

Lehrstuhl für Steuerungs- und Regelungstechnik  
Technische Universität München

Univ.-Prof. Dr.-Ing.(Univ. Tokio) Martin Buss

# High-Fidelity Haptics in Multimodal Human-Robot Interaction

**Zheng Wang**

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik  
der Technischen Universität München zur Erlangung des akademischen Grades eines

**Doktor-Ingenieurs (Dr.-Ing.)**

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Gordon Cheng, Ph.D.

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing.(Univ. Tokio) Martin Buss
2. Univ.-Prof. Dr.-Ing., Dr.-Ing.habil. Alois Knoll

Die Dissertation wurde am 21.06.2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 29.10.2010 angenommen.



# Foreword

This dissertation concludes four years of my research conducted at the Institute of Automatic Control Engineering (LSR), Technische Universität München. The work is supported by European Union FP6 project *Immersence*.

First of all, I would like to thank *mein Doktorvater*, my supervisor Prof. Dr.-Ing/Univ. Tokio Martin Buss, who not only offered me a research position in one of the best established robotics group, but also showed me the path through academia to where I am today. He had always shared with me his vision upon research topics while leaving me enough freedom for exploration and innovation.

My next most sincere appreciation goes to Dr. Angelika Peer, with whom I shared uncountable scientific discussions. She had always helped me with her experience and wisdom during the last two years of my research at LSR.

A special thank goes to Prof. Louis Phee, my current supervisor at Nanyang Technological University, Singapore, whose kindness and help made it possible for me to finish writing the dissertation.

Much of the work in this dissertation was conducted in collaboration with my Immersence partners within, and outside LSR. My sincere thanks to Jens Hoelldampf, Raphaela Groten, Ansgar Bittermann, for all the inspirations and help in LSR; to Elias and Mel at UPC, to Nicola, Pasquale, Mario, and other members of the Italian team I cannot name all, at UNIPI, to Benjamin, Juan, Paul, and Abder at LSC, to Christos at UBIRM, to Max and Marc at MPI, to Javier and Manuel at UPM, for all the good collaboration and achievements in Immersence.

A brief, but very special appreciation goes to Mr. Andreas Schweinberger. He had offered me so much help when I first started my life in Deutschland, and kept being a friend and supporter of me during my hardest times. *Little kindness goes a long way.*

Next I would like to thank my LSR colleagues, with whom I shared past four years of my life. Special thanks go to Chih-Chung Chen, Tingting Xu, Tianguang Zhang, Hao Ding, and Haiyan Wu, for lunch, dinner, and other uncountable events and help. Thanks to Kwang-kyu Lee, my first roommate and best friend, who shared with me beers and wisdom. I also would like to thank Daniela Feth, Carollina Passenberg, Thomas Schauss, and Nikolay Stefanov, for the fruitful discussions in the haptic group; to thank Tobias Goepel, Matthias Rungger, Iason Vittorias and Andreas Schmid, for sharing an office with me.

My next appreciations go to Dr. Dirk Wollherr, who had always kindly helped me in administrative issues, and the secretaries and technicians of LSR, Fr. Schmid, Fr. Werner, Fr. Renner, Hr. Jaschik, Hr. Gradl, Hr. Kubick, and Hr. Lowitz, without your help, nothing would have been possible.

I would also like to thank all students who had worked with me during the past years, especially Jun, Ziqing, Ji, Yang, Lei, Qixun, Mingxiang, Licheng, Rubens, Hong, and

---

Shuning, who contributed their hard work to the development of the systems as well as the experiments.

My final and most sincere thanks go to my family, and to the ones who had always put their faith in me, no matter how remote they might be. *Xiè Xiè Nǐ Mén.*

Singapore, June 2010

Zheng Wang

*To life...*



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem definition . . . . .	1
1.2	Main contributions and structure of the dissertation . . . . .	2
<b>2</b>	<b>Haptic rendering of arm dynamics 1: modeling and replay</b>	<b>4</b>
2.1	Physicality and challenges to haptic rendering . . . . .	4
2.2	A framework for human haptic modeling . . . . .	5
2.2.1	Handshake: a process-oriented human motor skill . . . . .	5
2.2.2	Related works . . . . .	6
2.2.3	A framework for human haptic modeling . . . . .	6
2.3	Developing a handshake robot with realistic arm behavior: overview . . . . .	8
2.3.1	Pilot study . . . . .	8
2.3.2	Robotic interfaces . . . . .	9
2.4	Basic controllers . . . . .	10
2.4.1	The first modeling iteration . . . . .	11
2.4.2	The second modeling iteration . . . . .	13
2.4.3	The third modeling iteration . . . . .	14
2.5	Conclusions . . . . .	16
<b>3</b>	<b>Haptic rendering of arm dynamics 2: towards an interactive controller</b>	<b>18</b>
3.1	Human behavior model . . . . .	18
3.2	Interactive controller . . . . .	20
3.2.1	Fast online parameter estimation . . . . .	21
3.2.2	Symbol abstraction and HMM intention estimation . . . . .	26
3.2.3	Trajectory planning and parameter adaptation . . . . .	27
3.2.4	Refined trajectory planning . . . . .	29
3.3	Performance validation tests . . . . .	31
3.3.1	HBP estimation . . . . .	31
3.3.2	HMM estimator . . . . .	36
3.3.3	Overall system performance . . . . .	37
3.4	Conclusions . . . . .	38
<b>4</b>	<b>Haptic rendering of hand dynamics</b>	<b>43</b>
4.1	Gesture data acquisition . . . . .	43
4.1.1	The CyberGlove . . . . .	44
4.1.2	Pisa glove . . . . .	45
4.2	Haptic data acquisition . . . . .	45
4.2.1	Tactile sensing technology . . . . .	46

4.2.2	Glove design and implementation . . . . .	47
4.2.3	Measuring handshakes with TSG gloves . . . . .	52
4.3	Robotic hand actuation . . . . .	57
4.3.1	BarrettHand . . . . .	57
4.4	Conclusions . . . . .	58
<b>5</b>	<b>Visual and sound rendering</b>	<b>63</b>
5.1	Visual rendering . . . . .	63
5.1.1	The rendering workflow . . . . .	64
5.1.2	Virtual human characters . . . . .	66
5.1.3	Virtual hand model . . . . .	67
5.1.4	Virtual environments . . . . .	68
5.1.5	Events . . . . .	69
5.2	Auditory rendering . . . . .	70
5.3	Conclusions . . . . .	71
<b>6</b>	<b>System integration and optimization</b>	<b>74</b>
6.1	User dependency . . . . .	74
6.1.1	User dependency in haptic subsystem . . . . .	75
6.1.2	User dependency in vision subsystem . . . . .	77
6.2	Real/virtual world integration . . . . .	80
6.2.1	Incorporating real world data . . . . .	80
6.2.2	Registration of real world object . . . . .	82
6.3	Integration of a second human input . . . . .	83
6.3.1	Problem definition . . . . .	84
6.3.2	Proposed remedy . . . . .	84
6.4	Optimization . . . . .	86
6.4.1	Improving natural haptic interaction . . . . .	86
6.4.2	Minimizing computational load . . . . .	87
6.4.3	Time delay . . . . .	87
6.5	Conclusions . . . . .	89
<b>7</b>	<b>Experiments and evaluation studies</b>	<b>90</b>
7.1	Plausibility and questionnaire . . . . .	90
7.2	Experiment 1: Robotic control algorithms comparison . . . . .	91
7.2.1	Haptic rendering algorithms for HLC controller . . . . .	92
7.2.2	The experiment and results . . . . .	93
7.2.3	Analysis and discussions . . . . .	94
7.2.4	Conclusions . . . . .	98
7.3	Experiment 2: Evaluation with haptics and audition . . . . .	98
7.3.1	Introduction . . . . .	99
7.3.2	Experimental setup . . . . .	99
7.3.3	Experimental procedures . . . . .	101
7.3.4	Analysis and results . . . . .	102
7.3.5	Conclusions . . . . .	105



7.4	Experiment 3: Study of haptic-visual integration . . . . .	106
7.4.1	Introduction . . . . .	106
7.4.2	Experimental setup . . . . .	107
7.4.3	Experimental design . . . . .	109
7.4.4	Results and discussion . . . . .	110
7.4.5	Conclusions . . . . .	111
7.5	Experiment 4: Evaluation with vision, haptics, and audition . . . . .	111
7.5.1	Introduction . . . . .	111
7.5.2	Experimental design . . . . .	111
7.5.3	Experimental procedure . . . . .	113
7.5.4	Results and analysis . . . . .	114
7.5.5	Conclusions . . . . .	115
7.6	Discussions and conclusions . . . . .	116
<b>8</b>	<b>Conclusions and future directions</b>	<b>118</b>
8.1	Concluding remarks . . . . .	118
8.2	Vistas . . . . .	119
	<b>Bibliography</b>	<b>121</b>

# Notations

## Abbreviations

DOF	Degree of freedom
HBP	Human behavioral parameter
HLC	High-level controller
HMM	Hidden Markov model
HRI	Human-robot interaction
LLC	Low-level controller
LS	Least-squares
LSMW	Least-squares with moving window
fMRI	Functional magnetic resonance imaging
OI	OpenInventor
PAC	Position-based admittance control
PbD	Programming by demonstration
PC	Position control
RLS	Recursive least-squares
SOA	State-of-the-art
VE	Virtual environment
ViSHaRD7	Virtual Scenario Haptic Rendering Device with 7 actuated DOF
ViSHaRD10	Virtual Scenario Haptic Rendering Device with 10 actuated DOF
VR	Virtual reality

## Conventions

$x$	variable scalar
$X$	constant scalar
$\mathbf{x}$	vector
$\mathbf{X}$	matrix
$f(\cdot)$	scalar function
$\mathbf{f}(\cdot)$	vector function
$\dot{x}$	time derivative
$\ddot{x}$	second order time derivative
$\hat{x}$	estimation of $x$
$\tilde{x}$	value of $x$ after calculation

## Subscripts and Superscripts

$x_0$	initial value of $x$
$x_d$	desired value of $x$
$x_h$	$x$ generated by human
$\mathbf{x}^T$	vector transpose
$\mathbf{X}^T$	matrix transpose
$\mathbf{X}^{-1}$	matrix inverse
$\mathbf{X}^\#$	matrix pseudo-inverse

## Symbols

$a$	acceleration
$A$	amplitude
$B$	damping coefficient
$c^*$	currently effective HMM
$D$	direction
$e$	error
$f$	force
$h$	human behavioral parameter
$i, j$	index
$K$	stiffness
$l, L$	data length
$M$	mass
$O$	observation
$S$	hidden state of HMM
$t$	time constant
$v$	velocity
$\mathbf{w}$	noise vector
$x$	displacement
$\mathbf{X}$	input matrix
$\mathbf{Y}$	output matrix
$\alpha, \beta$	parameters for $\lambda$ calculation
$\delta_i$	the $i$ -th HMM
$\eta$	coefficient for stiffness calculation
$\theta$	parameter vector
$\sigma$	boundary value of $\lambda$
$\lambda$	forgetting factor



## **Abstract**

Introducing haptics into human-robot interaction brings new challenges to control algorithm development as the robot stays in physical contact with the human user. When the aim is to design a system which interacts with the human in a natural and plausible way similar to two humans carrying out the same task, the main challenges are to estimate human intentions online and to plan the behavior of the robot accordingly. This work addresses the challenges of human intention estimation, online robot action generation and the integration of visual and sound feedback signals into the haptic system. Different stages of the system have been evaluated by a series of user studies.

## **Zusammenfassung**

Die Einführung von Haptik in die Mensch-Roboter Interaktion bringt neue Herausforderungen in Bezug auf die Entwicklung von Regelungsalgorithmen mit sich, da der Roboter in physikalischem Kontakt mit dem menschlichen Bediener steht. Wenn das System so gestaltet werden soll, dass es eine möglichst natürliche und plausible Interaktion mit dem Menschen erlaubt, ähnlich dem wie zwei Menschen die Aufgabe durchführen würden, bestehen die größten Herausforderungen darin, die menschlichen Intentionen online zu schätzen und das Verhalten des Roboters entsprechend zu planen. Diese Arbeit befasst sich mit den Herausforderungen der menschlichen Intentionsschätzung, der Generierung von online ausführbaren Roboteraktionen sowie mit der Integration von visuellen und auditiven Signalen in das haptische System. Verschiedene Systemstadien wurden anhand einer Reihe von Benutzerstudien evaluiert.

*"Once Zhuangzi dreamt he was a butterfly, a butterfly flitting and fluttering around, happy with himself and doing as he pleased. He did not know he was Zhuangzi. Suddenly he woke up and there he was, solid and unmistakable Zhuangzi. But he did not know if he was Zhuangzi who had dreamt he was a butterfly, or a butterfly dreaming he was Zhuangzi."*

- *"Equality of All Substances"*, Zhuangzi, *Ancient Chinese Philosopher, 369BC - 286BC*

It is the ultimate goal of a virtual reality system to provide a total plausible experience to the user, such that the distinction between the real world and the virtual world cannot be made without the assistance of an external reference.

# 1 Introduction

The term *haptics* is getting known to the public recently by means of touch-screen mobile phones and the newest generation of game consoles equipped with accelerometers. However, in most virtual environment (VE) applications, the virtual reality (VR) system measures haptic user input information while providing hardly any haptic feedback other than vibration.

This work focuses on the real-time interaction dynamics of a mechanically directly connected dyad, whether being two humans, or one human and one robot. Human intentions on how to complete the task are abstracted from measured haptic signals and used in generating robot behavior online. A multimodal VE is created and integrated with the haptic subsystem. A series of evaluation experiments with human participants were carried out to assess the performances of the developed systems with different sensory channel feedback signals provided to the participants.

## 1.1 Problem definition

In order to break down the general topic, a specific task is selected. Throughout this dissertation, handshaking is taken as a representative of human-human haptic interaction tasks with direct contact, i.e. without a tool or via an intermediate object. Handshaking is a common daily activity. In order to simulate human-human handshaking through human-robot interaction (HRI), problems such as signal measurement, control of robot interactive behavior, as well as evaluation must be tackled and solved. [50] created the first tele-handshake using a simple one degree-of-freedom (DOF) device while [68] generated handshake animations from a vision system. However, very few have viewed handshaking from a force/motion interaction aspect, until very recent works reported in [72], where the authors take the oscillation synchronization approach to realize human-robot handshaking; and in [108] the authors focused on the approaching and shaking motions of a handshaking robot. The study of handshaking in a haptic interaction context, however, is lacking in existing literature to the best of knowledge.

The aim of this work is to develop a high-fidelity VR system that can provide realistic handshaking experiences to human users. The haptic component of the system can recreate handshakes from a dynamic interaction point of view, close to real world human-human handshaking; while the vision and sound subsystems can comply with the haptic subsystem in creating a plausible VE interaction experience.

In order to achieve the above objectives, a wide range of problems related to VR and haptic interaction are to be tackled. Human handshaking behavior needs to be modeled. Moreover, in order to interact with the human handshake partner, the robot needs to be able to address human actions online. Therefore an advanced interactive controller is desirable to control the robot for dynamic interactions. Vision-wise, a VR system is needed

with the ability to generate animations online from real-time haptic input signals. Finally evaluation techniques are needed to validate the developed systems.

## 1.2 Main contributions and structure of the dissertation

The contributions of this work consist of three parts: haptic modeling, visual/haptic integration, and evaluation. Addressing all the aforementioned challenges, a handshake VR system with high-fidelity haptics is developed. The haptic interaction controller can estimate human partner intentions regarding handshake behavior online, and adapt its behavior and dynamical properties of the arm accordingly. A VE with virtual characters of high-fidelity facial details is created, with the ability to generate full body as well as facial animations online. The VE is also integrated with the haptic system and full body motion tracking system of the human user, such that the behavior of the human and the robot can be represented by virtual characters in the VE. Evaluation studies are carried out, jointly with the research group in UPC, Spain, on assessing the performance of the developed systems.

To the authors' knowledge, the handshake VR system described in this dissertation is world-unique not only in realizing high-fidelity human-robot handshaking, but also in haptic HRI while addressing human intentions. It is also novel for the evaluation technique to be applied to human haptic VR interaction tasks.

Although the discussions of this dissertation are made based on handshaking, a large proportion of the methods and findings are generalizable to other applications. For instance, the interactive controller with human intention estimation is also applicable to other haptic interaction tasks where robotic assistants need to consider human intentions such as in training or rehabilitation; visual/haptic integration studies can be applicable to entertainment or other scenario of multimodal integration, where haptic signal is integrated into the VE.

The dissertation is formulated in 7 chapters. A brief outline is given here starting from Chapter 2.

Modeling and recreation of the human handshake arm behavior onto the robot are the main contributions of this work, hence the discussions are divided into two chapters. In Chapter 2, the observation of human-human handshakes are presented, with a basic controller developed to replay the recorded human arm behavior by a robotic arm; while in Chapter 3, further studies are carried out, with a world-unique interactive controller of the robot to handle online interaction with human partners.

Besides the arm, hand also plays an important role in handshaking. Chapter 4 focuses on the modeling and recreation of hand behavior of a handshake. Devices to capture human gesture and force information during handshake are studied. A robotic hand is used to map the human motion online.

Based on the handshake partner robot, a multimodal VE is created in Chapter 5 with vision and sound feedbacks integrated with the haptic subsystem, while Chapter 6 focuses on the integration of the system and the improvement of the overall performance.

In Chapter 7 a series of user studies to evaluate the performance of the developed systems are carried out. In the four experiments with human participants, different candidate



realizations of the VR system are compared in terms of interaction plausibility. Questionnaires are used to assess the opinions of the participants, with statistical methods from psychological studies to analyze the results. The findings of the experiments suggest that the robot with the interactive controller feels significantly more realistic, in terms of interaction plausibility, than the robot with a basic controller.

Conclusions and future directions are drawn in Chapter 8.

## 2 Haptic rendering of arm dynamics 1: modeling and replay

### 2.1 Physicality and challenges to haptic rendering

An important goal of haptic research in the context of immersive virtual environments is to extend the range of multi-sensory feedback to participants in order to introduce physicality into the virtual reality experience. Physicality of haptic interaction, among other factors, is a critical issue to be considered in rendering realistic interaction. For instance, the participant should be able to feel the weight and elasticity of virtual objects as well as touch and interact with a virtual human partner in a real-life manner.

Towards this end a haptic rendering algorithm is presented in this chapter that aims towards achieving real-world-like experience to the human participant interacting with a virtual human partner. To further break down the problem, this chapter focuses on the arm dynamics during interaction, while the hand dynamics will be investigated in the next chapter.

A simple handshaking task is considered in this chapter, where a human participant grasps a haptic interface, representing the arm of a virtual human partner, and using this to perform a handshake with a virtual partner. Advanced controllers for the haptic interface are developed that implement a partner who adapts online to the human and gives her/him the illusion of taking part in a real human-human handshake.

New challenges for the controller design arise when rendering interactions with a virtual human partner compared to interactions with passive environments:

- The dynamic properties, for example, stiffness, of a human arm can vary over time, therefore the assumptions of constant mechanical properties of the environment as well as being independent of the actual state of interaction no longer hold;
- In case of direct contact (human and robot directly connected to each other, not via any object), such as in handshaking, interaction forces between the robot and the human are measured by a single force/torque sensor mounted at the end-effector of the robot, which only allows the measurement of one interaction force signal rather than two individual forces applied by each side, hence it is not possible to separate the force applied by one partner from the other;
- The human intention of how to carry on the interaction is not directly measurable, since the information is part of the internal mental processing of the participant.

To summarize, a human as an interacting partner has unmeasurable intentions/plans in the mind, time-varying dynamic properties of the arm (which are very difficult to measure accurately), with only the interaction force measurement available to the robot.

Bearing in mind these various challenges, two different approaches to design the robot controller can be considered (a) design a basic controller that carries out the handshake as pre-defined, ignoring the bilateral nature of human-human interaction and forgetting about the human-related information that is difficult to acquire; or (b) design an interactive controller that estimates the necessary human information and uses it to simulate a more interactive virtual partner. Both approaches are adopted for comparison.

The chapter is structured as follows. In Section 2.2, the challenges of haptic modeling are discussed, then a general scheme of the modeling workflow is proposed. In Section 2.3 and 2.4, the modeling of handshaking is discussed as an example of employing the proposed scheme. Conclusions are given at the end in Section 3.4.

Comparing with the state-of-the-art (SOA), the novelty of this chapter is the framework for human haptic modeling proposed in Section 2.2, which attempts to formulate the task specific workflows into a standard scheme. In Section 2.4 a large amount of human behavioral data is collected from human-human handshake experiments. On one hand the findings from human-human experiments are the basis of the basic controllers developed in Section 2.4, while on the other hand the collected human behavioral data is applicable to future studies.

## 2.2 A framework for human haptic modeling

Researchers have developed various methods in the past decades to let robots learn human behaviors. Works published so far ([1], [3], [4], [28], [50], [66]) are mostly task specific with distinct work-flows. Apart from the common learning methods for some cases such as reinforcement learning ([45], [66]) and programming by demonstration (PbD) ([58], [62], [73]), a systematic description of the process is yet to emerge. In this section, after addressing the common issues for human haptic modeling, a general framework is proposed, trying to address common issues in human haptic modeling and hence provide a guideline for future discussions.

### 2.2.1 Handshake: a process-oriented human motor skill

Humans reason in a symbolic level, see [106], leaving a gap between the symbolic commands and the muscle groups that work in a rather low level to make the actions. The skills that humans accumulate to fulfill this gap are often referred to as *motor skills*. There are certain motor skills that emphasize on haptic interactions. According to the objectives of the task, such skills can be further divided into two categories: goal oriented and process oriented.

*Goal-oriented* motor skills require the achievement of certain termination criteria. Examples of such skills are ping-pong playing, bottle opening, and hammering. The dynamics of the process by which the manipulator achieves the goal is often irrelevant to task completion but affects evaluation criteria such as completion speed or safety.

*Process-oriented* motor skills focus more on the dynamic process even before a final goal is reached. For instance, in violin playing the bowing force and motion must be carefully coordinated to yield the expected timbres; during handshaking, the dynamics perceived by the partner must follow certain patterns in order to make him/her feel comfortable. Some

skills in this category do not necessarily have a final goal to achieve, with the performance measure purely based on the dynamics of the process.

### 2.2.2 Related works

There are individual studies reported on modeling specific human motor skills and transferring them to robots. [1] and [3] reported teaching specialized robots to play ping-pong, while a number of robot manipulators are taught to carry out object manipulation tasks such as hammering by [4] and bottle opening by [28]. However, the above tasks are all goal-oriented. For process-oriented skills, [50] created handshake using a simple 1-degree-of-freedom device, but the device is tele-operated by another human, with little modeling involved; [58] considered robotic assisted laparoscopic surgery where robots are trained to tie knots using PbD; [85] reported teaching unskilled human Japanese calligraphy using a robot; while [49] reported a dancing robot playing the role of a female in a dance. A generic framework is missing from the above publications.

Another important issue is that the above works concerned with manipulation tasks with one human involved. When a second human is included in the tasks, namely human-human interaction tasks such as handshaking ([99], [102], [101]) and dancing ([37]), new challenges are raised. Instead of measuring the behavior of one human carrying out a manipulation task, now it is necessary to measure the interaction and adaptation between two humans. The resulting robot also needs to adapt to the interaction of the second human, instead of an object in manipulation tasks. In [49] the dancing robot plays the role of a female, which is led by a male dancer, hence interaction is not the major concern.

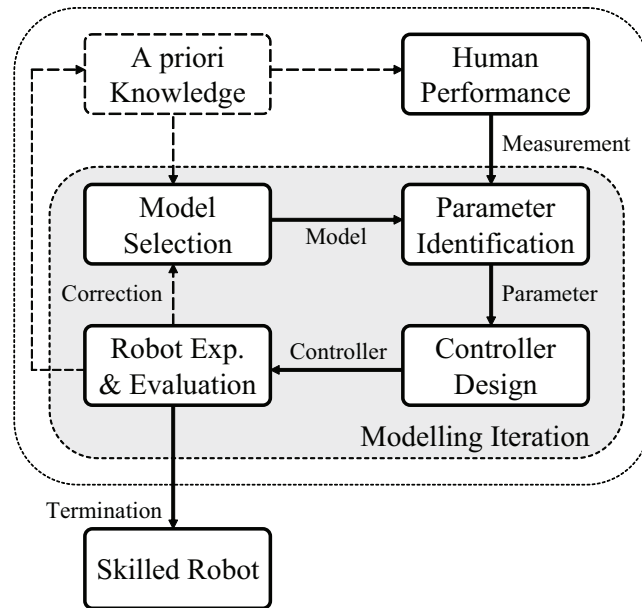
### 2.2.3 A framework for human haptic modeling

A modeling scheme is proposed as shown in Fig. 2.1. It shows a strong emphasize on the measurement of human-human interaction behavior. It is specifically important to process-oriented skill modeling, that the human-human performance measurement serves as a starting point, as the dynamical process is too complex to be defined by any performance index, such as a yes/no value in supervised learning.

Although throughout this dissertation handshaking is concerned, within this section the discussion is kept generic, in order to make the framework applicable to other haptic interaction tasks.

#### **A priori knowledge**

Although automated programming of robots is an active research issue, the learning environment needs to be well defined by experts. At the current stage a priori knowledge is necessary in designing experiments to measure human performance, selecting the initial model and hence the parameter identification method. After each of the modeling iterations the knowledge of the expert is updated by new information from the iteration.



**Figure 2.1:** A general scheme of human haptic modeling. Solid lines denote signals and data flows while dashed lines denote knowledge flows.

### Human performance measurement

This is the main argument of this scheme. Different from other modeling tasks, human haptic modeling often requires human performance measurements, both as a reference for analysis and as an evaluation guideline. It is particularly important for process oriented haptic skill, as the process in concern needs a reference from human performance. It is also important that human performance measurement is taken *before* and *independently* from the engineering of technical systems used to implement the model later on, although the design of the experiments for measuring needs to be guided by the a priori knowledge. By this approach the original information about human performance can be acquired, although subject to the means of measuring, be it by vision [59], force [111], or stochastic models [85], without being distorted by implementation constraints.

### Modeling iteration

The grey dashed box in Fig. 2.1 illustrates the modeling iteration, which consists of model selection, parameter identification, controller design, and robot experiment with evaluation. The model is not specified by the scheme, it can vary from deterministic to stochastic or even hierarchical. After a model is selected, human performance measurements are used to estimate the parameters to the model. Robot controllers are then designed and implemented. Technical constraints of implementation apply in this phase. However, with the previous parts of the modeling being unbounded, future improvement of the technologies can be implemented yielding a better representation of the skill while keeping the modeling results unchanged.

## Termination criteria

Unfortunately the termination policy of modeling can only be task specific. The various factors that affect human performance suggest the possible necessity of borrowing findings from other disciplines. In general when the performance of the robot is within the acceptable range and after iterations the objective does not improve significantly, the modeling can be terminated with a skilled robot capable of the modeled task.

In the next two sections, human handshaking is modeled by different approaches following the proposed scheme and implemented onto a robot. In modeling practice, the proposed scheme gives a clear guideline of workflow. The measured human performance plays an important role in the modeling iterations.

## 2.3 Developing a handshake robot with realistic arm behavior: overview

This section gives an overview of developing a handshake robot that provides the human partner a realistic handshaking experience. The horizon is limited to the haptic modality, meaning that the robot is not expected to *look* like a human, but instead should *shake* like one. Inspired by the proposed scheme, an outline of the workflow is given as follows:

- A pilot study is carried out to gather first knowledge about the handshake dynamics.
- Three iterations are carried out to develop a basic handshake controller.
  - In the first iteration, a large amount of human handshakes are measured. The handshake process is segmented and a switching controller model is used.
  - In the second iteration, desired trajectories are generated by a planner with a motion synthesis module. Trajectory playback is realized on the robot.
  - In the third iteration, new control methods are employed to achieve compliance on the robot. Realistic handshake is achieved with passive human partners.
- An interactive controller is developed based on the basic controller to remove the passive constraint.

For clearness of the text, this section will only cover the pilot experiment, as well as introducing the haptic interfaces that are involved in the discussion. The basic and interactive controllers will be discussed separately in the following sections.

### 2.3.1 Pilot study

The starting point in the proposed scheme is gathering a priori knowledge. Since very few reports can be found in mimicking or recreating handshake dynamics, a pilot experiment is necessary to get close to the problems. 300 handshakes are performed by volunteered male college students. The exchanged force is recorded using a haptic data glove, the TSG-1 as in [100], and the motion is recorded using a video camera. In spite of the limitations on measurement devices, the following findings are drawn from the pre-study experiments:

- Handshake is a dynamic process with intra- and inter- subject uncertainties. The pattern which helps a person to distinguish a realistic handshake from a fake one is yet to be identified. The actual motion trajectory is affected by the exerted force dynamics from both sides of the handshake pair.
- A typical handshake can be divided into three stages: approach, shake, and release. The switching conditions are the contact and lose of contact between the two hands of the participants.
- The shaking stage of handshake consists of both force and position exchanges, while the other two stages involve only free space motion.

The first findings of handshake are used to guide the design and modeling of the first iteration.

### 2.3.2 Robotic interfaces

In order to present the human user a realistic interaction experience, the haptic interfaces involved in such interaction tasks are required to have large workspaces matching those of the human, as well as the ability to carry large payloads running at considerably high speeds.

There are typically two types of haptic interfaces available: impedance type and admittance type. The impedance-type haptic devices, represented by the PHANTOM<sup>TM</sup>[67] family, provide force feedbacks when the user gives position input signals. Impedance-type devices have a big advantage of being capable to render very low impedances when using impedance control.

However when selecting suitable haptic interfaces for handshaking, the workspace and force output capability requirements cannot be fulfilled by impedance-type devices. Therefore admittance-type robotic interfaces are employed.

For admittance-type devices, the user gives force inputs, the device provides position feedback based on calculation. Sacrificing light-weightiness, admittance-type devices can achieve much larger workspace and higher force output capability than impedance-type devices.

Three robotic arms of two different designs are used in the scope of this work. Although both being admittance-type devices, they differ significantly in mechanical design as well as specifications.

#### **ViSHaRD10**

The ViSHaRD10 robotic arm is a 10 DOF redundant robotic device [95]. Two ViSHaRD10 arms are used in this work. One arm is mounted on the floor, while the other on the ceiling. Each arm has a workspace of 60cm\*60cm\*60cm. The end-effector can be placed at any position and orientation within the workspace. A 6 DOF force/torque sensor is mounted on the end-effector to measure force/torque signals during interaction. A picture of the ViSHaRD10 is shown in Fig. 2.2.



**Figure 2.2:** Robotic arm ViSHaRD10, floor mounted.

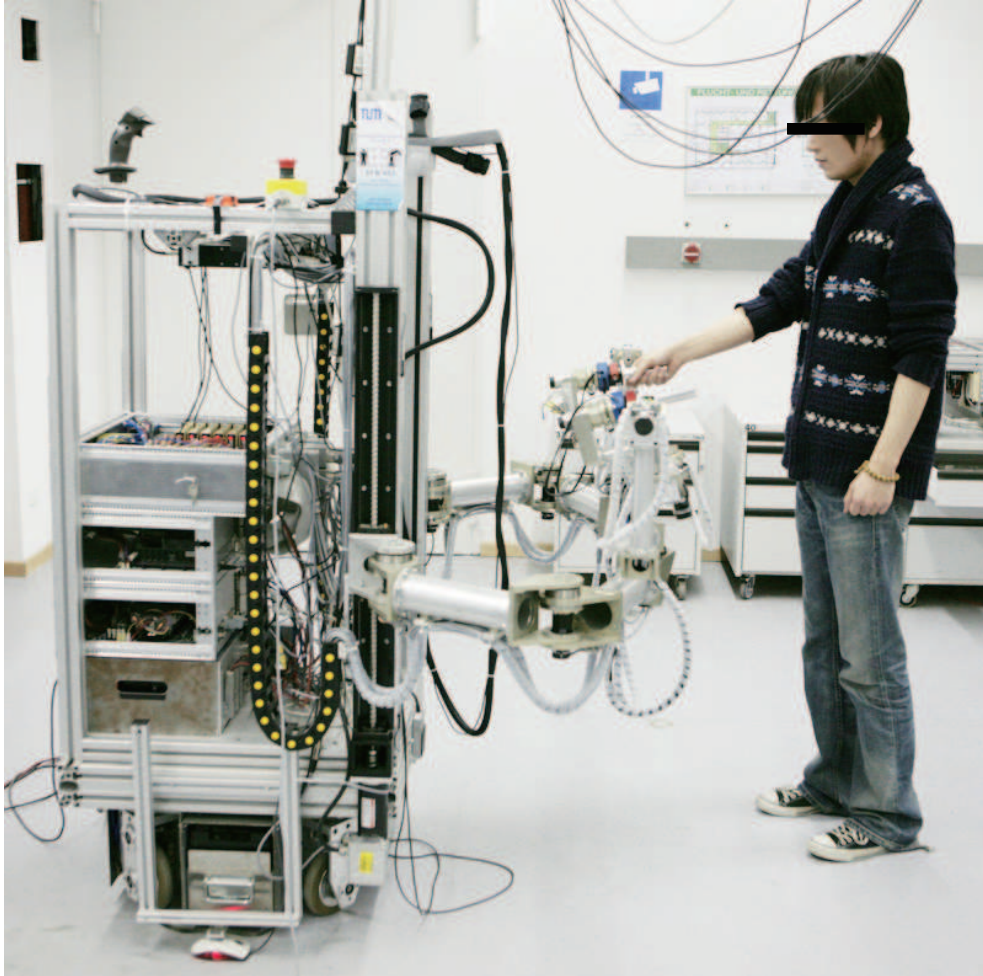
### **ViSHaRD7**

The ViSHaRD7 robotic arm is a 7 DOF redundant robotic device [63]. Two ViSHaRD7 arms are mounted on a mobile platform, while only one is used in this work. A 6 DOF force/torque sensor is mounted on the end-effector to measure interaction force/torque information. A picture of the ViSHaRD7 robotic arm is shown in Fig. 2.3. The mounting mechanism on the end-effector of ViSHaRD7 is identical to the one of ViSHaRD10, therefore the same end-effector tool can be interchangeable between the two robots. Hence it is possible to select a specific robotic arm to use in different tasks, while making no difference to the participant, since the end-effector in use remains the same.

## **2.4 Basic controllers**

In this section, three modeling iterations are carried out following the proposed scheme. Each iteration results in a controller of the handshake robot. With respect to the interac-





**Figure 2.3:** Robotic arm ViSHaRD7.

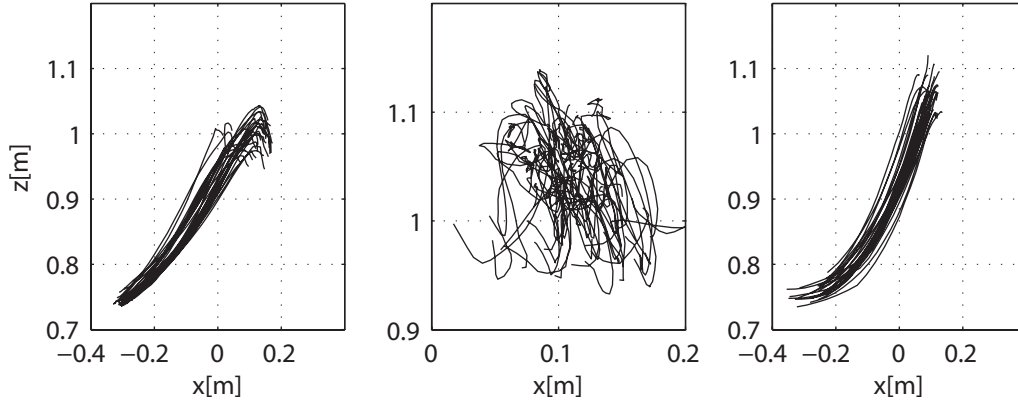
tive controller developed in next chapter, the controllers in this chapter are named basic controllers. The best performing basic controller is used as the basis to develop the interactive controller, while the simplest position controller is used as the comparison candidate in evaluating the performance of the system.

### 2.4.1 The first modeling iteration

With a priori knowledge from the pilot study, the next step to start the first iteration is to measure human performance. Considering the results from the pilot experiment, a larger number of handshake performances between two human participants are recorded. Grasping force and arm motion dynamics are measured. Optical markers are placed on the shoulder, elbow, as well as the back of the hand of both participants during each experiment. The haptic gloves used are TSG-2, with improved performance in measuring interaction forces during handshaking, technical details as in [100].

A total number of 900 handshakes are recorded from 24 male college students (mean age 28, standard deviation 2.292). As the aim of modeling is to teach a robot to shake hands which feels realistic to an arbitrary human counterpart, the participants are not given any

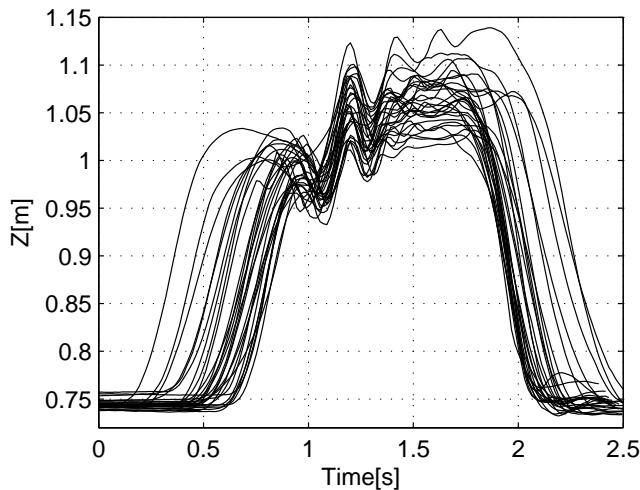
instruction on how they should conduct the handshake to achieve natural behaviors.



**Figure 2.4:** Segmented hand trajectory in 30 recorded handshakes in longitude( $x$ )-vertical( $z$ ) plane. Left to right: Approaching, shaking, retreating stages.

The handshake process is segmented according to the control methods in the robot. The optical tracking system used in the experiment provides 3-dimensional measurements of the trajectories of handshake dynamics. In Fig. 2.4, segmented trajectories are shown in longitude-vertical plane. The lateral motion is small comparing with the other two axes and is therefore neglected.

Inter-trial similarities can be observed for the trajectories of reaching and retreating stages, while the shaking stage curves differ significantly. However, if the vertical motion is plotted against time as shown in Fig. 2.5, similar pattern can be observed for different trial of handshakes.



**Figure 2.5:** Trajectories of 30 handshakes (3 stages). Vertical motion against time.

The human-robot system when in contact is modeled as two dynamical systems coupled together by force and position exchanges. Without contact, the robot has no interaction with the human and thus conducts free space motion.

A three state switching model is therefore constructed corresponding to the approaching, shaking, and retreating stages. The similar patterns of approaching and retreating stages support the hypothesis of modeling the two as free space motion following desired trajectories. For the shaking stage, the hypothesis is that it can be represented by the response of a second order linear time-invariant (LTI) system to certain inputs. As for the handshake, the arm can be regarded as being driven by the combined force from both partners.

In [103] we presented a first prototype that imitates the human arm behaviour during handshaking. A human-robot pair was modeled by two interconnected dynamical systems, namely two second order impedance models as shown in Equation (2.1), where  $M$ ,  $B$ , and  $K$  are the impedance parameters mass, damping, and stiffness, respectively,  $f$  and  $x$  denote force and position, and  $x_0$  is the equilibrium position of the human/virtual partner arms:

$$f(t) = M\ddot{x}(t) + B\dot{x}(t) + K(x(t) - x_0). \quad (2.1)$$

The LTI model is well studied in control theory, therefore standard methods can be applied for stability and performance analysis. Parameter estimation is also convenient with the LTI model. The 900 handshakes recorded are analyzed using standard least squares method to estimate parameters for the impedance model.

However, as the measured force is the combined force of both partners, without knowing the desired trajectory of each partner, the estimated parameters are for the coupled system rather than for each partner. Therefore the results cannot be implemented onto the robot. The iteration thus ends at this point, with correction information on the model selection.

## 2.4.2 The second modeling iteration

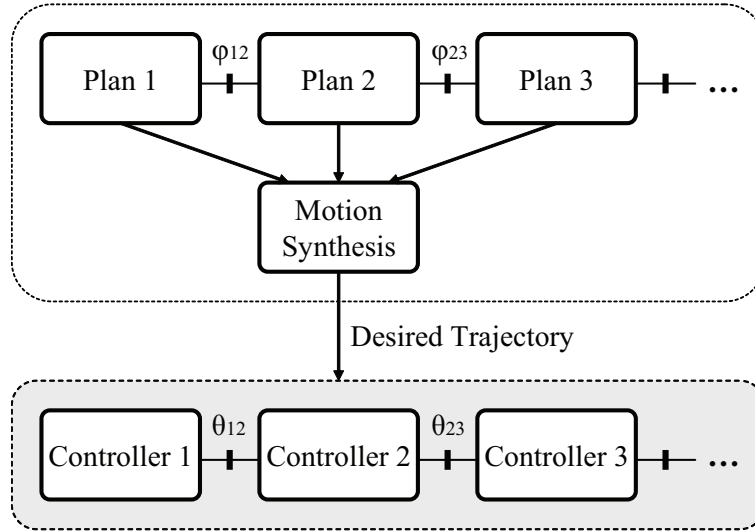
The second iteration starts with updating the a priori knowledge with the information from the first iteration. As there is no available method to measure the decoupled interaction force at the contact point of each partner, the simple impedance model with estimated parameters is not capable for the modeling and is changed.

Here an assumption is made on the human behavior that the human conducts a passive handshake, following the lead of the robot.

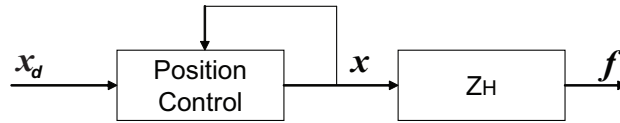
For controller design, as shown in Fig. 2.6, a switching model with motion synthesis is used to generate a planned trajectory, which serves as the reference signal to a controller to be implemented onto the robot. The method proposed in [5] is employed to ensure smooth position, velocity, and acceleration.

### Position control (PC)

The robot is controlled with standard position control. The desired trajectory is synthesized from the planners. The force feedback from the human partner is ignored in this controller. The scheme of position control (PC) algorithm is shown in Fig. 2.7. Assuming the robot has adequate actuation power, the actual position trajectory will be the same as the reference, regardless of the action of the human partner. This controller represents a non-compromising partner, or in other words, a partner with minimal compliance.



**Figure 2.6:** The model structure for handshake dynamics. The motion initiatives generated from the Plan blocks are synthesized in the Motion Synthesis block into a coherent trajectory and sent to appropriate controllers to actuate the robot.  $\theta$  and  $\phi$  stand for the transition conditions.



**Figure 2.7:** Position control scheme, with the human impedance model  $Z_H$ , the desired trajectory  $x_d$ , and the actual position  $x$ . Force interaction is not considered in this algorithm.

The PC algorithm is implemented onto ViSHaRD10 robot arm. The robot can perform handshakes following the generated desired trajectories. However, since force signals are not considered in the position controller, the robot arm cannot produce compliance and a completely stiff handshake is not realistic. The second iteration is terminated with corrections on the controller design.

### 2.4.3 The third modeling iteration

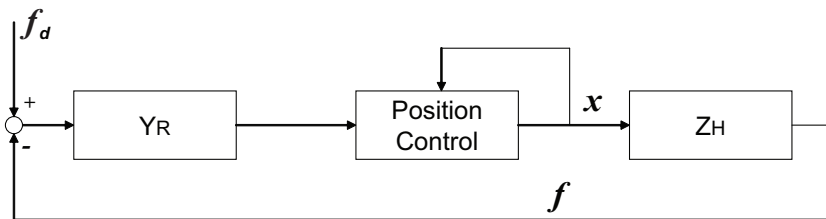
Enabling compliance to PC, the first step is considering the interaction force. Position-based admittance control (PAC) is essentially position control with an admittance filter to alter the desired position according to the measured force of the interaction. In the extreme case this controller will turn into PC, when the impedance parameters in the admittance filter are set to infinity. For finite impedance parameters, the admittance filter can be any constant value set within the dynamic constraints for the robot to maintain stability.

However, in reality the constant impedance parameters feel either too soft during shaking, when set to low values, or too stiff after shaking, when set to high values. This indicates

that constant parameters may not be the right approach for a handshake leader model. Therefore the following two variations are proposed to achieve better realistic feelings.

### Position-based admittance control: variant 1 (PAC1)

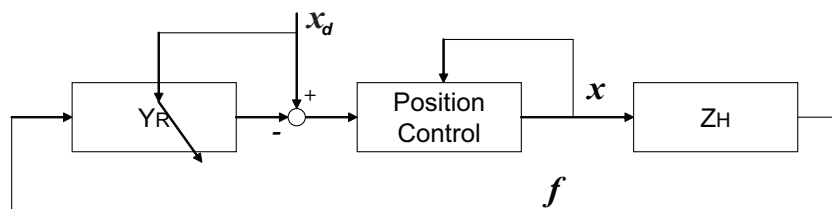
Different from PC and PAC, PAC1 uses force as the reference signal, as shown in Fig. 2.8, although sharing the admittance control structure. In PAC1, human interaction force is added directly to the force reference, instead of passing through the admittance filter in PAC. As discussed in [103], PAC1 achieves good results when the human follows the robot well. Obtaining the reference here, however, is difficult since force reference cannot be measured directly but can only be calculated. Detailed discussion on obtaining reference trajectories will be given by the end of this section. Note that due to the calculation of the force reference, the impedance parameters in the admittance filter are constant.



**Figure 2.8:** PAC1 control scheme, force is used as the reference trajectory instead of position, with the human impedance model  $Z_H$ , the robot impedance model  $Y_R$ , the desired force  $f_d$  and the actual interaction force  $f$ .

### Position-based admittance control: variant 2 (PAC2)

In reality, the contraction of the muscle groups is increased when the human is exerting higher forces. The stiffness of the arm is therefore increased. Inspired by this fact, an admittance filter with time-varying parameters is applied to improve the performance of the PAC controller. The stiffness of the admittance filter varies with the reference trajectory, as shown in Fig. 2.9.



**Figure 2.9:** PAC2 control scheme with time-varying impedance parameters in the admittance filter, with the human impedance model  $Z_H$ , the robot impedance model  $Y_R$ , the desired position  $x_d$ , the actual position  $x$ , and the actual interaction force  $f$ .

In reality, the contraction of the muscle groups is increased when the human is exerting

higher forces. The stiffness of the arm is therefore increased. Inspired by this fact, the time-varying virtual stiffness  $k$  consists of a constant term  $K_0$  in addition to a term proportional to the difference between the current desired position  $x_d(t)$  and the equilibrium position  $x_0$ :

$$k = K_0 + \eta(x_d(t) - x_0). \quad (2.2)$$

The intuitive explanation to this selection is that the more a participant wants to drive the partner away from the equilibrium, the stiffer the participant's arm should be in order to succeed. In other words, the robot with such a controller is compliant near the equilibrium while being stiffer when leaving it.

If the participant performs similarly in each handshake, the same desired trajectory  $x_d$  leads to similar position  $x$  and force  $f$  profiles for each handshake and hence provide the participant a similar handshaking experience compared to the real handshake where the reference trajectory  $x_d$  was recorded. If the participant fails to perform similarly, however, the admittance filter with the adaptation law will increase stiffness and try to maintain the desired trajectory, while still giving the human the feeling of a realistic handshake, although with a more dominant and less compliant partner.

Provided that participants are good followers, PAC2 can perform competent handshakes in human-robot handshaking experiments. However this approach has a fundamental limitation for the realization of full interactive handshakes since it lacks the ability to alter the reference trajectory, and therefore the robot can only playback the trajectory as pre-defined, with the human input applied to this. This is clearly different from human-human handshaking, where the arms can provide compliance during interaction, while the participant can select different strategies with respect to adaptation to their partner's form of handshake. Therefore a more advanced form of interactive handshaking controller has been developed, and is presented in the next section.

## 2.5 Conclusions

In this chapter physicality in HRI is discussed, as well as the challenges it brings to haptic modeling. Then the SOA of haptic modeling is given. Although existing works are available in modeling haptic manipulation tasks, very few works are reported in modeling interaction tasks where a second human is considered in the system.

To address the haptic interaction modeling problem, a framework for haptic modeling is proposed, attempting to provide a general guideline for the haptic interaction modeling workflow. The framework emphasizes on the inclusion of human-human experiment of the interaction task as the starting point of modeling, to collect behavioral data about the interaction. Then modeling iterations can be carried out based on the observations of the human-human experiment.

Following the proposed framework, three modeling iterations are carried out in modeling human handshake. As the starting point, a pilot study of human-human handshaking was carried out to obtain a first understanding about the problem. Based on the findings of the pilot study, a large amount of human-human handshake data was collected in the

human experiment redesigned from the pilot. Three controllers are developed to realize handshaking, each from a modeling iteration. An evaluation experiment was carried out in Section 7.2 with human participants in validating the controllers in terms of how plausible the robotic handshake is to the participant, with each controller implemented. The position controller with reference trajectory is the most simple and robust one, although not considering human interactions. It is chosen as the comparative algorithm to the interactive controller developed at a later stage of the work, in order to assess the effect of enabling interaction in the handshake process. On the other hand, the position-based admittance controller with time varying parameters is chosen as the basis for developing the interactive controller later on, for the reason that it is considered by the participant to be as plausible as the position controller, while it has the highest possibility of extensions.

## 3 Haptic rendering of arm dynamics 2: towards an interactive controller

Based on the findings of Chapter 2, the main contribution of this dissertation is presented in this chapter: the interactive controller that estimates and adapts to human intentions. To this end, a human behavior model is proposed first, and then the development of the interactive controller is discussed in detail, together with the necessary techniques for estimating human intention as well as adaptation. The performances of each module and the overall controller are validated by experimental results given in the end.

### 3.1 Human behavior model

To enable adaptation to human behavior, some knowledge of human intention is required. Unfortunately human intentions are not directly measurable; hence estimation of such information by indirect means is required.

In order to obtain a valid human intention estimation, the selection of input data is crucial. Recall from Fig. 2.9, the measurable force and position data result from the interaction of the human and the robot, hence any phenomenon observed in either signal can be the consequence of either the robot or the human. For this reason, neither of the two signals can be used to estimate human intentions. Instead, parameters obtained from inside the *Human* block in Fig. 2.9 are more suitable as they contain only information about the human.

The problem then becomes one of determining which parameters of the human behavior can be extracted from the measured data of force and position. In the basic controller, the human is assumed to be passively following the robot. Only the impedance parameters of mass, damping, and stiffness are considered, without further excitation signals to the system besides the reference trajectory of the robot. However, in the case of the robot adapting to the human, the human can no longer be expected to act passively: the desired trajectory of the human is now an additional input to the human/robot system.

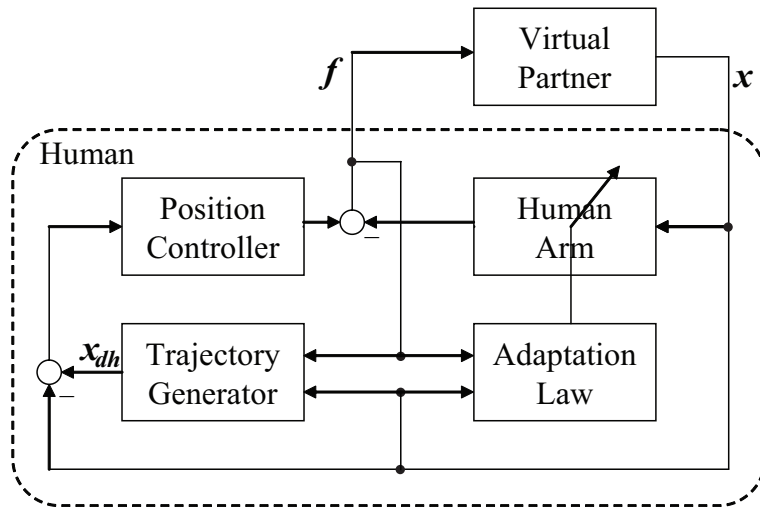
Considering the desired trajectory of the human, the following human behavior model is proposed:

1. The human is modeled by a position controlled arm with a trajectory planner and an adaptation module to adjust the impedance of the arm. The planner considers the actual interaction status and generates the desired signal for the position controller, while the adaptation law alters the arm impedance parameters at the same time.
2. The human behaves as a collaborative partner. In other words, the human planner generates desired signals based on the actual interaction status. It does not matter



whether the decision of the human is to follow or to change the current human/robot coupled position trajectory, the decision is made based on the actual measurements.

An illustration of the human model is given in Fig. 3.1. This model allows the human to change the arm impedance as well as the desired trajectory with respect to time. The assumption that the human will behave reasonably is a natural approach, since it is expected that a human plans the behavior of the next step based on the information gathered about the current step. Further information needed to generate the trajectory is integrated inside the planner block. On the other hand, this assumption makes the current position signal the only input to the human model, hence the input and output signals of the model are both known to the robot.



**Figure 3.1:** Human behavior model.  $x_{dh}$  is the desired trajectory generated by the human,  $x$  is the actual human/robot position, and  $f$  is the actual interaction force. The parameters of human arm impedance are variable in time.

The model in Fig. 3.1 is expected to be time-varying and non-linear. However, in practice the robot needs an easy-identifiable model for the interactive controller to estimate human intentions at each time instance. Hence the following approach is taken to linearize the human behavior model:

1. A linear differential equation is used to represent the relationship between position input and force output signals.
2. In this work the differential equations are limited to the second order, as shown in (3.1), where  $f$  and  $x$  are the position input and force output signals,  $h_2$ ,  $h_1$ , and  $h_0$  are the three parameters of the differential equation, each denoted as a *human behavior parameter* (HBP), representing the current human behavior that determines the force output based on the current position input. With the human planner and adaptation law in the module, the HBPs are time-varying.

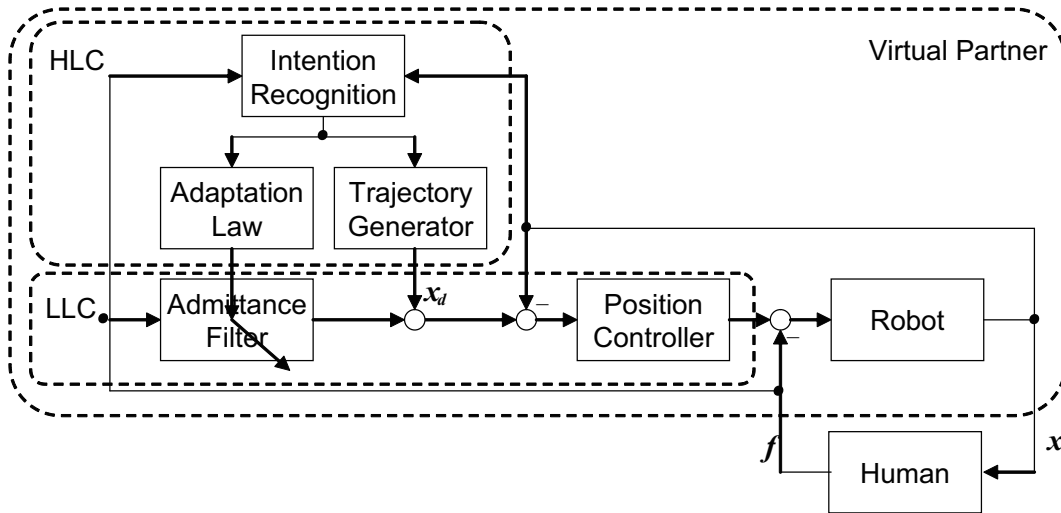
Comparing Equation (3.1) with (2.1), the HBP set  $(h_2, h_1, h_0)$  is similar to the impedance parameter set  $(M, B, K)$ , in the sense that they are both relationships between force and position signals. However, HBPs also take into account the influences of adaptation and the human planner, therefore they are not necessarily equivalent to the impedance parameters. In the case of the human being ideally passive (the human only acts as a position controller), HBPs will be equivalent to the impedance parameters.

$$f = h_2(t)\ddot{x} + h_1(t)\dot{x} + h_0(t)x. \quad (3.1)$$

In the remainder of this chapter, *impedance parameters* denote the parameters in the admittance filter of the robot controller, while *HBPs* denote the estimated behavior parameters of the human.

## 3.2 Interactive controller

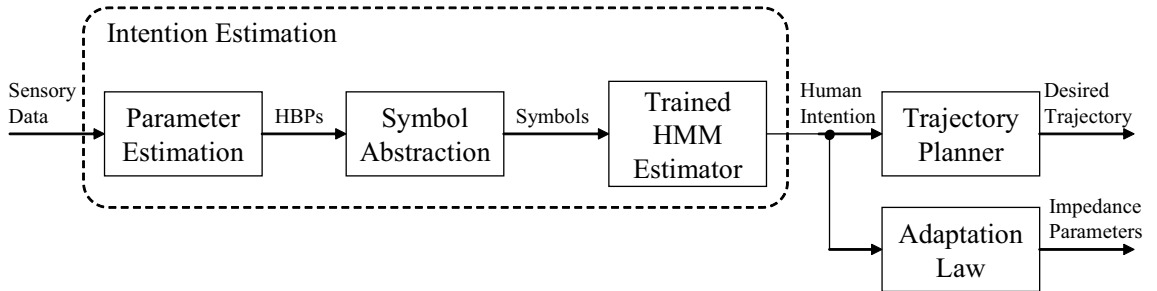
Based on the human behavior model of Equation (3.1), a new double-layered control scheme is proposed that not only alters the admittance parameters, but also the reference trajectory. A low-level controller (LLC) implements a position-based admittance controller and a high-level controller (HLC) updates the admittance parameters and adapts the reference trajectory, see Fig. 3.2. A preliminary version of the overall control has been presented in [102].



**Figure 3.2:** Overall controller scheme. The high-level controller provides reference trajectory for the basic controller while adjusts the impedance parameters in the admittance filter.

As the human intention is not directly measurable, a special estimator is needed. In the HLC, sensory data is fed into an intention estimator, the resulting estimations are then used by the planner to provide reference trajectories and control parameters of the already introduced low-level controller.

The problem for the HLC to solve is as follows: given predefined human intention states, use online sensory data to estimate current human intentions with respect to the handshake, generate reference trajectories for the low-level control algorithm using suitable policies and determine the virtual admittance parameters for the low-level controller. A detailed scheme of the HLC is shown in Fig. 3.3.



**Figure 3.3:** High-level controller scheme. Sensory data contains both force and position measurements.

The following approach is taken to solve the problem: taking the force and position data measured online, a parameter estimator identifies the current HBPs of the human who is handshaking with the robot. The parameters are abstracted into symbols and an HMM-based intention estimator uses them to give the current human intention estimation. The trajectory planner then generates the reference trajectory according to the different policies corresponding to the current human intention and the impedance parameters are adjusted according to the adaptation law.

The remainder of this section shows the details about the parameter estimator, the HMM estimator, the parameter adaptation, and the trajectory planner.

### 3.2.1 Fast online parameter estimation

The system frequency of the handshaking robot is 1kHz and therefore in order to achieve online estimation of the HBPs, a fast and robust estimation method is needed. The goal of the estimation is to provide information for the HMM module that estimates human intention. Offline estimation results show that all three HBPs change with respect to time. Therefore the estimation of all HBPs is required for the estimator.

Although HBP is newly proposed here, its estimation can be carried out in a manner similar to the impedance parameter estimation process, since the two are mathematically equivalent. Among the three impedance parameters of mass, damping, and stiffness, previous results are concerned mostly with the stiffness parameter estimation, see [6], [94], and [104], since this is essential to force tracking control, while only very few results are concerned with the damping coefficient estimation, see [26]. A commonly adopted class of estimation algorithms is based on least squares (LS), for instance, in [23] the recursive least squares (RLS) method with variable forgetting factors is used for online estimation, with the forgetting factor a function of the output error.

To achieve sub-second converging speed, as reported in [6] and [94], noise and speed

need to be compromised. For online estimation, as in [23], convergence and dynamic performance need to be compromised. Next, different methods for parameter identification are discussed and compared. Fundamental knowledge on system identification is not covered. Interested readers may take [53] as a reference.

### LS with moving window (LSMW)

When applying standard LS method to give online estimations, instead of using all available data from the past, a moving window method can significantly reduce calculation amount by only considering the most recent number of data points according to the specified window length. The procedure of the moving window LS is as follows:

For a given data set  $(\mathbf{X}, \mathbf{Y})$  of length  $L \in \mathbf{N}$ , such that

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{w}, \quad (3.2)$$

$\mathbf{X}^T = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L)$ ,  $\mathbf{Y}^T = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L)$ ,  $\boldsymbol{\theta} \in \mathbf{R}^{n \times m}$ ,  $\mathbf{x}_j \in \mathbf{R}^{n \times 1}$ ,  $\mathbf{y}_j \in \mathbf{R}^{m \times 1}$ ,  $j = 1, 2, \dots, L$ , and  $\mathbf{w}$  is the measurement noise.

Take a window of size  $l \in \mathbf{N}$  such that  $l < L$ , a total number of  $L - l + 1$  estimations of  $\boldsymbol{\theta}$  can be given using

$$\hat{\boldsymbol{\theta}}_i = \mathbf{X}_i^\# \mathbf{Y}_i, \quad \mathbf{X}_i^\# = (\mathbf{X}_i^T \mathbf{X}_i)^{-1} \mathbf{X}_i^T, \quad (3.3)$$

$\mathbf{Y}_i^T = (\mathbf{y}_i, \mathbf{y}_{i+l}, \dots, \mathbf{y}_{i+l-1})$ ,  $\mathbf{X}_i^T = (\mathbf{x}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_{i+l-1})$ ,  $i = 1, 2, \dots, L - l + 1$ , with the estimation error

$$\hat{\mathbf{e}}_i = \mathbf{Y}_i - \mathbf{X}_i \hat{\boldsymbol{\theta}}_i. \quad (3.4)$$

Note that the number of estimations is smaller than the data length.  $L$  estimations can be obtained by assigning zeros to the  $l - 1$  data points before the starting time.

The convergence or response speed in the ideal case is determined by the window length, as the method does not consider data points outside the window at all. For noise free data, (3.3) will converge to real  $\boldsymbol{\theta}$  in one step.

However, when the measured data is corrupted with noise, LSMW suffers significantly as it has no remedy to noise. For white noise signal, it can be proved that the estimation converges to the true value given a long enough data set [53]. As the measurement noise for robots are mostly sensor noise from motor encoders and force/torque sensors on the end-effectors, they can be approximated by white noise.

### Recursive LS

RLS is a widely used method for online application because of its computational efficiency. In order to identify time varying parameters, a forgetting factor is introduced into the RLS iterative algorithm such that newer data is more valued. A brief structure of the algorithm is given below.

For the data set of system in (3.2), given the data set and the corresponding parameter estimation at time  $t$

$$\hat{\boldsymbol{\theta}}_t = (\mathbf{X}_t^T \mathbf{X}_t)^{-1} \mathbf{X}_t^T \mathbf{Y}_t, \quad (3.5)$$

$\mathbf{Y}_t^T = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t)$ ,  $\mathbf{X}_t^T = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t)$ ,  
the estimation of time  $t + 1$  is calculated as

$$\begin{cases} \hat{\boldsymbol{\theta}}_{t+1} = \hat{\boldsymbol{\theta}}_t + \mathbf{Q}_{t+1}(\mathbf{y}_{t+1} - \mathbf{x}_{t+1}^T \hat{\boldsymbol{\theta}}_t), \\ \mathbf{P}_{k+1} = \frac{\mathbf{P}_t}{\lambda + \mathbf{x}_{t+1}^T \mathbf{P}_t \mathbf{x}_{t+1}}, \\ \mathbf{Q}_{t+1} = \mathbf{P}_{t+1} \mathbf{x}_{t+1}, \end{cases} \quad (3.6)$$

where  $\lambda$  is the forgetting factor.

The assignment of the forgetting factor needs to be carefully crafted, as improper values would lead to potential blow-up of the covariance matrix  $\mathbf{P}$ . Forgetting factor selection is also the main difference among different RLS methods. Time-varying forgetting factor values are often used to provide fast tracking performance during transients while achieving robustness to noise and disturbances in the slow changing periods of the tracked parameters.

In the case of time varying forgetting factors, setting  $\lambda$  to be a function of the estimation error  $\hat{\mathbf{e}}$  is often considered to be an effective and reasonable way of relating tracking speed, affected by  $\lambda$ , with the parameter varying speed, while tending to bound the forgetting factor value within a safe range. In [23] a RLS method is proposed specifically for impedance estimations in robot control. While following the common RLS structure, the forgetting factor is set to be a function of the estimation error

$$\lambda_t = 1 - \beta_1 \left( \frac{1}{\pi} \arctan(\beta_2(|\hat{\mathbf{e}}_t| - \beta_3)) + \frac{1}{2} \right) \quad (3.7)$$

where  $\hat{\mathbf{e}}_t$  is the estimation error at time  $t$  and  $\beta_1$ ,  $\beta_2$ , and  $\beta_3$  are tunable parameters.

In (3.7), due to the saturation characteristics of the arctan function,  $\lambda$  converges to  $1 - \beta_1$  in the case of large  $\hat{\mathbf{e}}$ , with  $\beta_3$  being the threshold of small and large error and  $\beta_2$  the control factor of the converging speed.

In [23], discussions concern more on proving convergence for the algorithm rather than the converging speed and tracking performance. Convergence can be achieved, but in some cases taking slower than a second. This fails to meet the requirement for fast estimation in robot control. Next a new method of assigning the forgetting factor is proposed, aiming for a faster speed of converging and tracking.

### Tracking parameter change

The RLS scheme of (3.6) does not involve any matrix inversions once initialized. The size of the matrices is fixed for all time steps. The computational efficiency of this method makes it suitable for online application. Moreover, the selection of the forgetting factor  $\lambda$  can lead to different properties of the algorithm. Small  $\lambda$  results in fast parameter change tracking but sensitivity to noise, while large  $\lambda$  means robustness to noise but slow convergence.

Previous methods use estimation error of the output signal as the main source of information to which the forgetting factor is calculated. However, in the concerned question, the output error is not the most fundamental factor.

Consider the force tracking scenario as an example. The goal is to maintain constant

force at different environment conditions, i.e. with different impedance parameters. Online estimation algorithms are intended to identify the *impedance parameter* at each time in order for the control algorithm to maintain force tracking. Therefore, it is the *impedance parameter estimation error* that should be minimized, with the *force tracking error* an overall objective for the force tracking controller. In more specific words, according to (3.5), the objective of RLS is to track the changes in  $\theta$ , not  $Y$ .

Therefore, a change in  $\theta$  should, in the ideal case, trigger a reduction in  $\lambda$  so that the tracking performance of the algorithm gets enhanced. However, according to (3.7),  $\lambda$  is only altered with an error accumulated in the output. In other words,  $\lambda$  is changed not *directly* by the change in  $\theta$ , but by the *indirect* change in  $e$  caused by the change in  $\theta$ .

Although it is an intuitive idea to determine the forgetting factor by the parameter change, there is no analytical measure of the parameter estimation error. The output error can be readily calculated from the measured output and the calculated output from the estimated parameters, while there is no accurate information on the true value of the parameters to be tracked at each time. Therefore calculation of the error of estimated parameters is not feasible.

On the other hand, although accurate information of parameter values is not available, there is the past information of parameter values estimated previously available for use.

Based on this non-accurate information, a new method of determining the forgetting factor is proposed next.

### RLS with redefined time-varying forgetting factors

Here a new method to calculate the forgetting factor is proposed. This method is based on the current and previous estimated parameters. By tolerating a potential error in estimation, the previous estimated parameter set is used as a reference set. The parameter set estimated at the current step is compared with the reference. If the difference is large, there is a potential parameter change; the forgetting factor is then degraded for faster tracking.

In addition, to relate the forgetting factor to the parameter change, the proposed method also introduces time varying functions to calculate the forgetting factor, see 3.8. Previous works employ fixed functions to calculate the forgetting factor at each time step. Although the forgetting factor is time-varying, the function of calculation is fixed, therefore performances in transient and settling periods have to be compromised. To achieve both tracking and robustness properties, two different functions are employed in the proposed method to calculate time-varying forgetting factors in the cases of high and low parameter changing rates.

The proposed algorithm is given bellow:

$$\lambda_t = \begin{cases} 1 - \frac{\alpha_3}{\pi_1} \arctan(|R_t - 1|), & \text{if } |R_t - 1| \geq \alpha_2; \\ \alpha_1 + \frac{1}{\pi}(1 - \alpha_1)(\arctan(1 - |R_t - 1|)), & \text{else;} \end{cases} \quad (3.8)$$

$$R_t = \begin{cases} \max \left( \left| \frac{\boldsymbol{\theta}_{t-k}^{ij}}{\boldsymbol{\theta}_t^{ij}} \right|, \left| \frac{\boldsymbol{\theta}_t^{ij}}{\boldsymbol{\theta}_{t-k}^{ij}} \right| \right), & \text{if } \boldsymbol{\theta}_{t-k}^{ij} \boldsymbol{\theta}_t^{ij} \neq 0; \\ \infty, & \text{else.} \end{cases}$$

$\forall i \in (1, 2, \dots, n), \forall j \in (1, 2, \dots, m)$ , with  $k, \alpha_1, \alpha_2, \alpha_3$  tunable parameters,  $\frac{1}{3} \leq \alpha_1 \leq 1$ ,  $\alpha_2 \geq 0, 0 \leq \alpha_3 \leq 2, k \in \mathbf{N}$ .

Here  $k$  is the number of steps by which the reference parameter estimation is selected,  $\alpha_1$  bounds the varying forgetting factor from below in the case of low changing speed,  $\alpha_2$  is the threshold of high and low changing speeds,  $\alpha_3$  sets the upper bound in the high changing rate case, and  $R_t$  is defined such that the highest change rate in all parameter entries is selected, which is always greater or equal than 1. If any entry turns to be zero,  $R_t$  is assigned to plus infinity.

### Parameter tuning

The forgetting factor  $\lambda_t$  has two possible varying ranges: at low parameter changing rates, the forgetting factor is set to vary very closely to 1, so that the algorithm is robust to noise and perturbations; while at high parameter changing rates, the forgetting factor is set to vary within a range with a lower mean value.

For both high and low parameter changing rate,  $\lambda_t$  is bounded by

$$\begin{aligned} \sigma_{min1} &\leq \lambda_t < \sigma_{max1}, & \text{if } |R_t - 1| \geq \alpha_2, \\ \sigma_{min2} &< \lambda_t \leq \sigma_{max2}, & \text{else,} \end{aligned} \quad (3.9)$$

$$\begin{aligned} \text{where } \quad \sigma_{min1} &= 1 - \frac{\alpha_3}{2}, \quad \sigma_{max1} = 1 - \frac{\alpha_3}{\pi} \arctan(\alpha_2), \\ \sigma_{min2} &= \alpha_1 + \frac{1 - \alpha_1}{\pi} \arctan(1 - \alpha_2), \quad \sigma_{max2} = \frac{1 + 3\alpha_1}{4}. \end{aligned}$$

Parameter selection is hand crafted for the proposed method, as for any similar methods of variable forgetting factors. The following guideline can be given on the selection of parameters, assuming that rough information of the tracked parameters is previously available.

- a)  $\alpha_1$  determines the range of variation in case of slow parameter change, it should be close to 1.
- b)  $\alpha_2$  controls the switching between the two functions, it should be selected according to the application. The faster the possible changing rate of the tracked parameters, the smaller  $\alpha_2$  should be, so that the algorithm is more sensitive to parameter changes, while on the other hand it is less robust to perturbations when the parameter changes at a lower rate.
- c)  $\alpha_3$  determines the range of variation as well as the changing rate of the forgetting factor with respect to the estimation error. In order to keep  $\lambda$  within the range of  $[0, 1]$ ,  $\alpha_3$  is limited to  $[0, 2]$ . Larger  $\alpha_3$  results in larger change in  $\lambda$  for the same estimation error. Careful selection should be made to maintain good performance of the algorithm.
- d)  $k$  is the number of steps by which a previous estimated parameter is chosen as the

reference. Small  $k$  yields faster response, but the smaller time interval of taking two samples will result in smaller difference in parameter value so the algorithm may always stay around a certain region. In robotic control, at a typical sampling rate of 1kHz, a value of 10 gives 0.01s time interval in between.

e)  $P_t$  is always larger than 1. The definition gives equal distributions for increasing and decreasing cases. The above ranges of parameters make sure that  $0 < \lambda \leq 1$ .

The new algorithm differs from the algorithm in [23] mainly in two aspects: defining the forgetting factor to be a function of the parameter changing rate; and using different functions in calculating the forgetting factor during parameter tracking and noise rejection. Validation results are given in Section 3.3.

### 3.2.2 Symbol abstraction and HMM intention estimation

In HRI, the robot cannot measure human mental states, but only the resultant actions. As a matter of fact, not only the human mental states, but also the mechanism of how these mental states lead to the resultant actions are *hidden*. In this approach, the human intention states are modeled by different hidden Markov models (HMM). An HMM has an underlying Markovian stochastic process that is not observable directly, but only through another stochastic process with respect to a certain probability of observation. This copes with the relationship between the hidden human intention states and the actions, such that a human intention is modeled by an HMM, while within each HMM, the observations are the measured information, and the hidden states are the hidden mechanism of how the human generates the measured action in the specific mental state. Therefore, for the estimation of  $n$  human mental states, there are  $n$  different HMMs, each having  $\sigma_n$  hidden states, the values of  $\sigma_n$  are determined by the training of the specific HMM. The estimation algorithm then decides which HMM is currently the best fit for the given sensory information.

This type of HMM estimators have been used for speech, handwriting, and other types of pattern recognition since the 80's, see [69]. Applications have been reported in human motion recognition by using image sequences, as in [107] and haptic signals, as in [93]. The application in [93] is to estimate human dancing steps using force signals measured by the force sensor on the robot during dancing. In [8], HMM is used to classify human behaviors based on haptic measurements when drawing on a 2D plain. In [16] HMMs with continuous inputs of force and velocity measurements are employed in recognizing human intention states in a joint object carrying task.

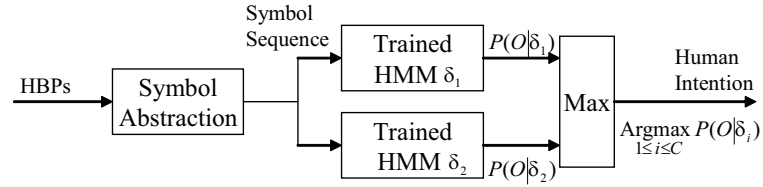
The method employed here is extended from the method described in [109]. The input data is changed from gesture paths as in [109] to haptic data, in this case the HBPs. The number of hidden states in each HMM is set to four. The method can be formulated as follows:

Given  $S = \{S_n\}, n = 1, 2, \dots, N$ , state  $S_n$  being the  $n$ th hidden state, and  $O = O_1, O_2, \dots, O_N$  the observed symbol sequence, choose the best matching HMM from  $\delta_i, i = 1, 2, \dots, C$ , i.e. calculate  $P(O|\delta_i)$  for each HMM  $\delta_i$  and select  $\delta_{c^*}$ , where

$$c^* = \operatorname{argmax}(P(O|\delta_i)). \quad (3.10)$$



Given the observation sequence  $O = O_1, O_2, \dots, O_N$  and the HMM  $\delta_i$ , the problem is how to evaluate HMM  $P(O|\delta_i)$ , the probability that the observation sequence was generated by HMM  $\delta_i$ . This probability can be calculated by using the *Forward-Backward algorithm*, as shown in [69]. The HMM with the highest likelihood is selected as the recognition result. The scheme of the HMM estimator is shown in Fig. 3.4. Here two HMMs are defined for the estimator: *Active* and *Passive*. Active indicates that the human is trying to lead the handshake, while passive means the human is following the lead of the robot.



**Figure 3.4:** Scheme of HMM-based human intention estimator. HMM  $\delta_1$  for *Active*, HMM  $\delta_2$  for *Passive*.

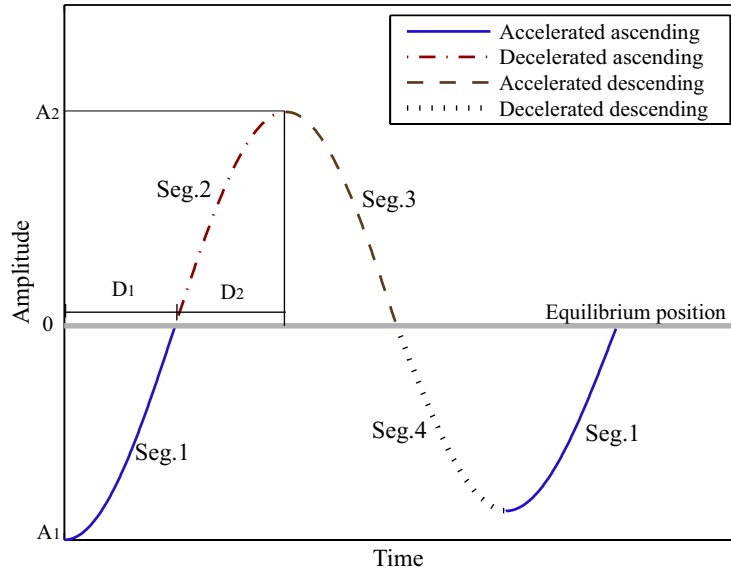
Here the HBPs are preferred to the interaction force signal as the input to the HMM estimator, for the reason that the interaction force contains the influences of both the robot and the human. For instance, a high measured interaction force can result from a strong robot motion acting against the human arm inertia in case of a passive human, or from an active human who is trying to change the current shaking status by either stiffening his arm or imposing a certain trajectory. In other words, high force measurements do not necessarily mean that it was the human who was overactive. Therefore it is not sufficient to estimate human intentions from the force measurement alone. The HBPs, however, consider the interaction trajectory and force signals at the same time. In case of high interaction forces resulting from a strong robot shaking the hand of a passive human, both position and force values would be high, resulting in moderate HBP estimations for  $h_0$  and  $h_1$ ; if the force results from an active human, the arm impedance or the imposed force would be high, which will be reflected in the HBP estimations as well. Hence HBPs are the better choice in representing human intentions.

For a given HBP sequence, 8 symbols are used in the abstraction as shown in Table. 3.1. The thresholds for *low* and *high* are set by heuristics: 1 for the  $h_2$  coefficient; 50 for  $h_1$  and 500 for  $h_0$ .

Performance validation tests have been carried out for the HMM estimator. Detailed discussions are given in Section 3.3.

### 3.2.3 Trajectory planning and parameter adaptation

The trajectory planner and the parameter adaptation module are the last modules in the HLC. The following information is available to generate the reference trajectory: force and position (also calculated velocity and acceleration) measurements, identified HBPs, and the estimated human intention state. In addition, history information of the above data can be stored if necessary. The task of the planner is to generate a trajectory on the basis of certain criteria which are to be defined in this section.



**Figure 3.5:** Segmentation of the shaking trajectory.

**Table 3.1:** Generated HMM observable symbols.

$h_2$	$h_1$	$h_0$	Symbol
low	low	low	1
low	low	high	2
low	high	low	3
low	high	high	4
high	low	low	5
high	low	high	6
high	high	low	7
high	high	high	8

There are many possible solutions for the selection of criteria, since handshaking contains a large amount of inter-cultural, inter-subjective, and even inter-trial variations. Therefore instead of defining one *standard* handshake, here the approach is taken to find boundaries for recreating a handshake trajectory that feels human-like to the user. In our case study of handshaking, see [103], 1800 recordings have been made on human-human handshaking. Averaged duration and amplitude of the position trajectories are shown in Table 3.2. The average of handshakes lasts about 1s, with the peak amplitude of about 0.1m.

Here a strategy of trajectory generation is proposed. Assuming that all information is available to the trajectory planner, the strategy of trajectory planning is formulated as

**Table 3.2:** Averaged duration and amplitude of human-human handshakes

Position	Min	Max	Mean
Duration/s	0.712	1.816	1.01
Amplitude/m	-	0.161	0.098

follows:

- **Segmentation:** The trajectory is divided into segments, see Fig. 3.5. Every subpart of the trajectory is calculated separately. An initial subpart is given at the beginning, the follow-ups are calculated according to the online interaction.
- **Passive:** When the human is in the *passive* state, the robot tries to take the lead. *Impedance parameters* in the *admittance filter* are set to *high* values, and no modification is made to the current reference trajectory: the robot continues as planned.
- **Active:** When the human is *active*, the robot tries to follow the lead. The *impedance parameters* are set to *low* values, and the reference *amplitude*  $A$  and *duration*  $D$  of the current segment  $m$  of the trajectory is set to:

$$A_{m+1} = A_m + 0.002\tilde{f}; \quad (3.11)$$

$$D_{m+1} = D_m + 0.01\tilde{f}; \quad (3.12)$$

$$\tilde{f} = \text{round}((f - f_{\text{threshold}})/5); \quad (3.13)$$

where  $A_{m+1}$  and  $D_{m+1}$  are the new amplitude and duration and  $A_m$  and  $D_m$  are current amplitude and duration;  $f_{\text{threshold}}$  is the force threshold to remove noise in zero measured force;  $\text{round}$  is the round up function which gives the smallest integer value no less than the input. The parameters are selected by heuristics, such that the resulting trajectory changes significantly when interacting with a human, while lying within the workspace capabilities of the robot.

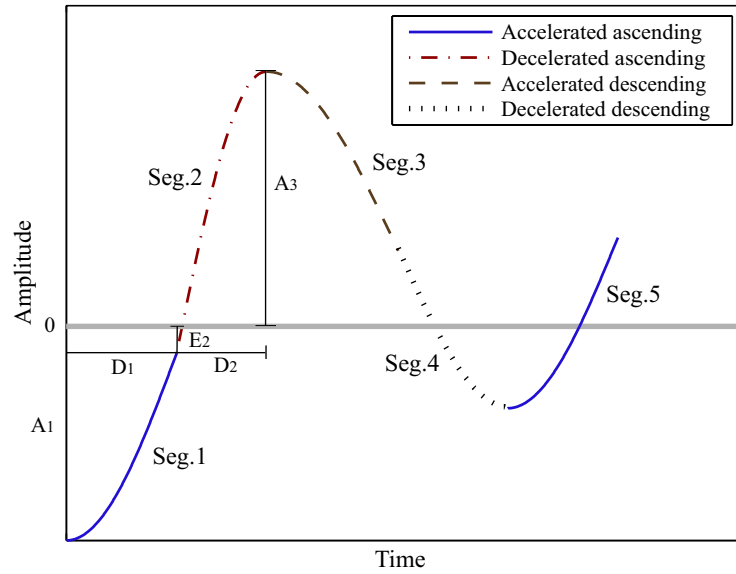
A sinusoidal curve is generated for each subpart. The curve is updated after each segment, as shown in Fig. 3.5 where 5 segments are illustrated. Currently the last 100 force measurements before the update time instance are averaged to be the force  $f$ , in correspondence to the length of input sequence for the HMM estimator.

### 3.2.4 Refined trajectory planning

In the subjective evaluation experiment employing the aforementioned trajectory planning algorithm, many participants reported that the interactive controller of the robot did not respond quickly enough to changes required by the human. The bottleneck of the interactive controller in terms of reactivity is the trajectory planner. Therefore, a refined trajectory planning strategy is proposed and implemented.

The main issue of the trajectory planner is dividing each handshake cycle into four segments and updating the planned trajectory only at the beginning of each segment. Therefore, no matter how much effort the human applied to change the current trajectory during the segment, the effect will only appear when the next segment starts. This delay of a quarter of a cycle left an irresponsive impression to the participants in the experiment.

To fix this latency issue, updating the trajectory at each time instance is desired. Similarly to the discussion in Section 3.2.3, the strategy of trajectory generation is not unique;



**Figure 3.6:** Refined segmentation of the shaking trajectory.

selected strategies clearly reflect different personalities of a robotic partner during handshaking. Here a general frame of online trajectory updating is proposed, with a maximum number of possible tunable parameters for future customization.

1. A handshake cycle is still divided into segments, each starts at peak/valley points (A) and ends at equilibrium points (E), or starts at E and ends at A, as shown in Fig. 3.6. However this segmentation is no longer used for trajectory updating, but to keep a sinusoidal form of the overall shaking trajectory.
2. As suggested by the human-human handshaking results in Table 3.2, a typical handshake lasts for 1s. Therefore each segment is initialized to be of time length  $D_1 = 250ms$ . A count  $t$  is introduced to note how much time passed since the beginning of the current segment,  $0 < t \leq D_n$ .
3. At each time instance  $t$ ,  $D_n$  is updated according to the interaction force and current position. The current segment can be immediately terminated once a significant force opposite to the current direction of motion is detected, which suggests the human wants to turn; with the current  $t$  set as  $D_{n+1}$ .
4. At each time instance, the destination amplitude  $A_{n+1}$  or  $E_{n+1}$  of the current segment is updated by (3.14) and (3.15) depending on the actual segment. In both cases  $dir$  is the current direction of motion,  $dir = 1$  for ascending,  $dir = -1$  for descending.  $f_t$  and  $x_t$  are the current measured force and position, respectively.  $\gamma_1$  and  $\gamma_2$  are tunable parameters determining the amount of adaptation to the interaction force. The sine function is introduced to take into consideration the feasible amount of change applied to the desired amplitude: the closer the current  $t$  is to the end of a segment, the less the allowed modification term will be, in order to avoid exceeding the speed limitations of the robot.

5. With the updated  $D_n$ ,  $A_{n+1}$ , and  $E_{n+1}$ , together with the measured current position, velocity, and acceleration, a 5th order polynomial trajectory can be calculated connecting the current and desired position smoothly (with continuous position and velocity).

$$E_{n+1} = -dir \cdot \gamma_1(\gamma_2 + f_t) \sin\left(\frac{\pi}{2}\left(\frac{t}{D_n} + 3\right)\right) + x_t \quad (3.14)$$

$$A_{n+1} = dir \cdot \gamma_1(\gamma_2 + f_t)\left(1 - \sin\frac{\pi t}{2D_n}\right) + x_t \quad (3.15)$$

The refined trajectory planner shares identical input and output signals with the original trajectory planner. The performance of the refined planner is superior to the original, providing improved response speed and a wider range of adaptability. The result of the interactive controller in Section 3.3 was achieved with the refined planner. Results of the interactive controller using the original planner can be found in an earlier publication of [102].

### 3.3 Performance validation tests

Validation tests with a small number of participants were carried out prior to the subjective evaluation experiments to assess the performance of each module of the system. All experiments in this section were carried out on the ViSHaRD10 robot with a metal rod as the end-effector.

#### 3.3.1 HBP estimation

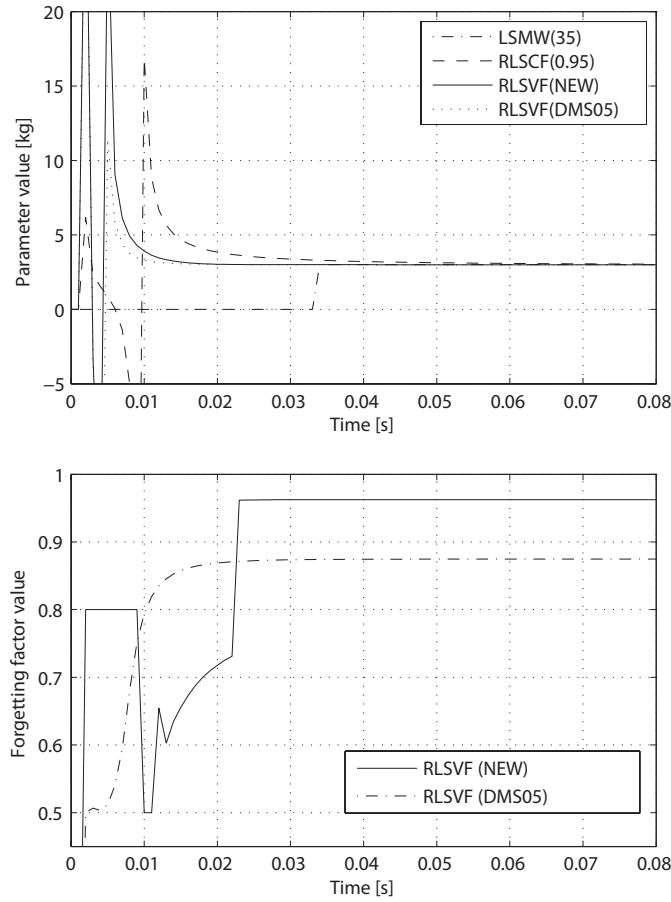
The aim of this experiment was to validate the performance of the fast online identification algorithm. Among the three impedance parameters, estimating mass is the most challenging task and the least reported one in literature, therefore the experiment is chosen to perform mass estimation.

#### Experimental design

Two experiments are carried out for each algorithm, both by simulation first and followed by robotic experiments on the ViSHaRD10 platform. In the first experiment, a constant mass of 2kg is mounted onto the end-effector. The robot arm is excited with a third order polynomial as input. Impedance estimations are made online. Signals are filtered using online IIR filters. In the second experiment, for the same condition as in the first experiment, the mass value is changed from 3kg to 2kg during the excited motion, to access the responding property of the algorithms. The position transient on the robot due to mass changing lasts about 0.1s.

Four different methods were used here for comparison:

- LSMW(35): Least squares with moving window length 35ms;
- RLSCF(0.95): RLS with constant forgetting factor of 0.95;



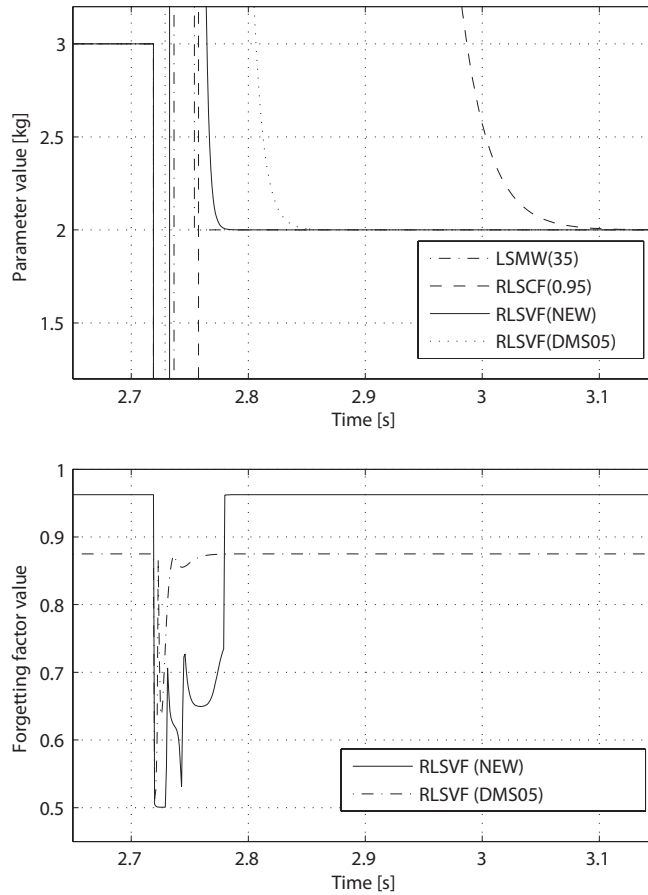
**Figure 3.7:** Simulation results comparison for constant 2kg mass estimation. LSMW(35): LSMW with window size of 35; RLSCF(0.95): RLS with a constant forgetting factor of 0.95; RLSVF(NEW): the proposed RLSVF algorithm; RLSVF(DMS05): the RLSVF algorithm by Diolaiti et al.; a (above): mass estimations; b (below): forgetting factors.

- RLSVF(DMS05): RLS with variable forgetting factors [23];
- RLSVF(NEW): The proposed new RLS method.

The parameters of the first two methods were tuned to provide the best trade-off between noise rejection and tracking performance in the specific task. The third method was tuned such that the noise rejection performance is identical to the proposed fourth method. Identical parameter sets are used for each algorithm in simulation and in robotic experiments.

## Results

In simulation with no noise corruption, all algorithms show good convergence speed. As shown in Fig. 3.7(a), all algorithms converge within 1s. The window size is 35 for LSMW, hence converges at 0.036s. RLSCF has a forgetting factor of 0.95 and has the lowest convergence speed. The two RLSVFs show similar performance.



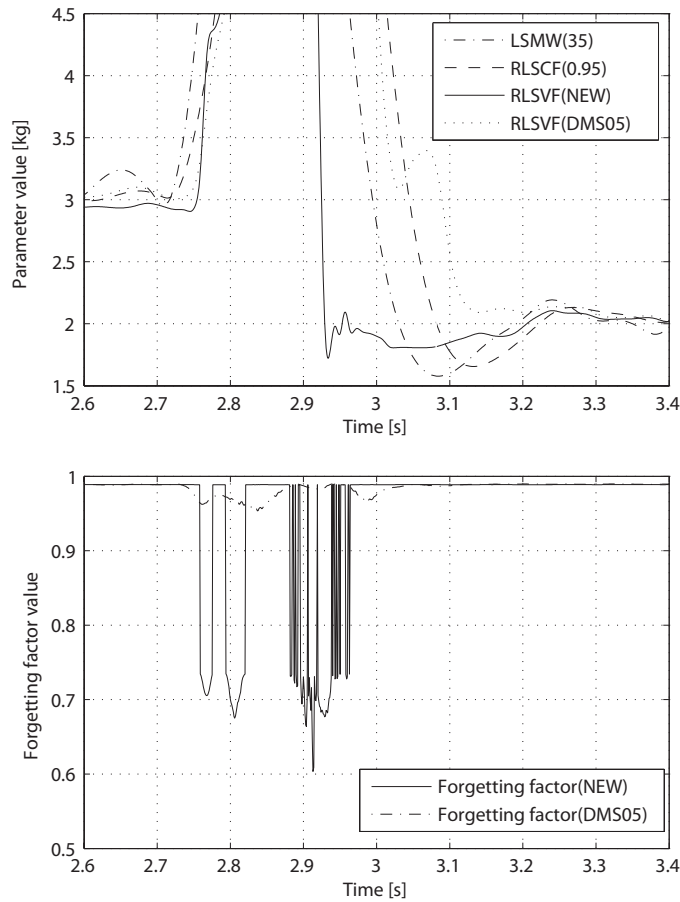
**Figure 3.8:** Simulation results comparison for mass changed from 3kg to 2kg; a (above): mass estimations; b (below): forgetting factors.

The forgetting factor trajectories are shown in Fig. 3.7(b). The parameters for calculating the forgetting factor are chosen to be:  $\alpha_1 = 0.98$ ,  $\alpha_2 = 0.1$ ,  $\alpha_3 = 1$ ,  $k = 10$ . The forgetting factor is assigned as 0.8 before the 10th estimation.

Next, a parameter change is applied in simulation to test the tracking performance. As shown in Fig. 3.8(a), both RLSVF algorithms show better tracking performance than the RLSCF. LSMW is the fastest in response. In Fig. 3.8(b), it can be observed that the proposed method has forgetting factor values that stay in the low range longer than the DMS05 method, resulting in the faster convergence speed.

Robotic experiments show different performances as in the simulations, as noise corrupts the measured signals. One trial is carried out for each algorithm, starting with a mass of 3kg and changed to 2kg in the movement. The change of mass is carried out after a sufficient length of time after all algorithms are converged. According to the simulation results, only the tracking of mass change is considered.

As shown in Fig. 3.9(a) and Fig. 3.10(a), when the robustness performance of the two RLSVF methods are tuned to be similar in the steady-state period, the tracking performance is very different. The proposed method is about 0.2s faster than the DMS05 method. This can be explained from the forgetting factor trajectory in Fig. 3.9(b) and



**Figure 3.9:** Experimental results comparison for the RLS module. Mass changed from 3kg to 2kg; a (upper): mass estimations; b (lower): forgetting factors.

Fig. 3.10(b), where similar forgetting factors are observed for the steady-state period while in the changing period the proposed method has much smaller forgetting factors to provide fast tracking.

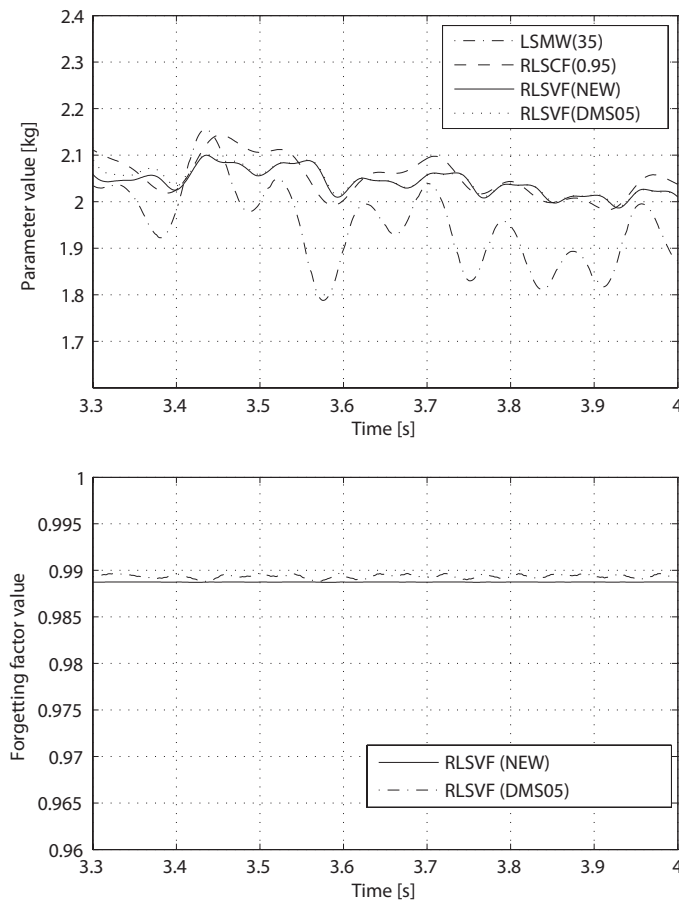
On the other hand, for the best tracking performance achievable of DMS05 method, as shown in Fig. 3.11(a) and Fig. 3.11(b), the proposed method is still about 0.1s faster, but in the steady-state period DMS05 method varies between 1.8-2.25kg, while the proposed method being within 2.0-2.1kg.

### Convergence and robustness

The proposed algorithm introduces a time-varying forgetting factor for the RLS method. Therefore, the convergence property is concerned, as well as the robustness of the scheme to perturbations and other uncertainty terms.

Proving convergence is a difficult issue for forgetting factor RLS schemes. For the proposed scheme, relating the forgetting factor to the parameter change does not change the fact that  $\lambda$  is always bounded within a user-defined range. Therefore in the worst case  $\lambda$  would be switching between two different values within  $(0,1]$ . With carefully tuned param-





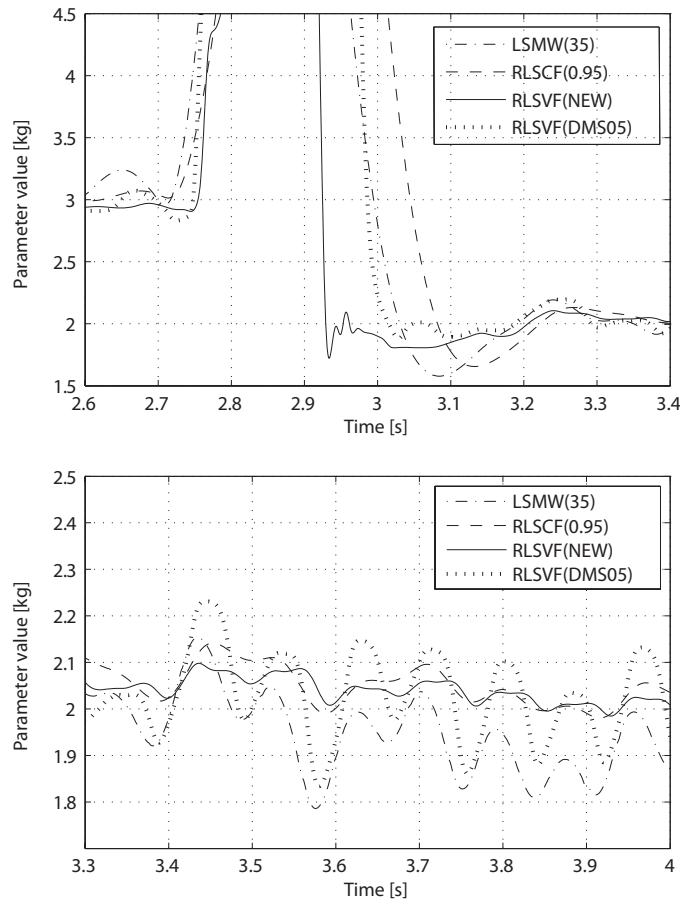
**Figure 3.10:** Experiment results comparison for the steady state mass at 2kg; a (above): mass estimations; b (below): forgetting factors.

eters, the two regions of forgetting factors can be connected with each other. Therefore, the proposed method has similar convergence property with other RLS schemes with time-varying forgetting factors such as the DMS05 method. The initial value selection of the targeted parameters plays an important role in convergence.

In practice the proposed scheme shows a good convergence property and robustness with an initial value of all estimated parameters set to zero. When mass changes from 3kg to 2kg, the algorithm converges to the new working point very quickly. Similar results are observed when applying disturbances to the system.

## Discussions

The constant forgetting factor RLS scheme, although already better than the standard RLS, shows poor tracking performance in the tests. The convergence time is around/above 1s. LSMW is fast, but suffers from oscillations in the steady-state period. RLSVF methods perform in general better than the other two. DMS05 RLSVF compromises performance in tracking with robustness. The proposed method solves this issue by changing the function of the forgetting factors.



**Figure 3.11:** Experiment results comparison for estimation with modified parameters in RLSVF(DMS05); a (above): with changed mass from 3kg to 2kg; b (below): with steady state mass at 2kg.

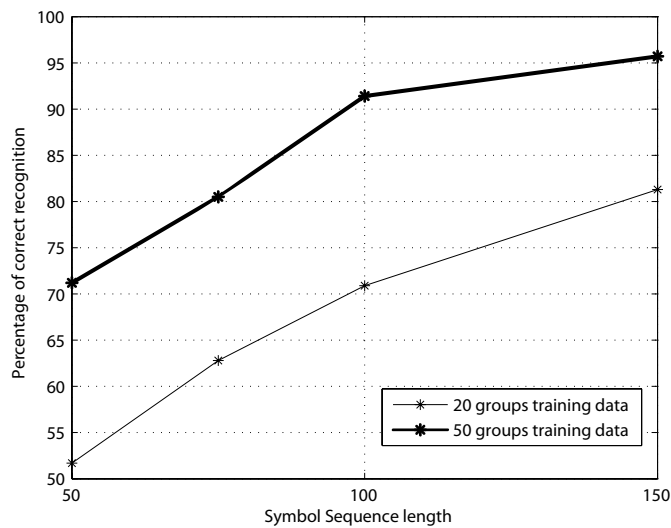
For absolute tracking performance, as the change of mass introduces transients to the robot arm motion, the actual convergence time is shorter than 0.2s as shown in Fig. 3.9.

The results shown are the estimation of mass. The performance in estimating stiffness or damping is even better for the proposed algorithm due to less noise affections. As the filtering and computation methods are standard, the algorithm application can be generalized to other applications.

### 3.3.2 HMM estimator

Validation tests were carried out for the HMM estimator. The training data was obtained by constructing a controlled environment, where 4 participants were instructed to either always take the lead or always follow the robot passively. 50 groups of training data were recorded for each case. The results of training using different data sets are shown in Fig. 3.12. For 50 groups of training data, sequence length 100 can obtain an accuracy rate of more than 90%.

The two trained HMMs for *Passive* and *Active* are implemented into the HLC which



**Figure 3.12:** HMM training results comparison for different symbol sequence lengths. A symbol sequence of length 100, corresponding to 100ms in time, provides an accuracy rate of more than 90% for 50 groups of training data.

executes in real-time (1kHz). The input sequence length of 100 is employed. The training data is obtained from a small number of participants, therefore the trained HMMs are potentially subject dependent. An extensive research will be carried out in the future for subject dependency.

### 3.3.3 Overall system performance

After testing the individual modules, they were integrated into the HLC of the interactive controller. Human-robot experiments were carried out using both the basic controller and the interactive controller. The participants were different from those in the training group, and results are shown in Fig. 3.13 and Fig. 3.14. For the basic controller, compliance was only provided by the virtual impedance model of the robot while for the interactive controller the robot could synchronize to the human. Another observation can be made from the force signals, that for basic controller, the human needs a relatively large force to drive the robot along the measured trajectory, while for the interactive controller, much lower forces are needed since the robot can detect the human trying to lead and hence adapts its reference trajectory to follow.

Fig. 3.15 shows the estimated results during the same handshake as in Fig. 3.14. The first observation from Fig. 3.15 is the occasionally negative HBP estimates. Stability of the system is not affected by the negative estimations, as they are not involved in the position control loop directly. The separation of *large* and *small* values, however, should take into account the possible negative values and use absolute values of each parameter. The thresholds are shown in Fig. 3.15 as dashed lines.

The results shown in Fig. 3.14 and Fig. 3.15 support the notion that HBPs are more suitable for human intention estimation than the measured force signal, since force mea-

surements contain information exerted by both the human and the robot. Between 1.6-1.9s in Fig. 3.14, for example, the measured force spike is applied by the robot driving the human, while the human arm is actually loose and acting as an additional inertia only, as can be observed in Fig. 3.15, therefore despite the high force measurement, the intention estimation result is *passive*.

On the other hand, the intention estimation results are also affected by the threshold values in HBPs. In the example in the above paragraph, if the threshold of  $h_1$  were set to 20 instead of 50, the resulting symbol between 1.6 and 1.9s would have been 3 instead of 0, the resulting human intention could have been different. Varying the threshold in HBPs can be interpreted to some extent as changing the *personality* of the virtual robot partner, where higher thresholds suggesting a more dominant one, while lower values being more gentle and adaptive.

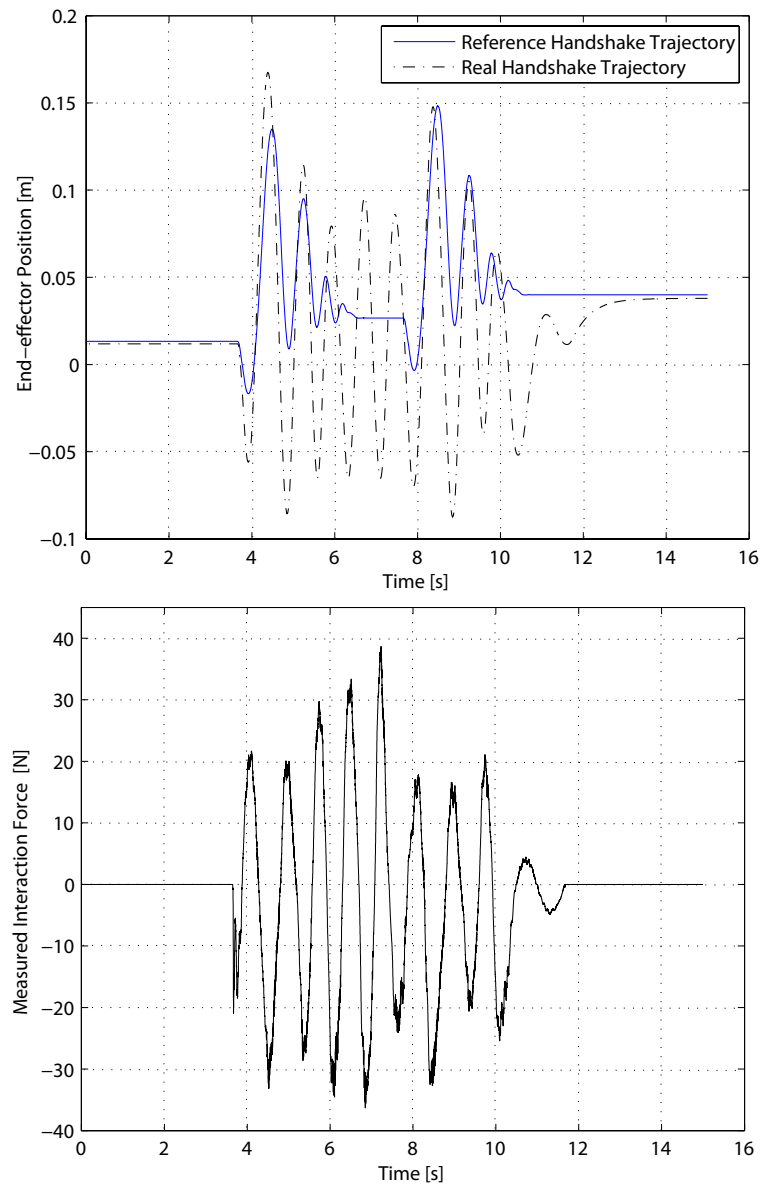
## 3.4 Conclusions

In this chapter different approaches to haptic rendering of arm dynamics on a robotic arm during handshaking were discussed. An interactive controller was designed to adapt the reference trajectory of the low-level controller for human-robot handshaking according to the actual interaction. HMMs with haptic inputs were employed for human intention estimation. A fast online parameter identification method has been proposed to provide estimations of human behavior parameters. An HMM-based estimator then estimated the current human intention state according to the identified human behavior parameters. Validation experiments were carried out on individual modules and on the overall controller. The proposed parameter estimation algorithm achieved best tracking and noise rejection performances among the comparative algorithms, both in simulation and in experiments. The HMM estimator achieved a correction rate of 90% for the selected input sequence length of 100. The overall controller could synchronize with human during handshaking, reducing the effort (interaction force) of the human.

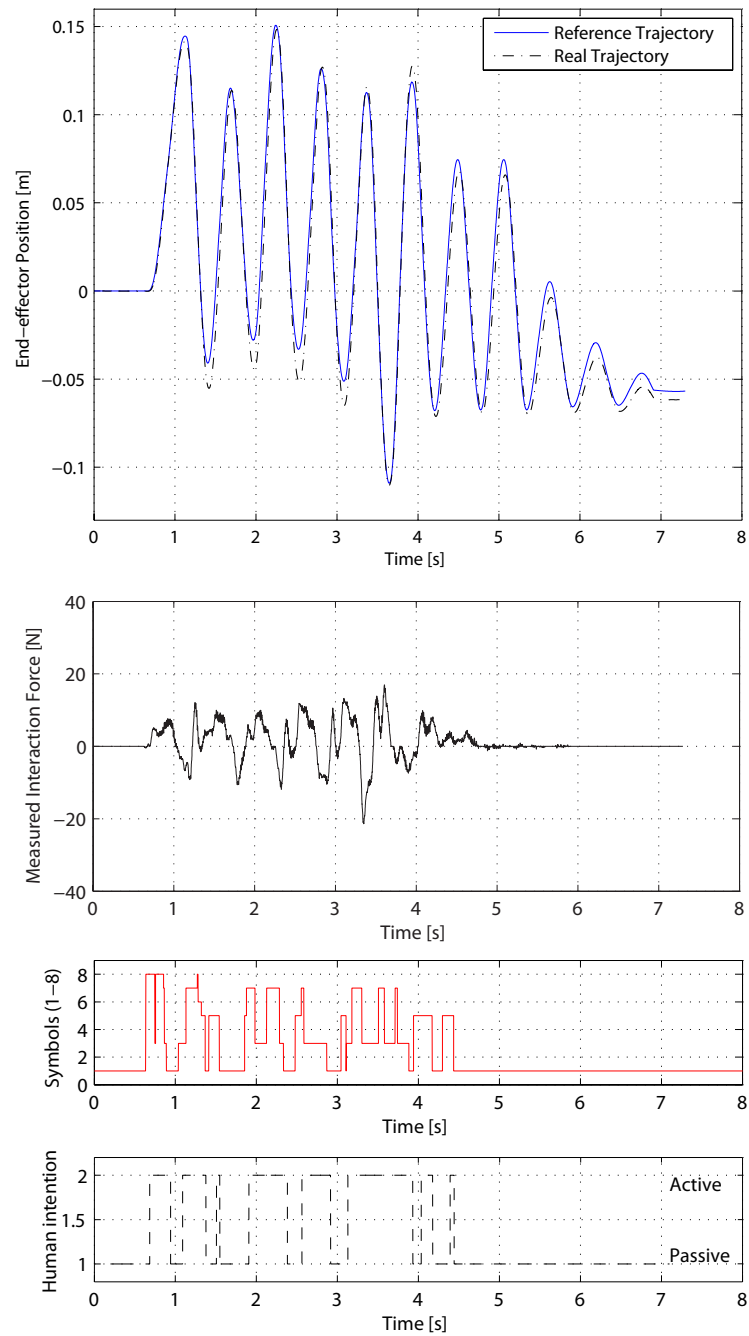
The main contribution of this chapter is Abstracting human related information from behavioral data, and making use of the information for online robot control. In experimental results, the estimated human intentions are capable to guide the robot when to lead and when to follow. This can be reflected in the much reduced interaction force while maintaining the interaction trajectory of similar amplitudes, comparing with the recorded handshakes using previous generations of robot controllers.

However the approach is not yet ideal. The estimation algorithm of HBP has a promising performance in experiments with controlled environments, where mass identification is carried out. In HBP estimation, no reference can be provided about the true HBP values, hence a trust index of the estimated parameters is not available. The HMM estimator suffers from a similar problem: the actual human intention information is not available as a reference, hence an accurate performance evaluation cannot be calculated. Currently both problems are avoided by approximations: parameter changing rate is used instead of estimation error for HBP estimation; for HMM estimator a trained human performs reference recordings consisting of purely active/passive data, correction rate is calculated as the percentage of time the estimator gives correct estimations.

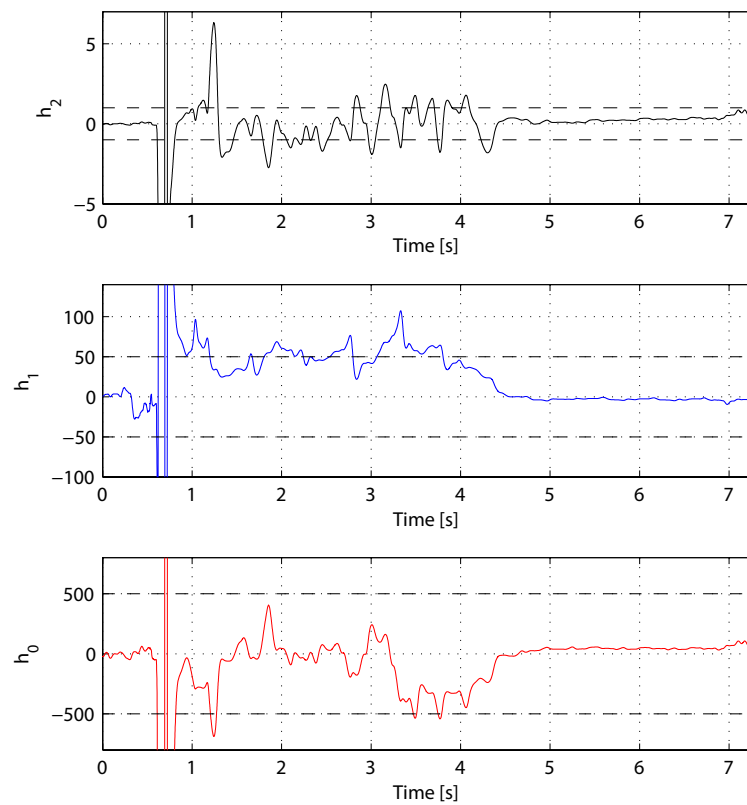
The trajectory planner considers a limited set of conditions, which can be further refined in the future. On the other hand, the above mentioned problems could be addressed by introducing physiology methods to measure human properties directly.



**Figure 3.13:** *Experimental results of human-robot handshaking using the basic controller. Upper: position; lower: force. Reference trajectory is not changed according to the human input. Human keeps applying large forces to drive the robot.*



**Figure 3.14:** *Experimental results of human-robot handshaking using the interactive controller. Upper to lower: position, force, symbol, estimated human intention. Reference trajectory is modified according to the human input. Force decreases as the robot synchronizes with the human.*



**Figure 3.15:** *Estimated impedance parameters of human-robot handshaking using the interactive controller. The impedance parameters remove the robot factor in the joint force measurements, considering only human performances.*



## 4 Haptic rendering of hand dynamics

The hand is a basic, yet natural tool for the human to explore and interact with the world. In social related haptic interaction tasks such as handshaking, actions as well as feedbacks of the hand are both vital to the overall experience. Consider a human-robot haptic interaction task, problems concerning the hand can be broken down to measuring the human hand gesture and gripping force for the robot, as well as generating appropriate robot actions to the human.

To measure the gesture and gripping force of the human hand, separate sensors are required. Gesture sensing is a more matured area with commercially available sensing gloves. Operational issues such as calibration and data processing are to be addressed in order for the device to be implemented in human handshake measuring. On the other hand, gripping force measurement is much less studied. The main challenge lies in the lack of suitable devices. Therefore the main focus of this chapter is placed in the development and implementation of a sensing glove to measure human gripping forces.

This chapter is formulated in four sections. First the measurement of human gesture data is discussed. Two measurement devices are presented, one commercially available, the other under development phase in lab. Then the measurement of human gripping forces is discussed in detail, from sensor selection, glove design, to implementation and validation experiments. Due to the nature of the sensors, the glove measures tactile information of contact pressure of the human hand. High spatial density sensing units cover the main contact area of the hand. Methods to estimate gripping force based on the tactile measurements are provided with validation results. Conclusions are drawn at the end.

This work develops a tactile sensing glove with high spatial density sensing units. The glove is based on flexible sensors, such that the human can carry out manipulation and interaction tasks wearing this glove. Further more, the tactile glove can be worn outside a gesture sensing glove to provide gesture measurements at the same time.

Comparing with the SOA, the main contribution of this work is the flexible tactile sensing glove. It serves as a low-cost alternative to customized tactile sensing gloves by using commercially available sensor units, while having advantages of high spatial resolution, compatibility to be used together with gesture gloves, and capability of gripping force estimation.

### 4.1 Gesture data acquisition

To measure human hand gesture data, commercially available devices such as the Cyber-Glove [21] have been widely adopted. Human grasping study has been an active area for decades. There have been reported works where it is used to categorize discrete human hand postures ([47]), as well as to provide continuous measurements ([13]).

Yet new sensing devices/gloves are being developed to suit specific needs that existing devices are not capable of. Here a recently developed position sensing glove is discussed [97]. The glove is fMRI (functional magnetic resonance imaging) compatible while maintaining very good wearability and flexibility.

### 4.1.1 The CyberGlove

CyberGlove is a commercially available data glove for hand motion capturing. It has been used in a number of different applications ranging from rehabilitation ([14], [89]) to teleoperation systems ([22], [65]). There are two versions of the CyberGlove, capable of measuring 18 and 22 joint angles, respectively. The device is real-time capable. The one involved in this work is a 22 sensor version, as shown in Fig. 4.1, featuring 3 flexion sensors per finger, 4 abduction sensors, 1 palm-arch sensor, and 2 additional sensors to measure the flexion and abduction of the palm. Each sensor is a bend-sensing resistor that is thin and flexible, offering minimum disturbance to the normal manipulation of the wearer.



**Figure 4.1:** CyberGlove with 22 sensors to measure human hand gestures.

The CyberGlove sensors have very good nominal performance specifications: 0.5 degree of typical sensor resolution, 1 degree of sensor repeatability, linearity 0.6% of maximum nonlinearity over full joint range, and with a typical update rate of 100 records per second.

#### Online calibration of CyberGlove

CyberGlove bend-sensing resistor sensors are located at each joint of the hand. For users of different hand sizes, the location of each sensor on the hand varies. In fact, each time a same user puts on the same glove, the exact sensor location could be different. Therefore proper calibration for every time a user wears the glove is vital to the measurement quality.

Unfortunately, due to the nature of the glove, all sensors are worn externally to the hand, hence the connections between sensors and the joints they measure are not solid

but affected by the fabrics of the glove. Hence even the sensors have good specifications, repeating a motion several times does not necessarily result in the same sensor curvature every time. Therefore, although universal perfect calibration routine is desirable, the factory-provided calibration process only provides compromised performance for all-purpose applications. Hence different customized calibration methods have been developed to suit specific applications. A good review of CyberGlove calibration methods can be found in [36].

In telemanipulation applications, CyberGlove is normally optimized for measuring fingertip distances. Since the measured signal from the palm-located sensors are not as good as the ones on the fingers, the common approach is to take a number of recordings over time and run optimization. Given the available computational power and the scale of the optimization problem to solve [36], the routine is currently not online-capable.

In this work, the angles of each joint are of major concern in order to render a handshake. Another concern is, when a participant joins an experiment and shakes hand with the robot wearing the CyberGlove, from time to time the glove could lose calibration because of repeated motions of the hand. In this case a quick recalibration without the need of stopping the virtual world simulation is desirable.

Based on the aforementioned discussions, an online calibration routine for the CyberGlove is developed in [29], optimizing for joint angle measurement of the fingers. The calibration routine can measure 14 angles, 2 for the joints on the thumb, 3 for each other finger. The whole process takes less than 30 seconds, requiring the user to make two gestures, and provides competent results to the specific visual and haptic rendering applications.

### 4.1.2 Pisa glove

Another candidate data glove is made by the research team from University of Pisa [97]. It is not yet commercially available. The sensor has a different layout than the CyberGlove. The calibration, as provided by University of Pisa, can also be made online. But subject to the nature of the sensor, currently 3 measurements are available, 1 angle of flexion of the thumb, 1 for the index and middle finger, and 1 for the ring and little finger.

While not matching the CyberGlove in terms of measurement DOF, the strength of this glove lies in its fMRI compatibility and being ultra thin and flexible. It can be worn inside another glove, for instance the haptic sensing glove, for combined measurements.

## 4.2 Haptic data acquisition

When measuring haptic data on the finger tips, wearable and measurement quality are the two factors to be compromised, given the available technology.

The design and adaptation of a haptic data glove based on the Tekscan grip sensor 4255N is introduced here. The gloves are used in experiments to measure human to human haptic interaction. Data analysis methods and initial results are illustrated. Evaluation results show that the pressure sensor based glove can provide force estimations for applications

where force measurement accuracy is not strictly required. The glove is also capable of measuring high spatial resolution force distribution at a fast sampling rate.

### 4.2.1 Tactile sensing technology

The sense of touch consists of cutaneous (tactile) perception and kinesthetic sense. The former is received by the receptors in the skin and the tissue underneath, about the stimulation on the surface of the human body; while the latter by the rest parts of the body about the static and dynamic gestures. Haptic perception is the combination of the two senses which includes the most cases of perception we encounter in our daily lives [55]. Measurements are also made from two distinctive directions: the kinesthetic information is acquired mainly from the joint of the device that the participant holds on to, while tactile information from the measuring mechanisms mounted on the contact surface. A detailed review can be found in [40].

Here only the contact pressure between two objects is considered. The physical basis of electrically measuring pressure is the piezoelectric feature of certain materials. The electrical feature such as capacity or resistance of the material in contact varies with the exerted pressure under certain rules. Using a read-out circuit, this variation can be transformed into a varying voltage or current signal, and thus sampled into digital computer compatible forms. Since the discovery of the piezoresistivity of semiconductors in the 1950s, a variety of semiconductor-based pressure sensors have been produced meeting the requirements of specific tasks ([25], [42], [57]). Recently as the developing of the integrated circuit industry, micro-machined pressure sensors have been developed. Integrated onto a printed circuit board (PCB), the new pressure sensor has marginally higher density. Multi-direction measurements are now possible as well as the distribution of force across the contact area ([2], [48]).

Another trend of pressure sensing is the flexible pressure sensitive material. It is used to provide an indicator of contact, rather than in accurate measuring devices, in research areas such as manufacturing, human machine interface, and robotics. The material can be shaped into various forms from thin lines to large surfaces to cover the area of concern, so that the output of the measure circuit changes when the area is applied with a detectable pressure. The simple setup and low cost make it possible to cover a large area with such devices. Robots with sensitive skin knowing which part they are being touched have already been developed ([34], [41], [92]).

Similarly to pressure sensing, force sensors are also based on the piezoelectric materials mentioned above. However, since the piezoelectric material is sensitive to the pressure applied to its surface, well controlled contact area becomes the key factor in force measurement.

One common approach is to build a pressure sensor into a mechanism such that the force to be measured is applied to a button higher than the base and hence transmitted to the sensing surface by the button with a fixed area. The pressure measurement can then be multiplied with this fixed area and results in the force applied. This is the basic idea of a one dimensional load cell [78]. By controlling the contact area, a pressure sensor can be used to give force estimations [17]. For multi-direction load cells, the force along each axis is measured by the differential signal from one set of piezoelectric materials. One load cell

consisting of several sets is capable of measuring multi-axis forces both translational and rotational. An example can be found in [110].

Another approach is by ensuring the sensing area larger than the contact area, so that the pressure can be read out from the pressure sensor while measuring the contact area by other methods such as a video camera. The force can be calculated from the two measurements afterwards. Some new approaches even estimate force directly from video image with a pre-trained decision model [90].

A critical point of pressure and force sensing of solid contact is that the sensor should have minimum effects to the original shape and deformation features of the surface; otherwise the measured contact is different from the original. Moreover, for pressure sensing, the sensitive area should be smaller than the contact area, so that the sensor gets fully engaged and gathers the most possible information from the contact. However, for force sensing, it is important that the contact area is not larger than the sensitive area, so that no force is distributed outside of the sensitive region and thus can not be detected by the sensor. Therefore the pressure sensors suffer from modest accuracy when used as force sensors, since the location of pressing yields a significant difference in the results. Load cells see much better performances in this aspect, but the housing makes them impossible for wearable devices which are getting more and more popular in presence related researches. However, as the increasing of sensor density, the sensor size is getting much smaller than the contact area so the force can be estimated at a better accuracy by the summation of the force values on each sensor cell [91].

## 4.2.2 Glove design and implementation

### Technical requirements

The gloves are used in measuring haptic information of human-environment interaction. The design guideline is to provide reliable, accurate measurements in a high frame rate while not affecting the process being measured. Therefore the evaluation criteria are as follows:

- Robust construction, invariant to temperature, good linearity and low hysteresis, good repeatability;
- High dynamic range, high sensor spatial resolution, provides force distributions of the whole hand;
- Low sensor response time, high system update rate;
- Flexible and wearable design.

The NASA/DARPA Robonaut hand employs the QTC [57] pressure sensor after trying the FSR approach [3]. The FSR glove gives indications of contact while the QTC providing force value when plastic beads are used as force concentrators.

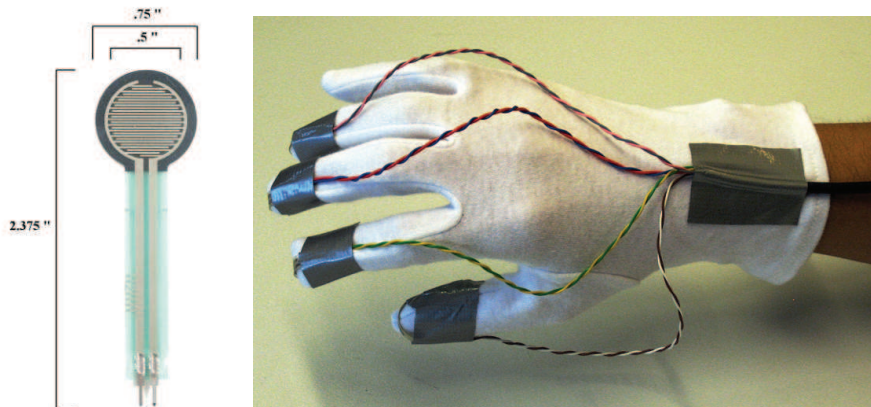
Similarly for the tailored sensing skin from the 90's ([42], [43]), the contact area needs to be defined when force values are concerned. For the multi-direction tactile sensor developed recently ([2], [48]), the dynamic range is yet to be improved until they can be implemented in grip force measurement.

In [17] a very similar glove was built using force sensing resistor (FSR) and evaluated with results similar to the above criteria. The FSR sensor is therefore taken as a starting point.

### The FSR prototype

A typical single FSR sensor consists of a polymer film sensing area varying in size and a flat extended cable for the signal read out, as shown in Figure 1. A prototype of the pressure sensing glove is made using 4 FSR sensors with 0.5inch (12.7mm) diameter from Interlink [75] as shown in Fig. 4.2.

In [17] the FSR sensors at the fingertips were covered with metal plates to control the contact area. However, here the mounting of metal plates makes the fingertip no longer flexible and thus reduce the realistic feeling of haptic exploration. Moreover, in a pretest of using metal plates in contact area control, it is observed that due to the elasticity of the FSR sensor, the applied force is not evenly distributed throughout the sensor surface even when a metal plate is placed on top of the sensor. The test results tally with the sensor specification that unevenly distributed pressure results in inaccurate measurement. For the same force applied to the center and the edge of the plate which is placed within the sensitive region of one sensor, up to 80% measurement drops are observed.



**Figure 4.2:** Left: Interlink FSR sensor, right: FSR glove prototype.

Another issue is sensor bending. When the sensor is mounted onto a deformable base such as a glove, the sensing area may be bended even without an intended external force applied. In this case, a non-zero output different from sensor noise can be measured.

It is very difficult to increase the spatial density of the sensors due to the wiring problem. Stacking cables reduce the flexibility and cause even less accurate measurements.

The evaluation of FSR sensor can be found in [17] for which reasons it can be suitable for the situations where cost and simplicity are more concerned than accuracy. However other solutions are considered since the FSR technology does not match the specifications requirements proposed in 3.1.1.

### Tekscan Flexiforce sensor

The Flexiforce sensors from Tekscan [74] are FSR sensors with improved repeatability and linearity. The manufacturer offers a special setup 4255N for measuring grip force. The sensors 4255N has 20 sensing blocks with size 20mm by 20mm, consisting of 16 sensors with size 2mm by 2mm on each block (or 6.25 sensors / sq-cm). The cables are also built into the setup and can be connected to a single interface (named *cuff unit*) and further to the data acquisition card in the computer. The thin film design of 0.15mm thickness provides the user with high flexibility while the sensor itself maintains a dynamic range of up to 200 kPa (30PSI). Fig. 4.3 shows the 4255N sensor from original package [91].

Although Flexiforce sensor is FSR based, the high density and high coverage rate of the sensor help solving the problem of force applied to the edge of the sensitive region. With the densely placed sensor, not only the force exerted to the entire contact area but also the force distribution can be measured.

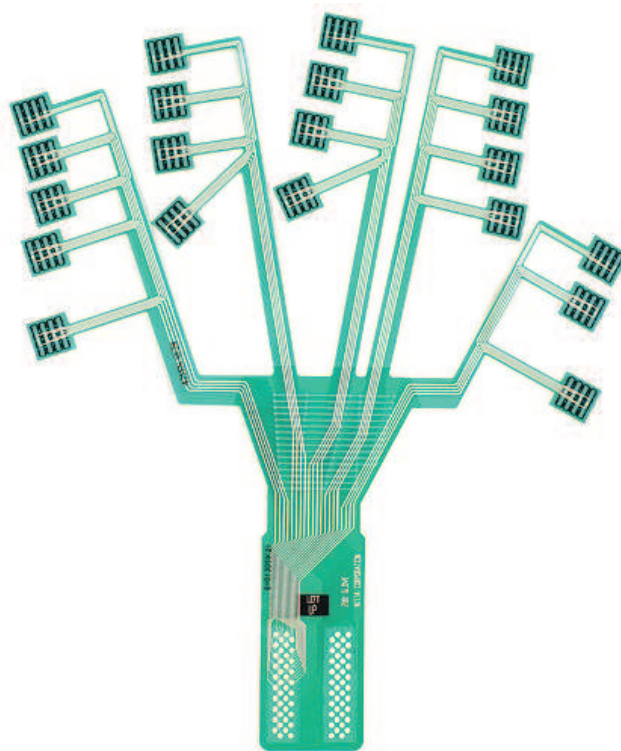


Figure 4.3: Tekscan 4255N grip sensor.

### Tactile Sensing Glove 1

Based on the Tekscan 4255N sensor, a first pair (two right handed) of sensing gloves are built and due to a later adapted version the first version is named the Tactile Sensing Glove 1 (TSG1).

TSG1 fits for the right hand with an artificial leather glove to provide better durability than the cotton glove and the surface is much easier to attach tapes. Double-sided tape is put between the sensing blocks and the glove to ensure firm and consistent connection

between them so that the pressure applied to the sensor is less possibly affected by glove elasticity. Stretchable plastic tape is put on top of sensors for protection. Fig. 4.4 shows the sensor positions on TSG1 before covered with protection tape.



**Figure 4.4:** TSG1 sensor position.

TSG1 is built following the manufacturer recommended setup of sensor 4255N: 3 sensor blocks on each finger and 5 on the palm. The sensing blocks cover the contact area in most grasping tasks. The dense sensor cells on each block can measure the force distribution during the contact. The update rate of the system is up to 120Hz and the measured data can be stored in ASCII format for post processing. The right hand setup can be transferred into left by copying and mirroring the sensor onto a left handed glove using an identical sensor and reverse the way of inserting the sensor into the cuff unit.

The sensor bending problem was proved to be a tension problem when a large hand wears a normal size glove. In adapted design TSG2, encountering bending became rare. The cable bending problem was also solved when building TSG2 by attaching supporting frames to the cables.

In general, TSG1 is a fast, sensitive, flexible while reliable device for tactile measuring. It gives high spatial resolution pressure distribution measures of main parts of the hand in grasping task in a high frame rate. The measured data is stored in a convenient and analysis-friendly way. However, calibration and force estimation problems may reduce the accuracy of the results.

### **Tactile Sensing Glove 2**

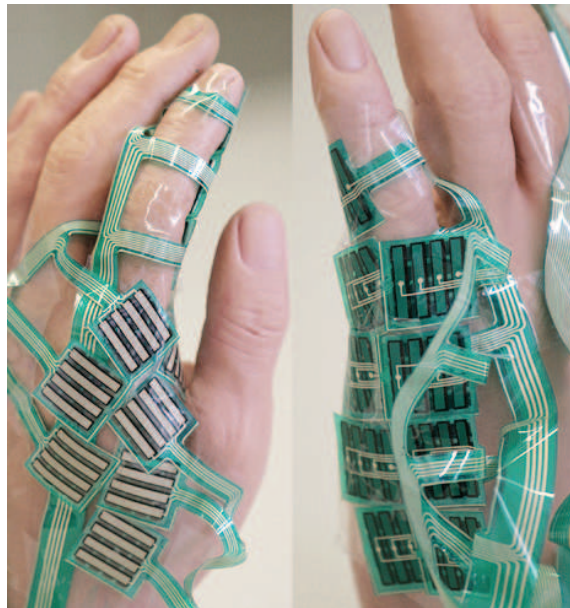
Based on the analysis of the handshake measurements using TSG1, changes are made to the glove by adjusting the area where the sensors are mounted. The resulting new version of the glove is called the Tactile Sensing Glove 2 (TSG2). TSG2 uses the same sensor 4255N from Tekscan. It is rather a specialized glove for measuring human to human handshake force exchange. Comparing with TSG1, the main improvements of TSG2 are:



The sensors are now more concentrated in the area of hand concerned with the particular application of handshaking.

Support frames are attached to the cables to avoid bending.

As Fig. 4.5 shows, 9 sensor blocks are mounted at the lower (small finger) part of the hand, while 8 at the upper (thumb) part which are the two parts that are passively being gripped during a human to human handshake. The other 3 blocks are mounted on each link of the index finger to measure the active gripping force exerted to the handshaking partner. The sensor blocks are closely aligned so that the sensors can measure the pressure distribution concerned in the handshake process and therefore provide us with more information on the interaction force during the process.



**Figure 4.5:** Sensing block position for TSG2 (on a dummy rubber hand), the thumb side on the left and the small finger side on the right.

The same artificial leather gloves are used as bases and the entire sensor is covered with tapes.

As the sensor is designed for mounting on the palm side, all cables are fixed for that setup. Changing mounting position results in an uneven configuration of cables. Moreover, aiming for measuring handshake, the cables must not obstruct the natural contacts of the hands of the handshaking partners, which further constrain the cable wiring. Some cables must be turned over to fit the mounting position. To protect the cables from being bended, supporting frames are used to provide rigid bases for the cables to avoid sharp bends. Two gloves are made following different methods of attaching supporting frames, one as an exoskeleton and the other as short sections. The participants for the handshake experiments provided comments showing that short section frames feels more natural. Fig. 4.6 shows from left to right the original leather glove, the TSG2 design 1 with exoskeleton supporting frame, and the TSG2 design 2 with short section frames.

After employing the supporting frame, the cables are better protected and less easily bended. During the first experiment of 300 handshakes, 2 sensors were worn out, while in



**Figure 4.6:** Base glove without sensor, TSG2 design 1 with manufacturer suggested sensor layout, TSG2 design 2 with customized sensor layout.

the second experiment of 900 handshakes the two TSG2s remained fully functional from the beginning to the end.

TSG2 suffers from the similar calibration and force estimation problems as TSG1. Detailed discussions are given in Chapter 4.

The successful adaptation from TSG1 to TSG2 shows that the design of TSG glove can be generalized. Provided that the region of the hand of concern is known, the TSG can be further adapted to fit other specific tasks with proper methods, such as supporting frames, to protect the bending cables.

### 4.2.3 Measuring handshakes with TSG gloves

The TSG gloves are used to measure force distribution and temporal information in human to human handshake experiments aiming to create realistic handshake in multi-modality virtual environment. A brief introduction of the experiment setup and the methods and initial results of haptic data analysis as well as the improvements of TSG itself are given here. The other aspects of the handshake experiments are not covered.

#### Handshake experiments

In order to create a virtual handshake that feels realistic to the human participants, certain behavioral principles are necessary so that they can be followed by the robot as a handshake interface. The data recorded from human to human experiments provide the information of how humans perform in a handshake. The haptic and position information recorded from the experiments are then used to generate models and principles for the handshake robot.

Two identical TSG1s were used in the first experiment. 30 participants formed 15 pairs. Each pair performed 20 handshakes which gives 300 handshakes resulting in 600 individual recordings. The participants were guided so that the initiator and follower of each handshake were well defined. This information was then considered in the data analysis.

Based on the findings from the first experiment, two TSG2s were made and used in the second experiment of handshake where 24 participants formed into 4 groups carried out 900 handshakes with 1800 recordings all together.

### **Sensor calibration**

In order to get accurate measurements, it is necessary to calibrate the sensors before using them for measuring. The calibration is divided into three steps according to the user manual [91]:

Conditioning: apply force to the new sensor for 20 times to activate the material.

Equilibration: assign a factor to each sensor cell so that identical measurements are obtained for each cell when the identical pressures are applied.

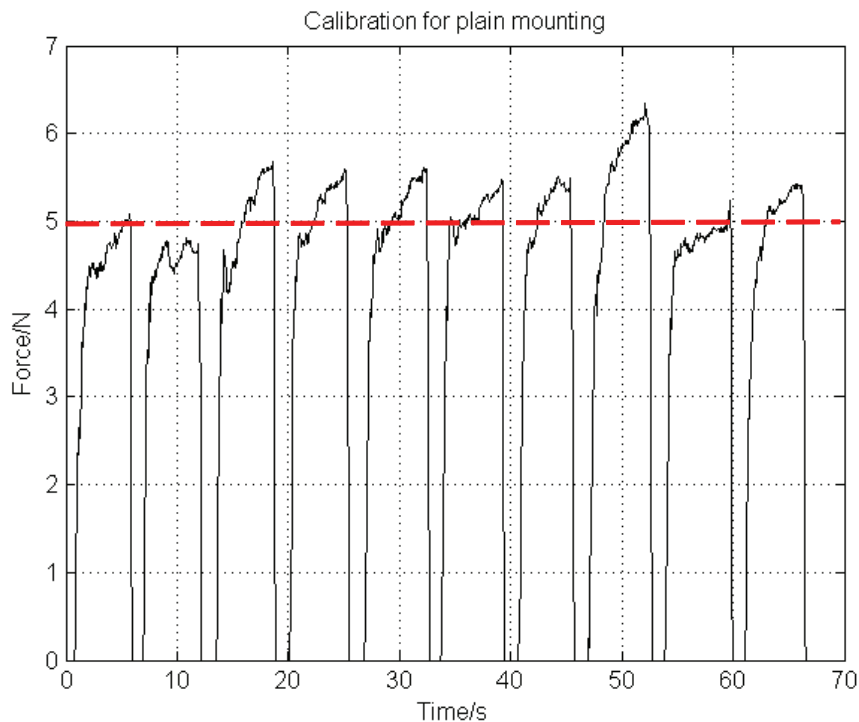
Calibration: obtain relationship between raw sensor values and the pressure values with units such as Pascal or mmHg.

Since in TSG the sensor is mounted on a deformable base and covered by tape, the calibration should be carried out after the glove is finished otherwise the characteristics of the sensor are distorted by the materials attached to the sensor. However, the calibration device provided by the manufacturer is designed to use only on a rigid plane surface.

After sensor conditioning, the calibration was divided into two steps. Firstly the sensor was mounted onto a rigid plane. Weights were used to apply well defined forces to the sensor. Repeatability of the 4255N is improved comparing with the FSR sensor. Fig. 4.7 shows the calibration tests results on one sensing block when 500g weight is put onto the sensor.

A noticeable drift is observed. As shown in Fig. 4.8, when a 500g weight is placed on the block and kept for about 40 seconds, the measured drifting is about 3N/min. However, releasing the load and reapply can remove the drift. Consider the application of this sensor in handshaking, each handshake lasts for a relatively short period of time, then the pressure on the sensor will be released. Therefore the drift will not affect the measurement result within the short time interval.

In the next step, the sensor is mounted onto the glove with cover tape. A normal force of 5N is applied to the sensing blocks. For 10 repeated tests, the standard deviation varies from 0.61N to 0.94N from block to block. It is observed that the measurement significantly decreases if the applied force is not perpendicular to the sensor surface, down to 2.9N lowest in the tests. However, when the force is kept perpendicular to the contact surface, the repeatability can be improved. For 5 tests with best performance out of the 10, the standard deviations are 0.36N to 0.45N. Fig. 4.9 shows the results on the glove. The repeatability degrees considerably when the sensor is mounted on a glove comparing with on a rigid plain. The force estimation is distorted by the elasticity of the glove and tape, bending and tension effects, leaning of contact force, etc. The measurement results are only acceptable when the force value accuracy is not strictly required.



**Figure 4.7:** Test result for rigid plane mounting.

### Data processing and results

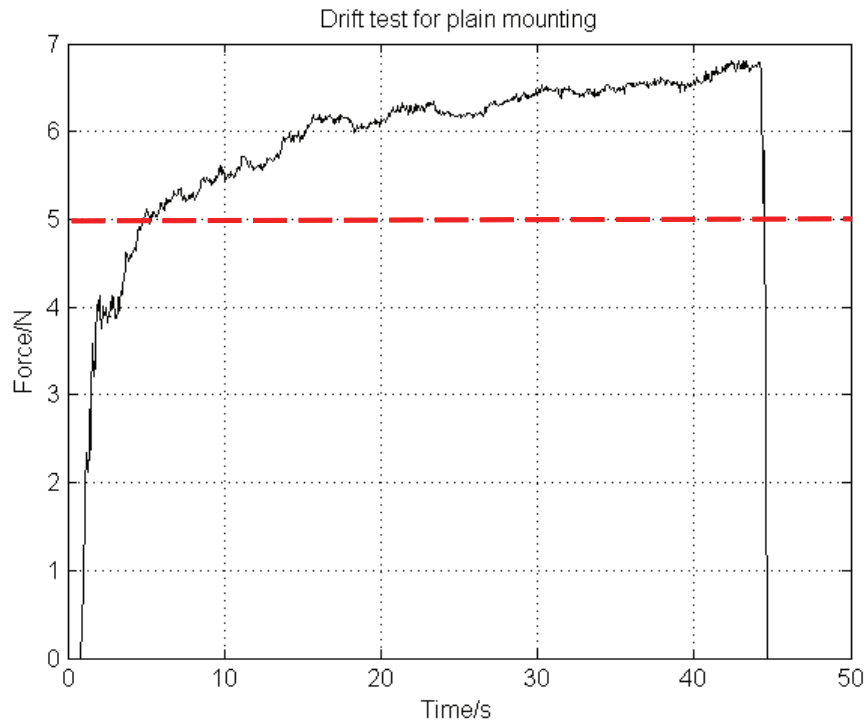
From the sensor, raw pressure measurements are stored as integers ranging from 0 to 255. The data are exported as ASCII files and then imported to MATLAB for analysis. The calibration results are used in transferring raw values to kPa unit values.

*Statistics of sensing blocks usage.* In the results from handshake experiment 1, some sensing blocks give very low outputs throughout the experiments. This fact indicates that not all the covered parts of the hand are active in a common handshake procedure. Inspired by this idea, a statistical process is carried out to the sensing blocks as follows:

- Estimate the force value of each sensing block at each time frame for each trial,
- Calculate the average of the force of each block for all time frames in each trial,
- Set a threshold and count the number of trials when the average force gets beyond it,
- Compare the counted numbers of each block being activated.

Fig. 4.10 and 4.11 give two samples of activation statistical results of blocks on the middle link of the ring finger and on the thumb side of the palm. Figure 11 shows the result of activation statistics.

The thumb appears to be used rarely in the above results. This should be caused by the sensor mounting position problem of the thumb, as in handshake experiment 2 the thumb resulted in the highest forces when the sensors are properly mounted.



**Figure 4.8:** Drift test result for plane mounting.

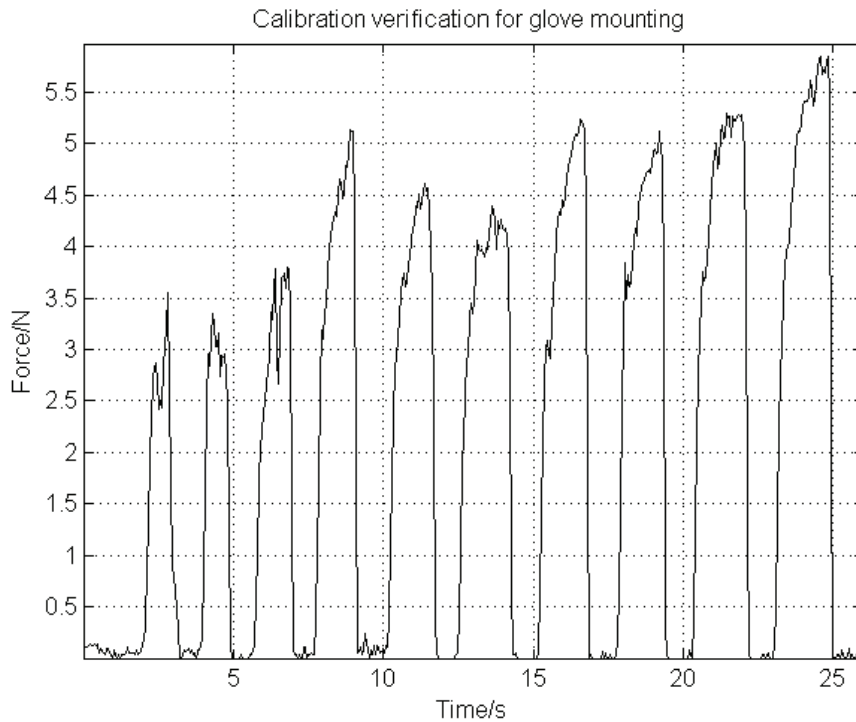
In Figure 11, the most activated sensing blocks align in the way that the hand of the handshaking partner is being hold. The gripping person decides how much force to exert, but where the force is mostly densely distributed is determined by where the hand of the partner is. Based on this conclusion, the TSG1 is adapted into TSG2. The sensor blocks are mostly mounted onto the two sides of the hand to measure passive gripped force, since those parts are gripped in most cases, no matter how the two participants are holding their hands. Only 3 blocks are mounted onto the index finger to measure the voluntary exerting force to the partner.

*Force estimation.* The 4255N grip sensor is sensitive to the pressure signal that is applied on it. An intuitive idea of deriving force value from the pressure data is to multiply the pressure with the area it is applied and get the corresponding force. However, in this particular case there are problems that prevent accurate results when doing so.

The grip force is distributed throughout the whole area in contact. Each small sensor unit can only pick up the pressure that is locally applied to it. As shown in Fig. 4.13 taken from [91] only the black areas are sensitive to pressure. All the other areas as well as the cables are not sensitive to pressure changes.

From the nominated values, the sensitive area is only one fourth of the total area of the sensing block. It is assumed that the pressure applied along the hand surface is continuous and slowly changing. Under the assumption the missing information of the insensitive areas can be obtained by interpolating the values of the nearest measurements.

The force estimation is then given by the sum of all pressure values multiplied by the corresponding area.



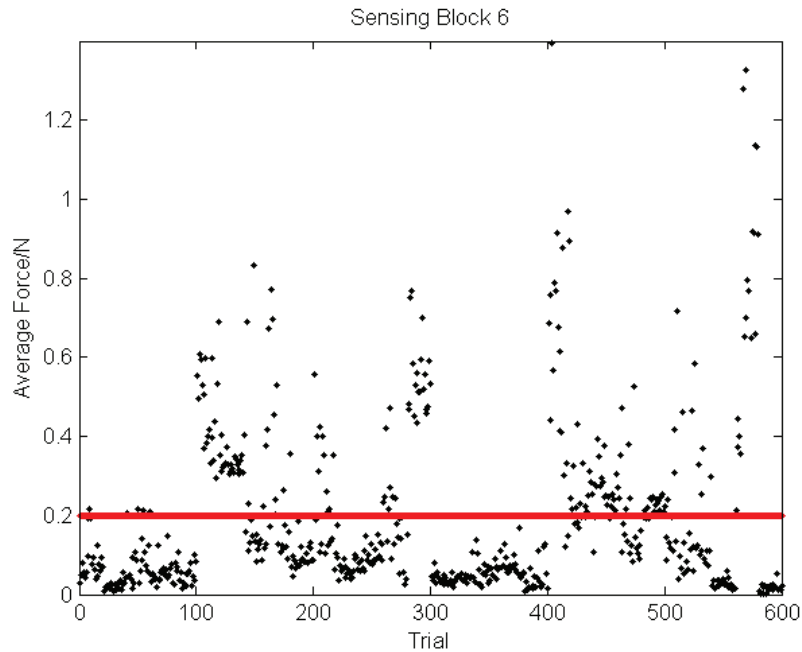
**Figure 4.9:** Calibration result of glove mounting.

Fig. 4.14 and 4.15 give two examples of estimated force from three blocks of TSG1 on the handshake initiator glove and on the follower glove. Figure 14 also shows the bending problem. Temporal information can be extracted, such that the middle finger grips earlier than the ring finger and the index finger grips the latest, for both participants.

For the large sensitive area covering one side of the hand, it is not sufficient to give only one over-all force estimation. In the later on realization phase, the robot needs more detailed information of how and where to grip with its fingers. Therefore, determination is needed on how many distinctive forces, where are they centered, and how large are they.

The data is clustered with the hypothesis that the upper part has one center resulting from the gripping by the thumb while the lower part has 4 due to the rest of the fingers. It is observed that the upper part of hand has more than one center visually identified, while the lower part has less than four. Two centers are tried for upper and lower part each with improved results. Fig. 4.16 gives examples of clustering results with force center and force sums in each cluster in Newton.

The argument for two force centers in the upper part for one thumb is that the two centers are applied by the two links of the thumb, while the two lower centers show that only two fingers are mainly active during handshake.



**Figure 4.10:** Activation statistics of the middle link of the ring finger.

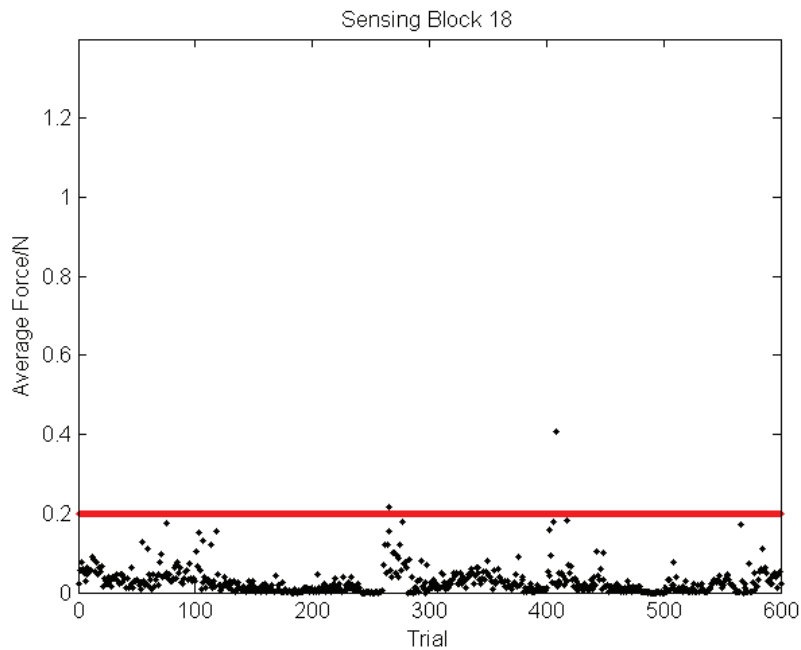
## 4.3 Robotic hand actuation

### 4.3.1 BarrettHand

After acquiring necessary information, the final step of haptic rendering is to generate appropriate actuation commands for the robotic hand to interact with the human user. The available robotic hand for this work is a BarrettHand. A BarrettHand [11] is a 3 fingered programmable robotic grasper. It has 4 DOF, 1 flexion per each finger, and 1 abduction between two fingers while the third finger is fixed in rotational axis. It is not an anthropomorphic hand, therefore finger joint angle mapping is not possible from human hand to the BarrettHand.

In telemanipulation, mapping algorithms from human hand to BarrettHand has been proposed, see a point-to-point mapping algorithm proposed in [64]. For handshaking, the distance between two fingers are of more concern than fingertip locations, therefore a modified algorithm of position mapping is proposed in [61]. The proposed mapping algorithm considers the distance between the middle link of the thumb and the opposite four fingers. In a handshake, the hand of the human is held in the robotic hand, the middle links of robotic fingers are the actual areas in contact with the human hand. Therefore the aim is to control the middle link distance properly, such that the robotic hand can apply grasping force without crushing the human hand.

Preliminary experiments of mapping the human hand motion to the robot have been carried out. As shown in Fig. 4.17, the human wearing a CyberGlove can shake hand with the ViSHaRD10 robotic arm with the BarrettHand mounted on the end-effector. The BarrettHand copies the human hand motion, such that the hard the human grasps the



**Figure 4.11:** Activation statistics of the thumb side of the palm.

BarrettHand, the harder it will grasp back. However the work is still at a very early stage, implementation to the handshake system remains future work.

## 4.4 Conclusions

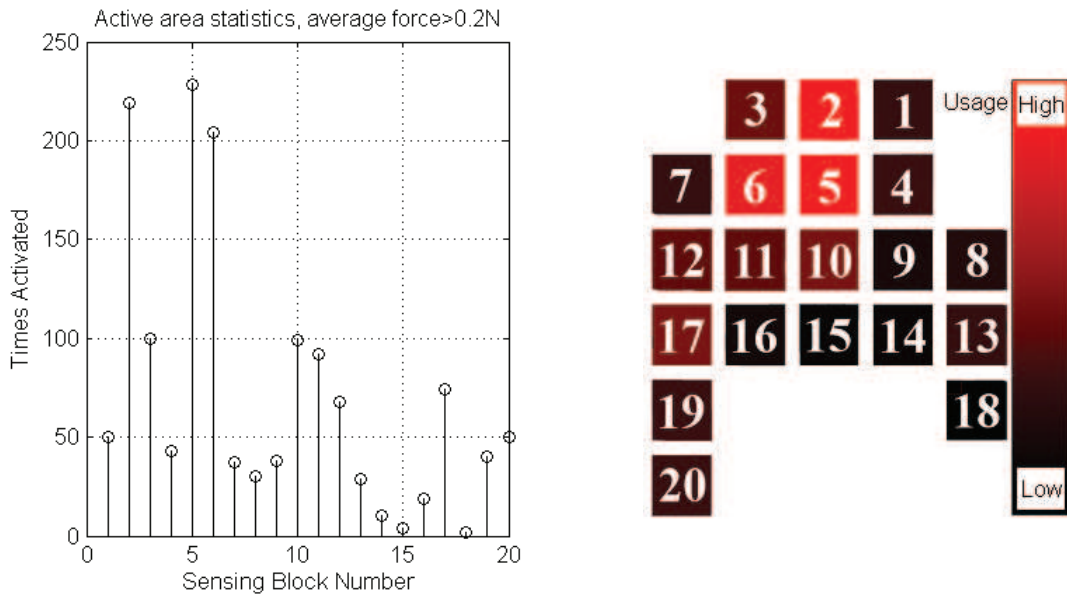
This chapter focuses on the haptic rendering of the hand, covering three aspects of the problem: position sensing, tactile sensing, as well as actuation.

Two candidate gloves, the CyberGlove and the Pisa glove, were studied to measure the motion/gesture data of human hands. Simplified human hand models were constructed to match the specifications of the two gloves. Calibration routines were developed to achieve quick online calibration for both gloves. The CyberGlove, with 14 DOF measurements after calibration, is suitable for both visual rendering of hand motion for the human user and real-time mapping of human hand to robot hand; while the Pisa glove, with 3 DOF measurements, is also capable for real-time mapping of the human hand motion to the robotic hand BarrettHand with 3 robotic fingers.

A tactile sensing glove was built to acquire haptic data of the human interacting with the environment. The mounting position of the sensor can be adjusted so that the sensor units are concentrated to measure the most concerned area. The pressure values can be interpolated to get force estimations. The tactile gloves were used in human-human handshake experiments. The measurements of human gripping forces as well as grasping spatial distributions were among the very first knowledge obtained about human handshaking. It was used in the modeling and realization of the handshaking robot.

The BarrettHand serves as the actuated robotic hand in this work. A real-time mapping





**Figure 4.12:** Activation statistics of sensing blocks and the visualization with block numbers.

algorithm from the human hand to the BarrettHand was proposed, which enables mirrored handshake: the participant wearing a position sensing glove can copy his/her own grasping motion onto the robot in real time.

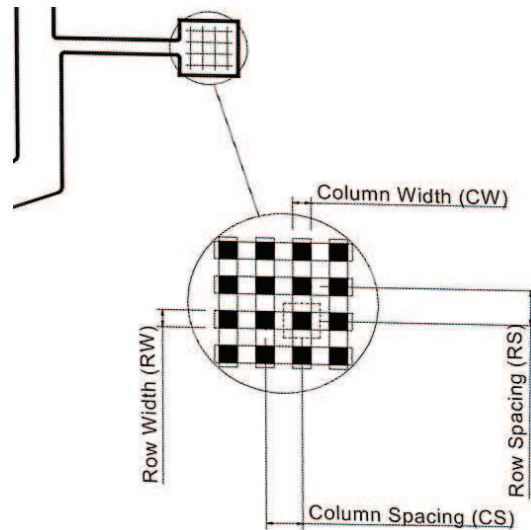
For this work, the CyberGlove, with the online calibration process, is used to map human hand motions into the virtual world. The Pisa glove is used to map human hand motions to the robot hand. The tactile gloves are used in the human-human handshake experiments.

The main drawback of the CyberGlove is the difficulty to fine calibrate close-loop kinematic chain from the thumb to the index finger. A time-consuming multi-variable optimization is required, which can not always guarantee decent results. In this work this problem is avoided since the fingertip distance is not required. The optimization is hence replaced by a one-time completion algorithm. However in other applications this may not be feasible, then other calibration algorithms are to be developed addressing specific needs.

For the tactile glove, the dilemma of flexibility and force estimation accuracy needs to be considered. Fine calibration cannot resolve this issue, since it is due to the characteristics of the sensor that any additional bending will cause unexpected measurements. Using stiff supporting surfaces underneath the sensing pads can improve the accuracy, but the tactile information will not be detectable by the human hand.

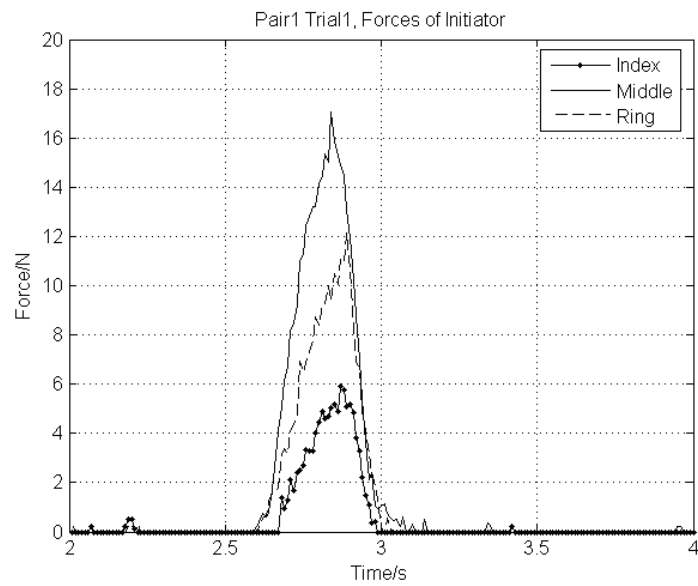
Future investigations are possible from the following directions:

- BarrettHand could be integrated to the robotic arm for handshaking will full fore-limb dynamics. A refined control algorithm is needed for human safety and better performance.
- Making use of the high spatial resolution of the tactile sensor, the tactile glove could be applied to analyze human grasp force distribution.

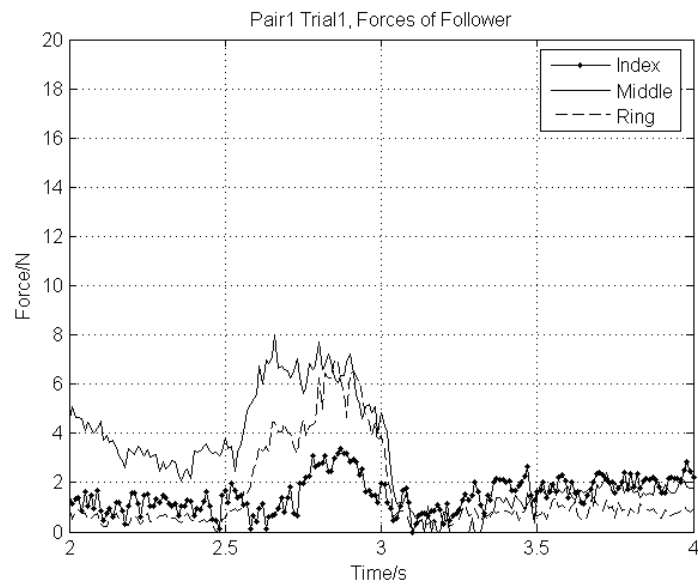


**Figure 4.13:** Detailed structure of one sensing block of 4255N, for 4255N,  $CW=RW=2\text{mm}$ ,  $CS=RS=4\text{mm}$ .

- Position sensing gloves could be integrated with the tactile glove, and the data measured by the two gloves could be fused to enable analysis of the object as well as the human behavior. Preliminary work has been carried out by using the combined glove to distinguish objects in a surgical context. The high spatial resolution of the tactile glove makes it possible to determine not only scissors or cup, but specifically which part of the scissor is in contact. A full discussion can be found in [29] This result shows a good compatibility of the two devices. However, it is not included here, as it does not comply with the main focus of this dissertation.



**Figure 4.14:** Force estimation of initiator using TSG1, fingertips of index, middle, and ring fingers.



**Figure 4.15:** Force estimation of follower using TSG1, fingertips of index, middle, and ring fingers.

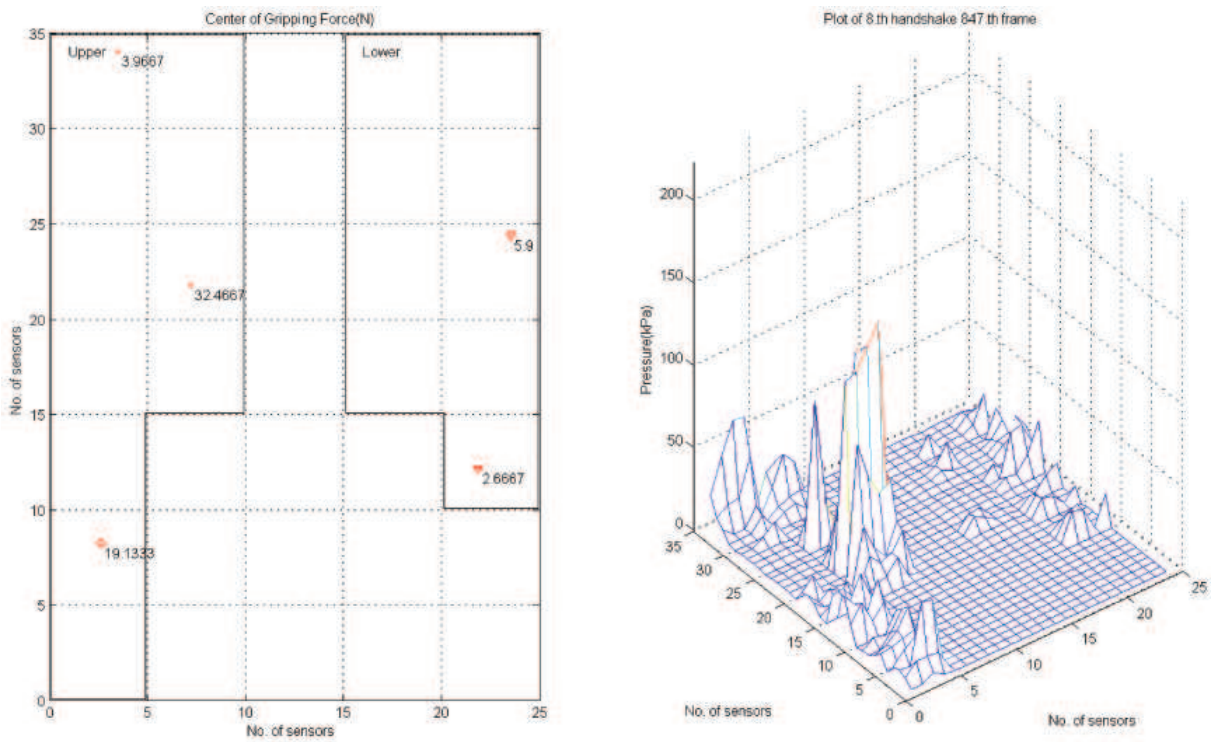


Figure 4.16: Clustering results of force centers.

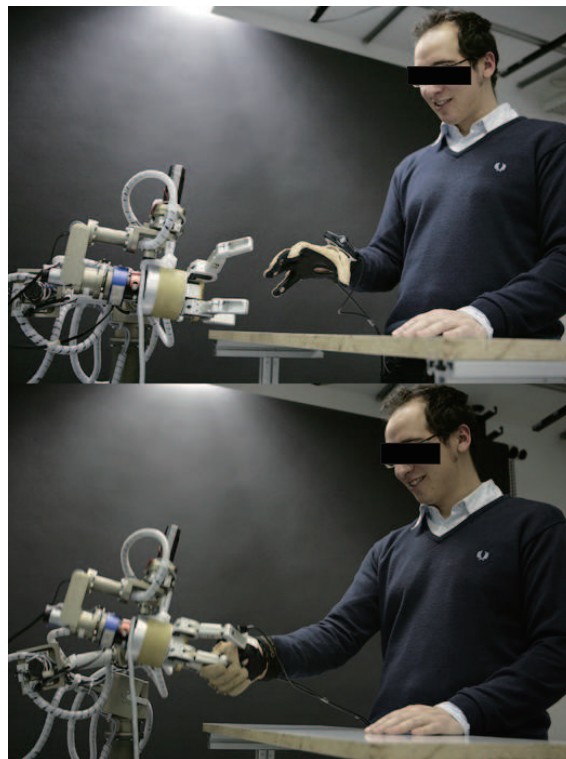


Figure 4.17: Preliminary experiment of hand motion mapping: a human conducts a mirrored handshake with the BarrettHand mounted on robotic arm ViSHaRD10.

## 5 Visual and sound rendering

A plausible VR scenario consists of a number of well-tuned components, ranging from the environment, person/objects, to the events that occur within the VR world, and most importantly, how they are positively added up. Adding haptics to the scenario brings new problems. For instance, the user has the haptic interface as an additional interaction point with the system, proper registration of the haptic interaction into the VR is crucial, otherwise instead of being *present* into the VR, the user will be constantly reminded of being in the lab by clumsy haptics. For visual rendering, to reflect the interaction of the user in the VR scenario, real-time animation generation is required based on the input data from haptic interaction.

To address the challenges, a VE with visual and sound components is created using real-time capable software environment. Haptic input data can be rendered as the motion of virtual characters. The highly detailed virtual characters can walk, speak, and handshake with the human participant in the VE.

This chapter is structured as follows: in Section 5.1, visual rendering is discussed. The workflow is compared with existing systems, and then virtual characters are created with different genders, clothings, different level of details as well as DOFs. Two virtual scenarios are created with possibly occurring events. In section 5.2 sound is integrated into the system. Voices of people hosting conversations are synchronized with the mouth movements of the virtual characters and mixed with the background music/noises online.

Comparing with the SOA, the main contribution of this work is real-time full-body animation generation of the virtual characters with haptic interaction data input. For the handshaking application, a highly detailed virtual hand model with online animation generation capability is integrated into the virtual characters.

For the sake of clearness, the discussions in this chapter are carried out in a logical way instead of a timeliness way. Therefore different stages of work will be labeled to differentiate their versions, since they will be referred to in evaluation experiments.

### 5.1 Visual rendering

There are a large number of reported VR systems with high fidelity virtual characters and environments ([86], [9], [82], [54]), although haptics is rarely considered by such systems. On the other hand, systems built around haptic tasks ([31], [88], [60], [87]) are often focused on a specific task with a simple VR setup.

When developing the visual rendering algorithm for handshaking, the main focus was placed on emphasizing the two main points that are lacking from existing systems:

1. Optimized for real-time animation generation of full human body.

2. With highly detailed hands on the virtual characters, since the handshake application.

In addition, the essence is also to lower the requirement of computational power, so that the resulting system can be run on a standard personal computer in any robotics lab.

### **5.1.1 The rendering workflow**

The virtual world presented by the vision channel is the main platform for the human to experience and interact with. Without it, the human had to be guided by an expert to carry out tasks in a haptic-only case. Therefore, unavoidably, the visual rendering subsystem takes over and addresses inputs from all user inputs from the physical world into the virtual world, a major part of which has to be done in online when haptics is integrated.

There are mainly three sources of data in a haptic-enhanced VR that can trigger events and hence animations in the virtual world:

1. Motion trackers worn by the human user or other movable objects in the real world, which can change, depending on where they are mounted, the viewing angle (mounted on HMD), body posture (on the human body), or position/orientation of objects (on objects).
2. Haptic interfaces operated by the human user. The position/orientation of the haptic interface needs to be registered into the virtual world.
3. System generated events. This includes pre-recorded events/animations, as well as online-generated ones resulting from the interaction with human.

For immersive virtual reality, head tracker on the human is rather a standard requirement. Haptic interface is naturally involved when speaking of haptic-enhanced VR. Therefore all the above three sources almost always exist in similar VR systems. Since the tracking system, haptic interfaces, and the virtual world normally have different coordination systems, integration and calibration is necessary.

When implementing such systems, however, the workflow is not unique. With the coexistence of multiple choices for every segment of the system, alternatives are available for specific requirements. Here two typical and yet closely related workflows in realizing a VR system are given first, then the workflow employed in this work is introduced.

#### **Workflow without motion tracker**

In [30] a VR system is designed to evaluate human weight perception when lifting virtual boxes of different weights. Two haptic interfaces are used for the human to operate with both hands when lifting a virtual object in the virtual world.

The virtual world consists of an empty table surface, two hands, and a virtual box. No motion tracker is used, the angle of view is fixed. Here the user-generated event is the motion of human hands, which is handled by a customized detailed hand model, which will

be discussed later on. Human hand positions are calculated reversely by the end-effector positions of the haptic interfaces. System generated event is the position of the virtual box, which is also calculated based on the haptic interface measurements. The system also generates haptic feedback information according to the weight of the object, and displays it to the human via the haptic interface.

Hence here the visual rendering only takes care of displaying the two hands, which are readily generated by the hand model, and a virtual object, which is also calculated when addressing the haptic interface measurements into hand postures.

This is a typical example of a VR system built around the haptic subsystem. The visual world consists of minimal items, relying greatly on the haptic subsystem.

### **Workflow without haptics**

To the opposite of the first example, here a VR system without haptics is given. In [86] a virtual fire experiment is reported, where human subjects experience a fire burst out in a bar with multimodal feedbacks presented. Although everything was completely virtual, similar reactions were observed as if they were in real fire emergencies.

The VR of the virtual fire experiment is much more complex than that of the weight perception experiment. Vision, sound, and even heat were controlled by the system and presented to the human users. The user input is mainly through head tracker, and the system generates the rest of events such as fire spreading, people screaming and running around, and even sound and heat of the fire.

Pre-recorded animations and sound clips are played back in the system. The user can change the angle of view by turning the head around. No haptic interaction was allowed.

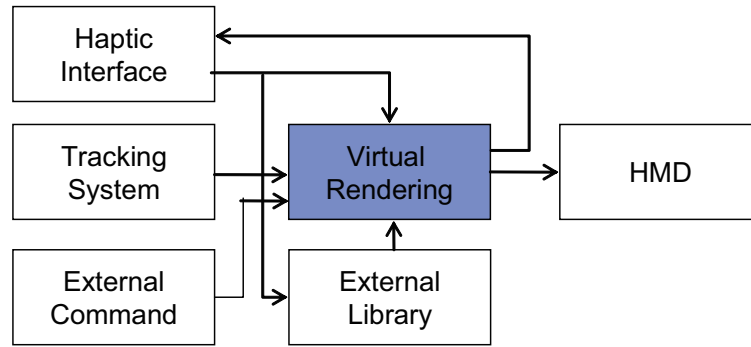
This is a typical example of a sophisticated VR system without haptics. Feature-rich and highly-detailed virtual events are available as pre-recorded clips played back online. The human is presented with all the system-generated feedbacks, being only able to control the viewing angle.

### **Workflow of this work**

The requirement of VR rendering of this work shares essence with both examples above: built around haptics, while requiring a good amount of complexity of system-generated events.

The diagram in Fig. 5.1 shows the structure of the VR system. With trackers and robot in the system, visual rendering needs to consider both inputs and generate animations online for both the virtual body of the human user and the virtual handshake partner. The online-generated animation needs to be of comparable standard with the rest parts of the system. On the other hand, the computational load needs to be controlled so that a standard lab PC can handle it within each cycle.

With this aim, the details of visual rendering and system integration/optimization are given next. To emphasize the line of thought, only main results are given for each segment. More details and discussions can be found in [56].



**Figure 5.1:** Structure of VR system.

### 5.1.2 Virtual human characters

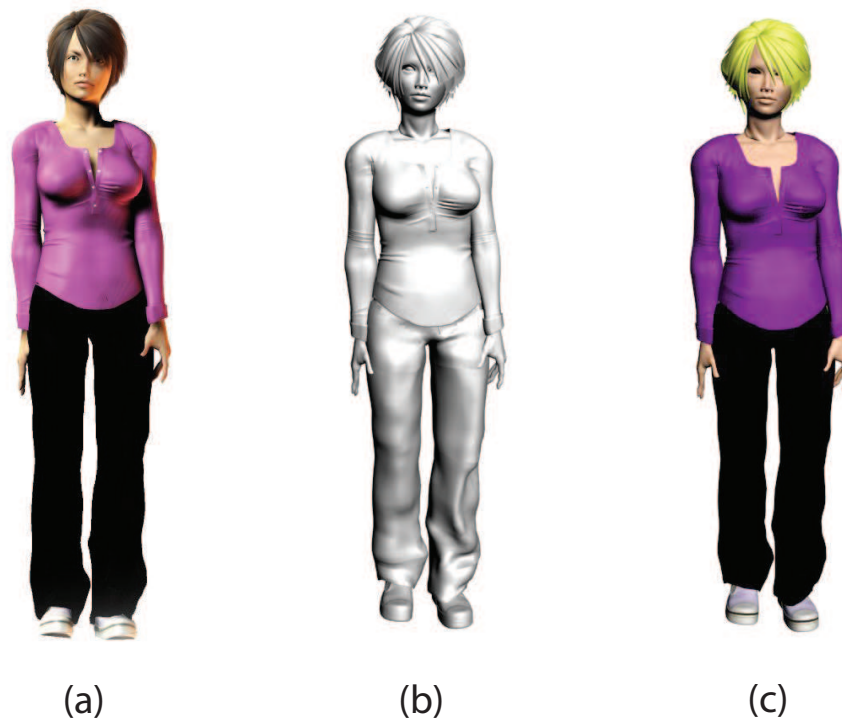
For VR applications, convincingly realistic human characters are the first step to implement. Luckily both ends of the problem are readily available: there is real-time-capable visual rendering software that can handle simple objects, and off-line rendering software that can generate highly realistic human character animations. However aiming at both ends, and without the requirement of super computational power, is a bit head-scratching. With real-time capability a hard constraint, compromises have to be made between fidelity and computational complexity.

OpenInventor (OI) [20] serves as the visual rendering environment, since its strength in real-time rendering of objects with simple geometric shapes, while being capable of importing complex models. The human characters are created in Poser [84], a popular and powerful software in creating high-fidelity human models and animations, but unfortunately offline only. In OI, an imported model can only be treated as a whole, with no further deformations possible. In order to realize full body animation, the human characters generated in Poser are first processed in 3DSMAX [7], where the lost color and texture information is restored, and later movable body parts are saved separately. The parts are then imported into OI as different objects and combined together in OI afterwards. Fig. 5.2 shows a female character generated in Poser, exported directly to OI, and processed via 3DS.

Here it is not difficult to recognize that a significant amount of information is lost due to the exporting between software platforms. Fortunately part of the lost information can be compensated by using extra textures in 3Ds. Fig. 5.3 shows a comparison between the initial virtual character and the one after decoration, significant improvements can be observed on the eye, eyebrow, lips, hair, etc. The undecorated character was used in evaluation experiment 3 in Section 7.4, and the decorated characters were used in experiment 4 in 7.5.

Another point to note is the virtual representation of the human participant. Since the focus of this task is on the virtual handshake partner, only a simple representation of the participant is developed, wearing a white coat and black shoes. Head is not designed for this virtual character, since it would always be behind the virtual camera and hence never visible. The hand of this virtual character will be very important due to the handshaking task. A high detailed virtual hand model is used for the right hand, the same as in the





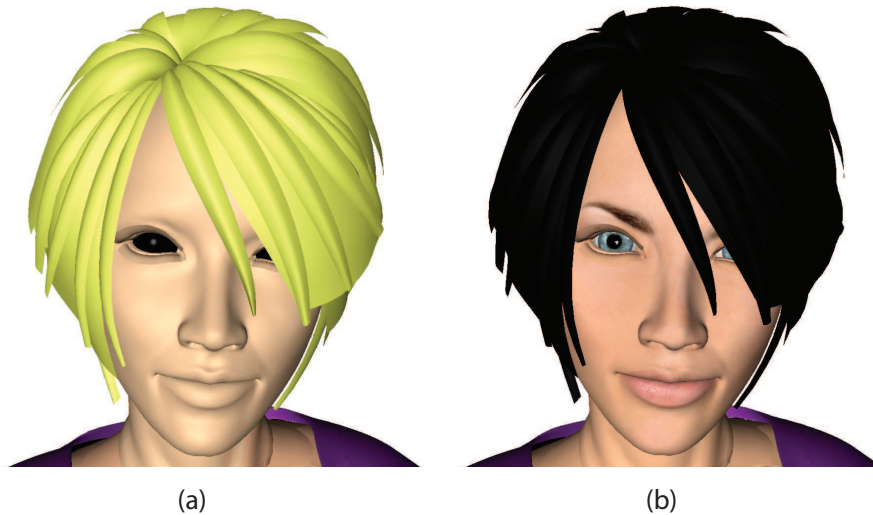
**Figure 5.2:** Comparison of virtual character details. (a) original model in Poser, (b) Poser directly exported to OI, (c) Poser to 3Ds then to OI.

virtual handshake partners.

### 5.1.3 Virtual hand model

For handshaking, as well as other hand-based haptic interaction tasks, it is important that the virtual hand is detail-rich, realistic, and capable of mapping the motions of the human hand. If the virtual hand motion is different from the ego-motion of the human, the plausibility of the VR will be affected. Since the human wears a data glove to measure hand gestures online, real-time capability of animation generation is essential to virtual rendering.

Here a 24DOF high-fidelity virtual hand model is used. It was developed by our project partner at UPC [19]. It is a OpenGL-based library that takes real-time finger-joint angle inputs and generates hand gestures that can be imported into OI for virtual rendering. The model has been cut to the wrist when provided by UPC, so that it replaces the right hand of the virtual character when loaded into OI. Fig. 5.4 shows the mapping results from a human hand wearing a data glove to the virtual hand model. In the evaluation experiments 3 and 4 where the hand model was used, almost all human participants reported that the virtual hand was *very realistic* and felt *like the real hand moving*.



**Figure 5.3:** Comparison of virtual character decorations. (a) undecorated character exported from Poser to 3Ds then to OI, (b) character of (a) after decoration in 3DS.

#### 5.1.4 Virtual environments

Two virtual environments are designed, for evaluation experiment 3 and 4, respectively.

The first environment, as used in Experiment 3 in Section 7.4, is very simple, consists of an empty space with a floor with black and white mosaic bricks. The more interesting part of this environment is how to register the robot end-effector position to the virtual human character hand position accurately. This will be discussed in system integration. To create a more realistic atmosphere, a single spot light was used to lighten the virtual world, as shown in Fig. 5.5.

The second virtual environment, as used in Experiment 4 in Section 7.5, is much more complex than the first one. Due to the requirement of the experiment, the virtual world resembles a cocktail bar. A number of virtual characters will approach the human participant to handshake, while the actual robot location is fixed in the physical world, which means the participant needs to remain still while the virtual characters come and leave, preferably in a plausible way. Once again computational complexity should be minimized. Fig. 5.7 shows the design plot of the virtual bar environment, about which the following remarks can be made:

1. A room is created of 1:1 size with decorations resembling a bar.
2. A table is placed right in front of the participant, such that the participant can reach the robot over the table. In accordance with the physical table, a virtual table of the same height is placed in the virtual world. The participant, as the bar tender, is hence constrained by this table, being able to see and shake hand, but not actually walk into the virtual world.
3. Two doors are placed on each side of the room, so that characters can enter and leave the room.



**Figure 5.4:** Gesture mapping from real human hand to virtual hand model.

4. To address the ambient noise of the bar, a number of additional virtual characters are placed in the room, near and far. To minimize system load, they are not movable.
5. To further increase the realistic atmosphere, lighting condition in the room is set to be dark, with rotating lights of time-varying colors.

### 5.1.5 Events

Apart from virtual characters and environments, the events that occur in the virtual world are also important to the overall plausibility.

There are three types of events regarding visual rendering in this context: walking, handshaking, and talking.

Walking of virtual characters is implemented in the simplest possible way, so as to minimize computation, and because of the table, it is very rare that the walking motion is even visible to the participant. The virtual character follows a pre-defined trajectory to walk from the door to in front of the participant, during which a pre-recorded walking animation is played back. After reaching the destination, the leg animation is switched to standing. After handshaking, the virtual character turns around and moves towards the other door to leave the room, with the same walking animation played back on the legs.

Handshaking is more complicated. During walking, the hand and arm play back a pre-recorded animation of swinging by the side of the body. After reaching the destination, the system switches from walking to handshaking-ready mode, where the right arm get raised to address the real location of the robot, and the arm position is now controlled by the actual position of the robot arm instead of the pre-recorded trajectory. Hence the virtual character is able to render the current handshake interaction. The hand motion of the virtual character is real-time mapped from the data glove worn by the human participant. After handshaking, the arm and hand switch back to walking mode again.



**Figure 5.5:** The first virtual environment, used in Experiment 3.

Talking is closely related to auditory rendering. Facial expression is different from arm or leg motion, it cannot be realized by changing the position of a fixed object. Unfortunately, as previously mentioned, the model imported into OI is treated as a fixed object, it is not possible to further change the facial expression after the head is imported into OI. Therefore several *key frames* of the talking animation are imported into OI, with everything else identical but the facial expressions. A program then switches among the key frames according to a certain time sequence to generate mouth motions online. Fig. 5.6 shows four key frames of a female character. More details are given in the next section.

## 5.2 Auditory rendering

The robots involved in this work are considerably noisy in operation. To cover the robot noise, a sound-isolating headphone is always worn by the participant. However for some cases, white noise or music clip is played. Here only the actual rendered sound conditions are covered.

There are two different auditory rendering approaches involved in this work. In the first approach, a sound clip is pre-recorded and played back in the experiment as a whole, being the timing of the system. In the second approach, one background sound clip is pre-recorded, with a number of other clips as add-ons to it, which are mixed-in online, triggered by other events.

The first approach is easy to implement. The sound clip is recorded before the experiment according to the plot of which event occurs at which time. During the experiment, the playback of the sound clip starts first, then everything else is operated by the timing



**Figure 5.6:** Four key frames of the talking animation.

of the sound. Experiment 2 used this approach.

The second approach involves synchronization with vision. The background noise of a bar is recorded. The dialogs of virtual characters are recorded separately. During the experiment, the background noise starts playing first. When the robot is ready, the experimenter who controls the events hits one key that triggers the talking mouth animation and the playback of a dialog at the same time. The mouth motion should fit with the dialog.

In order to achieve mouth synchronization with the four key-frames approach, Voice-O-Matic [98] is used to generate mouth motion information out of voice clips. Once this information is available, it is mapped to the 4 key-frame case, while keeping the timing information. For each dialog clip, such a sequence is created pre-experiment. An audio library FMOD [27] then reads-in this information and switches the key frames to generate the talking animation that is synchronized with the voice.

## 5.3 Conclusions

In this chapter, the workflow of creating a virtual scenario with both haptic input and human body motion tracking is presented. Comparing with the cases where either single input data is available, the development is more challenging in integrating different sources of information. Another contribution of this work is to render the virtual scene in real-time without the requirement of special computational power: the entire system can be

executed on a standard lab PC. In the mean time, a large portion of character details and kinematics are kept or restored in the final scene, providing high-fidelity rendering results.

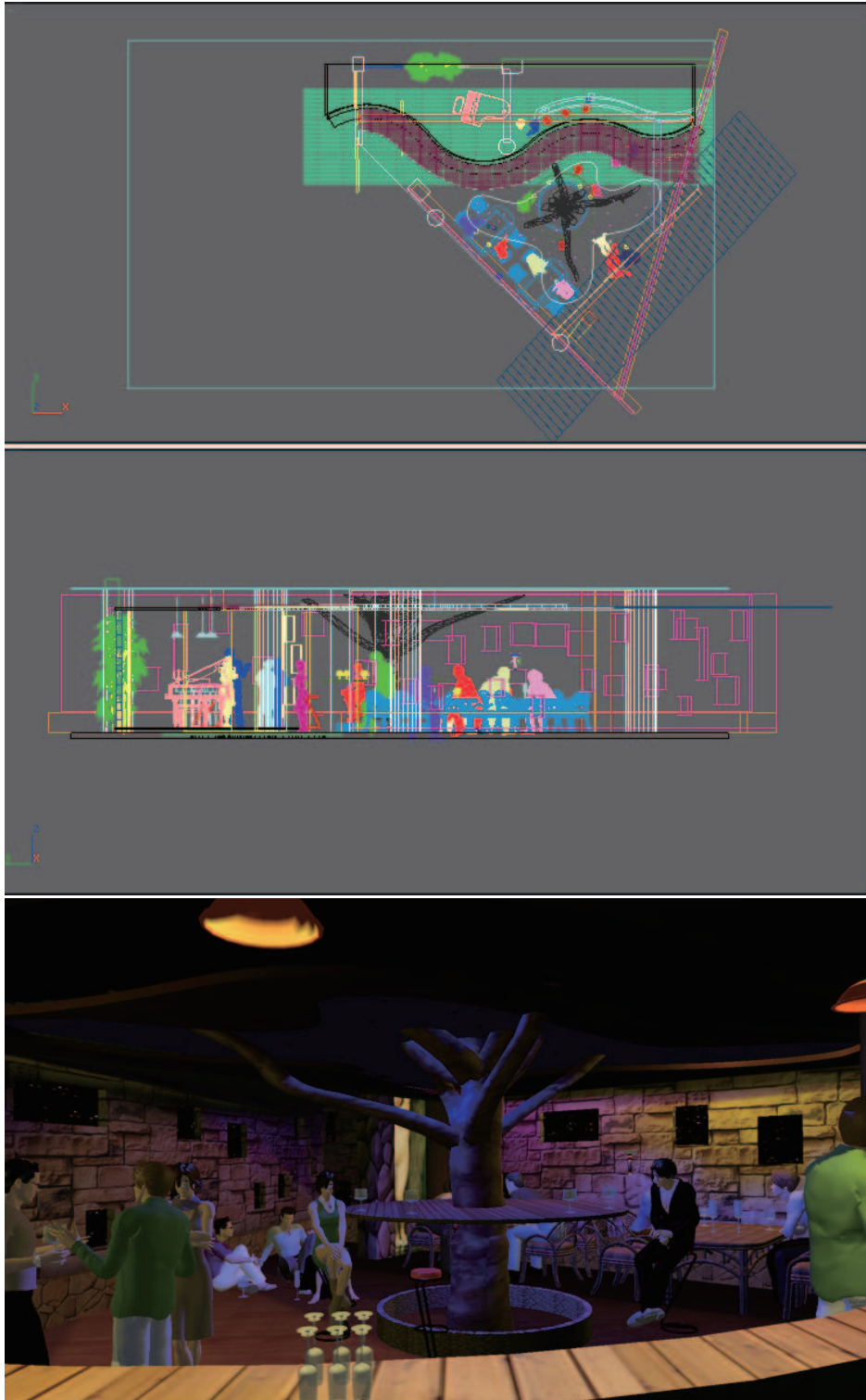
The integration of sound with vision is also covered. Pre-recorded voice clips are mixed with ambient noise of a cocktail bar, and synchronized with the mouth motion of the virtual characters. To minimize computational load, mouth motion is limited to four key frames. This could be improved by using more key frames of the same character to provide a smoother transaction of mouth motion.

Different stages of the system discussed in this chapter are used in the human evaluation experiments. 20 different characters were designed, of which 1 was used in evaluation experiment 3, and 18 were used in evaluation experiment 4. A unique recorded voice clip is assigned to each virtual character.

The system proves that a multimodal VR system consisting of haptic and tracking system inputs, highly detailed virtual characters, real-time full-body animation generation, and voice synchronization, is achievable. Moreover, no special computational power is required: the entire program mentioned above runs on a standard personal computer.

In the future the following directions could be investigated to improve the current drawbacks of the system: From Poser to OI, the restoration of lost kinematic and texture information is a forced action. The appearance of the virtual characters would be further improved if direct imported from Poser to OI with texture information preserved. The same statement is true for the kinematic data. Currently if miss-calibrated, different parts of the virtual character may detach. It would be more desirable that body parts of the virtual character are always linked together. This would require kinematic information from Poser to OI to be preserved.

Although developed for use with handshaking, the VR system can be easily applied to other applications. Necessary modification might be needed to address specific application needs. For instance, if used for HRI dancing, the left arm would also be linked to a robotic arm that provides online interaction data; the left hand of the virtual character would also need to be replaced by a highly detailed virtual model similar to the right hand.



**Figure 5.7:** Design plot and rendered result of the second virtual environment, used in Experiment 4. Upper: design plot top view, middle: design plot front view, lower: rendered result.

## 6 System integration and optimization

The haptic, vision, and sound subsystems of the handshake partner robot have been developed in the previous chapters. Adding the subparts into one system brings further problems. For instance, how to address the user difference, how to register virtual objects/events with real world objects/events, and how to optimize the system yielding best performance and minimum computational resource consumption.

This chapter covers a number of topics involved in integrating and optimizing the multimodal handshake system. Calibration processes are discussed to address user differences, with procedures proposed for HMM and data glove calibrations. Methods to register real world data/object into the virtual world are presented, with examples of incorporating a table and haptic interaction data of the hand into the VE. A second human is considered to handshake with the participant, different approaches are discussed, including a teleoperation approach. Finally optimization in terms of computational load and time delay is discussed, with conclusions drawn at the time. To maintain generality, each topic starts with the idea that applies to arbitrary multimodal HRI systems, then the specifics of this study are discussed in detail.

### 6.1 User dependency

Be it the calibration of a device, or setting the operational parameters of a controller, user dependency needs to be considered when designing technical systems wherever human users are involved.

User specifics are concerned by technical systems for two main reasons: either the functioning of the system relies on certain user specifics, or the desired performance/result of the system depends on the user in some way. For the former case, the system can only operate when adjusted according to the specification of the current user, a simple example of such case is using the head-mounted display, the diameter of which must be adjusted to the user in order for any virtual environment to be displayed. For the latter case, the system can be operational without considering the current user, however the user may not perceive the presented contents by the system in the intended way, unless proper adaptation according to the user is carried out. An example of such case is the data glove worn by the user in a virtual environment, unless calibrated properly according to the hand size of the current user, the glove cannot provide valid data for the system to register and render the hand motion of the user, hence the user can still move the hand, but may see the virtual hand moving in a different way.

Although both affecting the performance of the system, the priority of the two cases are clearly different. The *functional* related factors must be fulfilled for the system to run properly, while the *performance* related factors are negotiable: if certain remedies are available, it is possible to trade the complexity of calibration to slightly compromised



performance, since long calibration sessions could also result in fatigue and loss of interest.

In this work, user specifics are concerned of both cases in both haptic and vision subsystems. The remainder of this section will discuss each of them, starting from problem definition, then move on to solution and finally the implementation to the system.

### 6.1.1 User dependency in haptic subsystem

With a robotic haptic interface, the haptic subsystem involves direct physical interaction with the human user. Fortunately the impedance control algorithm in the low level controller can cope with user actions by providing compliant motion of the robot corresponding to the force input of human users. Therefore the system is operational without the need of calibration with every user.

However, one major concern of user dependency comes from the HMM module in the high level controller where human user intention information is estimated.

#### Problem definition

HMM is typically applied to robot learning applications where a reference signal is available to validate the HMM output ([16], [39], [85], [109]). For instance, in a peg-in-hole task, the performance of HMM-based controller can be clearly defined as if the peg is successfully placed in the hole. However, for the handshaking task of this study, to define an objective for the HMM performance is not straightforward. The actual intention information of the human is not directly measurable. This missing information will affect the entire HMM estimator, from acquiring training data to validating final output.

The HMMs used in the high level controller were trained priorly and then applied to the system. Because it is impractical to obtain the information of human intentions at each time instance, in order to get training data for both *active* and *passive* data, a small group (4 in this study) of participants were trained to consistently conduct handshake with a robot with either active or passive behavior. After a long training session and collecting a large number of measurements, the increased correction rate of the training results show the validity of assuming the trained participants could consistently perform a handshaking as instructed. The HMM implemented into the system achieved a correction rate of 90 percent using a foreign test data sequence other than the ones used in the training.

Next the following question needs to be answered: How to validate the trained HMMs with naive subjects in the evaluation experiments? Can the small number of trained participant who provided the training data for the HMMs represent the characteristics of arbitrary participants?

#### Proposed remedy

Before trying to answer the two questions, one thing to bare in mind is the limitation of not being able to measure the actual human intention at each time instance. Hence it is impractical to get an accurate reference to validate the intention estimation results directly by comparing the estimated intention with the actual intention at each time.

Here an indirect approach is proposed to tackle the above questions. The key is to obtain a reference for validating the estimation results. In order to obtain such reference, three steps can be taken. First, take certain trials of data measured by the same participant as test data, to validate the training result using the rest trials. Then take trials from a different trained participant to test the same training result. Finally, introduce naive subjects, instruct him/her to handshake following a consistent pattern, be it active or passive, and use the data to test the same training result.

The first and second steps have already been exercised when taking the HMM implemented in the system. The above mentioned validation result was achieved by using testing data measured from a different participant than the one used for training. When taking the third step, however, a dilemma was observed on the requirement of naive subjects: in order to record validation data, a participant must be instructed to act either actively or passively before the handshake, hence the actual handshake later on will no longer be a natural uninstructed one, as required by the general aim of the system. Moreover, as observed in the naive subjects validation experiment, without proper training, the participants may not act consistently during a handshake, hence the variation of estimation results are not affected solely by the performance of the HMM estimator, but also by how well the participant performed in always keeping active/passive.

Another observation made from the tests is the symbolization of the estimated HBPs has a large impact on the training results. With a same HBP sequence, defining a set of lower or higher threshold values results in a different symbol sequence, hence a different estimation result profile. For different participants, the thresholds for distinguishing *high* and *low* HBPs could be very different, for instance between a strong man and a small child. This factor is closely user dependent, and practical to implement as follows:

1. Before experiment, the participant is requested to handshake with the robot in a natural way as in real life. The control algorithm of the robot only provides a fixed reference trajectory with compliance to human input, in order that a handshake can proceed in both active and passive human intentions. A measurement is taken, while the participant is still not instructed on the actual handshake behavior other than being *natural*.
2. After the very first handshake, the measured HBP sequence is processed according to a certain law to calculate the threshold values for each HBP. For instance, 50 percent of the range between highest and lowest values, since HBPs can also be negative.
3. The thresholds can then be applied to the handshake system. The new thresholds are used to symbolize the HBPs into input sequences to the HMM estimator.

It is desirable to update only the HBP thresholds for each participant, while maintaining the same HMM parameter set, such that the participants are kept to be naive and uninstructed in their handshake behaviors. To verify this, 4 participants went through the above process. However, again, in order to get reference of their intentions, after the last step they went through the training for performing consistently active and passive handshakes. After the training, the correction rates of human intention estimation using a pre-trained HMM parameter set and updated HBP thresholds maintained above 90 percent for each participant.

## Discussions

The HMM training issue may affect the performance of the system, since if the HMM parameters are set improperly, estimated human intentions might always be *passive*, hence the robotic reference trajectory is never updated, hence the robot cannot adapt to human behaviors. Generally this issue can be compensated by updating the HMM and HBP symbolizing parameters for each participant by running a calibration process before the experiment. But in this study, the aim is to keep the participants uninstructed, hence any pre-experiment training is avoided. Fortunately only updating the HBP thresholds is proved to be competent in addressing the user dependency. As shown in the preliminary test results, the 90 percent correction rate is based on the comparison between the percentage of time length when the output of the HMM intention estimator is the same as the reference intention. Based on the assumption that the human is always active or passive as instructed, there is no validation if the human performed as instructed throughout the handshake. However, this result is at least as good as the correction rate when the measured data from the same participant from whom the training data is obtained.

On the other hand, in this study the role of the robot is the handshake partner to the human participant. What the human perceive is not the parameter itself, but the reactive behavior of the robot. In other words, the human perceives the effect of the estimation results. Therefore although not measurable directly, an error in human intention estimation will not cause the system to collapse, but result in a potential undesired behavior of the robot.

Consider a symbol sequence that will change the human intention estimation from passive to active. When the HBP thresholds are set higher, the require HBPs must also be higher to generate the same symbol sequence than with lower HBP thresholds. Hence the human will feel the robot to be less responsive in adapting to human changes, since the robot stays in *passive human* state longer than expected. In the opposite case if the thresholds are set lower than expected, the robot will become over-sensitive to human actions by entering *active human* state more easily. However this can be perceived by the human as partner characteristics instead of system error, since in real life some people are more firm and dominant than others in adapting to the intention of the handshake partner. As a matter of fact, deliberately tuning the HBP thresholds has the effect of introducing personality to the robot, in combination with tuning the parameters in the trajectory planner in the high level controller.

Based on the above discussion, in this study the HBP thresholds are set to the lowest among all experimentally obtained data sets, yielding a polite and compliant robot partner to human changes. Hence no pre-calibration is required regarding this issue before each handshake experiment.

### 6.1.2 User dependency in vision subsystem

The vision subsystem does not only render the visual scene to the human user, but also takes input signals from the human, as well as registers events from other modalities. Therefore, other than adjusting the diameter of the HMD, there are more user dependencies to consider.

### Problem definition

The main issue involved in the vision subsystem is the different body sizes of different participants. This involves the using of a number of devices that the participants have to wear when interacting with the system. For instance, in this study, the participant wears a data glove that measures the gesture data that can be rendered in the vision system so that the participant can see his/her own hand moving in the virtual scene. However in order for accurate angles of each hand joint to be measured online, an extensive calibration procedure has to be performed before using the glove.

The data glove can be customized in different sizes to match the hand sizes of different participants. However the sensors on finger joints only give raw data, which must be post-calculated to get angle information for each joint [21]. The coefficients of the calculation are user dependent, hence a calibration is needed for each participant.

If the glove is used uncalibrated, or the calibration is not done properly, the measured angle data can deviate severely from the true value, hence the rendered virtual hand will make wrong gestures than desired. Depending on application, this can cause various problems, ranging from mis-indication to breaking of presence feeling. In this study, since the application of the system is handshaking, if the rendered virtual hand cannot move consistently with the real human hand, the main function of the system will fail. Thus proper calibration of the glove is related to the functioning of the system.

### Proposed remedy

Existing calibration processes take two different approaches [44]. One approach is based on optimization. The aim of such algorithms are to obtain parameter sets that provide optimized finger-tip distances for manipulation tasks [35]. This involves closing the kinematic chain of the thumb and the opposite finger in consideration. The abduction and other DOFs of the thumb are therefore essential to the accuracy of the finger-tip distance. The optimization variables are increased because of the thumb. Given the multiple variable nature of the problem, such optimization processes are generally computationally heavy and hence time consuming. On the other hand, in order to gather the necessary input data for the optimization, the user has to follow a number of steps composing various hand gestures. The entire calibration process might take up to 20 minutes from the starting of the process to the optimization algorithm to converge.

Another approach is to take linear methods aiming for quick completion. However the accuracy might not be as good as the optimization approach, but the procedure can be significantly simplified ([46]).

If finger-tip distance is the main focus of the task, then the optimization process is desirable for better accuracy. The length of each experiment hence needs to be extended due to the glove calibration. However, in this study, the main function of the data glove is to provide real-time finger joint angle measurements to visualize the hand motion of the human user in the virtual world. Therefore finger-tip distance accuracy is not the main focus. On the other hand, quick calibration process and simple operations are more desired.

A quick calibration process has been proposed in [29]. The fundamental idea of the

proposed process is to simplify the calibration process by focusing on the finger angle optimization while leaving the complexity of the thumb abduction. After the simplification, an instant calculation of the parameters is desirable, while the user only need to carry out two simple hand postures, namely keeping the hand flat with fingers fully stretched on a plain surface, and making a full fist with the thumb touching the nail of the ring finger, as shown in Fig. 6.1. The process of the proposed calibration only takes 20 seconds, and the results are automatically incorporated into the system. During the experiment, if for certain reasons the glove slips or needs to be worn again by the user, a quick recalibration can be done conveniently without the need to reboot the rest part of the system.



**Figure 6.1:** Quick calibration procedure for the CyberGlove.

## Discussions

The proposed calibration process is much faster than the available one in operation, but compromising accuracy. For the application the results of the proposed calibration algorithm is adequate, since the visualization hand model already has limited DOF without thumb abduction. However depending on application, other calibration processes are desirable for different task specifications.

Glove calibration is one issue of user dependency in the vision subsystem. Other issues such as the adaptation to different user body heights and arm lengths when doing the reverse kinematics can be solved by pre-experimental measuring of user body profiles. For simplicity they are not discussed in detail here.

However it is worth pointing out the two approaches regarding user dependency in general: designing one system that works for all, or keeping possibilities to customize according to each individual cases. For one universal system, it is achievable by using the *average* value and enabling a safety margin for inter-user variances. The benefit is no pre-experiment calibration or measurements are needed, while paying the price of compromised system performance due to the difference in actual user profile and the assumption value

used in system design. The larger the inter-user variance, the higher risk of degraded system performance there is. The reverse statement is true for keeping parameters adjustable for individual cases: taking the time for pre-experiment calibration and measurement trades time and simplicity for improved system performance. For instance, it is possible to use one set of glove parameters for all users, provided that their hand sizes do not vary too much. But the more different the hand size, the higher possibility that the resulting measurements will deviate from the true angles. For different issues, it will be upon the wisdom of the designer which approach to take.

## 6.2 Real/virtual world integration

One of the key differences between haptic HRI and non-haptic HRI is that the virtual world and the real world at least emerge at the contact point where the user touches the haptic interface. With haptics taken into consideration, virtual world rendering is no longer self-contained, but needs to register online data and events from the real world.

In this study, a robotic arm is used as the haptic interface. Motion trackers are placed on the human body to provide real-time measurements of human motions. Therefore the virtual engine needs to consider and fuse the measured information with the visual rendering online.

Two examples are discussed in this section, the registration of real and virtual hand, and the use of a virtual table. The former topic addresses typical problems when incorporating online measured position data with virtual generated data, while the latter represents problems of matching a real world object into the virtual world.

### 6.2.1 Incorporating real world data

With a robotic arm as the haptic interface in the system, the online measured position and orientation information needs to be reflected into the virtual world as the position and orientation of the wrist of the virtual character during handshaking.

#### Problem definition

Three components of the system are involved in this procedure: the robotic arm, the motion tracker, and the virtual rendering engine. Around the three components, two major challenges are to be addressed, in order for the motion of the robotic arm to be incorporated into the virtual scenario:

- To map the position and orientation of the robotic arm end-effector from the robot coordination system to the coordination system of the virtual world.
- To generate animation for the arm of the virtual character according to the mapped robotic arm motion online.

There are three distinct coordination systems in the scope: that of the robot, of the motion tracking system, and of the virtual world. In addition, there is the camera view

point of the human user into the virtual world, which is assigned to be the position and orientation of the motion tracker placed on top of the head of the human user.

### Proposed remedy

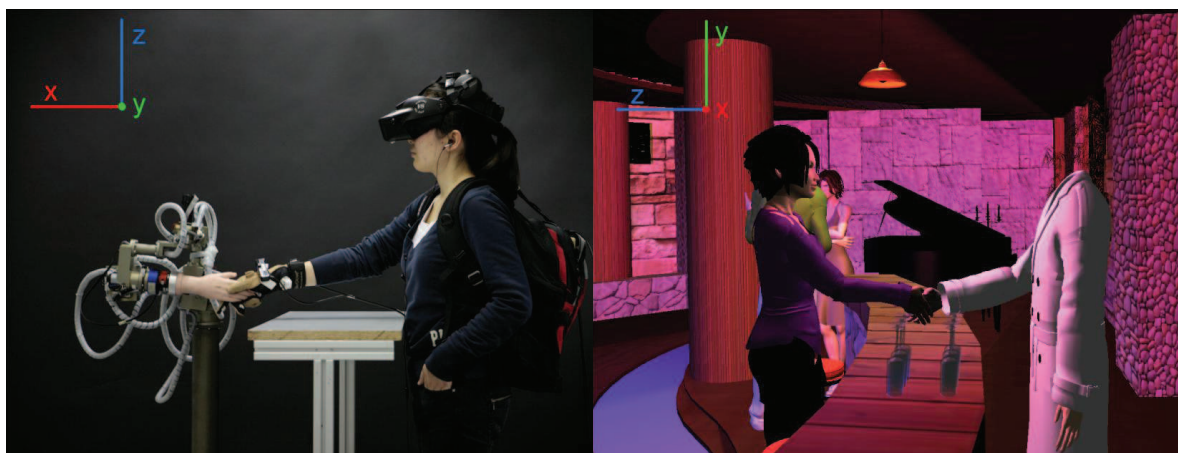
The robot platform is fixed to the ground, hence a measurable constant transformation matrix can be applied to map the robot coordinates to the coordination system of the motion tracking system.

On the other hand, the virtual world is designed to have the same size as the real world. After registering the origin of the virtual world to the origin of the tracking system, the virtual objects have the same scale and location as if in the real world.

The virtual character is then placed at the location of the robot end-effector. However, due to the different mechanical structure of the robot and the virtual character, the origin of the robot is the equilibrium point of the end-effector; while for the virtual character the origin is the center of two heels. Therefore placing the virtual character at the location where the two origins meet will result in mismatched positions for the virtual hand and the robotic end-effector.

Considering the aforementioned problem, an additional transformation is applied to the virtual character position. The hand location of the virtual character is first assigned to be the position of the robot end-effector. Then reverse kinematics is applied to the virtual character arm to get the location of the shoulder. The origin of the virtual character is then calculated according to the shoulder position and the body size.

This procedure ensures that the hand locations in the virtual world and the real world are identical, therefore where the human user sees a hand, a real physical hand can be felt when he/she reaches out, as shown in Fig. 6.2. Note the headless virtual representation of the participant.



**Figure 6.2:** Registration of real world data into the virtual world. The robot hand and the virtual hand in VE are registered in the same location during handshaking.

### Discussions

This idea can be generalized to arbitrary haptic interface, in the case that the virtual representation of the haptic device is different from its physical existence. In such applications, it is essential to register the contact point as the merging point of the virtual and real worlds, and adjust the rest of the involved system accordingly.

Depending on application, there might be additional difficulties induced by technical limitations. For instance, in this study the robot is fixed to the ground, hence coordination system mapping can be realized by applying a constant transformation matrix to the robot coordinates. However, if the location or orientation of the robot is changeable, for instance mobile platforms, an additional motion tracker needs to be placed at the origin of the robot coordination system to measure the location and orientation online. Under such situation, the mapping depends on the signal quality of the tracking system. In a prototype of the system developed in this study, the motion tracker approach has been applied. The resulting virtual character suffered from visible jumps caused by the noises in the position tracking data. Adding a dead-zone to the tracking data can eliminate the jumping problem, but results in jerky behaviors when the robot changes its location. Also when the tracking system fails to provide data, which may happen due to hardware performance issues, the virtual character will then suffer in finding its location. Therefore a stable position mapping method is necessary for maintaining accurate registration of real world data into the virtual world.

### 6.2.2 Registration of real world object

It is a common technique to mix real world objects with virtual rendered ones in augmented reality setups. Here a pure virtual reality setup is utilized, the discussion is about registering a virtual object in the virtual world to a real object in the real world. Comparing with incorporating real world data into the virtual world, registration of real world objects are more straightforward in procedure. However, it is the focus of this section that introducing an object in the virtual world that has its real world replica can serve multiple purposes.

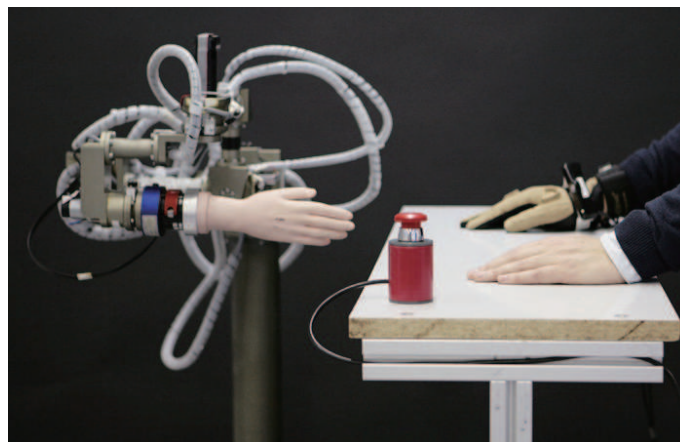
In this study, the participant has a restricted space on the platform during the experiment while he/she shakes hand with the robot. If the participant is placed in an open space in the virtual scenario, it is unavoidable that the participant tries to explore the environment by walking around. Even the participant gets instructed to remain still, it is a natural reaction to approach to the person approaching with a greeting handshake. Therefore a measure is taken to prevent the participant from leaving the platform and walking into the rest part of the virtual scene: the participant is assigned the identity of the bar tender of the cocktail party, a social scenario applied to the handshake experiment, and hence stays always behind the bar table. Each person approaching will stop by the table, and a handshake is carried out between the participant and the virtual character over the table.

The table has another function. The motion tracking system used in this study utilizes ultrasound. With the sensor bars mounted on the ceiling, the higher the marks are placed, the better and more robust the measurement results. Therefore the participant is requested



to place the right hand on the table whenever not engaged in a handshake, hence the motion tracker mounted on the right hand gets improved signal stability.

To register the table into the real world, a motion tracker can be mounted on the table to provide position/orientation data online. Then a virtual table can be rendered in the virtual world with the same size as the real one, and placed accordingly. Here the knowledge from incorporating real world data can be used, by keeping the edge of the table that the participant can touch identical to the virtual and real tables. The size of the actual table is not essential, provided that the part reachable by the participant has a sufficiently wide virtual representation. Fig. 6.3 shows the setup of the table. The participant is requested to place his/her hands on the table if not engaged in a handshake. The table constrains the participant from walking into the VE, while allowing handshakes with virtual characters. The virtual representation of the table is shown in Fig. 6.2.



**Figure 6.3:** Setup of the table. It constrains the participant from walking into the VE, while allowing handshakes with virtual characters.

This work gives an example about registering a real object to a virtual representation. In this study, the participant has two physical contact points with the virtual environment: the robotic end-effector and the table. The robot and the table are both fixed to the ground, with constant coordinates in the world coordinate system. In case that the object to be registered is movable, a motion tracker needs to be mounted onto the object, hence the registration will be prone to tracking signal quality.

### 6.3 Integration of a second human input

The system considers the scenario that a human participant shakes hand with a virtual character. However, it is necessary that in some cases the human participant shakes hand with a second human, in order to compare the performance with the virtual character case. While there are existing methods to integrate multi-user input signals in multi-user teleoperation studies, here the problem is more specific: the two users carry out one simple haptic task as opponents. Several prototypes have been developed to realize the second human input to the system. Finally a teleoperation approach is implemented into the

handshake system.

### 6.3.1 Problem definition

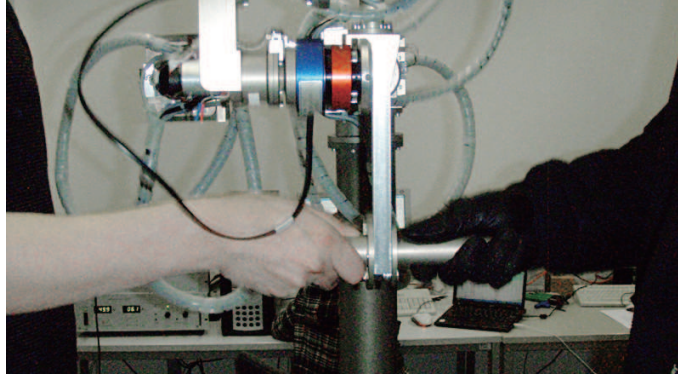
The aim of this task is to induce a second human input to the system to substitute the virtual character, and compare the human participant behavior when handshaking with the two.

Given the limitations of current technology, a computer mediated replica of a real world interaction task can only be a simplification. Therefore in order to compare one factor, the integration approach has to take measures to ensure that other factors of the second human and the virtual human are kept equal. For instance, in this study, the robot has a metal rod as the end-effector, hence the hand grasping part of handshaking is missing. In order to make a fair comparison, a second human should also conduct handshaking with the human participant via a similar metal rod, instead of making a direct hand-hand contact.

### 6.3.2 Proposed remedy

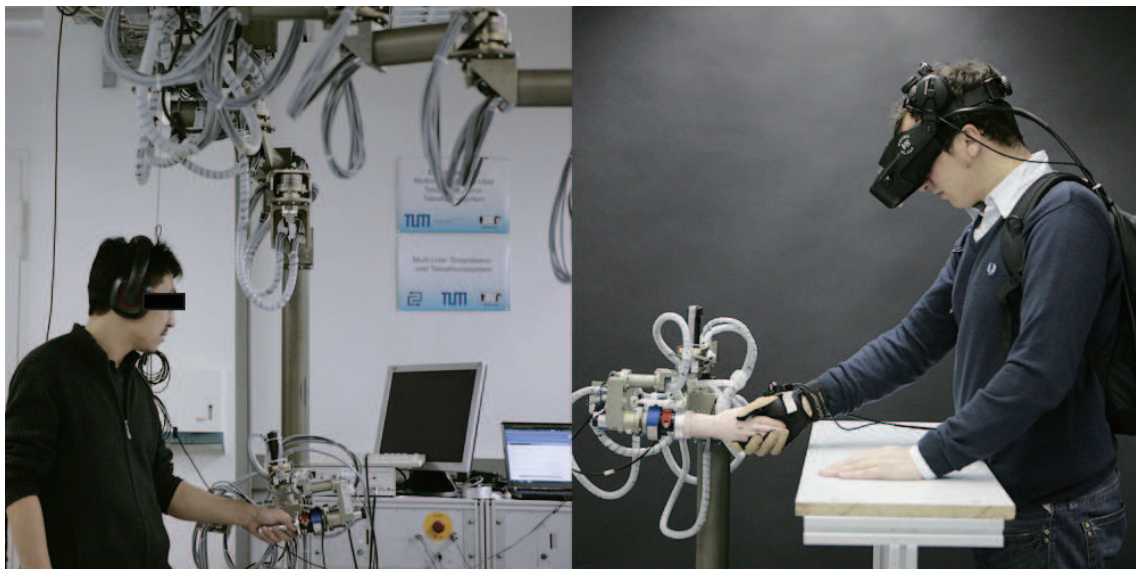
There are several candidate approaches to be considered, each introducing the second human input at a different stage of the process:

1. A natural consideration is to let the human participant interact with the second human directly, without any technical media. However this approach involves hand-hand contact, which is not a fair comparison given the robot has a metal rod as end-effector.
2. A modified approach to direct contact is to use a double-sided metal rod as a contact media to replicate the end-effector of the robot, with a motion tracking sensor and a force sensor mounted in between, such that position and force information can be measured. However in validation tests this approach shows another drawback. The robot has its technical limitations such as lower threshold of the displayable mass of the end-effector, motor noise, limited DOF, etc. Without connecting to the robot, the double-sided metal rod cannot introduce the limitations to the human-human interaction, hence making the comparison with the robot-human interaction unfair.
3. A further modification is to introduce the robot arm into play. The double-sided metal rod is mounted onto the end-effector of the robot. Fig. 6.4 shows the setup of this approach. The position and interaction force information is now measured by the encoders and force/torque sensor of the robot. Although the two humans are still holding the two ends of the rod, with the admittance controlled robot also connected, the interaction process is affected by the limitations of the robot.
4. The final concern of the process is that the force measurement during the interaction process. If one force sensor is used, as for the case of the double-sided bar, only one interaction force can be measured, it is not possible to distinguish between the forces applied by each side. To achieve separate force measurements, the two humans



**Figure 6.4:** A double-sided metal rod as robot end-effector to allow handshaking with two human partners.

are separated by interacting with two haptic interfaces remotely, and connected by teleoperation. With two robotic arms, each equipped with one force sensor, the force applied by either side can be accurately measured. Here universal virtual impedance is employed to govern the force/position relationship at both sides. Force measurements are sent to each other site to calculate the corresponding desired position for each robot [15]. Fig. 6.5 shows the teleoperation approach in operation. The expert (left) shakes hand with the participant (right) via haptic interfaces. Two ViSHaRD10 robotic arms are used in this case, one mounted on the ceiling, the other on the floor. Although other approaches of implementing teleoperation are possible, the chosen one best replicates the interaction with a double-sided bar, while maintaining system stability even if communication between the two sites fails.



**Figure 6.5:** Teleoperation approach to introduce another human into handshaking. The expert (left) shakes hand with the participant (right) via teleoperated haptic interfaces.

## Discussions

From interaction with direct contact to teleoperation via two haptic interfaces, the second human input is implemented from two entirely different approaches. The driving idea behind the change is to make a fair comparison between human-human to robot-human interactions. It is interesting to see how the human participants judge the second human input in comparison with the computer generated virtual partner in the evaluation experiments.

Different approaches of integrating the second human input have been implemented to the system and used in evaluation experiments. The double sided bar mounted on the robot has been used in [101], while the teleoperated human has been used in Experiment 4 in Section 7.5.

To generalize the study, direct contact is the most natural way of introducing a second human input, while teleoperation enables most possibilities for application. Be it in the same room, for more complete force measurements, or in a remote location, for geography restriction, properly implemented teleoperation can faithfully interconnect different sites of the system while providing similar functionalities as if they are locally connected.

## 6.4 Optimization

It is the fact for almost every integrated system that each subsystem does not fit perfectly together. And the performance of the overall system is subject to the worst performing subsystem. To improve and optimize the integrated system can be very much task specific and hence difficult to generalize.

Here in the last section before discussing about the evaluation of the developed systems, various topics of improving and optimizing the integrated system are covered, ranging from the extension of working DOF in the haptic modality to minimizing memory usage in vision.

### 6.4.1 Improving natural haptic interaction

Originally the arm motion for handshaking was modeled only along the vertical direction. Hence when implemented onto the robot, the other directions were fixed, with the human user only able to move the robot along the vertical translational DOF.

A human-human handshake is not subject to such constraint. To remove it, and hence improve natural interaction, the system needs to be extended from 1DOF to multiple DOF.

The force/torque sensor mounted on the robot end-effector is capable of measuring 6 DOF information, namely 3 translational forces and 3 rotational torques. All calculations are made in the Cartesian space. A virtual impedance model is introduced in each of the 5 DOFs, namely 2 other translational DOF and 3 rotational DOF. The impedance models enable the user to move the robot end-effector along that direction while maintaining stability. The robot is passive along the other 5 DOFs, with the reference trajectory generated by the virtual engine only applied to the vertical direction. Distinctive impedance parameters are set empirically for each DOF.

After extending 1 DOF to 6 DOF, the user no longer feels stiff walls along the other directions than the vertical one, but is free to drive the robot into them. The interaction hence becomes more natural than before.

For visual rendering, the new haptic system is still subject to the same data protocol. The 6 DOF motion of the robot is still within the same coordination system. Hence no adaptation in the vision subsystem is needed to address the change in the haptic subsystem.

### 6.4.2 Minimizing computational load

With the dropping of RAM prices, memory usage is becoming less and less of an issue to system design. On the other hand, specialized power-rich computers are commercially available for video rendering applications.

However, this study has a different vision of visual rendering: to develop a computationally efficient system with minimum requirement on the hardware platform.

A practical argument is, given an HMD with resolution of 1024\*768, or even 800\*600, it is a waste of computational power to render a virtual scene with high resolution textures, while reducing texture size can bring down memory usage significantly.

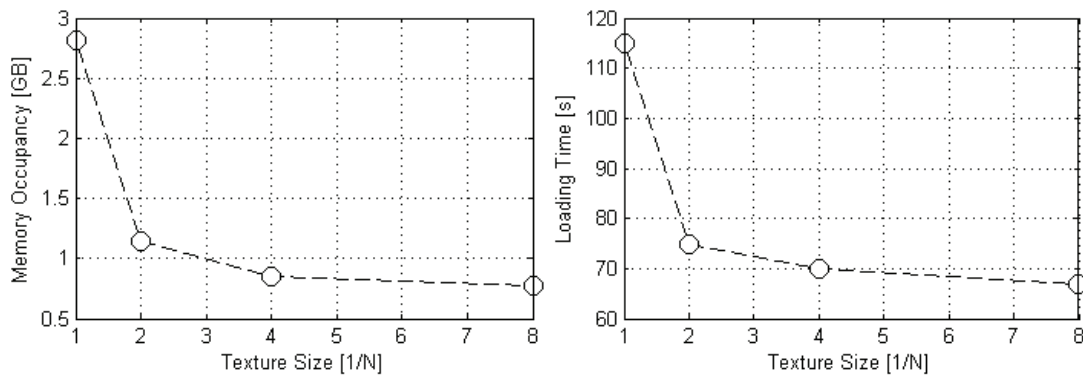
There are two main factors governing the memory usage of the virtual scene: the virtual environment and the virtual characters. The virtual environment has a rather steady memory usage, while the virtual characters can occupy a much larger share once the number of different characters involved in one experiment increases [56]. In one experiment, 18 different virtual characters are used, hence using uncompressed textures would cause a problem. Fig. 6.6 and Table. 6.1 shows the memory usage and loading time of the virtual scene consisting of the virtual environment and 18 characters. By reducing the texture size in half several times, both memory occupancy and initial loading time can be dramatically reduced. On the other hand, a clear turning point can be observed, therefore texture size of 1/4 the original size is chosen as the optimal solution. More details of the discussion can be found in [56].

Texture Size (1/N)	Memory Occupancy (MB)	Loading Time (s)
1	2.82	115
2	1.15	75
4	0.85	70
8	0.77	67

**Table 6.1:** Memory and loading time of 18 Virtual Characters

### 6.4.3 Time delay

The time delay from the moment the user makes a movement till the movement is rendered in the virtual world is concerned in this study, as it affects the experience of the user of the virtual system. The user motion is measured either by the haptic interface, or by the motion tracking system, then goes through pre-calculation and sent to the visual rendering machine. The visual rendering machine then modifies the virtual camera angle or virtual characters accordingly and displays the updated scene to the HMD. For the

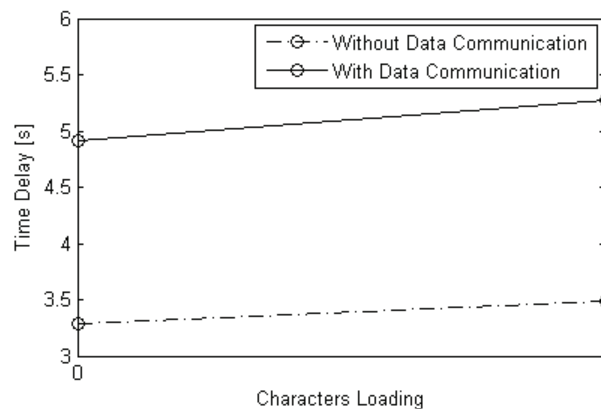


**Figure 6.6:** Memory and loading time of 18 Virtual Characters

haptic subsystem and the motion tracking system, this process is around millisecond level. The main cause of time delay is the vision subsystem.

To assess the problem, a simple test is carried out by quickly turning the head-mounted motion tracker by 360 degrees. The virtual camera is then supposed to turn also 360 degrees accordingly. This means re-rendering the entire virtual scene, since it is the approach of OI to only re-render the part of a scene when it is visible to the virtual camera. The time between tracker turning is finished and the virtual view finishes moving is measured for 10 times [56]. The averaged values are shown in Fig. 6.7. With or without loading the virtual characters does not make much difference, as not all virtual characters are visible in the virtual scene at the same time, and invisible characters do not affect in re-rendering. However with or without data communication with the haptic interface makes a significant difference in time delay measures. Reducing the data updating rate does not help much in reducing the time delay.

Even without data communication with the haptic interface, the smallest time delay is still more than 3 seconds, a level clearly noticeable to the human user. Luckily in this application the user focuses more to the front direction, without turning the head very often. With the head tracker fixed and moving the robotic arm, the rendering of the virtual character arm movement introduces time delay that is hardly noticeable.



**Figure 6.7:** Time delay of visual rendering in different conditions.

## 6.5 Conclusions

This chapter covered a wide range of topics concerning the integration and optimization of the multimodal system. A shared goal of the topics is to improve the overall performance of the integrated system by adjusting each individual subsystem according to the requirements of other subsystems.

Under the shared goal, user-specific HMM parameters and symbolizing thresholds for HBP parameters are introduced into the system; a quick calibration process is proposed for the data glove to address different hand sizes of human users; methods of registering real world object and data into the virtual world are discussed, with two examples of the hand and table; approaches to integrate a second human input are presented, with the teleoperation approach implemented into the system; and finally several task specific topics to optimize the system are presented, such as extending the operational DOF of the robot, minimizing memory and computational loads, reducing and balancing the time delay of different modalities.

Although the ideas and aims are kept general, due to the task specific nature of most topics in this chapter, the approaches and results are not directly transferable to other applications. However two general remarks can be made from this study: it is the worst performing subsystem that constrains the overall performance, be it about time delay, resource consumption, or even multi-DOF haptic interaction; if necessary, break down the encapsulated available procedure to yield simplified, efficient novel approach that better suits the specific application, as shown by the glove calibration process.

The integrated system is robust enough for dozens of participants to conduct continues handshake experiments lasting for days, while providing results as expected. The details of the experiments will be discussed in next chapter. On the other hand, several future directions about system integration can be given here. A reference for HMM training is not available till now. In the future, if physiological measurements can be employed, it is possible to search for a reliable reference data for HMM training. The motion tracking system is prone to noise, hence not suitable as the data provider for coordination system mapping. State-of-the-art computer vision systems employ various techniques for object mapping, future work could study methods for better methods or devices to map real world objects/data into the virtual world. The system is extended to 6 DOF interactions, with only 1 DOF having reference trajectory, the rest 5 DOF passive. In the future works could be done to extend the virtual handshake partner model to cover the other DOFs.

## 7 Experiments and evaluation studies

There is no better way to examine the performance of a human-robot interaction system other than inviting human participants to try it out and collect their opinions and monitor their behaviors. In this chapter, 4 experiments with human participants are carried out. The aim of the experiments is to assess the performance of the developed systems, in terms of whether the users feel that the handshaking experience with the system is plausible, or close to a handshake in real life. Further more, the experiments are to tackle the question that whether different realizations of the system, such as integrating multimodal feedback signals or using different rendering algorithms, can affect the perceived plausibility of the user.

Starting with the discussion of *plausibility* and the use of questionnaires to assess user experience in Section 7.1, each experiment will be introduced in each of the following sections. The systems involved in each experiment employ different versions of subsystems as well as different hardware devices described in the previous chapters, which will be clearly marked. Questionnaires were used to collect the opinion of the participants in all experiments. Statistical analysis results are given in each section, while a general discussion with conclusions is given in the end.

The experiments reported in this chapter are among the first to adopt the presence questionnaire ([96], [86]) to measure user responses to haptic VR systems. Experiment 2 and 4 were conducted in collaboration with the research group of Prof. Slater in UPC, Barcelona, Spain. The lessons learned from the experiments would benefit future studies.

### 7.1 Plausibility and questionnaire

Adding haptics enables physicality in VR systems. However the effect can be double-faced: the system is capable of rendering new tasks with haptics, but in case of poor haptics or even different parts of the system conflict, as observed in the experiments, the user experience may be degraded.

Plausibility, as in the context of HRI [83], refers to the quality that the entities and events of a virtual environment are experienced by the human user to be *reasonable* and in a favorable way: the former means the user believes the entity is actually there, the event is actually happening, while the latter means the entity and event cope with the priori knowledge of the user on their real-life representatives. For handshaking, reasonable means that although being in a lab and holding on to a robotic interface, the user should have the illusion of performing a handshake with a virtual person in the VR; favorable means the virtual handshaking dynamics is similar to a handshake that the user normally carries out with another human in real world. The term plausibility is derived from the term *presence*, meaning the illusion of being in an environment depicted by the virtual reality displays ([76], [77], [10]).



To assess plausibility, on the other hand, is not a trivial issue from an engineering approach. Recall the definition of *goal-oriented* and *process-oriented* human haptic skills in Section 2.2. For goal-oriented tasks ([1], [4], [28]), there is the Boolean value of *task completion*. Statistics can be applied to the collected Boolean values to obtain performance index for the developed system. For some process-oriented tasks, there are standard behaviors available as references to measure system performance, such as dancing steps for a dance-partner robot [49], and reference character for a writing robot [85]. Algorithms based on measured behavioral signals can then be defined to assess system performance comparing with the reference. However, for process-oriented haptic interaction tasks of social interaction, such as handshaking, a standard behavior can no longer be defined. Traditional methodology in engineering of measuring system performance therefore no longer applies.

Here questionnaires are used to measure the user responses to interaction plausibility of the VR systems. Questionnaires have been applied to measuring the degree of presence during experiments with VR systems ([105], [52], [79]). However there are concerns of questionnaires being used as the only assessment ([80], [81]). The questionnaires used in the following experiments are based on the questionnaires used in [86]. Addressing the concerns in [80] and [81], behavioral data such as position and force trajectories are also recorded for each experimental trial for further studies.

Four experiments were carried out focusing on evaluating plausibility. An overview is given here to highlight their differences:

- Experiment 1: Haptics only. Comparing three different control algorithms in haptic rendering on which one is the favorite to the human participants.
- Experiment 2: Haptics and sound. Comparing different conditions of haptics and sound, to see which composition is more plausible.
- Experiment 3: Haptics and vision. Comparing the plausibility of different conditions in vision and haptics to study their influences.
- Experiment 4: Haptics, vision, and sound. Comparing different conditions in haptics, to see if the plausibility of each condition is still differentiable when multi-modality signals are presented.

## 7.2 Experiment 1: Robotic control algorithms comparison

The aim of this experiment is to select the most suitable haptic rendering algorithms for the handshaking robot. HRI experiments are carried out to evaluate candidate controllers. The only sensory modality involved in this experiment is haptics, while vision and sound channels of the participant are blocked.

To achieve plausible interaction, a number of challenges need to be addressed in haptic rendering. For the controller design, a crucial issue is to select the fundamental structure of the controller, starting with the reference signal. The two candidate signals are position and force. Works are reported on the relation of position and force cues and their roles in human haptic perception when performing tactile exploration tasks. The authors of [70]

proposed that force can overcome object geometry in the perception of shape when the user actively explores a surface, the study was revisited in [24] with the finding that the weighting of importance for position and force cues is task-dependent. In [18] experimental results show that non-constant force degrades the perception of virtual surfaces.

While HRI handshaking is a kinesthetic task different from tactile exploration, here the plausible objective is proposed as a hypothesis that the smoothness of position trajectory is more important than of force in obtaining plausibility, provided that no other factor breaks plausibility.

### 7.2.1 Haptic rendering algorithms for HLC controller

In Section 2.4, several haptic rendering algorithms are proposed, with a resulting robot that is able to perform handshakes under a constraint that the human follows the lead of the robot. However in Section 3.2, the HLC controller is already based on the structure of position-based admittance control. This experiment fills the gap between the two, providing supporting materials of how the low-level controller approach is selected by human evaluation results.

#### Three algorithms in comparison

In this experiment, three candidate control algorithms are compared to see which is the best fit for HLC controller design. Recall the controllers proposed in Section 2.4. Position control (PC, Fig. 2.7, position-based admittance control: variant 1 (PAC1, Fig. 2.8), and position-based admittance control: variant 2 (PAC2, Fig. 2.9) are selected for comparison.

#### The comparison of position or force signal as reference trajectory

As shown in [103], all control algorithms implemented in this work need reference trajectories. The two candidate solutions are using position or force signals. However, in order to compare the two solutions, the reference trajectories in use should be comparable: in the ideal situation, the measured interaction position signal of the force-referenced controller should be identical to the position reference, and vice versa. In other words, the position and force references should be in pair, which turns out to be a nontrivial task.

For position reference, the optical tracking system used in the human-human tests gives results with 1mm resolution. However, measuring interaction forces between the robot and the human cannot provide the force reference that drives the robot. The human desired force is contained in the human mind and hence cannot be measured directly.

A series of experiments were carried out to acquire matching force and position reference signals. Two persons are involved in the experiments, referred to as the *participant* and the *expert*. In each experiment, a different participant handshakes with the same expert. The expert plays a more important role in each experiment as he/she needs to repeat the handshakes as closely as possible. The participant is instructed to handshake passively in each trial by following the lead of the expert in the most natural way. Each participant goes through the following procedure with the same expert:

1. First handshake: The Participant is led by the expert with positions recorded by the optical tracking system.
2. The measured position trajectory is implemented onto the robot using position control.
3. Second handshake: The participant is led by the robot playing back the recorded position trajectory.
4. The (constant) impedance model of the participant during the handshake is estimated from the position measured from the tracking system and force measured from the force/torque sensor on the robot.

By assuming similar behavior (impedance parameters) of the participant over several handshakes, the force signal needed to generate such a handshake can be calculated. This resulting force signal is used as the reference trajectory for the controller that matches the position reference. Using this approach, on the one hand assumptions on the human impedance parameters while passively handshaking are made, but on the other hand the approach only requires a known position reference and a virtual impedance model of the human arm. Measuring the desired force in the human mind is not required.

Next, an HRI experiment is reported to assess the performance of PC, PAC1, and PAC2. Subjective evaluations are measured, analyzed, and compared.

### 7.2.2 The experiment and results

A validation test was carried out to compare the performance of the three different control algorithms and to study which factors are considered by humans to be the main difference between a human handshake and a computer-generated one.

#### Experimental design

The experiment was carried out on the robotic arm ViSHaRD10 with a metal rod as the robot end-effector. The system was restricted to vertical translational motion. 16 participants took part in the experiment. They were all right handed, 8 male and 8 female, with an average age of 24.7. Only haptic feedback was provided to the human participant. They were blind-folded and wearing a sound-proof headphone playing noise loud enough to cover ambient sounds. A picture of the scene is shown in Fig. 7.1. Because of the blind-folding, an experimenter guided the participants in and out of the experimental room, as well as placed the hand onto the robot end-effector before each handshake.

Each participant was requested to perform a training session and the main experiment consisting of 8 handshakes. The 4 different control algorithms were repeated twice in a random order. In all handshakes the participant was asked to follow passively the lead of the robot. They were not aware of the number or the order of different algorithms.

First each participant was given an unlimited number of trials to shake hand with the robot to get familiar with the system. The algorithm implemented during training is a simple sinusoidal trajectory. After each handshake, the participant was asked to provide a



**Figure 7.1:** HRI experiment: a blind-folded subject wearing sound-prove headphone shakes hand with the robot.

**Table 7.1:** Bonferroni adjusted post-hoc tests for the differences between the three controllers (significant differences on a 5% level are marked with \*)

Condition	PC	PAC1	PAC2
PC	-	0.766*	0.016
PAC1	-0.766*	-	-0.750*
PAC2	-0.016	0.750*	-

score ranging from 1 to 5 (1 indicates the handshake feels completely machine-like, while 5 indicates the handshake feels plausible, i.e. human-like). After the experiment each participant was lead to another quiet room to fill in a questionnaire.

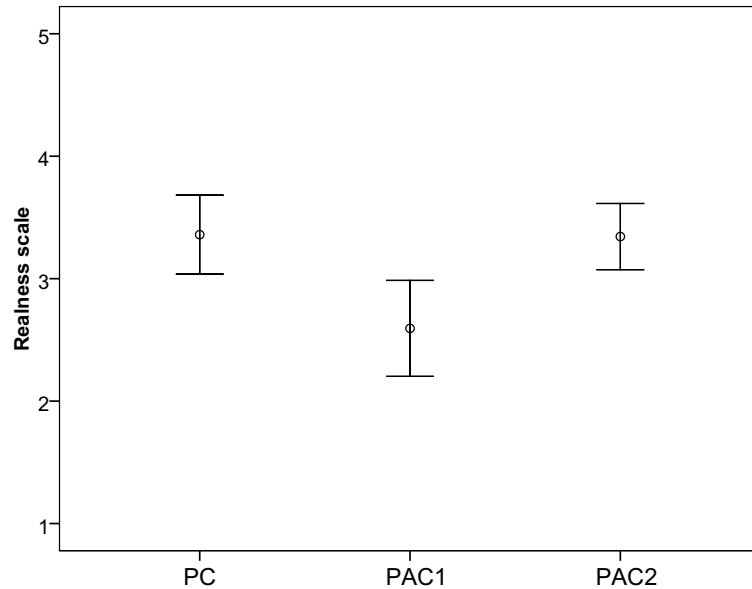
## Experimental results

The score for each condition is shown in Fig. 7.2. A one-factorial repeated measurement ANOVA was conducted to test for a difference between the tested control algorithms on the 5 point scale. A significant effect of this factor was found:  $F(1.802, 27.026)=8.422$ ;  $p<0.002$ ; partial  $\eta^2=0.360$ , Greenhouse-Geisser corrected. Thus, the Bonferroni-adjusted pairwise comparisons of the three analyzed conditions are reported, see Table 7.1. PC and PAC2 are both rated significantly better than the force-reference-based PAC1. No significant difference between PC and PAC2 was found. This indicates that position-reference-based algorithms are generally preferred to the force-reference-based algorithm.

### 7.2.3 Analysis and discussions

PC and PAC2 are based on position references. The differences in the rating of PC and PAC2 with respect to PAC1 show that position-reference-based algorithms are preferred.

One possible reason for this is that the force-reference-based algorithm PAC1 cannot provide the same level of smoothness in the position trajectory than the position reference

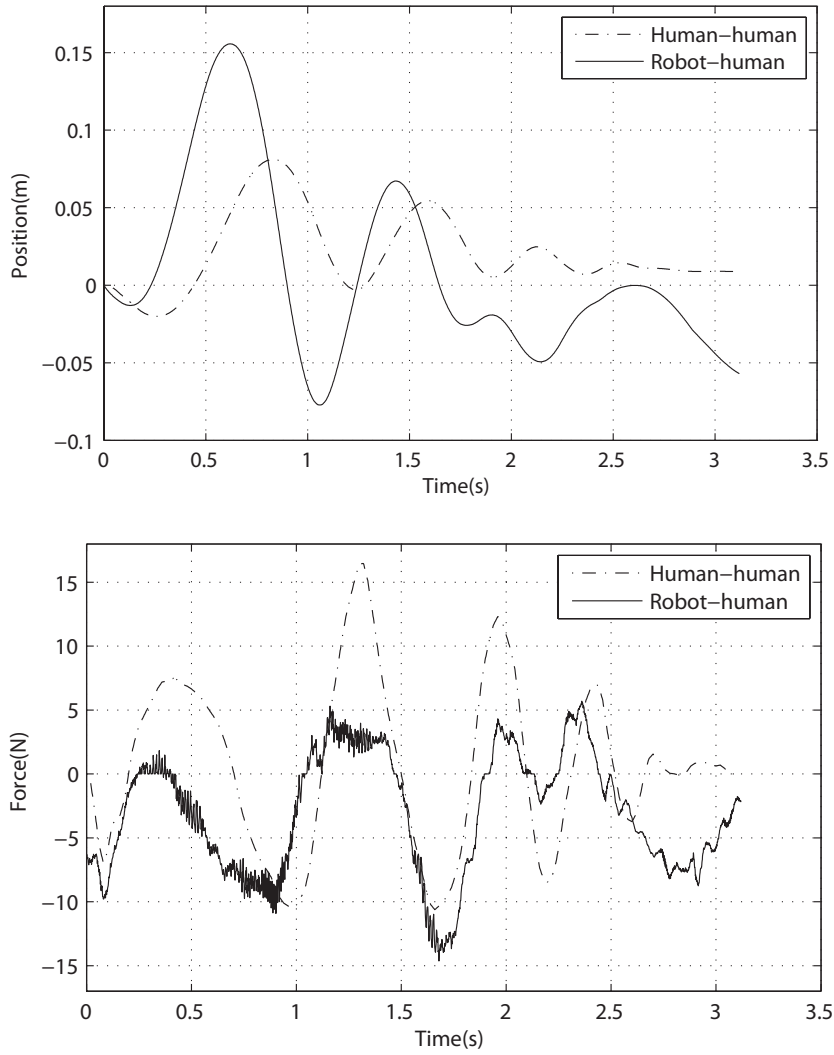


**Figure 7.2:** Subjective evaluation results of the passive condition showing mean values per participant on the 5 point plausibility score. Score values ranging from 1-5, higher values indicate higher feeling of realness.

based algorithms. The reference position trajectory used in the experiments are in pair with the force reference used in PAC1, provided that the human performs similarly as in the experiments where the reference trajectories are measured. Since the performance of each participant differs, the measured position trajectory in PAC1 is often deviating from the matching reference. The deviation often happens at the turning points of the position trajectory, when the controller tries to maintain the force reference trajectory, resulting in sudden turnings in the actual position. An example of experimental results of PAC1 when the human performance deviates from expected is shown in Fig. 7.3. In general the measured force followed the desired force, but the deviations resulted in the large changes in position trying to maintain the force profile. The resulted large amplitudes of shaking and the non-oscillatory motions after 1.7s were both unnatural to the human.

Note that for position-reference-based algorithms, especially for PC where force is not considered at all, the controller tries to maintain the position reference trajectory and hence results in sudden turnings in force when the human fails to follow perfectly. However, according to the evaluation results, people tend to focus more on the smoothness in position rather than force, as PC gets the highest ratings despite the force turnings.

Another possible reason is that, the participants, already knowing the partner being a robot, expect the handshake to be firm. Participants can best distinguish PC because of the firm leading, while being uncertain with the other algorithms. Similarly for PAC3, with the increased stiffness when the actual position differs from the reference position, the algorithm feels also firm and stiff if the participant fails to follow the lead of the robot. PAC1 and PAC2, on the other hand, incorporate a set of much lower impedance parameters and therefore feel much more compliant to the human.



**Figure 7.3:** Performance comparison: PAC1. Upper: Position; Lower: Force. Position trajectory driven away by unexpected human performance

### Improvement to PAC1

The current impedance parameters of  $Y_R$  for PAC1 are constants. Here a more in-depth discussion is given on PAC1, if the comparison between PAC1 and PAC2 is fair, and if there will be improvements when introducing time-varying parameters in  $Y_R$  of PAC1.

First, for a position controlled robot, using force reference is fundamentally different from using position reference. Starting with the PAC algorithm. Deriving the relationship between the measured signal  $x$ , the error  $e_x$  and the reference signal  $x_d$  yields:

$$x = \frac{1}{1 + Y_R Z_H} x_d, \quad (7.1)$$

$$e_x = x_d - x = \frac{Y_R Z_H}{1 + Y_R Z_H} x_d. \quad (7.2)$$

For PAC2, as in Fig. 2.8, deriving the relationship between the measured signal  $f$ , the

error  $e_f$  and the reference signal  $f_d$  yields:

$$f = \frac{Z_H Y_R}{1 + Z_H Y_R} f_d, \quad (7.3)$$

$$e_f = f_d - f = \frac{1}{1 + Z_H Y_R} f_d. \quad (7.4)$$

For PAC, according to (7.2), perfect tracking with any  $Z_H$  value is possible with zero  $Y_R$ , which is exactly the situation for PC. However for PAC1, according to (7.4) perfect tracking of  $f_d$  can only be achieved with infinite  $Z_H Y_R$ , which is practically impossible.

More over, introducing time-varying  $Y_R$  in (7.1,7.2) properly, as in PAC2, can limit the effect of the human uncertainty in  $Z_H$ , yielding small  $e_x$ . For PAC1, however,  $Y_R$  appears in the numerator in (7.3), a time-varying  $Y_R$  will hence further degrade the tracking performance in case of human perturbation. The analysis results support the findings of the HRI experiments.

### Factors that break presence

More information can be obtained from the post-experiment questionnaires, where each participant was asked about the reason of their preferences on the algorithms. Many of them stated that although the system generally felt fine, here and there some factors went wrong, which ruined the whole realness of the handshake. The most popular reasons include: too firm, too heavy, and first peak too large. Personal preference is also important, for example some people prefer shaking gently, while others like strong interactions. All these factors have higher importance level than trajectory error, as violating one of them breaks the experience, regardless of the actual trajectory.

In the implementation of an HRI interaction task, there are factors that do not give direct measure, but realness will break if any of them is not fulfilled. They serve as priori conditions for realness. For the handshaking task, such factors include triggering of the process, the restricted direction of motion, motor noise, the missing gripping dynamics, and many others. Other factors, on the other hand, give actual realness measure, hence determine the measured results. For handshaking such factors are not restricted to the stiffness of the handshake and the tracking errors, the underlying relationships are yet to be identified in future experiments.

At the current level of haptic technology, the technical systems only recreate a subset of the real world. However, it is already encouraging to see the reports by the participants on how much they felt like shaking a real person via the metal stick in the haptic-only HRI experiments. Most importantly, the study of suitable control algorithms and realness definition gives a valuable hint of the future development of more integrated haptic HRI systems.

### Remedy for active human

All above discussions are made based on the assumption that the human follows the lead of the robot passively. When the human tries to be active and to lead the robot, the models of the algorithms will no longer be valid, leading to degraded results in realistic ratings.

Addressing the problem in the active condition, studies have been carried out, see [102] where a high level controller is developed to generate reference trajectories online in the active case to adapt to the human actions. As PC does not consider human input, PAC2 is selected as the suitable algorithm for future developments of the interactive controller that could deal with both passive and active conditions. On the other hand, PC with selected trajectory reference signals is chosen to be the *basic robot condition*, where human input is deliberately ignored, in further evaluation experiments.

### 7.2.4 Conclusions

This experiment asks human participants about their preferences of three different haptic rendering algorithms in terms of plausibility. It is interesting to observe that although with haptics only, and even the haptic channel is nearly as restricted as possible (with a metal rod as the contact point, only 1 DOF motion), the results still showed statistical significances in some pairs. It is also helpful by looking into the comments of the questionnaire, on the actual reason of why the participants made their choices, as shown in Section 7.2.3.

As a result, the two most preferred algorithms are chosen for further studies: PC is selected as the basis of the *basic robot condition*, while PAC1 the basis of the *interactive controller* in future experiments.

## 7.3 Experiment 2: Evaluation with haptics and audition

This experiment evaluates the performances of difference haptic rendering algorithms in terms of plausibility/realness, with a simulated social scenario of different quality conditions in sound. The experiment was conducted in collaboration with Prof. Slater's research group in UPC [33].

For the evaluation a 'ground truth' approach is introduced, where the robot handshakes are compared with handshakes operated by a human via the same metal rod. For this, an experiment was carried out where the participants entered a virtual environment, i.e. a virtual cocktail party, and were asked to perform a number of handshakes, either with the robot operating with one of two control algorithms operating the metal rod - a basic one for comparison or the proposed new advanced one, or with a human operating the metal rod. The virtual environment was represented only through audio and haptics, without any visual representation, i.e. the subjects participated blindfolded. The evaluation of each handshake was achieved through the subjective scoring of each of the handshakes. The results of the study show that the demonstrator operating with the proposed new control scheme was evaluated significantly more human-like than with the demonstrator operating with the basic algorithm, and also that the real human handshake was evaluated more like a real human handshake than both types of robot handshakes. Although the difference between the interactive robot and human handshake was significant, the effect sizes are not very different, indicating substantial confusion of participants between the interactive robot and human operated handshakes.



### 7.3.1 Introduction

In this experiment the handshakes were mediated through a metal rod that was held at one end by the experimental participant with the other end attached to the robot, and manipulated either by the robot itself or by one of the experimenters moving it directly. After each handshake the participant was asked to infer whether they had just interacted with a person or with a robot, based solely on the haptic responses since they were blindfolded throughout. The purpose was to evaluate the degree of similarity of the robot handshake with a human one. The robot handshakes were controlled either by a basic robot without any attempt to model the true haptics involved in a handshake, or a more interactive robot that embodied our best current haptic model of human action under these circumstances.

### 7.3.2 Experimental setup

#### Human subjects

Upon obtaining an ethical approval from the ethical board of the clinical centre of Grosshadern, Ludwig-Maximilian University of Munich (LMU), a total of 35 participants was recruited for the experiment (29 male and 6 female). The participants were mostly students of the Technical University of Munich (TUM), aged between 19 - 30 years old. Prior to the experiment the participants were informed regarding the task they were to perform. After this the participants were asked to sign a consent form acknowledging their understanding of their tasks and approval of the terms of their participation in the study. Each experiment lasted about 30 minutes, after which they were compensated for their participation with 5 Euros.

#### The Experimental Scenario

The participants entered a virtual cocktail party which was represented only through audio and haptic cues that were provided to the participants, without any visual cues. The audio cues were rendered through a pair of noise-canceling headphones (for covering any machine or lab noise), whereas the haptics was via the handshake robot implemented on the floor-mounted ViSHaRD10.

The cocktail party sound was recorded through a stereo recording device in a bar in Munich. There were two types of sound recording -stereo and binaural, and the purpose of this was to examine whether these influenced overall reported place illusion. Stereo was recorded through the microphones of the stereo recording device alone. The binaural recording was performed by the stereo recording device, through two microphones placed inside the ear cavities of a custom made dummy head. Both recordings were exactly the same in terms of content, i.e. only the recording procedure was different.

The two types of recording used were distributed randomly amongst the participants, with half of the participants listening to the stereo recording and the other half to the binaural one. The recordings began with some music and people chatting in the background. After a few seconds a person approached and greeted the participant. This was repeated 21 times with 21 different voices, with a gap of approximately 15s between each one. The



**Figure 7.4:** The experimenter performing a 'human' handshake with a participant 'through' the robot.

voices consisted of 10 female and 10 male voices in the following sequence, where M=male and F=female:

M-F-M-F-F-M-F-F-M-F-M-M-F-F-M-M-F-M-F-M-M

For the haptic mode the handshake robot was used, utilizing only one degree of freedom, the vertical axis, i.e. up-down movement. The handshakes were exchanged between the participant and the handshake robot via a metal rod as the end-effector, as can be seen in Fig.7.4. Three haptic conditions were used:

- The handshake performed by the robot with the *interactive controller*, referred to as *interactive robot* condition. This consists of the robot operating according to the algorithm described in Section 3.2. In this condition the robot waited for the participant to initiate the handshake.
- The handshake was performed by the robot with a basic controller (Section 2.4) of PC algorithm with a selected position reference (Section 2.4), henceforth termed as *basic* robot condition. This consists of the robot performing a sinusoid motion of 3 cycles, upon initiation by the participant.
- The handshake performed by the human experimenter (henceforth termed as *human-driven* condition). This handshake was performed by the experimenter by directly manipulating the haptic robot, as shown in Fig. 7.4. The experimenter used the opposite end of the same end-effector as the participant. The reason for performing the human handshake through the robot was to try to equalize all the conditions apart from the factors of interest (i.e., the two machine algorithms and the actual human handshake).

Given the above description the purpose of the experiment can be restated specifically as investigating the extent to which the interactive robot can 'fool' the human participants into thinking they are shaking hands with a real human. The evaluation method can be

considered as comparison of the robot performance with 'ground truth' which was in this case the perceptions of the handshakes that were carried out by the humans.

### **7.3.3 Experimental procedures**

Upon arrival outside the laboratory, each participant was briefed regarding their task which involved exchanging some handshakes sometimes with a robot and sometimes with a human. Furthermore they were informed that throughout the experiment they would be blindfolded. Then the participants were blindfolded prior to entering the laboratory and guided by one of the experimenters onto the platform in front of the robot - the blindfolding ensuring that they had no idea regarding the physical appearance of the robot.

Once placed in front of the robot they were fitted with an ordinary glove to reduce the tactile sensation of the metal rod. They were also fitted with a pair of noise canceling headphones. They were then asked to perform a few handshakes in order to become familiarized with the handshaking procedure. They were informed that they should initiate a handshake after the experimenter had guided their hand to grasp the metal rod. They were to perform a handshake and then place their hand back on their side, until the next time their hand would be guided to the rod again. After these instructions, some background music was introduced through their headphones and they were guided by the experimenter to perform 4-6 handshakes.

After reporting that they felt confident with the haptic interface they were asked to perform the same task, however now they would give out a score after each handshake, evaluating it. They would do so by saying out loud an integer between 1 and 10, where 1 would indicate that the handshake they performed felt as if it were performed by a robot and 10 would indicate that it felt as if it were performed by a human. The in-between values should indicate a likelihood scale between the two extremes. Then they performed another set of handshakes (4-6) while listening to the background music, this time giving out a number after each handshake.

After they had finished with these handshakes they were informed that the training session had finished and the main experiment was about to begin, where they would have to perform the same procedure as in the last part of the training. Throughout the training session only the two robot conditions were used, i.e. the interactive and the basic robot condition.

The main experiment started with the cocktail party sounds playing through the headphones that faded in background music and chatting and after a while the first voice approached and greeted them. The experimenter at that point guided the hand of the participant to grasp the rod who then initiated a handshake. After placing their hand on the side they called out their score.

The sequence of handshakes consisted of 7 for each condition - 7 handshakes for the interactive robot, 7 with the basic and 7 with the human-driven - the order being random for each subject. The distribution of the types of handshakes was almost equal between male and female voices in the recording to avoid association of a specific type of handshake with a gender. The same voice sequence was used for all the participants.

The responses of the participants, along with the sequence of the handshake conditions were recorded by the experimenters. Upon completion of the experiment, participants were

guided into a separate room, where their blindfold was removed and they were asked to complete a post-experimental questionnaire, which consists of:

- 4 questions of basic demographic data (dominant hand, sex, age, nationality)
- 1 questions for evaluating the level of place illusion and the following 3 questions concerned more with the degree of plausibility experienced by the subject within the virtual environment:
  1. (real bar): Indicate your experience of being in a real bar on a scale from 1 to 7, where 7 represents a normal experience of being in a real place.
  2. (socializing) On a scale from 1 to 7, how often did you have the feeling that you were socializing with real people?
  3. (real person) On a scale from 1 to 7, how often did you have the feeling that you were shaking hands with a real person?
  4. (type of person) On a scale from 1 to 7, throughout the experiment how often did you think about the type of person you might be shaking hands with?
- 3 questions for evaluating the validity of the responses given throughout the experiment:
  1. (number of human-driven handshakes) In how many of the 21 handshakes did you think you were shaking hands with a human?
  2. (number of robot handshakes) In how many of the 21 handshakes did you think you were shaking hands with a robot?
  3. (number of different robots) How many different robots do you think you were shaking hands with?

After having answered these questions, they returned to the laboratory with the handshake robot and provided further comments.

### 7.3.4 Analysis and results

#### Statistical methods

The data were considered as a three way analysis of variance with response variable being the scores and the factors as type of sound (two levels, stereo or binaural), the gender of the speaker (2 levels), and the type of handshake (basic robot, interactive robot or human-driven). The latter is the fundamental variable of interest. It should be noted that although there were 7 trials of each of the three types of handshake, 4 of these scores were not recorded, and therefore the design is not strictly balanced. However, the analysis of variance (ANOVA) technique allowed for this.

Since the gender of the speaker did not prove to be significant at all, this is not considered further.

**Table 7.2:** Means and standard deviations of the scores in the three conditions. An individual score is an assessment out of 10, where 10 means that the participant fully believes that the hand shaker was human and 0 that it was a robot.

Condition	Mean score (max 10)	Standard deviation	n
Basic robot	3.3	2.1	242
Interactive robot	5.9	2.2	244
Human-driven	6.8	2.1	245

**Table 7.3:** Analysis of Variance of the variable  $y = \sqrt{\text{score}^3}$  with factors type of handshake and type of sound.

Source	SS	df	Mean SS	F-ratio	Sig. level
Handshake	16879.2	2	8439.62	170.36	0
Sound	17.04	1	17.04	0.34	0.5577
Handshake*Sound	326.8	2	163.41	3.3	0.0375
Residual	35916.6	725	49.54	-	-
Total	53045.2	730	-	-	-

The ANOVA requires that the residual errors of the model fit follow a normal distribution. This was attained by a transformation of the response variable (score) by using  $y = \sqrt{\text{score}^3}$ . Since this transformation is monotonically increasing with respect to score, any inferences based on  $y$  apply equally to *score*.

## Results

Table 7.2 shows the means and standard deviations of the scores under the three conditions. It is clear that the difference between the basic robot and the other two conditions is large, and although the difference between the interactive robot and human-driven is smaller, the difference between these two is still significant.

This is confirmed by an analysis of variance (Table 7.3) on the transformed score variable ( $y$ ), with two factors: type of handshake (basic robot, interactive robot and human-driven) and type of sound (stereo or binaural).

The results show overwhelmingly that there is a difference in scores amongst the three conditions. Moreover there is some evidence of an interaction effect between type of handshake and type of sound. A multiple contrast analysis (at an overall 0.01 level of significance) shows that

- There is a significant difference between each of the handshake means with  $\text{mean}(\text{human-driven}) > \text{mean}(\text{interactive robot}) > \text{mean}(\text{basic robot})$
- When the basic robot is used there is no significant difference between the scores for the different types of sound. This is also true for the interactive robot and the human-driven handshake. However, the mean score when the interactive robot is used with the binaural sound is significantly less than for the human-driven handshake for

**Table 7.4:** Means and Standard deviations of the subjective scores for the two sound conditions.

Condition	Basic robot		Interactive robot		Human-driven	
	Mean	Std.	Mean	Std.	Mean	Std.
Binaural	3.6	1.4	5.7	1.3	6.7	1.5
Stereo	3.0	1.4	6.1	1.8	6.9	1.3

**Table 7.5:** Means and Standard deviations of the subjective scores for the four post-experimental questionnaire, for the two sound conditions. The maximum score in each case was 7.

Condition	Read bar		Socializing		Real person		Type of person	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Binaural	4.3	1.4	3.2	1.7	3.3	1.9	4.7	1.9
Stereo	4.9	1.5	4.2	1.4	3.8	1.3	4.8	1.6

either type of sound. This is illustrated in Table 7.4, which shows the breakdown of scores by sound type. This is what is responsible for the significant interaction term.

From Table 7.4 it is also clear that although the differences between the human-driven and interactive robot is formally significant, the effect size difference is small. The interactive robot was almost judged the same as the human-driven, and in fact the scores for the human-driven are far from the maximum of 10. Since there was considerable uncertainty in rating the human-driven as human, this also adds to the point that the interactive robot was ranked nearly the same as the human-driven.

Table 7.5 shows the mean and standard deviations of the subjective scores provided in the post-experimental questionnaire. There are no significant differences between binaural and stereo. If we consider the proportion of people who gave relative high scores (scores of 5, 6 or 7 out of a maximum of 7) then there is evidence that overall participants tended to subjectively have the illusion of being in a bar. Considering the question 'real bar' 21/35 people gave a score of at least 5, and against the null hypothesis that the scores were randomly assigned, this has a significance level of 0.03 (using the binomial distribution). However, they did not have the illusion of socializing with real people (here there were 11 out of 35 who recorded high scores), nor shaking hands with a real person (9 had high scores). However, 23 of them did give high scores in relation to often thinking about the type of person they were shaking hands with ( $P = 0.005$ ) - but this was not surprising since it was their actual task.

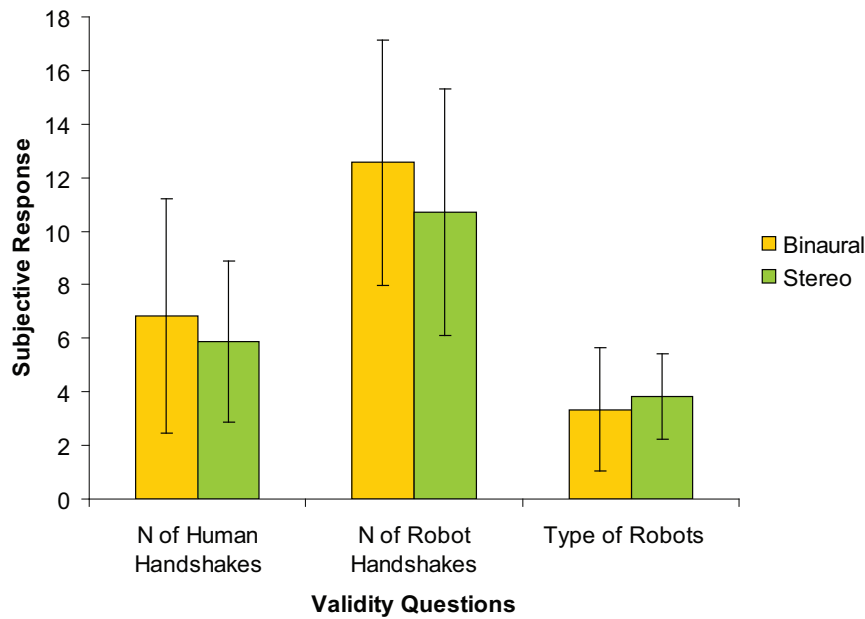
Table 7.6 shows the means and standard deviations of the scores provided by the participants in the post-experimental questionnaires regarding the three questions regarding their overall view of the numbers of human-driven and robot handshakes. There were 3 people who said that the number of human-driven handshakes was more than 10 out of the 21 possible handshakes, but 21 who said that there were more than 10 robot handshakes. In other words there is a clear overestimate of the number of robot handshakes, which is again evidence that the difference between some of the real human-driven handshakes and

**Table 7.6:** Means and Standard deviations of the subjective scores for the three last questions of the post-experimental questionnaire, for the two sound conditions.

Condition	N of Human-driven handshakes		N of Robot handshakes		Type of robots	
	Mean	Std.	Mean	Std.	Mean	Std.
Binaural	6.8	4.4	12.6	4.6	3.3	2.3
Stereo	5.9	3.0	10.7	4.6	3.8	1.6

robot handshakes were blurred. It is not that the robot felt like a human but that the human felt like a robot.

Fig. 7.5 illustrates the results from Table 7.6. From this we can see that the subjects under the binaural sound condition, seem to have responded more accurately than the subjects under the stereo conditions, as the actual number of human-driven handshakes is 7, the actual number of robot handshakes is 14 and the real number of types of robot is 3 (basic, interactive and human-driven).

**Figure 7.5:** Means and standard deviations of the subjects' responses in the three post-experimental questionnaire regarding the three validity questions, for the two sound conditions.

### 7.3.5 Conclusions

The purpose of this experiment was to provide a method for evaluating the ability of a haptic interface in inducing the feeling that the participants were actually shaking hands with a human. The results from this experiment show that the human-driven handshake is perceived as more human than the robot handshakes and also that the robot operating in its

interactive mode is scored considerably higher than the robot operating in its basic mode. However, there is also evidence that there was considerable confusion between the human-driven and interactive robot. This can be attributed to the fact that the haptic feedback was implemented through a metallic rod instead of an expected hand when performing a handshake, even for the human. It also reflects the fact, as some of the participants remarked, that the motion of the handshake was purely up and down rather than with no lateral movement at all. Furthermore, the sound condition did not affect significantly the results, which might be attributable to the quality of the binaural recording itself.

The results of this experiment are to be used as feedback for an improved version of the control algorithm of the handshake robot. Furthermore, the next step would be to introduce visual cues in the experiment through an immersive projection system such as a Head Mounted Display, as this would eliminate the use of an experimenter guiding the hands of the participants to reach the 'hand' of the robot and may increase the levels of plausibility experienced by the subjects. Furthermore, to increase the level of place illusion, an improved binaural recording should be considered, through the use of an appropriate algorithm to enhance spatial auditory localization. Also, an alternative to the metal rod end-effector of the handshake robot is under construction in order to bring the tactile interface closer to reality and increase the plausibility of the experience.

### 7.4 Experiment 3: Study of haptic-visual integration

This experiment focuses on the effects of visual and haptic feedback on the experienced plausibility of social interaction in a virtual reality scenario, where participants were asked to perform handshakes with a virtual, visually and haptically rendered partner. A 3D virtual environment was created and integrated with a handshaking robot, enabling the participant to see the virtual partner while shaking hands. To assess the effect of visual and haptic rendering strategies on plausibility, an experiment with human subjects was carried out. The results indicate that adding vision and improving the quality of haptics, both improve plausibility. No interaction between the two modalities was found, suggesting that vision and haptics are equally important to the perceived plausibility of a virtual handshake task.

#### 7.4.1 Introduction

Adding haptic feedback to multimodal virtual reality (VR) systems enables users to not only see, but also physically interact with objects and characters in a virtual world. Knowledge about the influence of haptics and vision on plausibility of the interaction is an important factor when developing VR systems and is especially critical when assigning limited resources for system development.

Different from the previous two experiments, with vision in the system, plausibility now refers to the perception that an event/object in the virtual world is actually occurring/existing, although knowing it is only computer mediated [83]. It is different from the illusion of "being there", often referred to as *presence*. While the experience of sharing a room with a virtual character, e.g., can be achieved by visual feedback only, plausible



handshaking with a virtual character not only requires rendered limb motions similar to a real handshake, but the physicality of interaction also calls for an adequate haptic feedback.

There are studies of the integration of haptics and vision in presence ([82], [38]) as well as human perception [51], and studies on social interaction that focus on the effect of haptics on performance ([12], [71], [32]) and the level of presence ([12], [71]). To the authors' knowledge, however, there is no report on the interaction of haptics and vision and their influences on plausibility when performing a social interaction task. But, based on common findings it is expected that: i) enabling both, haptics and vision, improves plausibility compared to providing a single modality alone, ii) improving the quality of haptics improves plausibility, iii) haptics and vision influence each other.

To assess the validity of these assumptions, an experiment was carried out, where subjects were asked to perform handshakes with a virtual character.



**Figure 7.6:** (a) Experimental setup; (b) 3D virtual scenario

### 7.4.2 Experimental setup

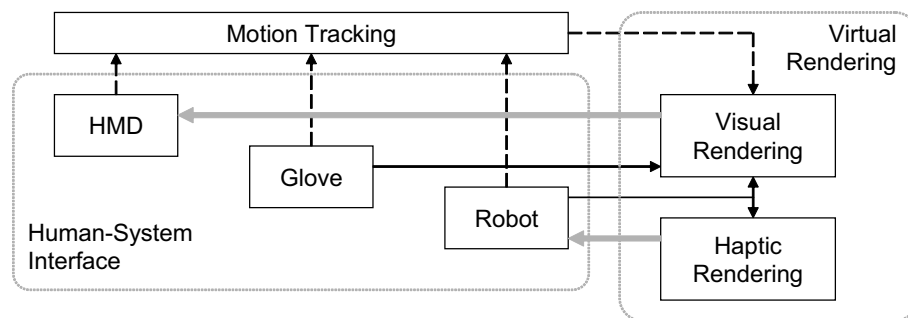
An immersive VR system was used, see Fig. 7.6. The user wears a head-mounted display (HMD) that provides visual feedback. A motion tracker is mounted on the HMD that allows her/him to look around in the virtual environment just by turning her/his own head. To provide haptic feedback ViSHaRD7 robotic arm [63] with a rubber artificial hand mounted at the end-effector was used. A second motion tracker is mounted to the base of the robot platform to map the position and orientation of the end-effector to the world coordinate system for visual rendering. A 6 DOF force/torque sensor located between rubber hand and robot end-effector allows measuring interaction forces. The rubber hand is placed coincidentally with the hand of the virtual character such that it is waiting when the user reaches out to the virtual hand. The robot arm is registered with the virtual environment and measurement data is used to update position and orientation of the rubber hand in the virtual environment. Human hand posture is measured using a CyberGlove [21] worn by the participant. A third motion tracker is placed on the glove

to provide the actual position and orientation of the user's wrist for the visual rendering engine. The overall structure of the experimental setup is shown in Fig. 7.7.

Haptic and visual rendering is handled by separate subsystems. After fine calibration, the two virtual hands touch each other then, when in the physical world the hand of the participant touches the rubber hand mounted on the robot. Because of the existence of the rubber hand, the hand of the participant does not move further and the two virtual hands do not penetrate.

### Haptic rendering:

To investigate whether the quality of haptic feedback has an effect on the experienced plausibility of interaction, two types of haptic feedback, basic and human-driven, were provided and compared with each other. The basic form (HR1) aims at providing a very poor resemblance of the real world. The robot implements virtual impedance and acts as a passive mass-spring-damper model with mass 10kg, damping 20 Ns/m, and stiffness 20 N/m. Thus, the human needs to continuously accelerate the robot, otherwise it stops quickly. The second, human-driven haptic feedback (HR2) uses a trained human expert to drive the robot arm and shake with the participant.



**Figure 7.7:** Structure of the experimental system. Thick gray lines indicate feedback signals, solid black lines indicate the measurements from the devices, dashed black lines indicate the position and orientation measurements by the motion trackers on each device.

### Visual rendering:

The virtual scene consists of an empty space with a floor of size 2m by 4m, a female virtual character, and the virtual representation of the participant's own forelimb. To achieve a high level of realism, fine details and vividness of the virtual character are expected. The detail-rich hand postures of both the virtual character and the participant are rendered by a realistic hand model with 24 degrees of freedom (DOF) [19]. The virtual handshake partner and the virtual body of the participant are created using commercially available virtual characters from Poser [84]. Poser is not capable of rendering the virtual characters online, therefore Open Inventor (OI) is employed in real-time visual rendering. From Poser to OI, color and texture information is lost, as well as the links between body parts. To solve this problem the virtual human model is first exported to 3Ds Max [7] to separate

the necessary body parts and add color and texture information before loaded into OI. Only the right hand is substituted by the 24 DOF hand model.

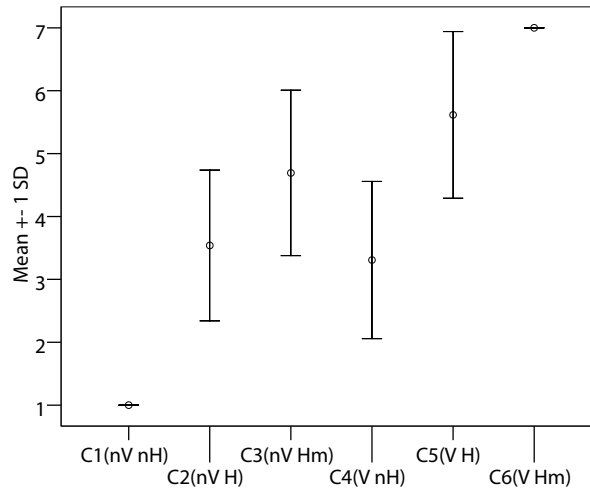
### 7.4.3 Experimental design

In the experiment two different levels in vision, *without vision* (nV) and *with vision* (V) as well as three levels in haptics, *without haptics* (nH), *with basic haptics* (H), and *human driven* (Hm) were distinguished. This results into 6 different conditions to be compared: (C1) nV+nH, (C2) nV+H, (C3) nV+Hm, (C4) V+nH, (C5) V+H, and (C6) V+Hm.

A black screen is presented in nV, while the normal VR environment is provided in V, see Fig. 7.6b. The participant does not hold anything and shakes air in nH, while holding the rubber hand and basic haptic feedback is provided in H, while the human drives the robot in Hm. Since only arm dynamics is considered in this work, the human condition is rendered as teleoperated, so that the participant always grasps onto the same unactuated end-effector. Given the technical constraints, the best achievable condition C6 and the worst condition C1 are repeated in every handshake group to enforce the comparison boundaries. The participants are actually evaluating of *how close* is the handshake in question to the best case, in terms of plausibility. A number of 13 (4 females, 9 males, averaged age 26.9) college students participated in the experiment.

The experiment consists of three parts: pre-experiment briefing, the training session, and the main experiment. The procedure is designed such that the participant does not see the robot throughout the experiment, avoiding the appearance of the robot to reduce the plausibility of the interaction. The training session is crucial for getting the participant naive to the experiment familiar to the system, and to focus them to the plausibility of haptic/vision rendering instead of other factors. A trained expert guides the participant through the entire procedure.

1. Pre-experiment briefing. When the participant arrives at the site, the expert first introduces the experiment and gets necessary documents signed outside the experiment room and then blind-folds and guides the participant into the room. The participant puts on the HMD with sound-proof headphones which covers robot noise, as well as the CyberGlove and runs calibration.
2. The training session. The participant gets familiar with the system by shaking hands with the most unrealistic handshake C1 and the most realistic handshake C6 to define the upper and lower limit of the plausibility scale. A scoring system is introduced, with 1 being *most unrealistic*, and 7 being *most life-like*. C1 was assigned the score 1 as the worst condition with "seeing nothing while shaking air". C6 with visual feedback and a human driving the robot was the best achievable condition, hence assigned score 7. Then the participant performed 6 groups of handshakes, each group consisting of three handshakes arranged in the following order: C1, C6, and C(x), where C(x) is one of the conditions C1-C6 to be evaluated. In each handshake group C1 and C6 were carried out to remind the participant of the worst and best handshakes before C(x) was presented. The participant evaluates the plausibility of social interaction for the condition C(x) and gives an integer score between 1 and 7.



**Figure 7.8:** Mean values and standard error among different conditions

3. Main experiment. After training, the main experiment took place and again the participant performed 6 groups of handshakes, giving a score for the actual presented  $C(x)$ . Conditions were presented in randomized order.

#### 7.4.4 Results and discussion

In the following only data of the main experiment is analyzed. Descriptive results of the effect of the six conditions on the experienced plausibility of social interaction are depicted in Fig. 7.8. As can be seen the two extreme conditions C1 and C6 could be clearly identified and categorized compared to the other conditions. Since the two extreme conditions C1 and C6 have zero variance, no ANOVA (analysis of variance) could be calculated because of the violation of its assumptions. To analyze effects of the single modalities, conditions in vision (nV, V) and haptics (nH, H, Hm) were grouped and single t-tests were calculated to analyze whether adding the visual modality or adding/improving the haptic modality has a significant effect on plausibility. To keep the overall alpha error on a 5% level, the significance level for the individual tests was adjusted to 0.0125 according to Bonferroni ( $\alpha/n$ , where  $n$  is the number of performed tests). Results show significant differences between the two conditions in vision ( $t(12) = 9.54$ ;  $p < .0125$ ;  $r = .940$ ) and between the three conditions in haptics (nH-H:  $t(12) = 7.584$ ;  $p < .0125$ ;  $r = .909$ ; nH-Hm:  $t(12) = 12.660$ ;  $p < .0125$ ;  $r = 0.965$ ; H-Hm:  $t(12) = 3.926$ ;  $p < .0125$ ;  $r = .752$ ).

Thus, it can be concluded that adding vision, adding and improving haptic feedback to a social interaction scenario can all improve the plausibility of interaction. The missing interaction between the two factors further indicates that visual and haptic feedbacks are two independent components to be taken into account for system design. This finding is also supported by similar effect sizes found for the influence of vision and haptics on the experienced plausibility illusion. There is a trend of balanced vision/haptics (C5) being favored over improving either modality (C3 and C4), but the difference is not significant.

### 7.4.5 Conclusions

The influence of visual and haptic feedback on plausibility of social interaction was investigated. To this end, an experimental approach was employed, where human participants were invited to perform handshakes with a virtual character while experiencing different visual and haptic rendering strategies. The results suggest that adding vision as well as improving haptics have a positive effect on experienced plausibility of a handshake and that the two modalities contribute equally and independently to this illusion.

## 7.5 Experiment 4: Evaluation with vision, haptics, and audition

### 7.5.1 Introduction

This experiment is a continuation of Experiment 2 in Section 7.3, on how do the haptic rendering algorithms perform in terms of plausibility, when visual feedback is presented.

Major changes have been applied to the virtual system comparing with the one used in Experiment 2. A visual scenario is developed, the haptic modality is extended from 1 DOF to 6 DOF, the human-driven condition is realized by teleoperation via another haptic interface remotely.

The *ground truth* approach and the cocktail party social scenario remain unchanged. The experiment was conducted in collaboration with Prof. Slater's research group in UPC.

### 7.5.2 Experimental design

A visual scenario of the cocktail party is created with virtual characters, as introduced in Section 5.1. In accordance with the sound recordings made from a real cocktail bar, the participant is emerged into the virtual environment with visual and sound feedbacks throughout the experiment.

Once in a while, a virtual character enters the room and walks towards the participant. Once the virtual character stops, he/she greets the participant with a simple line and reaches out the hand to the participant for a handshake. When the participant reaches out for the virtual hand, it is the robot waiting at the same location in the real world. After handshaking the virtual character turns around and leaves the room. The procedure is repeated until the end of the experiment.

Two different visual environments are used in this experiment. A basic setup is used in the training session, where a single virtual character stands in an empty room performs handshakes repeatedly with the participant, as used in Experiment 3, see Fig. 7.6(b). This is only to get the participant familiar with the system, without actually introducing the cocktail party scenario. The actual experiment employs the full visual setup with 18 different virtual characters. No comparative conditions are introduced in vision and sound during the actual experiment.

There are three conditions in haptics to be compared: *basic robot*, *interactive robot*, and *human-driven*. In basic robot condition, the robot is controlled by a position controller

with a fixed sinusoidal reference trajectory; in interactive robot condition the interactive controller with the updated trajectory planner is employed; while in human-driven condition the second human handshake with the participant via teleoperation. In all three conditions the robot in use is the robotic arm ViSHaRD10, with 6 DOFs enabled. A picture of the experimental scene is shown in Fig. 7.9, and the view of the participant into the virtual world is shown in Fig. 7.10.

Totally 21 college students (11 females, 10 males, averaged age 26.4) participated in this experiment.



**Figure 7.9:** Experimental 4 setup.



**Figure 7.10:** Participant view of the virtual system in experiment 4.

### 7.5.3 Experimental procedure

This experiment consists of four parts: pre-experiment briefing, the training session, the main experiment, and filling up the post-experiment questionnaire. The participants are not allowed to see the robot throughout the experiment, avoiding the appearance of the robot to reduce the immersion into the VR environment. A trained experimenter guides the participant during the entire procedure.

1. Pre-experiment briefing. When participant arrives at the site, the trained expert introduces the experiment and get necessary documents signed outside the experiment room, preventing the participants to view the robot prior to the conclusion of the experiment, then blind-folds and guides the participant into the experiment room. The participant is requested to put on the HMD with sound-proof earphones to cover the robot noises, also wear motion trackers, as well as the data glove and then run the calibration, as described in Section 6.1.2.
2. The training session. The participant performs 6 handshakes of the basic condition. At this stage, the participant sees a female virtual character in front of him/her with her hand protruded as if awaiting a handshake. The participant is asked to reach for the hand of the virtual character and perform a handshake. At this point, the expert explains to the participant about the rating, that after each handshake a score from 0 to 10 needs to be given by the participant, about how close the handshake felt to shaking hands with a real person. Score 1 means *most unrealistic*, and 10 being *totally real*. The participant then continues on the training session for another 6 handshakes, giving a score after each one. In this training session, the handshakes performed are the basic robot condition and the interactive robot condition, presented in random order.
3. The main experiment. This session consists of 18 handshakes. Each of the three conditions in comparison is repeated 6 times in random order. The visual environment presented to the participant in this stage is the cocktail bar room, which is foreign to the participant. Therefore initially the participant is given some time to look around the environment to get accustomed to it and also to look down at the virtual ego-representation and get familiar with moving the virtual hand. A pre-recorded cocktail party background noise with people chatting and background music is played back. At intervals of 15 seconds a virtual character enters the room, walks towards the participant, performs a handshake, and then leaves the room. After each handshake the participant is supposed to rate according to the experience. No further interaction between the expert and the participant is expected, the participant is supposed to be fully instructed by the virtual events occurring in the virtual world.
4. After the experiment, the participant is led out of the experiment room to fill in a post-experiment questionnaire.

After filling out the questionnaire, the participant is allowed back in the experiment room for further comments.

	N	Mean	Std. Deviation	Std. Error
With Basic Haptics	126	3.03	1.595	.175
With Advanced Haptics	126	5.30	2.001	.178
Human Tele-driven	126	6.78	2.400	.214

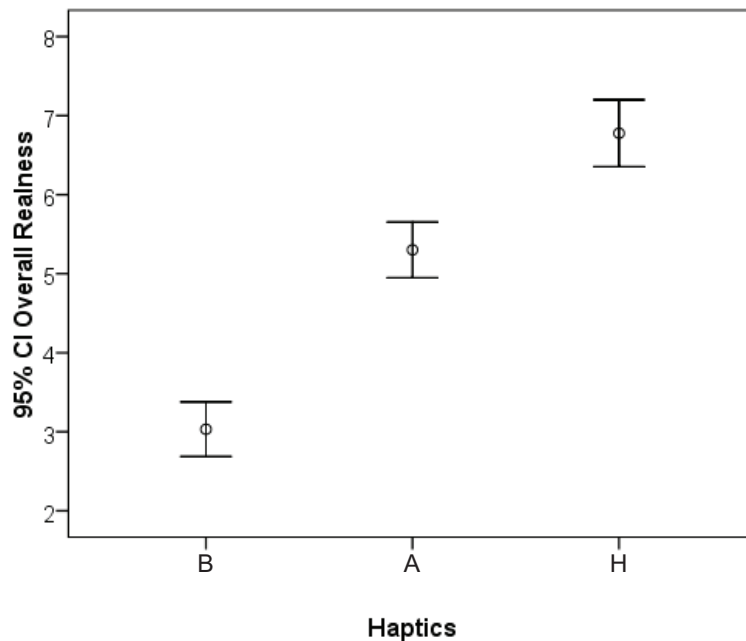
**Table 7.7:** Description of the Haptic in Experiment 4

### 7.5.4 Results and analysis

In this experiment, the analysis focuses on the influence of haptics on the overall realness. Descriptive statistics of the overall realness based on different haptic conditions are reported, then the analysis of variance (ANOVA) and multiple comparisons are presented.

#### Descriptive analysis

A total number of 378 samples are collected in this experiment, 126 for each condition in haptics ( $B$  for basic robot,  $A$  for interactive robot,  $H$  for human driven). Descriptive results of the effect of the three conditions on the experienced plausibility of social interaction are depicted in Table. 7.7 and Fig. 7.11. It can be observed from the t-test results that no overlap about the overall realness is found between all three conditions.



**Figure 7.11:** Mean Values and Standard Errors of the Haptic Conditions in Experiment 4

#### ANOVA

ANOVA is used to examine the relationship between haptics and the overall realness. As shown in Table. 7.8 and Table. 7.9, there is a significant effect of haptics condition on the overall realness ( $F(2, 375) = 98.96, p < .001$ ). Post hoc comparisons via Bonferroni tests reveal that the 3 different haptic algorithms are completely discriminated in



	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	897.291	2	448.646	98.955	.000
With Groups	1700.190	375	4.534		
Total	2597.481	377			

**Table 7.8:** Results of ANOVA for the Overall Realness in Experiment 4

		95% Confidence Interval				
(I) Condition	(J) Condition	(I-J) Mean Difference	Std. Error	Sig.	Lower Bound	Upper Bound
B	A	-2.270*	.268	.000	-2.91	-1.62
	H	-3.746*	.268	.000	-4.39	-3.10
A	B	2.270*	.268	.000	1.62	2.91
	H	-1.476*	.268	.000	-2.12	-.83
H	B	3.746*	.268	.000	3.10	4.39
	A	1.476*	.268	.000	.83	2.12

\*. The mean difference is significant at the 0.05 level.

**Table 7.9:** Post-hoc Comparisons on the Haptic Conditions in Experiment 4

this experiment, although the virtual social environment is more complicated than former experiment.

### 7.5.5 Conclusions

Before the experiment a major concern was raised about if introducing vision into the system eliminates the discrimination in the haptic modality, making all conditions indifferent. Although the findings of experiment 3 suggest that vision and haptics are independent, since only a basic visual scenario was used in that experiment, it is still unclear what effect will the complex cocktail party scenario have on the discrimination of haptic conditions. Although no specific instruction on where to focus was given to the participant during the experiment, the results turn out that all three conditions in haptics could significantly be distinguished. This may due to the fact that handshaking is a task that emphasizes on haptics, but at least it can be claimed that for such tasks where haptic interaction is concerned, vision does not dominate haptics.

Since the extension of DOF and the visual feedback are applied to all conditions, the human-driven condition as the *ground truth* is different from the one applied in previous experiments. Hence the resulting scores are not directly comparable and no conclusion can be drawn on whether the improvements of the system have made the human-driven condition closer to the real-world interaction. However it can be observed that the current human-driven condition is still scored lower than 10, *totally real*. This can be expected, given the missing hand grasping and all the technical limitations of the robot.

## 7.6 Discussions and conclusions

The four evaluation experiments covered in this chapter were presented in a timeliness manner. Each experiment validates one stage of the system, with its findings fed back into system design and improvement. Experiment 1 was carried out at a very early stage of the work, which concerns about the most suitable control algorithm for the virtual handshake partner control. Experiment 3 deviates slightly from the main line of evaluating the evolutionary haptic modality of the system, while focusing more on the integration of the haptic subsystem to the newly developed vision subsystem. The findings of experiment 3 made important contributions to the design of experiment 4, where the final version of the integrated system gets evaluated.

All experiments concerned with one overall question: How can plausible experience of a real world human-human interaction task be achieved by various configurations of technical systems. To this end, the *ground truth* approach is taken in all experiments. The human-human interaction (human-driven condition) is always presented in the experimental design as a reference of the comparison with virtual partner-human interaction conditions. In experiment 3, different fidelity levels of visual and haptic rendering were presented to the human participant to compare with modality plays a more important role in plausibility. In experiments 2 and 4, the best virtual partner available at that stage (interactive robot condition) was compared to the human-driven condition, as well as another virtual partner with only basic functionality (basic robot condition). The aim of the comparison is to validate the superiority of the interactive robot over the basic robot, and if the interactive robot can match the human driven condition in terms of plausibility of interaction.

The results of experiment 3 show that haptic and vision are two independent factors, improving one cannot enhance or eliminate the level of fidelity of the other. The results of experiments 2 and 4 all suggest that interactive robot condition is rated significantly superior to the basic robot condition, while still being surpassed by the human-driven condition.

Here two remarks are drawn for future considerations:

### **What is wrong with the human-driven condition?**

In all experiments, the human-driven condition was rated by the participants significantly lower than the highest mark. Before experiment, all participants were instructed that the highest mark should be given to a most natural interaction experience as if the handshake was carried out in the real world with a real human partner. In the experiments, the human-driven condition is realized as another human driving a media device to handshake with the participant, instead of direct hand-hand contact. This is necessary to maintain a fair comparison given the technical limitations applied to the robot-partner conditions. Hence to the participant, a human-driven handshake with motor noise, residual robot inertia, highest acceleration limitation, missing hand grasping, is not quite matching a real-world handshake experience. Therefore it is rated lower than the highest mark.

**What is still different between interactive robot and human-driven conditions?**

Although the human-driven condition takes into consideration the technical limitations and constraints of the robot, the interactive robot condition is still rated significantly lower. The reason lies in the trajectory planner. Even if a perfect estimation of the participant intention can be obtained, the trajectory planning module of the robot controller, with a handful set of mathematical rules, cannot match the human with much more intelligence.

The results obtained from the experiments carried out in this work are specifically made for handshaking application. Cautions are needed when generalizing the results to other application scenarios. However, the methodology and lessons learned are easily transferable. For instance, the factors that can break presence are not only true for handshaking, but also apply to other haptic HRI tasks.

Future studies could be direction from the following aspects: as already pointed out in [80] and [81], using questionnaires as the only method of assessment may lead to incomplete knowledge about the system performance. During the experiments behavioral data such as trajectory and interaction force were recorded for each trial of each participant. The large amount of collected data could be studied together with the questionnaire scores, to find out what really happened when a participant gives a low score for a handshake. Another approach is to introduce physiological measurements of human participants such as anxiety during handshaking, as another complement to questionnaire results.

## 8 Conclusions and future directions

### 8.1 Concluding remarks

This thesis tackles the problems of how and how well can a human-human haptic interaction task be modeled and replicated by a technical system, with the aim of recreating a plausible interaction experience to the user.

An experiment-driven approach is taken for the system development. Starting from measuring human-human performances of the task, the obtained behavioral data is studied, modeled, and implemented into the technical systems. Studies are carried out of the forelimb of the human involved in a haptic interaction task including the arm and the hand. Besides haptics, vision and sound feedbacks are also provided to the human user. Evaluation experiments with human participants are carried out, with statistical analysis results to validate the performance of the developed systems.

There are three major contributions of this work: modeling and technical recreation of human arm behavior during handshaking; real-time integration of haptic data into multimodal virtual reality environment; and user response evaluation of haptic-enhanced virtual reality systems, with all development work focused on the human-human haptic interaction task of handshaking. Up to the date of publication, the systems created in this work are world-unique.

Chapters 2 and 3 are dedicated to the modeling and recreation of human arm behavior during handshaking by a technical system. In Chapter 2 human-human handshakes were observed and measured. A basic controller was developed to replay the recorded human behavior by a robotic arm. In Chapter 3, an interactive controller was developed to enable the robot handshake partner to interact with the human user proactively. The novelty of this work is that the robotic handshake partner can estimate human intentions of handshaking online and hence adapt its behavior to them. To achieve this, first the definition of human behavioral parameters were proposed to represent true human characteristics under mixed signal measurements; then a novel fast online estimation algorithm was proposed to obtain fast converging estimations to the HBPs online; an HMM based intention estimator then takes the symbolized HBPs to obtain human intention estimations, being world-unique in taking haptic behavioral data as HMM inputs; finally the estimated intentions are used in the adaptation of robot reference trajectory of handshaking, and setting the virtual dynamical characteristics of the robotic arm.

Chapter 4 is focused on the modeling and recreation of hand behavior of a handshake. Measurement devices are studied to obtain gesture and grasping force information to study human hand behavior. Studies of the actuation of a robotic hand to render handshaking are also presented.

Based on the handshake partner robot, a multimodal virtual environment was created, with vision and sound feedbacks integrated with the haptic subsystem. Chapter 5 is

concerned with the development of the vision and sound subsystems, while Chapter 6 focusing on the integration of the system and the improvement of the overall performance. The main achievement and novelty of the two chapters is real-time generation of high fidelity full body virtual human animation based on multi-source behavioral data, as well as the integration and registration of real-world data and objects into the virtual world. Improvements to the haptic modality, such as introducing a teleoperated human as a comparative handshake partner, extending the working DOF from 1 to 6, are also included.

A series of user studies to evaluate the performance of the developed systems were carried out. They are discussed in detail in Chapter 7. Four experiments were carried out with human participants. Questionnaires were used to assess the opinions of the participants, with statistical methods from psychological studies to analyze the results. The first and third experiments aimed at tackling specific problems of the system, namely to select a suitable controller and to study the effect of integrating haptics and vision, respectively. The second and fourth experiment, on the other hand, aimed to evaluate the handshake system at two different stages. They were conducted in collaboration with the research group of Prof. Slater in UPC, Spain. The results suggest that the advanced handshake robot with the ability to interact with human is rated significantly better than the basic one in term of interaction plausibility, however not good enough comparing to the reference condition where a real human conducts the handshake through a robotic interface. This relationship is not affected by enabling visual feedback. Although the techniques are standard, the experiments that evaluating haptic interaction systems from the aspect of plausibility have never been reported before. The experience and findings of the experiments could be helpful for future works.

## 8.2 Vistas

This work involves a broad range of topics related to human behavior modeling, virtual reality, and user evaluation. In this section, possible extensions and future directions are provided.

The haptic subsystem can be extended in several aspects. The interactive controller for the robotic arm can be extended to other applications where human intentions are concerned, with the estimation parameters and the HMM structure, for instance the number of hidden states, modified accordingly. The trajectory planner can be further refined to recognize more situations in the measured human action. The study of hand grasping is not yet integrated into the system, future work could be lead into this direction with new user validation studies.

The vision subsystem is developed as a balance between high-fidelity and real-time applicability. With the growing computational power and new edge in computer graphics, better and better virtual scenarios are desirable in the future. On the other hand, the current work only considered the matching between the gender of the virtual character and the voice clip. In the future the gender, and even characteristics, differences could be addressed also in haptics. Technically different characteristics of the robot can be realized by assigning different parameters during symbol abstraction, in the trajectory planner, and for the virtual arm dynamics.

The evaluation studies are carried out based on statistical analysis of questionnaires. At the same time, a large amount of behavioral data has been collected during the experiment. It could be worthwhile to investigate into the recorded position and force data, to exploit the relationship between a score in the questionnaire and what actually happened during the interaction.

In general, handshaking is a haptic interaction task specifically between two humans. However, the techniques in estimating human intentions and adapt the system behavior accordingly can be generalized to any robotic assistant that needs to comply to human users. For instance, in rehabilitation and automated training, the degree and profile of system assistance can be determined according to the user intentions online. On the other hand, the integration of haptics and vision/sound concerned in this work is generalizable to other haptic interaction tasks.

## Bibliography

- [1] L. Acosta, J.J. Rodrigo, J.A. Mendez, G.N. Marichal, and M. Sigut. Ping-pong player prototype. *IEEE robotics & automation magazine*, 10:44–52, 2003.
- [2] M. Adam, E. Vasonyi, I. Barsony, G. Vasarhelyi, and C. Ducso. Three dimensional single crystalline force sensor by porous si micromachining. In *Sensors, Proceedings of IEEE*, volume 1, pages 501–504, 2004.
- [3] R.L. Andersson. Dynamic sensing in a ping-pong playing robot. *IEEE Transactions on Robotics and Automation*, 5(6):728–739, 1989.
- [4] A.M. Arsenio. Developmental learning on a humanoid robot. In *Proceedings of IEEE International Joint Conference on Neural Networks*, pages 3167–3172, Hungary, 2004.
- [5] E. Asarin, O. Bournez, T. Dang, A. Pnueli, and O. Maler. Effective synthesis of switching controllers for linear systems. In *Proceedings of the IEEE 88, No.7*, pages 1011–1025, 2000.
- [6] C.G. Atkeson, C.H. An, and J.M. Hollerbach. Estimation of inertial parameters of manipulator loads and links. *The International Journal of Robotics Research*, 5(3):101–119, 1986.
- [7] Autodesk. 3ds Max. <http://www.autodesk.com>.
- [8] C.A. Avizzano. Cognitive human-computer communication by means of haptic interfaces. In *Proceedings of the 16th International Conference on Robot&Human Interactive Communication*, pages 75–80, Jeju, Japan, 2007.
- [9] W. Barfield and T. Furness. *Introduction to virtual environments and advanced interface design*. Oxford university press, New York, USA, 1 edition, 1995.
- [10] W. Barfield and S. Weghorst. The sense of presence within virtual environments: A conceptual framework. In G. Salvendy and M. Smith, editors, *Human-Computer Interaction: Software and Hardware Interfaces*, pages 699–704. Elsevier Publisher, 1993.
- [11] Barrett Technology: BarrettHand. <http://www.barrett.com/robot/products-hand.htm>.
- [12] C. Basdogan, C. H. Ho, M. A. Srinivasan, and M. Slater. An experimental study on the role of touch in shared virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(4):443–460, 2000.

- [13] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann. A sensor fusion approach for recognizing continuous human grasping sequences using hidden Markov models. *IEEE Transactions on Robotics*, 21(1):47–57, 2005.
- [14] R. Boian, A. Sharma, C. Han, A. Merians, G. Burdea, S. Adamovich, M. Recce, M. Tremaine, and H. Poizner. Virtual reality-based post-stroke hand rehabilitation. *Studies in Health Technology and Informatics*, 85:64–70, 2002.
- [15] M. Buss, K.K. Lee, N. Nitzsche, A. Peer, B. Stanczyk, and U. Unterhinninghofen. Advanced telerobotics: Dual-handed and mobile remote manipulation. In M. Ferre, M. Buss, R. Aracil, C. Melchiorri, and C. Balaguer, editors, *Advances in Telerobotics: Human System Interfaces, Control, and Applications*. Star Series, Springer, 2007.
- [16] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar. Learning collaborative manipulation tasks by demonstration using a haptic interface. In *Proceedings of the 14th International Conference on Advanced Robotics (ICAR)*, Munich, Germany, 2009.
- [17] M. C. F. Castro and A. Cliquet. A low-cost instrumented glove for monitoring forces during object manipulation. *IEEE Transactions on Rehabilitation Engineering*, 5(2):140–147, 1997.
- [18] S. Choi, L. Walker, and H.Z. Tan. Force constancy and its effect on haptic perception of virtual surfaces. *ACM Transactions on Applied Perception*, 2(2):89–105, 2005.
- [19] S. Cobos, M. Ferre, J. Ortego, and M.A. Sanchez-Uran. Simplified hand configuration for object manipulation. *Proceedings of EuroHaptics 2008*, LNCS 5024:730–735, 2008.
- [20] Coin3D. Openinventor. <http://www.coin3d.org>.
- [21] CyberGlove. <http://www.cyberglovesystems.com/>.
- [22] M. A. Diftler, C. J. Culbert, R. O. Ambrose, R. Platt, and W. J. Bluethmann. Evolution of the nasa/darpa robonaut control system. In *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, volume 2, pages 2543–2548, 2003.
- [23] N. Diolaiti, C. Melchiorri, and S. Stramigioli. Contact impedance estimation for robotic systems. *IEEE Transactions on Robotics*, 21(5):925–935, October 2005.
- [24] K. Drewing and M.O. Ernst. Integration of force and position cues for shape perception through active touch. *Brain Research*, 1078:92–100, 2006.
- [25] W. P. Eaton and J. H. Smith. Micromachined pressure sensors: review and recent developments. *Smart Material Structure*, 6:530–539, 1997.
- [26] D. Erickson, M. Weber, and I. Sharf. Contact stiffness and damping estimation for robotic systems. *The International Journal of Robotics Research*, 22(1):41–57, 2003.
- [27] FMOD. audio library. <http://www.fmod.org>.



- 
- [28] R. Fukano, Y. Kuniyoshi, T. Kobayahi, T. Otani, and N. Otsu. Statistical manipulation learning of unknown objects by a multi-fingered robot hand. In *Proceedings of the 4th IEEE/RAS International Conference on Humanoid Robots*, pages 726–740, 2004.
- [29] Y. Gao. *Multimodal data fusion of human grasping*. Technische Universität München, Master Thesis, 2010.
- [30] C. Giachritsis, J. Barrio, M. Ferre, A. Wing, and J. Ortego. Evaluation of weight perception during unimanual and bimanual manipulation of virtual objects. In *Proceedings of the World Haptics 2009*, pages 629–634, Salt Lake City, USA, 2009.
- [31] C. Giachritsis and A. Wing. Unimanual and bimanual weight discrimination in a desktop setup. In *Proceedings of EuroHaptics 2008*, pages 378–382, Spain, 2008.
- [32] E. Giannopoulos, V. Eslava, M. Oyarzabal, T. Hierro, L. González, M. Ferre, and M. Slater. The effect of haptic feedback on basic social interaction within shared virtual environments. In *Proceedings of EuroHaptics 2008*, pages 301–307, Madrid, Spain, 2008.
- [33] E. Giannopoulos, Z. Wang, A. Peer, M. Buss, and M. Slater. Comparison of people’s responses to real and virtual handshakes within a virtual environment. *Brain Research Bulletin (Under review)*, 2010.
- [34] D. Göger, K. Weiss, and C. Burghart abd H. Wörn. Sensitive skin for a humanoid robot. In *URL: <http://sfb588.uni-karlsruhe.de>*, 2006.
- [35] W. Griffin, R. Findley, M. Turner, and M. Cutkosky. Calibration and mapping of a human hand for dexterous telemanipulation. In *Proceedings of the ASME IMECE Conference Haptic Interface for Virtual Environments and Teleoperator Systems Symposium*, 2000.
- [36] Weston B. Griffin, Ryan P. Findley, Michael L. Turner, and Mark R. Cutkosky. Calibration and mapping of a human hand for dexterous telemanipulation. In *ASME IMECE 2000 Conference Haptic Interfaces for Virtual Environments and Teleoperator Systems Symposium*, 2000.
- [37] R. Groten, J. Hölldampf, M. Di Luca, M. Ernst, and M. Buss. Kinesthetics of dancing vs. cyclical movements. In *Proceedings of EuroHaptics 2008*, Madrid, Spain, 2008.
- [38] H.G. Hoffman. Physically touching virtual objects using tactil augmentation enhances the realism of virtual environments. In *Proceedings of the IEEE Virtual Reality Annual International Symposium, USA*, pages 59–63, 1998.
- [39] G. Hovland, P. Sikka, and B McCarragher. Skill acquisition from human demonstration using a Hidden Markov model. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2706–2711, 1996.

- [40] R. Howe. Tactile sensing and control of robotic manipulation. *Journal of Advanced Robotics*, 8(3):245–261, 1994.
- [41] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu. Robovie: an interactive humanoid robot. *Industrial Robot: An International Journal*, 28(6):498–504, 2001.
- [42] J. Jockusch, J. Walter, and H. Ritter. A tactile sensor system for a three-fingered robot manipulator. In *Proceedings of the 1997 IEEE International Conference on Robotics & Automation*, volume 4, pages 3080–3086, 1997.
- [43] D. Johnston, P. Zhang, J. Hollerbach, and S. Jacobsen. A full tactile sensing suite for dextrous robot hands and use in contact force control. In *Proceedings of the 1996 IEEE International Conference on Robotics & Automation*, volume 4, pages 3222–3227, 1996.
- [44] R. Kadouche, M. Mokhtari, and M. Maier. Modeling of the residual capability for people with severe motor disabilities: Analysis of hand posture. In *Proceedings of the International conference on user modeling*, pages 231–235, Edinburgh, 2005.
- [45] L. Kaelbling, M. Littman, and A. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [46] F. Kahlesz, G. Zachmann, and R. Klein. Visual-fidelity dataglove calibration. In *Proceedings of International Conference on Computer Graphics (CGI)*, Greece, 2004.
- [47] N. Kamakura, M. Ohmura, H. Ishii, F. Mitsubosi, and Y. Miura. Patterns of static prehension in normal hands. *The American journal of occupational therapy*, 34:437–445, 1980.
- [48] A. Kis, F. Kovács, and P. Szolgay. 3D tactile sensor array processed by CNN-UM: a fast method for detecting and identifying slippage and twisting motion. *International Journal of Circuit Theory and Applications*, 34:517–531, 2006.
- [49] K. Kosuge, T. Hayashi, Y. Hirata, and R. Tobiyaama. Dance partner robot -ms dancer-. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3459–3464, 2003.
- [50] Y. Kunii and H. Hashimoto. Tele-handshake using handshake device. In *Proceedings of the 21st IEEE International Conference on Industrial Electronics, Control, and Instrumentation*, volume 1, pages 179–182, USA, 1995.
- [51] S.J. Lederman, G. Thorne, and B. Jones. Perception of texture by vision and touch: Multidimensionality and intersensory integration. *Journal of Experimental Psychology: Human Perception and Performance*, 12(2):169–180, May 1986.
- [52] J. Lessiter, J. Freeman, E. Keogh, and J. Davidoff. A cross-media presence questionnaire: The itc-sense of presence inventory. *Presence: Teleoperators and Virtual Environments*, 10:282–297, 2001.

- 
- [53] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 2 edition, 1999.
- [54] J. Llobera, B. Spanlang, G. Ruffini, and M. Slater. Proxemics with multiple dynamic characters in an immersive virtual environment. *ACM Transactions on Applied Perception*, 2010.
- [55] J.M. Loomis and S.J. Lederman. Tactual perception. In K. Boff and L. Kaufman, editors, *Handbook of Perception and Human Performance*, pages 1–41. 1986.
- [56] J. Lu. *Study of visual and haptic cues in human-robot interaction*. Technische Universität München, Master Thesis, 2010.
- [57] T. B. Martin, R. O. Ambrose, M. A. Diftler, R. Platt, and M. J. Butzer. Tactile gloves for autonomous grasping with the NASA/DARPA robonaut. In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, pages 1713–1718, 2004.
- [58] Hermann Mayer, Istvan Nagy, and Alois Knoll. Skill transfer and learning by demonstration in a realistic scenario of laparoscopic surgery. In *Proceedings of IEEE International Conference on Humanoids*, 2003.
- [59] F. Miyazaki and R. Mori. Realization of ball catching task using a mobile robot. In *Proceedings of IEEE International Conference on Networking, Sensing and Control*, pages 58–63, Taipei, Taiwan, 2004.
- [60] F. Newell, M. Ernst, B. Tjan, and H. Buelthoff. Viewpoint dependence in visual and haptic object recognition. *Psychological Science*, 12(1):37–42, 2001.
- [61] L. Pan. *Real-time position mapping of human hand and robot hand*. Technische Universität München, Bachelor Thesis, 2010.
- [62] G. Paynter and I. Witten. Applying machine learning to programming by demonstration. *Journal of Experimental and Theoretical Artificial Intelligence*, 16(3):161–188, 2004.
- [63] A. Peer and M. Buss. A New Admittance Type Haptic Interface for Bimanual Manipulations. *IEEE/ASME Transactions on Mechatronics*, 13(4):416–428, 2008.
- [64] A. Peer, S. Eidenkel, and M. Buss. Multi-fingered telemanipulation - mapping of a human hand to a three finger gripper. In *Proceedings of the 17th International Symposium on Robot and Human Interactive Communication*, pages 465–470.
- [65] A. Peer, U. Unterhinninghofen, and M. Buss. Tele-assembly in wide remote environments. In *Proceedings of the 2nd International Workshop on HumanCentered Robotic Systems*, 2006.
- [66] J. Peters and S. Schaal. Reinforcement learning for parameterized motor primitives. In *Proceedings of the International Joint Conference on Neural Networks*, pages 73–80, 2006.

- [67] Sensable: PHANTOM. <http://www.sensable.com>.
- [68] N.S. Pollard and V.B. Zordan. Physically based grasping control from example. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 311–318, USA, 2005.
- [69] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.
- [70] G. Robles-De-La-Torre and V. Hayward. Force can overcome object geometry in the perception of shape through active touch. *Nature*, 412, 2001.
- [71] E. Sallnäs, K. Rasmus-Gröhn, and C. Sjöström. Supporting presence in collaborative environments by haptic force feedback. *ACM Transactions on Computer-Human Interaction*, 7:461–476, 2000.
- [72] T. Sato, M. Hashimoto, and M. Tsukahara. Synchronization based control using online design of dynamics and its application to human-robot interaction. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pages 652–657, 2007.
- [73] Stefan Schaal. Learning from demonstration. *Advances in Neural Information Processing Systems*, 9:12–20, 1997.
- [74] Tekscan Flexiforce sensor online specifications. <http://www.tekscan.com/flexiforce.html>.
- [75] Interlink Force Sensors. <http://www.interlinksensors.com/>.
- [76] T. Sheridan. Musings on telepresence and virtual presence. *Presence: Teleoperators and Virtual Environments*, 1:120–126, 1992.
- [77] T. Sheridan. Further musings on the psychophysics of presence. *Presence: Teleoperators and Virtual Environments*, 5:241–246, 1996.
- [78] M. Shimazoe and Y. Matsuoka. Load cell. *US Patent 4,454,771*, 1984.
- [79] M. Slater. Measuring presence: A response to the witmer and singer presence questionnaire. *Presence: Teleoperators and Virtual Environments*, 8:560–565, 1999.
- [80] M. Slater. How colorful was your day? why questionnaires cannot assess presence in virtual environments. *Presence: Teleoperators and Virtual Environments*, 13:484–493, 2004.
- [81] M. Slater and M. Garau. The use of questionnaire data in presence studies: Do not seriously likert. *Presence: Teleoperators and Virtual Environments*, 16:447–456, 2007.
- [82] M. Slater, D.P. Marcos, H.H. Ehrsson, and M.V. Sanchez-Vives. Inducing illusory ownership of a virtual body. *Frontiers in Neuroscience*, 3(2):214–220, September 2009.

- 
- [83] Mel Slater. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical Transactions of the Royal Society of London B*, Available at: <http://www.cs.ucl.ac.uk/staff/m.slater/Papers/rss-prepublication.pdf/>, In Press.
- [84] SmithMicro. Poser. <http://www.smithmicro.com/poser>.
- [85] J. Solis, S. Marcheschi, A. Frisoli, C. Avizzano, and M. Bergamasco. Reactive robot system using a haptic interface: an active interaction to transfer skills from the robot to unskilled persons. *Advanced robotics*, 21(3-4):267–291, 2007.
- [86] B. Spanlang, T. Fröhlich, V.F. Descalzo, A. Antley, and M. Slater. The making of a presence experiment: Responses to virtual fire. *Presence*, 2007.
- [87] M. Srinivasan and C. Basdogan. Haptics in virtual environments: taxonomy, research status, and challenges. *Computer graphics*, 21(4):393–404, 1997.
- [88] M. Srinivasan, G. Beauregard, and D. Brock. The impact of visual information on the haptic perception of stiffness in virtual environments. In *Proceedings of the ASME Dynamic Systems and Control Division*, pages 555–559, 1996.
- [89] S. Subramanian, L. Knaut, C. Beaudoin, B. McFadyen, A. Feldman, and M. Levin. Virtual reality environments for post-stroke arm rehabilitation. *Journal of Neuro-Engineering and Rehabilitation*, 4(20), 2007.
- [90] Y. Sun, J. M. Hollerbach, and T. Kesavadas. Measuring fingertip forces by imaging the fingernail. In *Haptics, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 125–131, 2006.
- [91] Tekscan 4255 tactile sensor. *User Manual*.
- [92] R. Tajima, S. Kagami, M. Inaba, and H. Inoue. Development of soft and distributed tactile sensors and the application to a humanoid robot. *Advanced Robotics*, 16:381–397, 2002.
- [93] T. Takeda, K. Kosuge, and Y. Hirata. HMM-based dance step estimation for dance partner robot-MS DanceR. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS) 2005*, pages 3245–3250, Canada, 2005.
- [94] T. Tsuji, P.G. Morasso, K. Goto, and K. Ito. Human hand impedance characteristics during maintained posture. *Biological Cybernetics*, 72(6):475–485, 1995.
- [95] M. Ueberle, N. Mock, and M. Buss. Vishard10, a novel hyper-redundant haptic interface. In *Proceedings of HAPTICS '04, 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 58–65, 2004.
- [96] M. Usuh, E. Catena, S. Arman, and M. Slater. Using presence questionnaires in reality. *Presence: Teleoperators and Virtual Environments*, 9(5):497–503, 2000.

- [97] N. Vanello, V. Hartwig, M. Tesconi, E. Ricciardi, A. Tognetti, G. Zupone, R. Gassert, D. Chapuis, N. Sgambelluri, E. P. Scilingo, G. Giovannetti, V. Positano, M. F. Santarelli, A. Bicchi, P. Pietrini, D. D. Rossi, and L. Landini. Sensing glove for brain studies: Design and assessment of its compatibility for fMRI with a robust test. *IEEE / ASME Transactions on Mechatronics*, 13(3), 2008.
- [98] Voice-O-Matic. <http://www.voice-o-matic.com/>.
- [99] Z. Wang and M. Buss. Preliminary results in human-robot handshaking. In *Proceedings of The 2nd Annual Workshop of CoTeSys, "Cognition for Technical Systems"*, Germany, 2008.
- [100] Z. Wang, J. Hoelldampf, and M. Buss. Design and performance of a haptic data acquisition glove. In *Proceedings of the 10th Annual International Workshop on Presence*, pages 349–357, Spain, 2007.
- [101] Z. Wang, J. Lu, A. Peer, and M. Buss. Influence of vision and haptics on plausibility of social interaction in virtual reality scenarios. In *Proceedings of EuroHaptics 2010*, Amsterdam, Netherlands, 2010.
- [102] Z. Wang, A. Peer, and M. Buss. An HMM approach to realistic haptic human-robot interaction. In *Proceedings of World Haptics*, pages 374–379, USA, 2009.
- [103] Z. Wang, J. Yuan, and M. Buss. Modelling of human haptic skill: a framework and preliminary results. In *Proceedings of the 17th IFAC World Congress*, pages 14761–14766, Korea, 2008.
- [104] M. Weber, K. Patel, O. Ma, and I. Sharf. Identification of contact dynamics model parameters from constrained robotic operations. *J. of Dynamic Systems, Measurement, and Control*, 128(2):307–318, 2006.
- [105] B.G. Witmer and M.J. Singer. Measuring presence in virtual environments- a presence questionnaire. *Presence: Teleoperators and Virtual Environments*, 7(3):225–240, 1998.
- [106] D.M. Wolpert and Z. Ghahramani. Computational principles of movement neuroscience. *Nature Neuroscience*, 3:1212–1217, 2000.
- [107] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 379–385, 1992.
- [108] Y. Yamato, M. Jindai, and T. Watanabe. Development of a shake-motion leading model for human-robot handshaking. In *Proceedings of the SICE Annual Conference 2008*, pages 502–507, Japan, 2008.
- [109] J. Yang, Y. Xu, and C. Chen. Human action learning via hidden markov model. In *IEEE Transactions on Systems, Man, and Cybernetics*, pages 34–44, 1997.

- 
- [110] R. B. Zipin. Multi-axis load cell. *US Patent 3,939,704*, 1976.
- [111] R. Zoellner, O. Rogalla, and R. Dillmann. Integration of tactile sensors in a programming by demonstration system. In *Proceedings of IEEE International Conference on robotics and automation*, pages 2578–2583, Seoul, Korea, 2001.