustrated in Figure 3-1.

For simplicity the interaction that the operator has with the master device is not considered here and the position of the master device, $X_m(s)$, is considered to be the reference trajectory. The desired dynamic relation between the robot and the environment is often chosen to be a linear second order system, because of the well known behaviour of such a system. This can be represented as follows:

$$C_s(s) (X_m(s) - X_s(s)) = F_e(s) \quad (3-1)$$

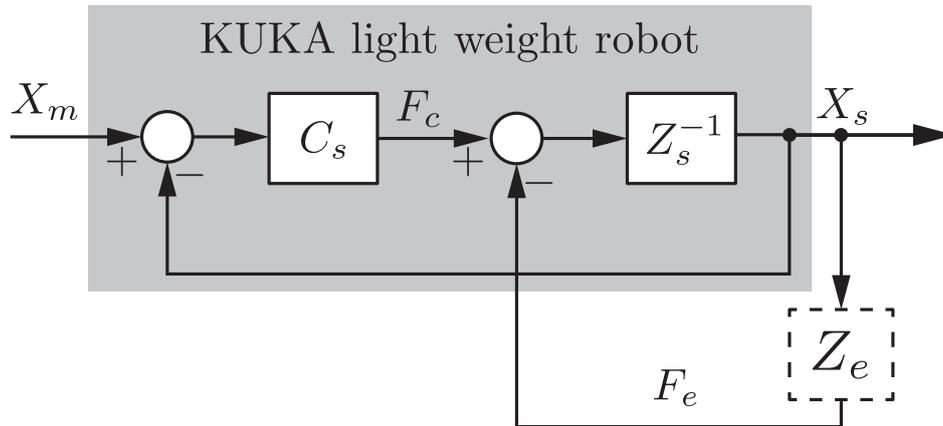


Figure 3-1: High level impedance control architecture.

With $C_s(s)$ the desired impedance:

$$C_s(s) = M_s s^2 + B_s s + K_s \quad (3-2)$$

Here M_s represents the desired mass, B_s the desired damping and K_s the desired stiffness of the robot. The forces are obtained by using a force-torque sensor, which is normally attached to the wrist of the robot. When performing a peg-in-hole task, uncertainties can cause a mismatch between the model of the environment and the real environment, together with friction and other unmodelled behaviour of the manipulator, these uncertainties result into errors in the reference that can be divided into translational and rotational offsets, as illustrated in Figure 3-2

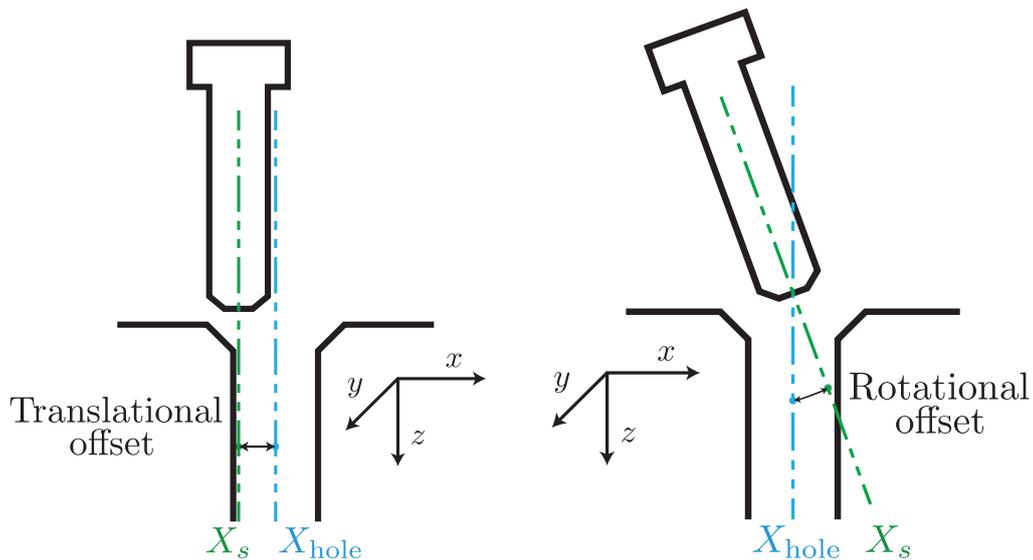


Figure 3-2: **Left:** Example of a translational offset in $-x$ direction. **Right:** Example of a rotational offset around the y axis.

In previous research [39] mating of rigid parts has been described and successful insertion of depends greatly on the forces and moments applied to the peg. Jamming of the peg occurs

if the forces and moments that are being applied to the peg are not in the right proportion resulting in the peg being stuck in the hole. If wedging occurs, the peg also appears to be stuck in the hole. However, wedging has a geometrical cause where the friction between the parts plays an important role.

For the robot to still be able to insert the peg in presence of uncertainty, it is desired to have a low stiffness of the robot so it complies with the environment to prevent jamming. However, a low stiffness results in poor position tracking in free air. A high stiffness results in better position tracking in free air. However, in contact with the environment, high stiffness results in high forces and therefore poor compliance which result in jamming of the part. So in practical applications, like assembly tasks, impedance control has limited performance.

3-2 Approach: Pseudo-admittance control

If Equation (3-1) and Equation (3-2) are combined, the target impedance formula is revealed:

$$M_s \left(\ddot{X}_m(t) - \ddot{X}_s(t) \right) + B_s \left(\dot{X}_m(t) - \dot{X}_s(t) \right) + K_s (X_m(t) - X_s(t)) = F_c(t) \quad (3-3)$$

This shows the dynamic behaviour between the force and the target impedance. In steady state, the output of the impedance controller, $F_c(t)$ is given by:

$$\lim_{t \rightarrow \infty} F_c(t) = K_s (X_m(t) - X_s(t)) \quad (3-4)$$

The environment, in contact with the manipulator, can be modelled by a linear spring-damper model [40]:

$$F_e(t) = B_e \dot{X}_s(t) + K_e (X_s(t) - X_e) \quad (3-5)$$

Here B_e represents the damping of the environments, K_e the stiffness of the environment and X_e the position of the environment which in steady state results in:

$$\lim_{t \rightarrow \infty} F_e(t) = K_e (X_s(t) - X_e) \quad (3-6)$$

Since in steady-state $\lim_{t \rightarrow \infty} F_c(t) = \lim_{t \rightarrow \infty} F_e(t)$, Equations (3-5) and (3-6) can be combined, resulting in:

$$\lim_{t \rightarrow \infty} F_e(t) = \frac{K_s K_e}{K_s + K_e} (X_m(t) - X_e) \quad (3-7)$$

Here $\frac{K_s K_e}{K_s + K_e}$ is equivalent to two springs, with different stiffness, connected in series.

Jamming can be prevented by minimizing the force upon contact. From Equation (3-7) it can be seen that contact forces can be influenced by either changing the target stiffness, K_s , or the reference trajectory, $X_m(t)$.

In literature, two main approaches to achieve good position tracking in free air, as well as compliant behaviour in stiff contact, can be identified:

1. **Adaptive impedance control.** Methods that change the impedance parameters, K_s and B_s , based on the contact forces.
2. **Force tracking impedance control.** Methods that change the set-point, $X_m(t)$, of the impedance controller based on contact forces.

Adaptive impedance control

Adaptive impedance control is able to achieve these two conflicting objectives by changing the impedance parameters on-line, resulting in desired dynamical behaviour for different situations. Lee and Buss [41] proposed a method that follows human-like approach, where the impedance parameters are adapted according to force tracking errors as is illustrated in Figure 3-3. Here $F_d(s)$ represents the desire force to be tracked.

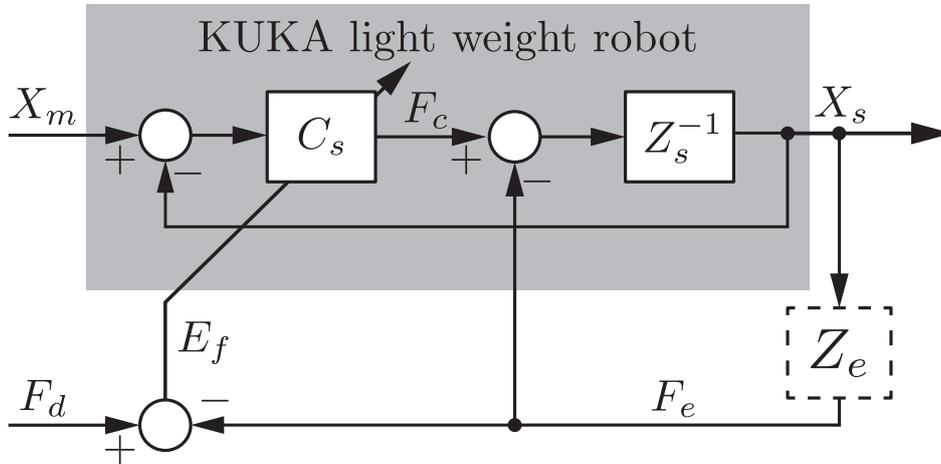


Figure 3-3: High level adaptive impedance control architecture.

Force tracking control

In [42, 43, 44] force tracking is achieved with an impedance controlled manipulator by generating a reference motion that is based on contact forces and torques. In [45, 46] a industrial position controlled manipulator is equipped with a six DOF controlled impedance device that modifies the desired position trajectory to achieve force tracking. Moreover, peg-in-hole assembly tasks have been successfully executed.

An example of the transient of $\hat{X}_e(s)$ can be seen in Figure 3-18 in the results section. $\hat{X}_e(s)$ can be used to update the model that is used by the haptic shared controller online in order to improve haptic guidance that is provided to the operator.

3-2-1 Theoretical analysis

In this section, the closed loop stability bounds, as illustrated in Figure 3-4, are assessed using the Routh-Hurwitz stability criterion [47]. Moreover, the steady state-error is evaluated to gain insight in the closed-loop system performance.

$$X_c = Y F_e \quad (3-10)$$

$$F_e = Z_e X_s \quad (3-11)$$

$$X_s = Z_s^{-1} F_s \quad (3-12)$$

$$F_s = C_s X_m - C_s X_s - C_s X_c - F_e \quad (3-13)$$

$$X_s = \frac{C_s}{Z_s + C_s} X_m - \frac{C_s}{Z_s + C_s} X_c - \frac{1}{Z_s + C_s} F_e \quad (3-14)$$

$$F_e = \frac{Z_e C_s}{Z_s + C_s + Z_e} X_m - \frac{Z_e C_s}{Z_s + C_s + Z_e} X_c \quad (3-15)$$

$$X_c = \frac{Y Z_e C_s}{Z_s + C_s + Z_e + Y Z_e C_s} X_m \quad (3-16)$$

For the position steady state error, the transfer function $\frac{E_x(s)}{X_m(s)}$, with $E_x(s) = X_m(s) - X_s(s)$ is evaluated with the final value theorem [48] for a unit step.

$$E_x(s) = \frac{1}{Z_s(s) + C_s(s) + Z_e(s) + Y(s)Z_e(s)C_s(s)} X_m(s) \quad (3-17)$$

$$e_{ss,x} = \lim_{s \rightarrow 0} \{s E_x(s)\} = \lim_{s \rightarrow 0} s \frac{1}{Z_s(s) + C_s(s) + Z_e(s) + Y(s)Z_e(s)C_s(s)} \frac{1}{s} = 0$$

The force steady state error can be found with the same approach, but now by looking at the transfer function $\frac{E_f(s)}{X_m(s)}$.

$$E_f(s) = \frac{C_s(s)Z_e(s)}{Z_s(s) + C_s(s) + Z_e(s) + Y(s)Z_e(s)C_s(s)} X_m(s) \quad (3-18)$$

$$e_{ss,f} = \lim_{s \rightarrow 0} \{s E_f(s)\} = \lim_{s \rightarrow 0} s \frac{C_s(s)Z_e(s)}{Z_s(s) + C_s(s) + Z_e(s) + Y(s)Z_e(s)C_s(s)} \frac{1}{s} = 0$$

For the stability analysis, the closed loop transfer function $\frac{X_s(s)}{X_m(s)}$ is evaluated.

$$\frac{X_s(s)}{X_m(s)} = \frac{C_s(s)}{Z_s(s) + C_s(s) + Z_e(s) + Y(s)Z_e(s)C_s(s)} \quad (3-19)$$

Table 3-1: Routh-Hurwitz array of $\frac{X_s(s)}{X_m(s)}$.

s^3	$M_{\text{slave}}B_y$	$(K_s + K_e)B_y + B_sK_e + B_eK_s$
s^2	$(B_s + B_e + B_{\text{slave}})B_y + B_sB_e$	K_sK_e
s^1	R	0
s^0	K_sK_e	0

By filling in for $C_s(s) = B_s s + K_s$, $Z_s(s) = M_{\text{slave}} s^2 + B_{\text{slave}} s$, $Z_e(s) = B_e s + K_e$ and $Y(s) = \frac{1}{B_y s}$, the Routh-Hurwitz array is constructed as can be seen in Table 3-1. The slave robot is modelled as a mass damper system with M_{slave} the mass and B_{slave} the damping.

For stability, the necessary and sufficient condition is that all of the coefficients in the first column must be positive in order for all poles to be in the left-half plane. From this analysis, two bounds on the admittance control gain B_y , can be extracted by looking at the first columns of s^3 and s^2 . It is assumed that $B_s, K_s, B_e, K_e, M_{\text{slave}}, B_{\text{slave}} > 0$.

$$s^3 : B_y > 0 \quad (3-20)$$

$$s^2 : B_y > -\frac{B_s B_e}{B_s + B_e + B_{\text{slave}}} \quad (3-21)$$

The first bound found on B_y is a tighter bound compared to the second, so the second bound can be discarded. For more bounds, R in the first column of s^1 needs to be investigated. R is given by:

$$\underbrace{(K_s + K_e)}_a B_y - \frac{\overbrace{M_{\text{slave}} K_s K_e}^b B_y}{\underbrace{(B_s + B_e + B_{\text{slave}})}_c B_y + \underbrace{B_s B_e}_d} + \underbrace{B_s K_e + B_e K_s}_e > 0 \quad (3-22)$$

From Equation (3-22), bounds on B_y can be derived under certain conditions of the other parameters of the system.

$$\text{In the case that: } K_s + K_e \geq \frac{M_{\text{slave}} K_s K_e}{B_s B_e} \quad (3-23)$$

$$B_y > 0$$

$$\text{In the case that: } 0 < K_s + K_e < \frac{M_{\text{slave}} K_s K_e}{B_s B_e} \text{ and } e > \frac{-2c\sqrt{\frac{abd}{c^2}} + ad + b}{c} \quad (3-24)$$

$$B_y > 0$$

$$\text{In the case that: } 0 < K_s + K_e < \frac{M_{\text{slave}}K_sK_e}{B_sB_e} \text{ and } e = \frac{-2c\sqrt{\frac{abd}{c^2}} + ad + b}{c} \quad (3-25)$$

$$0 < B_y < \frac{c\sqrt{\frac{abd}{c^2}} - ad}{ac}$$

$$\text{In the case that: } 0 < K_s + K_e < \frac{M_{\text{slave}}K_sK_e}{B_sB_e} \text{ and } 0 < e < \frac{-2c\sqrt{\frac{abd}{c^2}} + ad + b}{c} \quad (3-26)$$

$$\frac{ac\sqrt{\frac{a^2d^2 - 2abd - 2acde + b^2 - 2bce + c^2e^2}{a^2c^2}} - ad + b - ce}{2ac} < B_y < \frac{-ac\sqrt{\frac{a^2d^2 - 2abd - 2acde + b^2 - 2bce + c^2e^2}{a^2c^2}} - ad + b - ce}{2ac}$$

3-2-2 Simulated performance evaluation

The bounds on the outer admittance loop gains give some insight of theoretical stability, but the analysis focuses only on continuous-time systems. The robot control architecture, namely the discrete-time processing and internal delays, also poses limitations on stability and performance.

System architecture

The KUKA LWR 4+ robot is controlled using the ‘Robot Sensor Interface (RSI)’ that is programmed in Cartesian impedance control mode at a frequency of 1 kHz. Torque control on joint level is running at a frequency of 5 kHz. The Robot and the force-torque sensor communicate to the local control computer through a proxy at a frequency of 500 Hz. The inputs can be provided to the proxy using a human machine interface. Due to the limitations of the RSI, the communication between the proxy is limited to a frequency of 100 Hz. The system architecture can be seen in Figure 3-5.

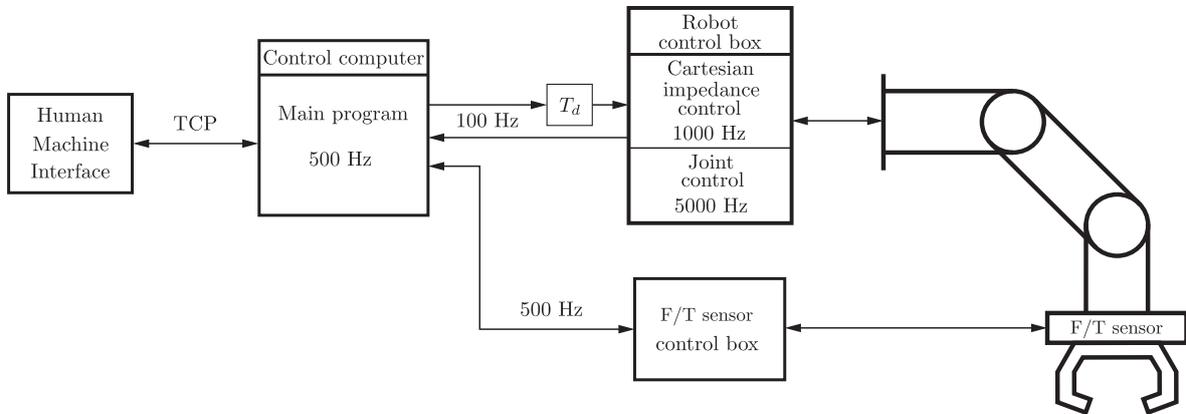


Figure 3-5: The system architecture overview.

The control inputs to the Cartesian impedance controller are affected by a time-delay, T_d . This imposes limitations on the performance and stability of the system. In order to investigate these limitations, the impedance controlled robot is identified using the nonlinear least squares method. Input/output data from a series of step responses, using different impedance parameters, is used for estimating a second order system. All fits are evaluated by how well the response of the model fits the estimation data, expressed as a percentage using the Normalized Root Mean Squared Error (NRMSE). All estimated models fit the estimation data with a NRMSE > 90%. The time-delay has identified to be $T_d = 0.09$ s and is incorporated in the simulated control architecture, as illustrated in Figure 3-6. Here τ_d represents the desired torque and τ_s the measured torque in the inner torque control loop of the slave robot. The step responses in free air for different slave stiffness, can be seen in Figure 3-7.

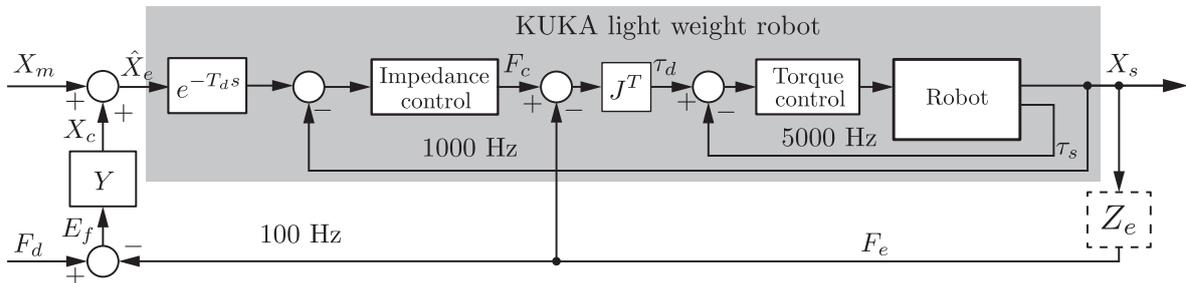


Figure 3-6: Impedance control architecture with outer admittance control loop with delay.

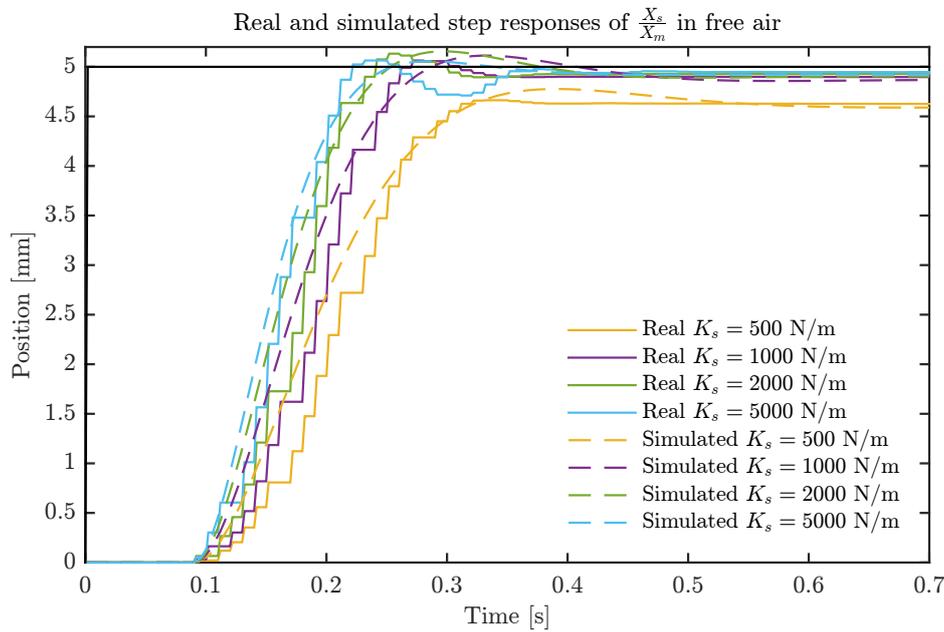


Figure 3-7: Step responses of the impedance controlled robot with different impedance parameters, with the responses of the estimated models for a step of 5 mm at $t = 0$.

In contact with different environments the step response of the impedance controlled slave behaves like illustrated in Figure 3-8. With increasing stiffness of the environment, the slave manipulator’s motion is restricted sooner, resulting in less penetration and higher contact forces.

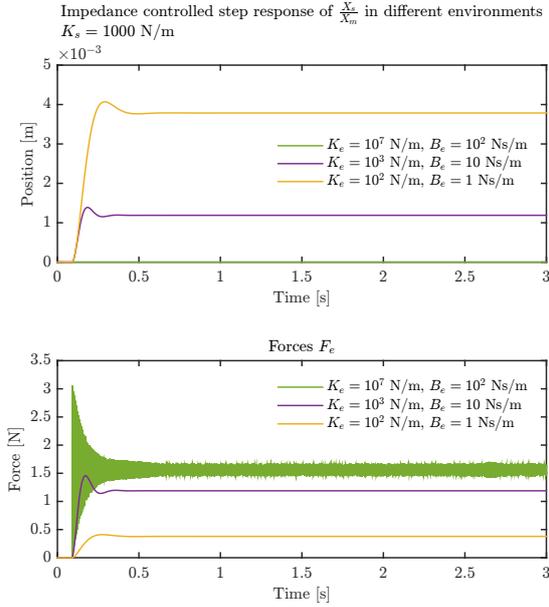


Figure 3-8: Step responses of the impedance controlled robot in contact with different environments.

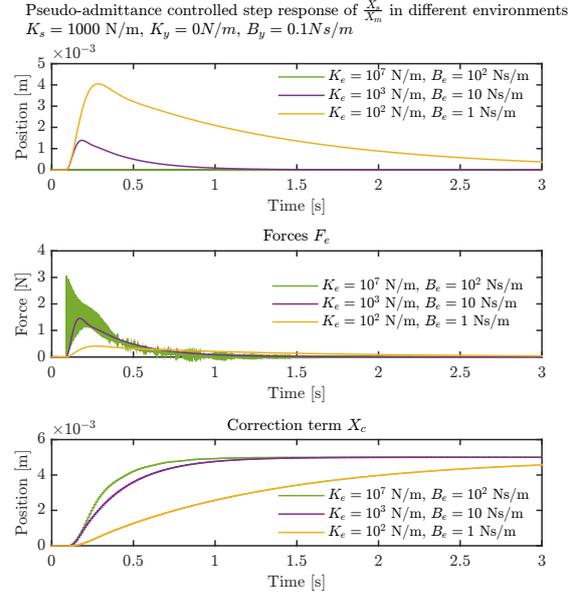


Figure 3-9: Step responses of the pseudo-admittance controlled robot in contact with different environments.

By closing the outer admittance loop, as illustrated Figure 3-6, the step response in contact with different environments is illustrated in Figure 3-9. It can be concluded that for the pseudo-admittance controlled robot, stiffer environments initially lead to higher contact forces. However, the higher contact forces result in a faster response of the correction term X_c , resulting in a faster decrease of contact forces.

The effects on the response of the pseudo-admittance controlled robot with different impedance parameters, are illustrated in Figure 3-12. It can be seen that the effects of a stiffer robot are the same as the effects that have been observed when increasing the stiffness of the environment.

The effects on the response of the pseudo-admittance controlled robot with different admittance control parameters, B_y , are illustrated in Figure 3-10. By decreasing the value of B_y , the actual gain on the force error $E_f(s)$, is increased. Smaller values for B_y result in a faster convergence of $X_c(s)$. Although, too small values for B_y render the system unstable.

For completeness, the response for different values of admittance control parameter K_y is illustrated in Figure 3-11. By increasing the values of K_y , the response shows faster convergence. However, increasing K_y leads to steady state error in both force as position, which are both non desired for smooth insertion. Moreover, the steady-state value of $X_c(s)$ does not show the real position of the environment X_e , and will result in a incorrect value of $\hat{X}_e(s)$ which updates the model that is used in the haptic shared controller.

Pseudo-admittance controlled step response of $\frac{X_c}{X_m}$ with different values for B_y
 $K_e = 10^4$ N/m, $B_e = 10$ Ns/m, $K_s = 1000$ N/m

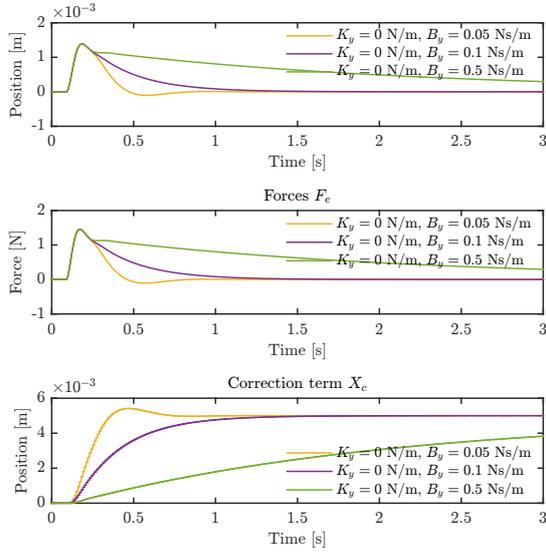


Figure 3-10: Step responses of the pseudo-admittance controlled robot in contact with different admittance control parameters, B_y .

Pseudo-admittance controlled step response of $\frac{X_c}{X_m}$ with different values for K_y
 $K_e = 10^4$ N/m, $B_e = 10$ Ns/m, $K_s = 1000$ N/m

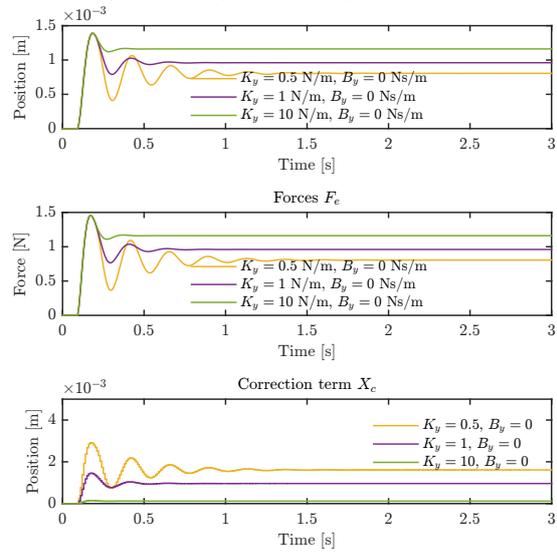


Figure 3-11: Step responses of the simulated pseudo-admittance controlled robot in contact with different admittance control parameters, K_y .

3-3 Experimental validation

For the experimental validation of the pseudo-admittance controlled robot, the system architecture that is described in Section 3-3-1 is used.

3-3-1 Experimental setup

The experimental setup that is used for validation consists of the 7-DOF KUKA light weight robot 4+ [8], with an ATI Gamma 6-DOF force-torque sensor attached to the wrist, as illustrated in Figure 3-13. The force-torque sensor measures the actual contact forces and torques with a resolution of 0.01 N and 0.0005 Nm at a frequency of 500 Hz. It is measuring 3 forces and 3 torques, $F_e = [F_{e_x}, F_{e_y}, F_{e_z}, \tau_{e_x}, \tau_{e_y}, \tau_{e_z}]$, in the reference frame $\Psi_{f/t}$, as illustrated in Figure 3-14. Furthermore it is important to note that the robot is equipped with a peg that is attached to the force-torque sensor. The peg has a length of 155 mm and a diameter of 14.0 mm and the tip of the peg is rounded. The hole has a diameter of 14.85 mm and the entrance of the hole is also rounded. Due to the tight tolerance, jamming can easily occur after an insertion depth of 10.0 mm if the peg is not well aligned with the hole.

The robot is commanded in the tool frame Ψ_{tool} , which is positioned at the end of the peg. The force-torque sensor measures forces and torques in the f/t frame $\Psi_{f/t}$, which is positioned at the edge of the sensor as illustrated in Figure 3-14. The measured forces and torques are transformed to act in the tool frame. Moreover, when the robot is initialised the forces and torques are set to 0 and they are compensated for forces that occur when not in contact due

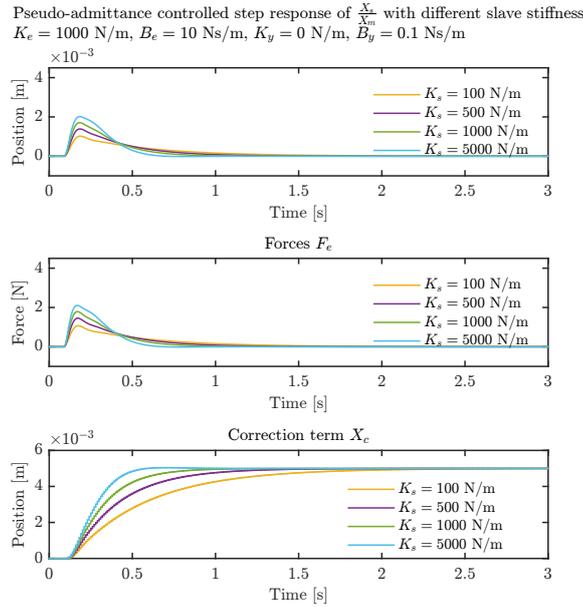


Figure 3-12: Step responses of the simulated pseudo-admittance controlled robot with different impedance parameters.

to the change of center of mass of the tool with respect to the gravitational vector, when rotating the tool.

3-3-2 Task and metrics

In the presented experimental scenario, the peg is commanded to go in z direction with a constant velocity of 8.0 mm/s until it reaches a depth of 80 mm. The entire insertion has a duration of 10 seconds, after which the peg is retracted with the same speed.

Independent variables

The independent variables can be divided in two groups:

- The control parameters. The tuning of the control parameters involves changing the impedance parameters of the robot and changing the admittance parameters of the outer admittance loop.
- The offsets. The offsets simulate the effects of the uncertainty. These offsets are divided in translational and rotational offsets.

The experiments that are performed are summarized in Table 3-2 and Table 3-3, respectively the rotational offsets versus different control parameters and the translational offsets versus different control parameters. Examples of rotational and translational offsets can be seen in respectively Figure 3-15 and Figure 3-16. The sets of control parameters are chosen after

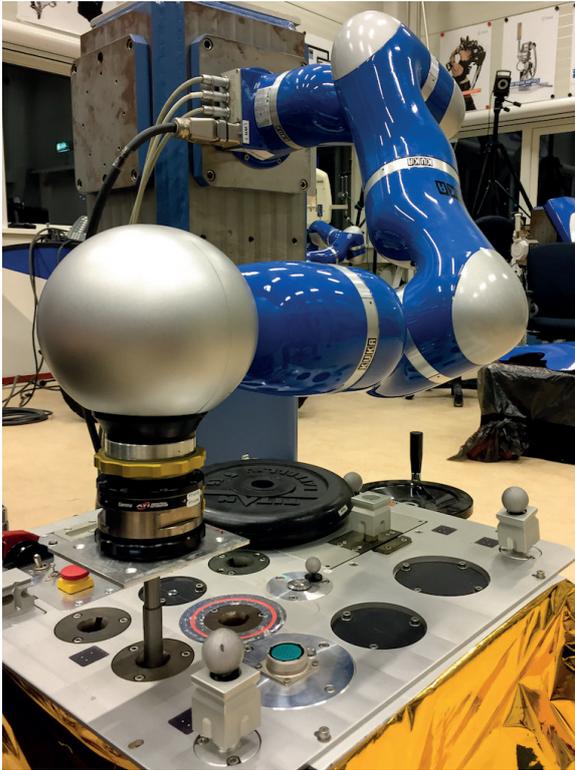


Figure 3-13: Experimental setup with KUKA LWR 4+ as slave robot, ATI Gamma force-torque sensor with attached peg and the task-board.

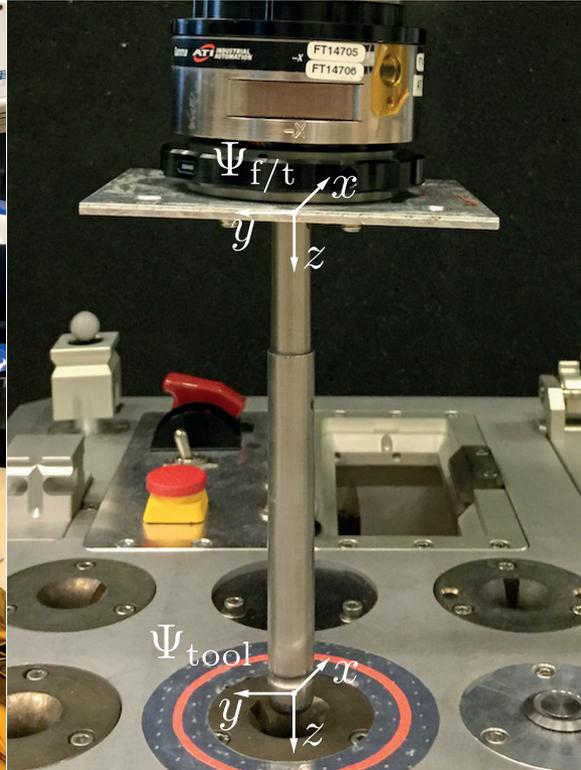


Figure 3-14: Tool frame Ψ_{tool} and f/t frame $\Psi_{\text{f/t}}$ on the robot peg and force-torque sensor.

preliminary experiments. The programmed Cartesian stiffness parameters of the impedance controller are specified in N/m for translations and Nm/rad for rotations in all directions. The Cartesian damping parameters follow from the damping ratio ζ and are given in Ns/m for translations and Nms/rad for rotations. The values for the Cartesian damping parameters for the impedance controller are left out in figures and tables for clarity. For these experiments a damping ratio of $\zeta = 0.7$ is used to calculate the damping coefficient:

$$B_s = 2\zeta\sqrt{K_s} \quad (3-27)$$

The parameters of the admittance controller are specified in the same units. However, for the experiment the admittance control is only active in the x and y directions, since the z is direction of insertion which is assumed to not be constrained.

For every offset/control parameter pair, 10 repetitions are performed in order to get sufficient data.

Dependent variables

The performance of the insertion is evaluated by the dependent variables. The dependent variables are the maximum reached depth, maximum exerted force, maximum exerted torque

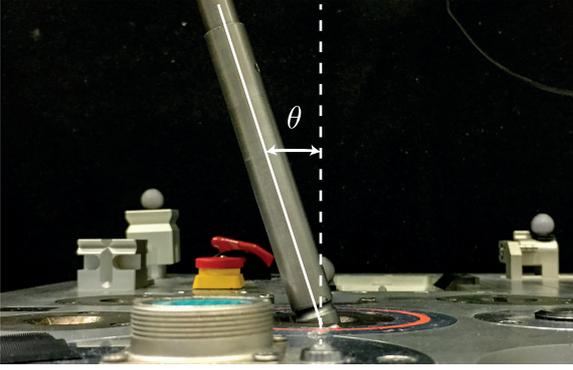


Figure 3-15: Example of a rotational offset illustrated on the INTERACT task-board.



Figure 3-16: Example of a translational offset illustrated on the INTERACT task-board.

Table 3-2: Experiments performed for different control parameters and **rotational** offsets.

		Robot stiffness (K_s)		
		500 N/m 50 Nm/rad	1000 N/m 100 Nm/rad	2000 N/m 200 Nm/rad
Pseudo- admittance controller gains (K_y, B_y)	Impedance control			
	0 N/m; 0 Nm/rad 0.15 Ns/m; 0.075 Nms/rad			
	0 N/m; 0 Nm/rad 0.10 Ns/m; 0.050 Nms/rad			
	0 N/m; 0 Nm/rad 0.067 Ns/m; 0.033 Nms/rad			
	0.10 N/m; 0.050 Nm/rad 0.10 Ns/m; 0.050 Nms/rad			

and average exerted weighted forces and torques. An insertion is considered to be successful if the maximum reached depth ≥ 60 mm and if the weighted sum of forces and torques < 40 . The maximum reached depth of ≥ 60 mm is depicted in the figures with a horizontal line. The weighted sum of forces and torques at each time step n , is calculated as follows:

$$F_{\text{score}}(n) = \sqrt{F_{e_x}(n)^2 + F_{e_y}(n)^2 + F_{e_z}(n)^2 + \frac{1}{0.10} \tau_{e_x}(n)^2 + \frac{1}{0.10} \tau_{e_y}(n)^2 + \frac{1}{0.10} \tau_{e_z}(n)^2} \quad (3-28)$$

Here 0.10 m represents the approximate distance between the force-torque sensor and the contact point of the peg in an inserted state.

- The maximum reached depth is calculated by taking the maximum value of the slave position in z direction, X_{s_z} .

$$X_{\text{depth}}(n) = \max_n X_{s_z}(n) \quad (3-29)$$

Table 3-3: Experiments performed for different control parameters and **translational** offsets.

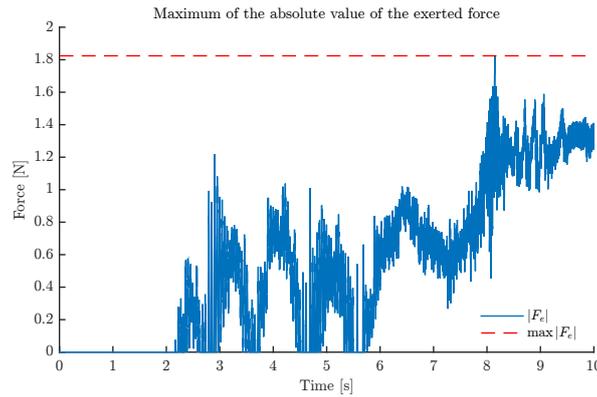
		Robot stiffness (K_s)		
		500 N/m 50 Nm/rad	1000 N/m 100 Nm/rad	2000 N/m 200 Nm/rad
Pseudo- admittance controller gains (K_y, B_y)	Impedance control			
	0 N/m; 0 Nm/rad 0.15 Ns/m; 0.075 Nms/rad			
	0 N/m; 0 Nm/rad 0.10 Ns/m; 0.050 Nms/rad			
	0 N/m; 0 Nm/rad 0.067 Ns/m; 0.033 Nms/rad			
	0.10 N/m; 0.050 Nm/rad 0.10 Ns/m; 0.050 Nms/rad			

- The average exerted weighted forces and torques are calculated by summing the weighted sum of forces and torques at each sample and the result is divided over the total number of samples recorded during an insertion, N .

$$F_{\text{average}} = \frac{1}{N} \sum_{n=1}^N \sqrt{F_{e_x}(n)^2 + F_{e_y}(n)^2 + F_{e_z}(n)^2 + \frac{1}{0.10} \tau_{e_x}(n)^2 + \frac{1}{0.10} \tau_{e_y}(n)^2 + \frac{1}{0.10} \tau_{e_z}(n)^2} \quad (3-30)$$

- The maximum exerted force is calculated by taking the maximum value of the absolute force exerted on the environment. In time domain this is illustrated by Figure 3-17.

$$F_{\text{max}} = \max_n |F_e(n)| \quad (3-31)$$

**Figure 3-17:** The absolute force exerted on the environment and the maximum value of the absolute force.

- A similar strategy is used to calculate the maximum torque exerted on the environment.

$$\tau_{\text{max}}(n) = \max_n |\tau_e(n)| \quad (3-32)$$

3-3-3 Results

In this section, the results of the experimental validation are presented in the form of box plots. The box plots show the median, which are connect to each other using a line and the 25th and 75th percentiles, respectively q_{25} and q_{75} . The whiskers extends up to the most extreme data point that is not an outlier. A data point is considered an outlier if $x > q_{75} + w(q_{75} - q_{25})$ or if $x < q_{25} - w(q_{75} - q_{25})$, where $w = 1.5$. Outliers are depicted in the boxplots as small dots. Experiments have been performed up to 20° rotational offset, and 6 mm translational offset with variable intervals. If no bars are present in the next interval, the combination of independent variables failed due to instability, excessive forces and/or not reaching the proper depth. In pseudo-admittance control a configured impedance of $K_s = 500$ N/m is not considered, since after each trial reset the jump in set-point due to resetting the integrator of the admittance control loop, caused the robot shut down for safety reasons. Moreover, in the results on impedance control, $K_s = 500$ N/m shows undesired free air position tracking. The results of the maximum torque dependent variable has been left out of this report, since they are similar to the maximum force dependent variable results and they are used to prove the same point of excessive forces and torques being exerted on the environment. The torques are still considered in the average force exerted on the environment dependent variable.

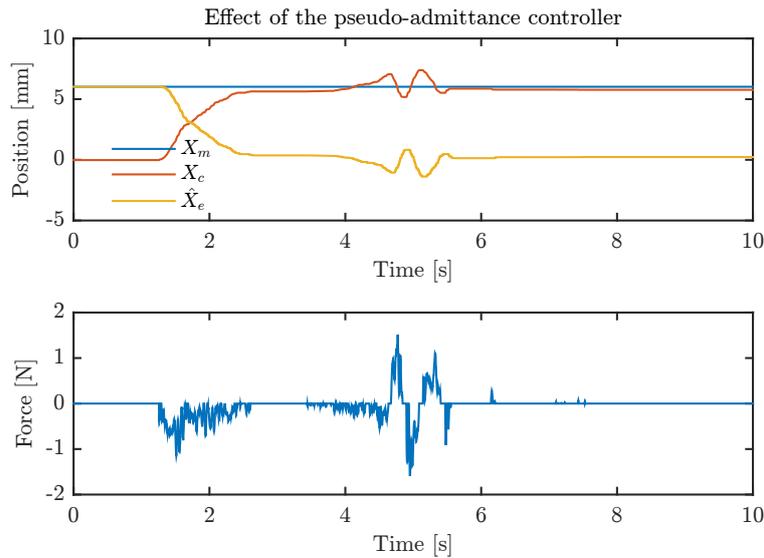


Figure 3-18: Change of $\hat{X}_e(s)$ and $X_c(s)$ when performing a peg-in-hole insertion with a translational offset of 6 mm. Note that $\hat{X}_e(s) \neq 0$ because it represents the estimated value of the position of the wall of the hole, not the centerline.

Offsets versus different impedance control parameters

The presented results show the effects of changing the impedance control parameters for both the impedance controlled as the pseudo-admittance controlled task. The experiments that are considered are summarized in Table 3-4. The results with rotational offsets are illustrated in Figure 3-19, Figure 3-21 and Figure 3-23. The results with translational offsets are illustrated in Figure 3-20, Figure 3-22 and Figure 3-24.

For both rotational and translational offsets it can be observed that a more compliant slave robot decreases the maximum reached depth. For rotational offsets, the average forces and maximum forces exerted on the environment decrease as well. For translational offsets, deviant behaviour can be observed. The maximum reached depth is similar as at the rotational offsets, however, the average exerted force of IMP05 turns out to be higher and shows more spread with increasing translational offset. Moreover, the maximum exerted forces show the same behaviour for both IMP05 and IMP10.

Table 3-4: Experiments used to show the effects of rotational and translational offsets versus impedance control parameters.

		Robot stiffness (K_s)		
		500 N/m 50 Nm/rad	1000 N/m 100 Nm/rad	2000 N/m 200 Nm/rad
Pseudo-admittance controller gains (K_y, B_y)	Impedance control	IMP05	IMP10	IMP20
	0 N/m; 0 Nm/rad 0.15 Ns/m; 0.075 Nms/rad			
	0 N/m; 0 Nm/rad 0.10 Ns/m; 0.050 Nms/rad		ADM10_10	ADM20_10
	0 N/m; 0 Nm/rad 0.067 Ns/m; 0.033 Nms/rad			
	0.10 N/m; 0.050 Nm/rad 0.10 Ns/m; 0.050 Nms/rad			

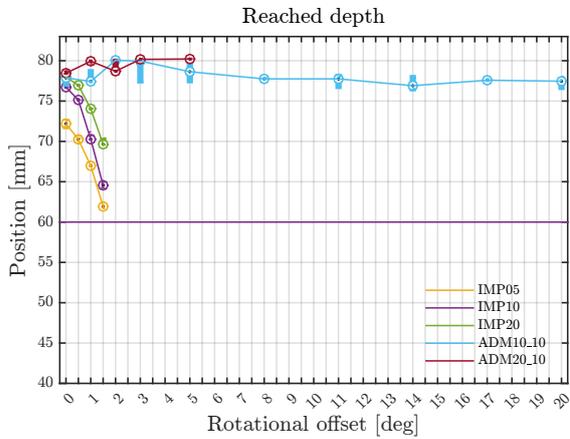


Figure 3-19: Boxplot of the maximum insertion depth for rotational offsets versus different impedance control parameters.

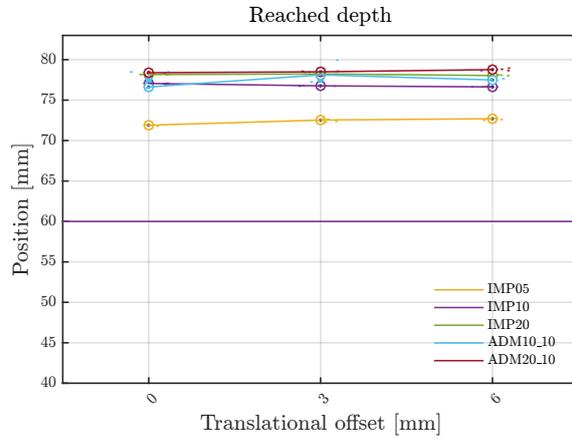


Figure 3-20: Boxplot of the maximum insertion depth for translational offsets versus different impedance control parameters.

Offsets versus different admittance control parameters

The presented results show the effects of changing the admittance control parameters for the pseudo-admittance controlled task compared to the impedance controlled task. The experi-

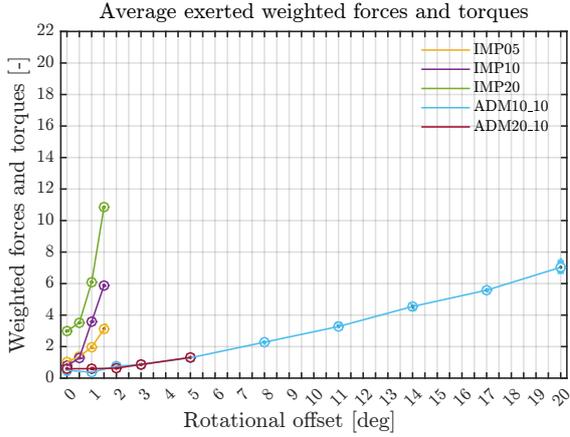


Figure 3-21: Boxplot of the average weighted forces and torques exerted on the environment for rotational and translational offsets versus different impedance control parameters.

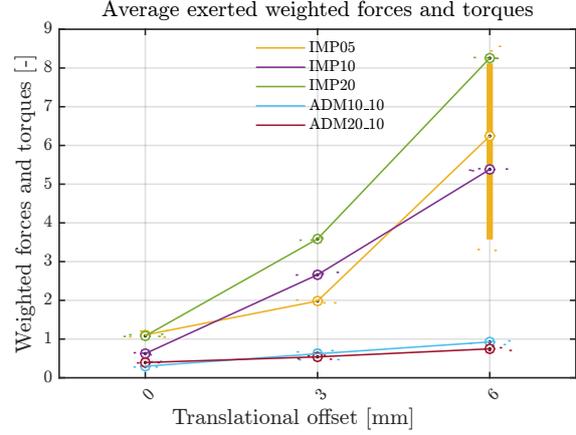


Figure 3-22: Boxplot of the average weighted forces and torques exerted on the environment for translational offsets versus different impedance control parameters.

ments that are considered are summarized in Table 3-5. The results with rotational offsets are illustrated in Figure 3-25, Figure 3-27 and Figure 3-29. The results with translational offsets are illustrated in Figure 3-26, Figure 3-28 and Figure 3-30.

It can be observed that for all tested admittance parameters B_y , the maximum reached depth increases. Moreover, a decrease in maximum force is expected for a small value of B_y , however it can be observed that the maximum force exerted is higher compared to a large value of B_y for rotational offsets, with an exception of ADM_10_07. For translational offsets this is the other way around, as expected. An increase in average exerted force can be observed for rotational offsets, while for translational offsets the average exerted force is very similar for each offset.

Table 3-5: Experiments used to show the effects of rotational and translational offsets versus different admittance control parameters.

		Slave stiffness (K_s)		
		500 N/m 50 Nm/rad	1000 N/m 100 Nm/rad	2000 N/m 200 Nm/rad
Pseudo- admittance controller gains (K_y, B_y)	Impedance control		IMP10	
	0 N/m; 0 Nm/rad 0.15 Ns/m; 0.075 Nms/rad		ADM10_15	
	0 N/m; 0 Nm/rad 0.10 Ns/m; 0.050 Nms/rad		ADM10_10	
	0 N/m; 0 Nm/rad 0.067 Ns/m; 0.033 Nms/rad		ADM10_07	
	0.10 N/m; 0.050 Nm/rad 0.10 Ns/m; 0.050 Nms/rad			

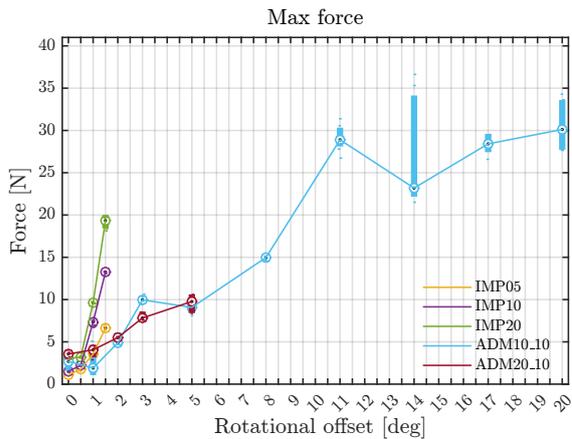


Figure 3-23: Boxplot of the maximum forces exerted on the taskboard for rotational offsets versus different impedance control parameters.

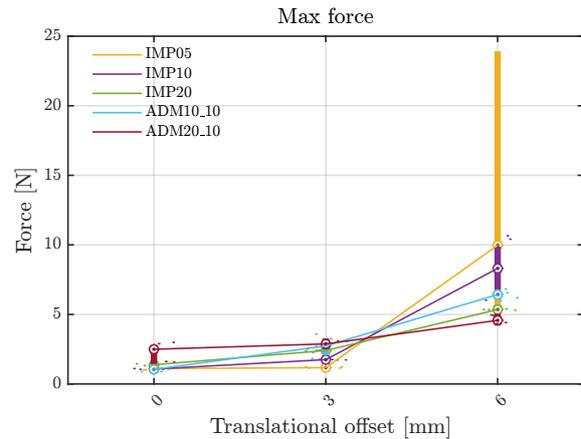


Figure 3-24: Boxplot of the maximum forces exerted on the taskboard for translational offsets versus different impedance control parameters.

Rotational offsets versus added position dependency in the admittance control loop

The presented results show the effects of added position dependency in the admittance control loop for the pseudo-admittance controlled task compared to no position dependency and the impedance controlled task. The experiments that are considered are summarized in Table 3-6. The results with rotational offsets only include the average force exerted on the environment, as illustrated in Figure 3-31. A time-domain plot of an insertion with added position dependency can be seen in Figure 3-33.

It can be observed that the average weighted forces and torques that are exerted on the environment becomes larger when K_y is introduced, compared to the admittance control without K_y . Furthermore, by adding K_y there is only a slight increase of 2.5° rotational offset, while the pseudo-admittance controller without K_y shows a much larger increase up to 20° of rotational offsets that can be successfully inserted.

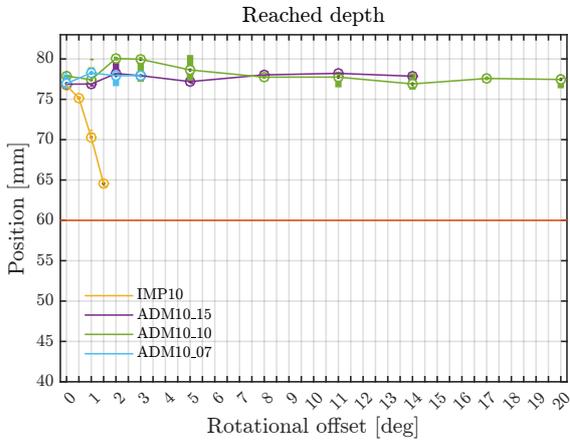


Figure 3-25: Boxplot of the maximum insertion depth for rotational offsets versus different admittance control parameters.

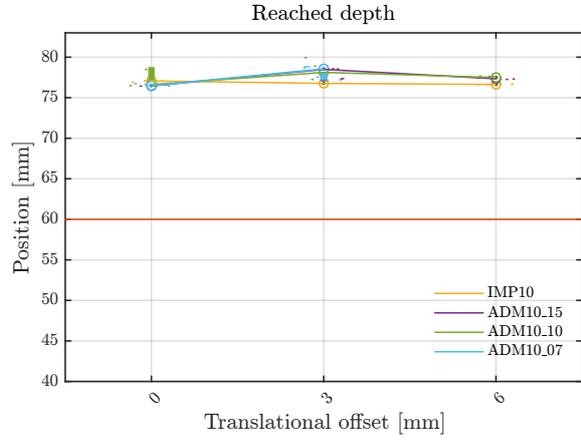


Figure 3-26: Boxplot of the maximum insertion depth for translational offsets versus different admittance control parameters.

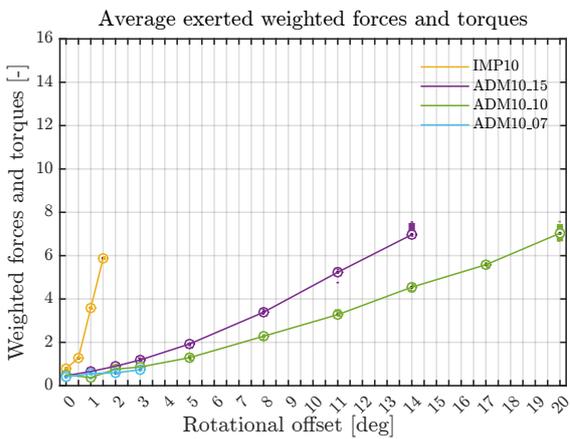


Figure 3-27: Boxplot of the average weighted forces and torques exerted on the environment for rotational offsets versus different admittance control parameters.

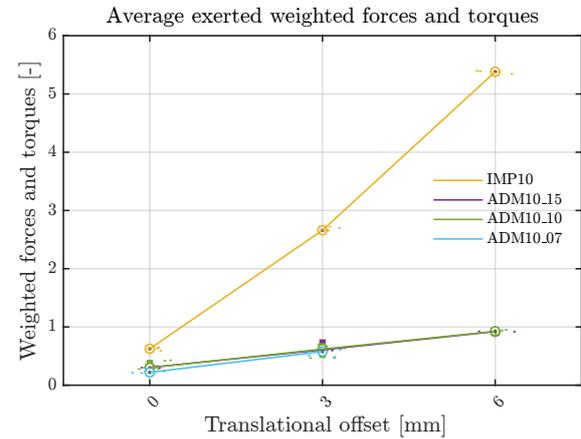


Figure 3-28: Boxplot of the average weighted forces and torques exerted on the environment for translational offsets versus different admittance control parameters.

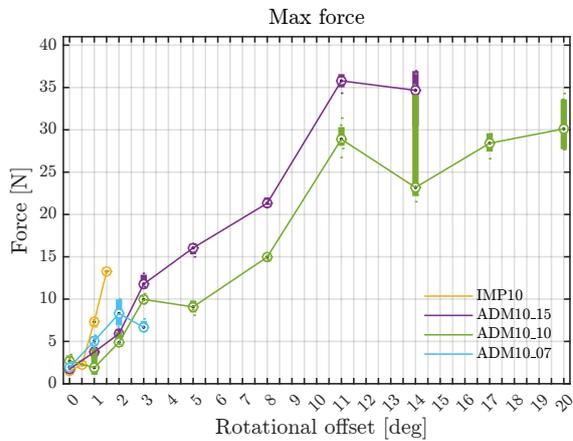


Figure 3-29: Boxplot of the maximum forces exerted on the taskboard for rotational offsets versus different admittance control parameters.

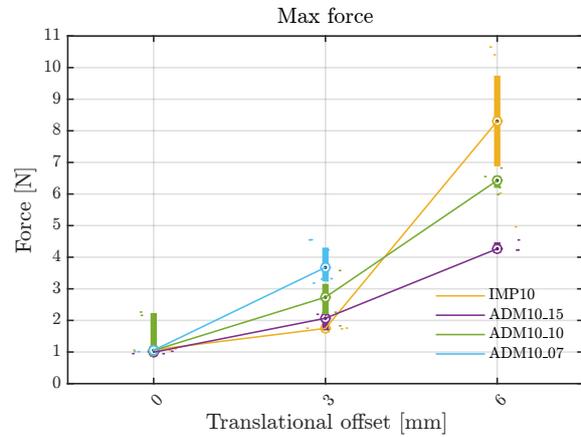


Figure 3-30: Boxplot of the maximum forces exerted on the taskboard for translational offsets versus different admittance control parameters.

Table 3-6: Experiments used to show the effects of rotational offsets versus adding position dependency (K_y) in the admittance control loop.

		Slave stiffness (K_s)		
		500 N/m 50 Nm/rad	1000 N/m 100 Nm/rad	2000 N/m 200 Nm/rad
Pseudo-admittance controller gains (K_y, B_y)	Impedance control		IMP10	
	0 N/m; 0 Nm/rad 0.15 Ns/m; 0.075 Nms/rad			
	0 N/m; 0 Nm/rad 0.10 Ns/m; 0.050 Nms/rad		ADM10_10	
	0 N/m; 0 Nm/rad 0.067 Ns/m; 0.033 Nms/rad			
	0.10 N/m; 0.050 Nm/rad 0.10 Ns/m; 0.050 Nms/rad		ADM10_10_10	

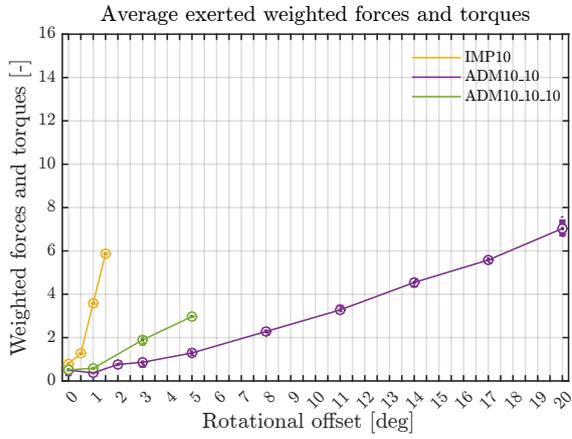


Figure 3-31: Boxplot of the average weighted forces and torques exerted on the environment for rotational offsets for methods with and without position dependency.

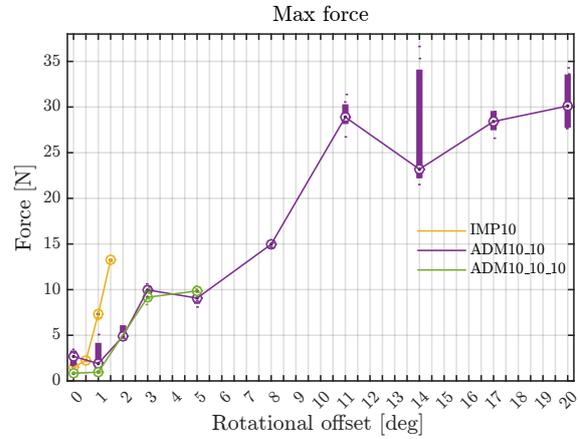


Figure 3-32: Boxplot of the maximum force exerted on the environment for rotational offsets for methods with and without position dependency.

Pseudo-admittance controlled insertion with K_y
 $K_y = 0.1 \text{ N/m}$, $B_y = 0.1 \text{ Ns/m}$, Rotational offset = 3°

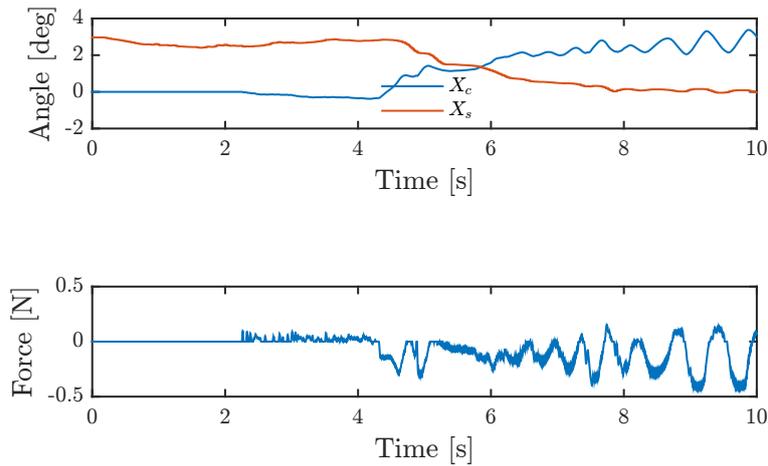


Figure 3-33: Pseudo-admittance controlled insertion with a rotational offset of 3° and $K_y = [0.1 \text{ N/m}, 0.05 \text{ Nm/rad}]$.

3-4 Discussion

Increasing the compliance of the impedance controlled robot does not increase the maximal admissible rotational error

In the result section, it can be observed that the maximum reached depth decreases when the robot becomes more compliant. Moreover, a decrease in forces exerted on the environment has been observed when the robot becomes more compliant. The decrease in forces is however not sufficient to increase the maximal admissible rotational error, since for the three different impedance parameters tested, no increase in successfully inserted rotational offset has been observed. This results shows that no robustification of task performance in presence of microscale uncertainty using solely adaptive impedance controllers is achieved, as described in Section 3-2.

Behaviour deviance of compliant impedance controlled slave explained

The behaviour that is observed when changing the compliance of the impedance controlled slave in translational offsets, can be explained by the decrease in position tracking. Especially for the case where $K_s = 500$ N/m, the peg makes contact with the chamfer upon insertion, resulting in an increase of contact forces. The same goes for $K_s = 1000$ N/m, however a larger offset is needed to make contact with the chamfer upon insertion. For the largest translational offset, 6 mm, both $K_s = 500$ N/m and $K_s = 1000$ N/m in some cases make contact on the edge of the chamfer and the task-board. This initially prevented insertion, but the commanded position $X_m(s)$ in z direction increased and so did the forces in that direction until they were large enough to still force the peg in the hole and reach the desired depth.

This behaviour can be considered undesired, because of the increase in the maximum force exerted on the environment, however they meet the bounds that were set on the maximum force and the reached depth. This behaviour illustrates the bound on the translational offsets, when the peg reaches the horizontal surface of the task-board no insertion is possible by only commanding the slave robot in z direction.

System sensitivity to B_y

A small value for B_y results in a high gain on the force-torque errors, this results in overshoots in the correction term $X_c(s)$. The overshoot in $X_c(s)$ results in high forces/torques, which cause the slave to hit the other side of the hole and the same effect occurs, resulting in a higher maximum force. For both rotational and translational offsets that are too large, the overshoots in $X_c(s)$ cause the system to become unstable.

On the other hand, a large value for B_y results in a low gain on the force-torque errors and a slower response of the correction term $X_c(s)$. Since the commanded position keeps moving in z direction, this leads to higher forces being exerted on the environment, explaining the behaviour of ADM10_15. For large rotational offsets, it takes the correction term too long to correct the misalignment enough for the insertion to continue. By the time the peg is well enough aligned, the forces exerted on the environment have already reached the safety-limit and the insertion is considered unsuccessful. The reason why this behaviour is not seen for

translational offsets is because the peg does not get stuck at the chamfer but keeps moving in z direction preventing a build-up of forces.

System sensitivity to K_y

By adding a position dependency in the admittance loop, the slave constantly tries to undo the effects of the correction $X_c(s)$. Even when inserted, the position dependency continuously pulls the slave to the commanded position $X_m(s)$ until it is being pushed back to the boundary of the hole due to the forces that are exerted, explaining the increase of average exerted force on the environment.

For larger errors this behaviour results in unstable behaviour in the constraint environment of the hole, as illustrated in Figure 3-33. The peg ‘bounces’ off the walls of the hole with increasing velocity, resulting in higher contact forces until the response blows up. In this scenario, a position dependency is undesired.

However, adding a position dependency could be advantageous in situations where the operator is commanding the slave robot with use of a master device that has a limited workspace and is not able to recalibrate its position by use of a ‘clutch’. An example of a master device would be an exoskeleton without the ability to clutch and recalibrate its position. When the operator would perform a peg in hole insertion with a large rotational offset and by only commanding the exoskeleton to go in the direction of insertion, the B_y would correct for the offset and make insertion successful. When the operator now retracts from the hole in the same direction as he inserted it, the orientation of the slave will be corrected by the degree of the rotational offset which could eventually result in the operator being at the limits of his workspace and not being able to command the slave robot any longer to perform a task. By adding a position dependency, the slave would restore itself to the orientation of the slave device before insertion.

On the bounds of the rotational error

For translational errors a bound has been found that can be physically explained. Such a bound also exists for rotational errors and depend on the geometry of the peg and the hole, since the admittance controller needs torques in order to correct for the rotational offset. The bound that has been found of 20° offset could be exceeded by tuning the admittance parameters, but more importantly by imposing some admittance on the z direction. When the value of B_y is high, it takes the correction term too long to have the peg aligned with the hole, resulting in a high forces as described above. By limiting the commanded value in z direction, the forces can stay within the safety-bounds. In order to ensure insertion, a position dependency in z direction could be added.

Using the information on the error

The estimated location of the environment, $\hat{X}_e(s)$, can be extracted online and used to update the model that is used by the haptic shared controller. By doing this, the guidance will immediately be corrected and the operator will be guided to the real centerline of the hole.

Moreover, after the peg has been inserted there will be no forces acting on the environment that are masked from the operator, as was pointed out by [3].

In bilateral teleoperation, this approach would limit the effects of force feedback, since the admittance controller dissolves all forces. The operator would only be able to feel it when contact is made, but is no longer able to exert a certain amount of force for a certain period of time.

Chapter 4

Conclusions

In this thesis, the performance of intention prediction methods is evaluated and compared in order to eliminate the macroscale error. Moreover, a pseudo-admittance controller is used to improve task performance in the presence of microscale uncertainty. The same method is used to reduce the microscale uncertainty. The main conclusions of this research are:

- Memory-based prediction is a feasible method to eliminate macroscale uncertainty, i.e., predict the intended goal of the operator, in a teleoperated reaching task. On pre-recorded teleoperation data, for 84% ($\pm 8\%$ CI_{95}) of the total trajectory, the predictor correctly identified the intended goal.
- For the tested peg-in-hole insertion task, robustness to microscale uncertainty is not increased by making the slave more compliant. Maximal acceptable rotational and translational errors were, for all tested impedance parameters, 1.5° and 6 mm, respectively. Moreover, the performance of position tracking is decreased by doing so, which makes peg-in-hole insertion an even more tedious task to perform.
- Pseudo-admittance control is a feasible method to robustify task performance against microscale uncertainty of the peg-in-hole insertion task. For rotational errors the maximal acceptable error is found to be 20° , which is 13 times better compared to impedance control. The average weighted forces and torques exerted on the environment is reduced with a factor of 5 compared to the best impedance controlled case with the maximal offset of 1.5° . For translational errors the maximal acceptable error is found to be 6 mm, which is the same compared to impedance control. The average weighted forces and torques exerted on the environment is reduced with a factor of 6 compared to the best impedance controlled case with the maximal offset of 6 mm.
- The proposed system provides improved knowledge of the environment X_e (by reducing the microscale uncertainty), and as such can provide an updated environment model \hat{X}_e , for the support system.

4-1 Future work

Although promising results have been found towards making a teleoperation support system robust to uncertainties in haptic guidance, this does not complete or exhaust this topic. There are many more developments that can be explored in future studies. The following topics pose as candidates for future research:

- The prediction methods have now been validated using pre-generated teleoperation data. In order to show the performance of the prediction methods it would be valuable to implement these methods on a setup where haptic guidance is provided to the operator, based on the predicted intended goal. In such a research other interesting topics reveal themselves, like how do different methods or intensities of haptic guidance influence the prediction. When to provide haptic guidance and how to arbitrate between no guidance and guidance.
- The pseudo-admittance controller has been validated when only considering the slave side of the teleoperation system. In order to gain more insight in the performance and stability, the pseudo-admittance controller should be validated on the entire teleoperation system, so by taking into account the human, master device and communication channel. This can be extended by looking at time-delays in the communication channel from master to slave. Moreover, as already has been discussed this method poses limitations on the force feedback that is provided to the operator. The effects of this could also be a topic for future research.
- One of the limitations of the pseudo-admittance method is the error in slave and master position ($X_s - X_m$) due to the time it takes to correct for misalignments. This error results in high contact forces and if aligned, a large jump in position. A future research direction would be to investigate whether admittance control in the unconstrained direction robustifies task performance even more. It should also be considered to add position dependency in this direction, so the insertion will still reach the desired depth.

Appendix A

Other bilateral teleoperation architectures

In this appendix the position-position and the position-force bilateral teleoperation architectures will be treated [49, 50].

A-1 Position-position

The most basic bilateral controller is being used since the beginning of the nuclear industry. It makes use of a position-position scheme where only the positions of the master device and slave robot are needed, so no force sensors are required. The master position is passed to the position controller on the slave side and the slave robot's position is returned to the master device. The block diagram of this scheme can be seen in Figure A-1. $Z_m(s)$ and $Z_s(s)$ represent the dynamics of the master device and the slave robot respectively. The controllers are denoted by $C_m(s)$ and $C_s(s)$ and can be in the form of a gain or a PD controller. The impedance of the environment is denoted by $Z_e(s)$. The system equations are conventionally expressed in terms of velocities [6] but here positions are used instead.

The input force, $F_h(s)$, represents the force that the operator applies to the master device. The force that is acting on the master device, $F_m(s)$, is proportional to the difference in the slave robot's position $X_s(s)$, and the position of the master device $X_m(s)$, with respect to the master control gain $C_m(s)$, with representing the slave robot's position. In the same way the force exerted by the slave robot, $F_s(s)$, is also proportional to the difference in the slave robot's and the master device's position, with respect to the slave control gain $C_s(s)$.

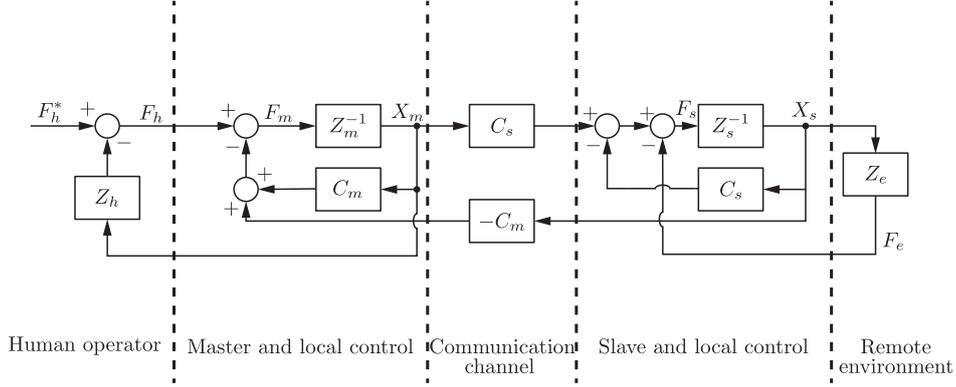


Figure A-1: Two-channel position-position architecture.

Mathematically the position-position bilateral scheme can be described as:

$$\begin{aligned}
 F_m(s) &= C_m(s)(X_m(s) - X_s(s)) \\
 F_s(s) &= C_s(s)(X_m(s) - X_s(s)) \\
 X_m(s) &= Z_m^{-1}(s)(F_h(s) - F_m(s)) \\
 X_s(s) &= Z_s^{-1}(s)(F_s(s) - F_e(s)) \\
 F_e(s) &= Z_e(s)X_s(s)
 \end{aligned} \tag{A-1}$$

Where $F_e(s)$ represents the force applied by the environment on the slave robot. By looking at the transfer function from $F_m(s)$ to $X_m(s)$, insights can be gained on how the forces from the remote environment are reflected on the master device. This relation is given by:

$$H(s) = \frac{F_m(s)}{X_m(s)} = C_m(s) \frac{1 + Z_e(s)Z_s^{-1}(s)}{1 + (Z_e(s) + C_s(s))Z_s^{-1}(s)} \tag{A-2}$$

When looking at the case where the slave robot is not in contact with the environment ($Z_e(s) = 0$), $H(s)$ is given by $\frac{C_m(s)}{1 + Z_s^{-1}(s)C_s(s)}$. Meaning that in free motion the operator feels the friction and inertial forces associated with the slave robot, although no external forces are acting on the slave robot. When looking at the opposite case, where the slave robot is contact with a rigid object ($Z_e(s) \rightarrow \infty$), $H(s)$ is given by $C_m(s)$. Meaning that the contact with a rigid contact is perceived as an elastic contact with an impedance of $C_m(s)$, so the operator is able to move the master device in this case. Furthermore, if the slave robot is not back-drivable then the environmental forces acting on the slave robot do not cause a position error and are thus not experienced by the operator. From above mentioned insight it can be concluded that this position-position control scheme, although straightforward to implement, offers the operator poor perception of the remote environment.

A-2 Position-force

An improvement on the position-position scheme, can be seen in Figure A-2. This scheme is known as the position-force scheme.

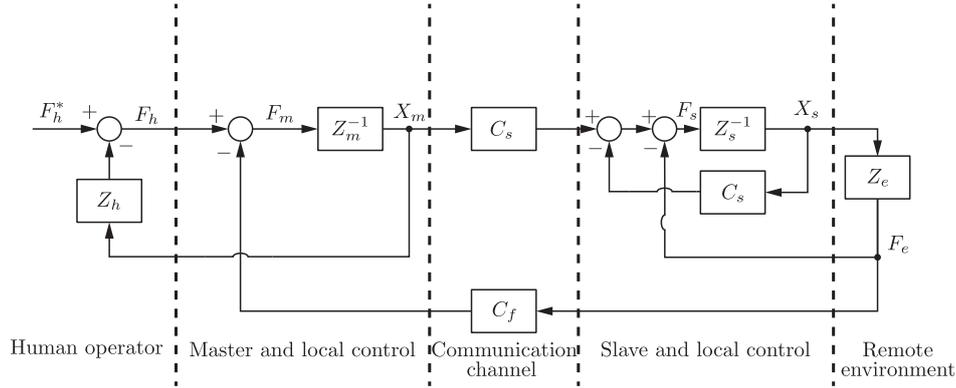


Figure A-2: Two-channel position-force architecture.

By placing a force-torque sensor between the slave robot and the environment, the contact force between the slave robot and the environment can be measured. By providing the master device with a force $F_m(s)$, which is proportional to this contact force $F_e(s)$, the operator has a much more clear sense of the environment. Here $C_f(s)$ denotes the control gain in the force channel. Mathematically the position-force bilateral scheme can be described as:

$$\begin{aligned}
 F_m(s) &= C_f(s)F_e(s) \\
 F_s(s) &= C_s(s)(X_m(s) - X_s(s)) \\
 X_m(s) &= Z_m^{-1}(s)(F_h(s) - F_m(s)) \\
 X_s(s) &= Z_s^{-1}(s)(F_s(s) - F_e(s)) \\
 F_e(s) &= Z_e(s)X_s(s)
 \end{aligned} \tag{A-3}$$

By looking again at the transfer function from $F_m(s)$ to $X_m(s)$, new insights can be gained on how the forces from the remote environment are reflected on the master device. This relation is given by:

$$H(s) = \frac{F_m(s)}{X_m(s)} = C_f(s) \frac{Z_e(s)Z_s^{-1}(s)C_s(s)}{1 + (Z_e(s) + C_s(s))Z_s^{-1}(s)} \tag{A-4}$$

When looking again at the case where the slave robot is not in contact with the environment ($Z_e(s) = 0$), $H(s)$ is now given by 0. This means that the undesired controller forces that were present in the position-position control scheme, are no longer present in the position-force scheme, which is an improvement over the position-position scheme. When looking at the opposite case, where the slave robot is contact with a stiff object ($Z_e(s) \rightarrow \infty$), the maximum value for $H(s)$ is equal to $C_f(s)C_s(s)$ which means that $H(s)$ is still limited and the operator can still move the master device in this case. However, the position-force control scheme has better performance than position-position due to lack of the slave robot controller forces perceived by the operator.

Appendix B

Memory-based prediction derivation

In this appendix the derivation of memory-based prediction is presented from [51]. In memory-based prediction, besides using the current slave robot configuration also the set of past slave robot configurations ($\xi_{X_{\text{init}} \rightarrow X_s(t)}$) is taken into account:

$$X_{G^*}(t) = \arg \max_{X_{G_i} \in \mathcal{G}} P(X_{G_i} | \xi_{X_{\text{init}} \rightarrow X_s(t)}) \quad (\text{B-1})$$

By applying Bayes' theorem this results in:

$$X_{G^*}(t) = \arg \max_{X_{G_i} \in \mathcal{G}} \frac{P(\xi_{X_{\text{init}} \rightarrow X_s(t)} | X_{G_i})P(X_{G_i})}{P(\xi_{X_{\text{init}} \rightarrow X_s(T=t)})} \quad (\text{B-2})$$

And since $P(\xi_{X_{\text{init}} \rightarrow X_s(t)}) = 1$, since this denotes the already travelled trajectory, Equation (B-2) results in:

$$X_{G^*}(t) = \arg \max_{X_{G_i} \in \mathcal{G}} P(\xi_{X_{\text{init}} \rightarrow X_s(t)} | X_{G_i})P(X_{G_i}) \quad (\text{B-3})$$

Here, $P(X_{G_i})$ is the probability on the goals that is set beforehand. When there is no prior knowledge, the probability can be uniform over all goals. $P(\xi_{X_{\text{init}} \rightarrow X_s} | X_{G_i})$ is the probability that the slave robot will actually follow $\xi_{X_{\text{init}} \rightarrow X_s(t)}$ when it is going for goal X_{G_i} . This is computed as the ratio of all trajectories from X_{init} to X_{G_i} , that went through $\xi_{X_{\text{init}} \rightarrow X_s(t)}$, to all trajectories from X_{init} to X_{G_i} :

$$P(\xi_{X_{\text{init}} \rightarrow X_s(t)} | X_{G_i}) = \frac{\int_{\xi_{X_s(t) \rightarrow X_{G_i}}} P(\xi_{X_{\text{init}} \rightarrow X_s(t) \rightarrow X_{G_i}})}{\int_{\xi_{X_{\text{init}} \rightarrow X_{G_i}}} P(\xi_{X_{\text{init}} \rightarrow X_{G_i}})} \quad (\text{B-4})$$

By assuming that the trajectories are separable, Equation (B-4) results in:

$$P(\xi_{X_{\text{init}} \rightarrow X_s(t)} | X_{G_i}) = P(\xi_{X_{\text{init}} \rightarrow X_s(t)}) \frac{\int_{\xi_{X_s(t) \rightarrow X_{G_i}}} P(\xi_{X_s(t) \rightarrow X_{G_i}})}{\int_{\xi_{X_{\text{init}} \rightarrow X_{G_i}}} P(\xi_{X_{\text{init}} \rightarrow X_{G_i}})} \quad (\text{B-5})$$

To calculate these probabilities, a model is needed of how operators move the slave robot to reach a certain goal. The assumption has been made that the slave robot configurations optimize a goal dependent cost function $J_{X_{G_i}}$. By using the principle of maximum entropy [52], a model can be made that makes increasing cost still possible, but exponentially less probable:

$$P(\xi | X_{G_i}) \propto e^{-J_{X_{G_i}}(\xi)} \quad (\text{B-6})$$

This results in:

$$P(\xi_{X_{\text{init}} \rightarrow X_s(t)} | X_{G_i}) = e^{-J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_s(t)})} \frac{\int_{\xi_{X_s(t) \rightarrow X_{G_i}}} e^{-J_{X_{G_i}}(\xi_{X_s(t) \rightarrow X_{G_i}})} d\xi_{X_s(t) \rightarrow X_{G_i}}}{\int_{\xi_{X_{\text{init}} \rightarrow X_{G_i}}} e^{-J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_{G_i}})} d\xi_{X_{\text{init}} \rightarrow X_{G_i}}} \quad (\text{B-7})$$

In high-dimensional spaces like in three dimensional manipulation, evaluating the integrals is too computationally expensive. The integrals can be approximated by using Laplace's method, starting by approximating $J_{X_{G_i}}(\xi_{X_{\text{from}} \rightarrow X_{\text{to}}})$ around $\xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*$ by its second order Taylor series expansion:

$$J_{X_{G_i}}(\xi_{X_{\text{from}} \rightarrow X_{\text{to}}}) \approx J_{X_{G_i}}(\xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*) + \quad (\text{B-8})$$

$$\frac{\partial J_{X_{G_i}}}{\partial \xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*} J_{X_{G_i}}(\xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*)^T (\xi_{X_{\text{from}} \rightarrow X_{\text{to}}} - \xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*) + \quad (\text{B-9})$$

$$\frac{1}{2} (\xi_{X_{\text{from}} \rightarrow X_{\text{to}}} - \xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*)^T \frac{\partial^2 J_{X_{G_i}}}{\partial^2 \xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*} (\xi_{X_{\text{from}} \rightarrow X_{\text{to}}} - \xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*) \quad (\text{B-10})$$

Where $\frac{\partial J}{\partial \xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*}$ is the gradient, which will be denoted as ∇ . $\frac{\partial^2 J}{\partial^2 \xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*}$ is the Hessian, which will be denoted as $\mathbf{H}_{\text{from} \rightarrow X_{\text{to}}}$. By filling in for the integral and by noting that $\nabla J_{X_{G_i}}(\xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*) = 0$:

$$\int_{\xi_{X_{\text{from}} \rightarrow X_{\text{to}}}} e^{-J_{X_{G_i}}(\xi_{X_{\text{from}} \rightarrow X_{\text{to}}})} d\xi_{X_{\text{from}} \rightarrow X_{\text{to}}} \approx e^{-J_{X_{G_i}}(\xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*)} \quad (\text{B-11})$$

$$\int_{\xi_{X_{\text{from}} \rightarrow X_{\text{to}}}} e^{-\frac{1}{2} (\xi_{X_{\text{from}} \rightarrow X_{\text{to}}} - \xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*)^T \mathbf{H}_{\text{from} \rightarrow X_{\text{to}}} (\xi_{X_{\text{from}} \rightarrow X_{\text{to}}} - \xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*)} d\xi_{X_{\text{from}} \rightarrow X_{\text{to}}} \quad (\text{B-12})$$

The latter integral can be assumed to be a Gaussian integral, because the exponential decays very fast away from $\xi_{X_{\text{init}} \rightarrow X_s(t)}^*$, and can be calculated:

$$\int_{\xi_{X_{\text{from}} \rightarrow X_{\text{to}}}} e^{-\frac{1}{2}(\xi_{X_{\text{from}} \rightarrow X_{\text{to}}} - \xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*)^T \mathbf{H}_{\text{from} \rightarrow X_{\text{to}}} (\xi_{X_{\text{from}} \rightarrow X_{\text{to}}} - \xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*)} \approx \frac{\sqrt{2\pi^k}}{\sqrt{|\mathbf{H}_{\text{from} \rightarrow X_{\text{to}}}|}} \quad (\text{B-13})$$

By filling this into Equation (B-12), this results into:

$$\int_{\xi_{X_{\text{from}} \rightarrow X_{\text{to}}}} e^{-J_{X_{G_i}}(\xi_{X_{\text{from}} \rightarrow X_{\text{to}}})} \approx e^{-J_{X_{G_i}}(\xi_{X_{\text{from}} \rightarrow X_{\text{to}}}^*)} \frac{\sqrt{2\pi^k}}{\sqrt{|\mathbf{H}_{\text{from} \rightarrow X_{\text{to}}}|}} \quad (\text{B-14})$$

This can be used to approximate Equation (B-7):

$$P(\xi_{X_{\text{init}} \rightarrow X_s(t)} | X_{G_i}) = e^{-J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_s(t)})} \frac{e^{-J_{X_{G_i}}(\xi_{X_s(t)}^* \rightarrow X_{G_i})} \sqrt{|\mathbf{H}_{X_s(t) \rightarrow X_{G_i}}|}}{e^{-J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_{G_i}}^*)} \sqrt{|\mathbf{H}_{X_{\text{init}} \rightarrow X_{G_i}}|}} \quad (\text{B-15})$$

For the cost function $J_{X_{G_i}}$, trajectory length has been chosen, which penalizes long trajectories when there is a shorter available. By choosing the cost function to be quadratic, the Hessian is constant resulting in:

$$P(\xi_{X_{\text{init}} \rightarrow X_s(t)} | X_{G_i}) = e^{-J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_s(t)})} \frac{e^{-J_{X_{G_i}}(\xi_{X_s(t)}^* \rightarrow X_{G_i})}}{e^{-J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_{G_i}}^*)}} \quad (\text{B-16})$$

With as cost function:

$$J_{X_{G_i}}(\xi) = \sum_i \|\xi(i) - \xi(i-1)\|^2 \quad (\text{B-17})$$

The resulting prediction problem follows by filling Equation (B-16) into Equation (B-3):

$$X_{G_i}^* = \arg \max_{X_{G_i} \in \mathcal{G}} e^{-J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_s(t)})} \frac{e^{-J_{X_{G_i}}(\xi_{X_s(t)}^* \rightarrow X_{G_i})}}{e^{-J_{X_{G_i}}(\xi_{X_{\text{init}} \rightarrow X_{G_i}}^*)}} P(X_{G_i}) \quad (\text{B-18})$$

Bibliography

- [1] A. Schiele, "Towards the Interact Space Experiment: Controlling an Outdoor Robot on Earth's Surface from Space," in *13th symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2015.
- [2] E. Coleshill, L. Oshinowo, R. Rembala, B. Bina, D. Rey, and S. Sindelar, "Dextre: Improving maintenance operations on the international space station," *Acta Astronautica*, vol. 64, no. 9, pp. 869–874, 2009.
- [3] J. Smisek, M. van Paassen, and A. Schiele, "Haptic Guidance in Bilateral Teleoperation: Effects of Guidance Inaccuracy," in *IEEE World Haptics Conference (WHC)*, 2015.
- [4] A. D. Dragan and S. S. Srinivasa, *Formalizing assistive teleoperation*. MIT Press, July, 2012.
- [5] H. El-Hussieny, S. F. Assal, A. Abouelsoud, and S. M. Megahed, "A novel intention prediction strategy for a shared control tele-manipulation system in unknown environments," in *2015 IEEE International Conference on Mechatronics (ICM)*, pp. 204–209, Mar. 2015.
- [6] D. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 624–637, Oct. 1993.
- [7] S. Kimmer, J. Smisek, and A. Schiele, "Effects of Haptic Guidance and Force Feedback on Mental Rotation Abilities in a 6-DOF Teleoperated Task," in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Oct 2015.
- [8] A. Albu-Schäffer, S. Haddadin, Ch. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, "The DLR lightweight robot: design and control concepts for robots in human environments," *Industrial Robot: An International Journal*, vol. 34, pp. 376–385, Aug. 2007.
- [9] "Robotic refueling mission." http://ssco.gsfc.nasa.gov/robotic_refueling_mission.html. Accessed: 11-08-2015.

- [10] J. van Oosterhout, J. Wildenbeest, H. Boessenkool, C. Heemskerk, M. de Baar, F. van der Helm, and D. Abbink, "Haptic Shared Control in Tele-manipulation: Effects of Inaccuracies in Guidance on Task Execution," *IEEE Transactions on Haptics*, vol. PP, no. 99, pp. 1–1, 2015.
- [11] M. H. Raibert and J. J. Craig, "Hybrid Position/Force Control of Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, pp. 126–133, June 1981.
- [12] N. Hogan, "Impedance Control: An Approach to Manipulation: Part I—Theory," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 1–7, Mar. 1985.
- [13] A. Bettini, P. Marayong, S. Lang, A. Okamura, and G. Hager, "Vision-assisted control for manipulation using virtual fixtures," *IEEE Transactions on Robotics*, vol. 20, pp. 953–966, Dec. 2004.
- [14] C. Ott, R. Mukherjee, and Y. Nakamura, "Unified Impedance and Admittance Control," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 554–561, May 2010.
- [15] R. J. Anderson and M. W. Spong, "Asymptotic Stability for Force Reflecting Teleoperators with Time Delay," *The International Journal of Robotics Research*, vol. 11, pp. 135–149, Apr. 1992.
- [16] B. Hannaford, "Stability and performance tradeoffs in bi-lateral telemanipulation," in *1989 IEEE International Conference on Robotics and Automation, 1989. Proceedings*, pp. 1764–1767 vol.3, May 1989.
- [17] Y. Yokokohji and T. Yoshikawa, "Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 605–620, Oct. 1994.
- [18] T. B. Sheridan, *Telerobotics, automation, and human supervisory control*. MIT press, 1992.
- [19] W. Ferrell and T. Sheridan, "Supervisory control of remote manipulation," *IEEE Spectrum*, vol. 4, pp. 81–88, Oct. 1967.
- [20] T. B. Sheridan and W. L. Verplank, "Human and computer control of undersea teleoperators," tech. rep., DTIC Document, 1978.
- [21] T. B. Sheridan, *Humans and automation: System design and research issues*. John Wiley & Sons, Inc., 2002.
- [22] D. A. Abbink and M. Mulder, "Neuromuscular analysis as a guideline in designing shared control," 2010.
- [23] D. A. Abbink and M. Mulder, "Exploring the dimensions of haptic feedback support in manual control," *Journal of Computing and Information Science in Engineering*, vol. 9, no. 1, p. 011006, 2009.
- [24] S. Park, R. D. Howe, and D. F. Torchiana, "Virtual fixtures for robotic cardiac surgery," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2001*, pp. 1419–1420, Springer, 2001.

-
- [25] P. Marayong, A. Bettini, and A. Okamura, "Effect of virtual fixture compliance on human-machine cooperative manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002*, vol. 2, pp. 1089–1095 vol.2, 2002.
- [26] J. J. Abbott and A. M. Okamura, "Virtual fixture architectures for telemanipulation," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2, pp. 2798–2805, IEEE, 2003.
- [27] D. Kragic, P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, "Human-Machine Collaborative Systems for Microsurgical Applications," *The International Journal of Robotics Research*, vol. 24, pp. 731–741, Sept. 2005.
- [28] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic Virtual Fixtures for Robot-Assisted Manipulation," in *Robotics Research* (D. S. Thrun, D. R. Brooks, and D. H. Durrant-Whyte, eds.), no. 28 in Springer Tracts in Advanced Robotics, pp. 49–64, Springer Berlin Heidelberg, 2007.
- [29] L. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *1993 IEEE Virtual Reality Annual International Symposium, 1993*, pp. 76–82, Sept. 1993.
- [30] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive Virtual Fixtures for Machine-Assisted Teleoperation Tasks," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005*, pp. 1139–1144, Apr. 2005.
- [31] H. Boessenkool, D. Abbink, C. Heemskerk, F. van der Helm, and J. Wildenbeest, "A Task-Specific Analysis of the Benefit of Haptic Shared Control During Telemanipulation," *IEEE Transactions on Haptics*, vol. 6, no. 1, pp. 2–12, 2013.
- [32] N. Stefanov, C. Passenberg, A. Peer, and M. Buss, "Design and Evaluation of a Haptic Computer-Assistant for Telemanipulation Tasks," *IEEE Transactions on Human-Machine Systems*, vol. 43, pp. 385–397, July 2013.
- [33] P. G. Griffiths and R. B. Gillespie, "Sharing control between humans and automation using haptic interface: primary and secondary task performance benefits," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 47, no. 3, pp. 574–590, 2005.
- [34] M. Mulder, D. A. Abbink, and E. R. Boer, "Sharing Control With Haptics Seamless Driver Support From Manual to Automatic Control," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 54, pp. 786–798, Oct. 2012.
- [35] "Vicon motion capture systems." <http://www.vicon.com>. Accessed: 12-10-2015.
- [36] N. Mol, "Teleoperation Support System Robust to Uncertainties in Haptic Guidance, literature survey," (Delft), 2015.
- [37] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *The Journal of Neuroscience*, vol. 5, pp. 1688–1703, July 1985.

- [38] M. E. Andani and F. Bahrami, "COMAP: A new computational interpretation of human movement planning level based on coordinated minimum angle jerk policies and six universal movement elements," *Human movement science*, vol. 31, no. 5, pp. 1037–1055, 2012.
- [39] D. E. Whitney, "Quasi-Static Assembly of Compliantly Supported Rigid Parts," *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, pp. 65–77, Mar. 1982.
- [40] A. Haddadi and K. Hashtrudi-Zaad, "A new method for online parameter estimation of Hunt-Crossley environment dynamic models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008*, pp. 981–986, Sept. 2008.
- [41] K. Lee and M. Buss, "Force tracking impedance control with variable target stiffness," in *Proceedings of the 17th IFAC World Congress*, vol. 17, pp. 6751–6756, 2008.
- [42] H. Seraji, "Adaptive admittance control: an approach to explicit force control in compliant motion," in *1994 IEEE International Conference on Robotics and Automation, 1994. Proceedings*, pp. 2705–2712 vol.4, May 1994.
- [43] H. Seraji and R. Colbaugh, "Force Tracking in Impedance Control," *The International Journal of Robotics Research*, vol. 16, pp. 97–117, Feb. 1997.
- [44] S. Jung, T. Hsia, and R. Bonitz, "Force tracking impedance control of robot manipulators under unknown environment," *IEEE Transactions on Control Systems Technology*, vol. 12, pp. 474–483, May 2004.
- [45] F. Almeida, A. Lopes, and P. Abreu, "Force-impedance control: a new control strategy of robotic manipulators," *Recent advances in Mechatronics*, pp. 126–137, 1999.
- [46] A. Lopes and F. Almeida, "A force-impedance controlled industrial robot using an active robotic auxiliary device," *Robotics and Computer-Integrated Manufacturing*, vol. 24, pp. 299–309, June 2008.
- [47] B. Friedland, *Control system design: an introduction to state-space methods*. Courier Corporation, 2012.
- [48] K. Ogata, *Modern control engineering*. Prentice Hall PTR, 2001.
- [49] S. Hirche, M. Ferre, J. Barrio, C. Melchiorri, and M. Buss, "Bilateral Control Architectures for Telerobotics," in *Advances in Telerobotics* (M. Ferre, M. Buss, R. Aracil, C. Melchiorri, and C. Balaguer, eds.), no. 31 in Springer Tracts in Advanced Robotics, pp. 163–176, Springer Berlin Heidelberg, 2007.
- [50] M. Ferre, J. Barrio, C. Melchiorri, J. M. Bogado, P. L. Castedo, and J. M. Ibarra, "Experimental Results on Bilateral Control Using an Industrial Telem manipulator," in *Advances in Telerobotics* (M. Ferre, M. Buss, R. Aracil, C. Melchiorri, and C. Balaguer, eds.), no. 31 in Springer Tracts in Advanced Robotics, pp. 177–190, Springer Berlin Heidelberg, 2007.
- [51] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on*, pp. 301–308, IEEE, 2013.

- [52] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “Maximum Entropy Inverse Reinforcement Learning.,” in *AAAI*, pp. 1433–1438, 2008.

Glossary

List of Acronyms

ESA	European Space Agency
NASA	National Aeronautics and Space Administration
CSA	Canadian Space Agency
ISS	International Space Station
RRM	Robotic Refuelling Mission
SPDM	Special Purpose Dexterous Manipulator
DOF	Degree of Freedom
GUI	Graphical User Interface
AMBOT	American Robot Company
RSI	Robot Sensor Interface
ZMWN	Zero Mean White Noise
NRMSE	Normalized Root Mean Squared Error

List of Symbols

\mathcal{G}	Set of accessible goals
$\Psi_{f/t}$	F/t frame
Ψ_{tool}	Tool frame
$\xi_{X_{\text{init}} \rightarrow X_{G_i}}^*$	Optimal trajectory from X_{init} towards goal X_{G_i}
$\xi_{X_s(t) \rightarrow X_{G_i}}^*$	Optimal trajectory to be travelled from $X_s(t)$ towards goal X_{G_i}

$\xi_{X_{\text{init}} \rightarrow X_s(t)}$	Set of past slave robot configurations
$\hat{X}_e(s)$	Estimated location of the environment
τ_d	Desired torque in inner torque control loop
τ_s	The measured torque of the slave robot
ζ	Damping ratio
B_e	The damping of the environment
B_s	The desired damping of the slave robot
B_y	Damping of the admittance controller
B_{slave}	The damping of the slave robot
$C_1(s)$	Forward velocity channel controller
$C_2(s)$	Backward force channel controller
$C_3(s)$	Forward force channel controller
$C_4(s)$	Backward velocity channel controller
$C_f(s)$	Force channel controller
$C_g(s)$	Shared control gain
$C_m(s)$	Local master controller
$C_s(s)$	Local slave controller
$E_f(s)$	Force error
$E_g(s)$	Error between the inaccurate reference position and the current slave robot position
$E_x(s)$	Position error
$F(s)$	General representation of forces
$F_c(s)$	Output force of the impedance controller
$F_d(s)$	Desired force
$F_e(s)$	Force and torques exerted on the environment
$F_g(s)$	Guidance forces that are provided by the support system
$F_h(s)$	Input force from human operator
$F_m(s)$	Force acting on master device
$F_s(s)$	Controlled motor force on the slave robot
$F_{\text{fb}}(s)$	Reflected force from the slave robot to the master device
$H(s)$	The hybrid matrix
$J_{X_{G_i}}$	Cost function where trajectories are evaluated in
K_e	The stiffness of the environment
K_h	Stiffness of the operator
K_s	The desired stiffness of the slave robot
K_y	Stiffness of the admittance controller
M_s	The desired mass of the slave robot
M_{slave}	The mass of the slave robot
T	End time
t	Current time

T_d	Time delay
$X(s)$	General representation of motions
$X_c(s)$	Correction term of the admittance controller
X_e	The position of the environment
$X_g(s)$	Position of the centerline determined by the support system
$X_m(s)$	Configuration of the master device
$X_r(s)$	Reference motion
$X_s(s)$	Configuration of slave robot
X_{hole}	Real centerline position of the hole (Peg-in-hole task)
$X_{\text{input}}(s)$	Control input to the system that is to be controlled in an input-mixing architecture
$X_{G_{\text{intended}}}$	Intended goal by the operator
X_{G_i}	Goal location
$X_{sd}(s)$	Set-point configuration for the slave robot
$Y(s)$	Admittance
$Z(s)$	Impedance
$Z_m(s)$	Dynamics of master device
$Z_s(s)$	Dynamics of the slave robot
$Z_t(s)$	Transmitted impedance to the operator

