

Lehrstuhl für Steuerungs- und Regelungstechnik
Technische Universität München
Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss

Training in Virtual Environments via a Hybrid Dynamic Trainer Model

Hasan Esen

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. Dr.-Ing. Alexander W. Koch

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss
2. Associate Prof. Dr. Ken'ichi Yano, Gifu Univ./Japan

Die Dissertation wurde am 21.06.2007 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 17.09.2007 angenommen.

Foreword

This thesis is a result of four years of work in the research group of my thesis adviser Prof. Martin Buss. The seed of the work was planted in Berlin in 2003, while Prof. Ken'ichi Yano was a guest at the Control Systems Group of Technische Universität Berlin; it was full-fledged and completed at the Institute of Automatic Control Engineering, Technische Universität München.

First of all, I would like to thank my *Doktorvater* Prof. Martin Buss not only for the excellent research environment, his invaluable scientific advice and encouragement, but also for giving me the research assistant position which brought me financial comfort to organize my multi-cultural private life.

This work would not be possible without the fruitful discussions with Prof. Ken'ichi Yano. My cordial thanks to him for his bright ideas and advice, and for his friendship that was giving me additional power.

Many thanks to my colleague Dr. Marc Ueberle for his hard and fine work on developing ViSHaRD devices, which played an important role in this thesis. I am thankful to Angelika Peer for her help on any hardware or software problems concerning ViSHaRD10. Sincere thanks to Dr. Stanczyk, for always being there as a friend and also for extending my knowledge on Polish folk music. Many thanks to Raphaela Groten and Klaas Klasing, who shared the office with me in the last six months and were very understanding about my "Dissertation Final Phase" anguish.

To all my students, especially to Andreas Sachsenhauser, Kai Börner, Elvin Eren, Dylan Ilg and Serkan Önder, I thank you very much for the extraordinary work and support.

I feel very lucky to have such a great family. My mother and my sisters were the greatest power for me, although being thousands of kilometers away. It is hard to find words to express my gratitude to them. *Sizlere herşey için çok çok teşekkür ediyorum.* My father is further away than thousands of kilometers, but felt like closest to me during the difficult moments.

My beloved wife Kasia, to whom I got married twice during my work, deserves an individual paragraph. Her proof-reading of my articles and dissertation undoubtedly increased the quality of my writing. Her joy of living, her never ending energy kept me alive and optimistic. *Dziękuję bardzo Kochanie.*

Munich, 2007.

Hasan Esen

anneme ve babama

...

Abstract

This thesis presents a novel virtual reality (VR) training concept that integrates the trainer or the trainer model into training sessions. As an extension to conventional VR training systems that rely only on realistic interaction, the students are given the chance to be corrected by the trainer in a multi-user schema. The trainer is connected to the same virtual environment as the student via an individual haptic display. As an alternative, task performing skill of the trainer is captured with hybrid identification methods and the trainer is replaced with the identified model allowing for a single-user training schema. Two different identification approaches are successfully applied and presented in this thesis: The weighted K-means clustering-based method and the stochastically switching dynamics method. Observations and corrections of the trainer or trainer model are multi-modal, i.e. can be represented in the form of visual, acoustic and/or haptic signals. The combination of these possible signals allows for the definition of different training strategies. Enhancing the training systems with extra features that are not available in a real task is investigated as well. Two different VR scenarios are developed as test-beds: A bone drilling medical training system and a push button system. The efficiency of the different training strategies is checked through a series of user tests. To assess the training results objectively, a metric depending on the distance between the trainer and student in n dimensional Euclidean space is introduced and applied. The results validate the efficiency and usability of the training strategies and hybrid identification methods.

Zusammenfassung

Vorliegende Dissertation stellt ein neuartiges Virtual Reality (VR) Trainingkonzept vor, das den Trainer oder ein Modell des Trainers in eine Trainingssitzung integriert. Als Erweiterung zu klassischen VR Trainingssystemen, die nur auf realistischer Interaktion basieren, haben die Studenten die Möglichkeit vom Trainer bzw. vom Trainermodell korrigiert zu werden. In sogenannten Multi-User Szenarien sind Trainer und Schüler durch zwei individuelle haptische Displays mit der virtuellen Umgebung verbunden. Die sensorische Fähigkeit des Trainers wird mittels der Methode der hybriden Identifikation erfasst. In dieser Arbeit werden zwei unterschiedliche Ansätze vorgestellt und erfolgreich angewendet: Die sogenannte gewichtete K-means Clustering Methode und die stochastisch umschaltende Dynamik Methode. Nachdem das hybride dynamische Modell des Trainers entworfen worden ist, besteht die Möglichkeit den Trainer durch das Modell zu ersetzen. Beobachtung und Korrektur durch den Trainer bzw. das Modell erfolgen multimodal, d.h. in Form von visuellen, akustischen sowie haptischen Signalen. Die Kombination dieser Signale ermöglicht die Konzeption unterschiedlicher Trainingsstrategien. Desweiteren wird die Erweiterung der Trainingssysteme durch zusätzliche Funktionen, die in realen Systemen nicht existieren, diskutiert. Zwei VR-Szenarien werden als Testumfeld verwendet: Das Bohren von Knochen im Rahmen eines medizinischen Trainingssystems sowie das Betätigen virtueller Knöpfe. Die Wirksamkeit der Trainingsstrategien wird durch eine Reihe von Benutzerstudien untersucht. Um die Ergebnisse objektiv zu bewerten, wird eine Evaluierungsmethode eingeführt, die auf einem Maß im n dimensionalen Euklidischen Raum basiert, das die Distanz zwischen Trainer und Schüler beschreibt. Die Ergebnisse bestätigen die Effizienz und Anwendbarkeit der in dieser Dissertation vorgestellten Trainingssysteme.

“The mythical story of Icarus and Daedalus is usually related to a warning about flying too high because the heat of the sun would melt the glue used to hold together the feathers of the wings - this is probably not a correct interpretation of the warning. The more probable warning was about the danger of flying too high before adequate training...”

Ray L. Page

Contents

1	Introduction	1
1.1	Problem Definition and Motivation	1
1.2	State of the Art	4
1.2.1	Medical Simulators with Haptics	4
1.2.2	VR Training Systems with Haptics	6
1.3	Main Contributions and Outline of the Thesis	8
2	Virtual Environments for Training	11
2.1	State of the Art	12
2.1.1	Collision Detection	12
2.1.2	Haptic Rendering	13
2.2	General Principles of VR Development	15
2.2.1	Collision Detection and Haptic Rendering	15
2.2.2	Bounding Boxes and Binary Search Trees	18
2.2.3	Physically Deformable Objects	18
2.2.4	Stability of Interaction with Virtual Environments	22
2.3	Multi-User Virtual Environments	23
2.4	Bone Drilling Medical Training System	25
2.4.1	Bone Drilling with a Pedal-Like Haptic Display	25
2.4.2	Bone Drilling with ViSHaRD3	26
2.4.3	Bone Drilling with ViSHaRD10	27
2.4.4	Kinematics Mapping	33
2.5	Push-Button Training System	34
2.6	Summary	35
3	Trainer Skill as a Hybrid Dynamical System	37
3.1	State of the Art: Methods to Model Human Skill	37
3.1.1	Linear and Non-linear Modeling Methods	38
3.1.2	Skill Segmentation and Skill Primitives	38
3.1.3	Stochastic Methods	39
3.1.4	Representing Skill as Impedance and Hybrid Automata	39
3.2	Human Demonstration Data Clustering	40
3.2.1	K-Means Clustering	40
3.2.2	Weighted K-Means (WKM) Clustering in Parameter Space	41
3.2.3	Overview of Other Clustering Methods	43
3.3	Hybrid Identification of the Trainer Skill	44
3.3.1	Skill as Switching Generalized Impedances	44

3.3.2	Least Squares Optimization for Impedance Models	45
3.3.3	WKM-Based Approach	46
3.3.4	Stochastically Switching Impedances (SS-Imp) Approach	47
3.3.5	Comparison of the Methods	52
3.4	Stability of the Hybrid Models	54
3.5	Applications	55
3.5.1	Hybrid Trainer for a Bone Drilling Medical Training System	55
3.5.2	Hybrid Trainer for a Push-Button Training System	64
3.6	Discussion	69
4	Training Methods to Transfer Human Skill via Virtual Environments	73
4.1	Enhanced Training Methods	75
4.1.1	Visual and Acoustic Enhancement	75
4.1.2	Haptic Enhancement	75
4.2	Multi-User Training Methods	76
4.2.1	Overview	77
4.2.2	Haptic-Visual Observation (HVO), Verbal Influencing	78
4.2.3	Force Demonstrating (FD) Method	79
4.2.4	Force/Velocity Leading (FVL) Methods	80
4.3	Training via a Trainer Model	82
4.3.1	Overview	82
4.3.2	HVO, FD and FVL Methods using the Trainer Model	82
4.3.3	Simulation of FVL Method Including Trainer Model	86
4.3.4	Real-Time Experiment Results of Trainer-in-the-Loop	88
4.3.5	Including Student Psychology: Adaptive Trainer Model	91
4.4	Discussion	94
4.4.1	Comparison of the Methods	95
4.4.2	Application of the Methods in VR Training Systems	96
4.4.3	Application of the Methods in Telepresence	97
5	User Tests and Their Objective Evaluation	99
5.1	Evaluation Methods	99
5.1.1	Using Pre-defined Skill Levels	100
5.1.2	Euclidean Distances Between Trainer and Student	100
5.2	Integration of Virtual Environments, Hybrid Skill Identification and Training Methods	102
5.3	User Tests	103
5.3.1	Experimental Conditions	103
5.3.2	Tests on Enhanced Training Methods	106
5.3.3	Tests on Multi-User Training Methods	110
5.3.4	Tests on Training via a Hybrid Trainer Methods	115
5.4	Discussion	120
6	Conclusions and Future Work	123
6.1	Concluding Remarks	123
6.2	Future Work	127

A	Defining Polygonal Objects in SOLID	129
B	Overview of Other Developed Virtual Reality Applications	131
B.1	Saw Blade Deformations Using FEM	131
B.1.1	Saw Blade FE Model	131
B.1.2	Finite Element Deformation of Quadrilaterals	131
B.2	Deformable Box Using Mass-Spring Network	132
B.3	Rigid Wall - Virtual Stairs	132
B.4	Multi-User VR Scenarios	133
B.4.1	Pull a Spring	133
B.4.2	Load Transportation	133
B.5	A Graphical User Interface for VR Applications	134

Notations

Abbreviations

AABB	Axis-Aligned Bounding Boxes
BD-MTS	Bone Drilling Medical Training System
BST	Binary Search Trees
DoF	Degree(s) of Freedom
EE-CS	End-Effector Coordinate System
FD	Force Demonstrating Method
FEM	Finite Element Modeling
FVL	Force/Velocity Leading Method
GUI	Graphical User Interface
HIP	Haptic Interface Point
HIDP	Hybrid Identification Problem
HMM	Hidden Markov Model
HVO	Haptic Visual Observation
RT	Real-Time
SS-Imp	Stochastically Switching Impedances
VE	Virtual Environment
VR	Virtual Reality
W-CS	World Coordinate System
WKM	Weighted K-Means

Conventions

Trainer and Expert are used as synonyms.

Trainee and Student are used as synonyms.

Scalar shown human - virtual reality interaction parameters underline that the interaction is possible or assumed to be 1D.

Scalars, Vectors, and Matrices

Scalars are denoted lower case letters in italic type. *Vectors* are denoted by lower case letters in boldface type, as the vector \mathbf{x} is composed of elements x_i . *Matrices* are denoted by upper case letters in boldface type, as the matrix \mathbf{A} is composed of elements a_{ij} (i -th row, j -th column).

x	scalar
$x, x(t)$	continuous signal
$x^*, x(k)$	discrete signal
\mathbf{x}	vector
\mathbf{X}	matrix
$f(\cdot)$	scalar function
$\mathbf{f}(\cdot)$	vector function
$\text{var}(\cdot)$	variance
$\text{cov}(\cdot)$	covariance
$\dot{\mathbf{x}}, \ddot{\mathbf{x}}$	equivalent to $\frac{d}{dt}\mathbf{x}$ and $\frac{d^2}{dt^2}\mathbf{x}$
\bar{x}	mean of x
$P(O \lambda)$	probability of observing O in HMM model λ

Subscripts and Superscripts

$G(s)$	linear transfer function in Laplace domain
x_{vr}	value x associated with the virtual environment
x_e	value x associated with the expert (the same as x_t)
x_t	value x associated with the trainer (the same as x_e)
x_s	value x associated with the student
x_d	desired value x
x_h	value x associated with the human operator
x_m	measured value x
x_{min}	minimum value of x
x_{max}	maximum value of x
${}^A\mathbf{R}_B$	mapping from coordinate system B to A

Symbols and Abbreviations

α	HMM forward algorithm
\mathbf{A}	state transition probability distribution
b_{md}	inherited mechanical damping of the haptic display
B_{dg}	digital damping
\mathbf{B}_{e1}	strain-displacement matrix of element-1
\mathbf{b}	observation probability distribution
β	HMM backward algorithm
c_s	scaling factor for HMM iterations
d_i	i^{th} damping coefficient
D_i	i^{th} cluster
D_{ts}	distance between the trainer and the student in n dimensional Euclidean space
\mathbf{E}	material matrix
E_{ym}	Young's modulus
E_u	specific cutting energy
\mathbf{e}	model error
$\boldsymbol{\varepsilon}_{e1}$	strain vector of element-1 in FEM
ε	threshold
\mathbf{f}	force
f	force along the collision direction
Δf	force difference between the trainer and the student
$f_{i,0}$	offset force of i^{th} impedance model
\hat{f}_i	output force of i^{th} impedance model
\mathbf{f}_{ij}	internal force on node i through the spring between nodes i and j
\mathbf{f}^e	external force
\bar{F}_w	force mean value for whole simulation duration
\bar{F}_{corr}	force mean value, while the applied force is correct
\bar{F}_{dr}	force mean value after collision occurs, until the task is completed (when drilling starts/finishes in BD-MTS)
γ_1	scaling factor in FVL training for impedance type VE
γ_2	scaling factor in FVL training for admittance type VE
\mathbf{J}	Jacobian matrix
J_c	cost function
k_i	i^{th} spring coefficient
K_{max}	maximum achievable stiffness
\mathbf{K}_{e1}	stiffness matrix for element-1
\mathbf{K}_g	global stiffness matrix
K_p	proportional gain
$K_{p,nom}$	nominal proportional gain
λ	hidden Markov model
L	degree of student's success
L_{st}	skillful student level
m	mass
$\hat{\boldsymbol{\mu}}_i$	randomly initialized cluster centers
$\bar{\boldsymbol{\mu}}_i$	cluster centers
$\tilde{\boldsymbol{\mu}}_i$	weighted cluster centers

\mathbf{n}_t	normal vector
ν	Poisson's ratio
n_m	model order
n	number of data points
N_i	i^{th} shape function
\mathbf{O}	observation sequence
$\boldsymbol{\pi}$	most probable initial state
\mathbf{p}_t	trainer point in n dimensional Euclidean space
\mathbf{p}_s	student point in n dimensional Euclidean space
\mathbf{Q}_j	covariance matrix for data points in j^{th} local model
\mathbf{q}	robot link angles
\mathbf{r}_i	position vector of node i
σ_i^2	variance of the i^{th} column parameters of \mathbf{X}
$\hat{\sigma}_j^2$	estimated variance for the j^{th} local model parameters
$\boldsymbol{\Sigma}$	weighting matrix
s	number of discrete states
sw_{ij}	switching condition to switch from state i to state j
S_r	student reaction to the trainer's correction signal T_c
$\boldsymbol{\theta}$	model parameters vector
T	refresh rate period
T_c	correction signal
T_w	whole simulation duration
T_{corr}	duration, while the applied force is correct
T_{dr}	duration between collision occurs and task is completed (when drilling starts/finishes in BD-MTS)
$T_{simstart}$	time point when the collision occurs
$T_{drillstart}$	time point when the task execution starts (drilling start in BD-MTS)
$T_{simstop}$	time point when the task is completed
\mathbf{u}	input vector
\mathbf{u}_g	global deformation vector
\mathbf{u}_{e1}	displacement vector for element-1
\mathbf{U}	state sequence
\mathbf{v}_i	velocity vector of node i
V	energy storage function
\mathbf{V}_j	parameter covariance matrix for j^{th} local model
v	thrust velocity
\bar{v}_w	velocity mean value for whole simulation duration
\bar{v}_{corr}	velocity mean value, while the applied force is correct
\bar{v}_{dr}	velocity mean value after collision occurs, until the task is completed (when drilling starts / finishes in BD-MTS)
ω	weighting factor
ξ, η	natural (quadrilateral) coordinates
$\boldsymbol{\xi}_j$	feature vector for clustering
x, y, z	cartesian coordinates
\mathbf{x}	position vector
\mathbf{X}	regression matrix
\mathbf{X}_j	j^{th} row of the regression matrix
\mathbf{y}	output vector

List of Figures

1.1	The Anatomy Lesson of Dr. Tulp, Rembrandt - 1632	2
1.2	Training in virtual environments concept	4
1.3	The research triangle and the target.	8
2.1	Main modalities and rendering issues explored in this work to create a VE.	11
2.2	Evolution of representing the human hand in VEs.	12
2.3	Point-triangle collision detection in 3D space	16
2.4	Ray-triangle collision detection in 3D space.	17
2.5	Bounding boxes	18
2.6	Axis aligned bounding boxes and binary search trees	19
2.7	A simple geometric object divided into four quadrilaterals.	20
2.8	Two-port communication network	23
2.9	Multi-user VR concept	24
2.10	Bone drilling with a pedal-like haptic display	26
2.11	Bone drilling with ViSHaRD3	27
2.12	Flow chart for acoustic feedback	28
2.13	The concept of BD-MTS	28
2.14	Non-linear haptic rendering function for BD-MTS	30
2.15	Bone drilling with ViSHaRD10	31
2.16	Three main blocks of the control structure in BD-MTS	32
2.17	Coordinate systems for ViSHaRD10 and virtual scene	34
2.18	Virtual push-button task	34
3.1	Main methods to model human skill.	37
3.2	Trainer skill as switching impedances.	44
3.3	A left-to-right HMM representation	47
3.4	Example ARX system clustered by K-means algorithm	53
3.5	Example ARX system clustered by WKM algorithm	54
3.6	BD-MTS, normalized absolute model errors	56
3.7	BD-MTS, intensity analysis results	57
3.8	Identified expert impedance parameters for BD-MTS	58
3.9	Identified switching conditions via WKM method for BD-MTS	60
3.10	WKM model error in BD-MTS	60
3.11	Switching conditions obtained using SS-Imp in BD-MTS	62
3.12	Clusters identified via SS-Imp method in BD-MTS	63
3.13	SS-Imp model error in BD-MTS	63
3.14	Clustering costs	65
3.15	Identified expert impedance parameters for push-button example	66
3.16	Identified clusters using WKM algorithm for push-button system	67

3.17	WKM model error in push-button system	67
3.18	SS-Imp switching in push-button example	68
3.19	SS-Imp model error in push-button system	68
4.1	Main training strategies	74
4.2	A training system for drilling with visual enhancement	76
4.3	Training in a multi-user manner	77
4.4	HVO method control scheme.	78
4.5	FD method control scheme.	79
4.6	FVL method control scheme	81
4.7	Training systems including hybrid dynamic trainer model	83
4.8	HVO training with a trainer model	83
4.9	FD training with a trainer model	84
4.10	FVL training with a trainer model	85
4.11	FVL method simulation results for push-button example	87
4.12	Error comparison when the FVL trainer is in feed-forward and feedback way.	88
4.13	FD training results for push-button example	89
4.14	FVL training results for push-button example	90
4.15	FD and FVL training results for bone drilling example	91
4.16	Student gain for adaptive HVO method	93
4.17	Student gain for adaptive FVL method	94
5.1	Expected Euclidean distance change	101
5.2	Integrated system	102
5.3	Vienna sensorimotor coordination test setup	104
5.4	Offline graphs to inform students	105
5.5	Skill improvement via haptic and visual enhancement	107
5.6	Improvement in applied force	107
5.7	Skill improvement via haptic, visual and acoustic enhancement	108
5.8	Comparison of enhanced training strategies	109
5.9	Parameter distance to define Euclidean space	110
5.10	Different student performances during multi-user training	113
5.11	Comparison of the first trials after first training session	114
5.12	Comparison of multi-user training methods	114
5.13	Different student performances during training via a trainer model	115
5.14	Comparison of average skill improvement	116
5.15	Skill improvement with HVO training including hybrid trainer model	117
5.16	Skill improvement with FD training including hybrid trainer model	118
5.17	Skill improvement with FVL training including hybrid trainer model	118
5.18	Impedance parameters for students after the last training	119
B.1	Deformation of a saw blade using FEM.	131
B.2	Deforming a virtual box using mass-spring network.	132
B.3	Virtual stairs.	133
B.4	A virtual task for two users: pulling a spring.	133
B.5	Collaborative load transportation	134
B.6	Graphical user interface for VR applications.	135

List of Tables

2.1	The numerical integration points and weights	22
2.2	Specifications of ViSHaRD3.	27
2.3	Attributes of skull and drill models	29
2.4	Specifications of ViSHaRD10	30
2.5	The parameters for virtual push-button application	35
3.1	Example ARX system parameters identified by K-means method	53
3.2	Example ARX system parameters identified by WKM method	53
3.3	Example ARX system parameters identified by SS-ARX method	54
3.4	Expert model parameters obtained via WKM method for BD-MTS	61
3.5	Expert model parameters obtained via SS-Imp method for BD-MTS	64
3.6	Expert model parameters obtained via WKM method for push-button	66
3.7	Expert model parameters obtained via SS-Imp method for push-button	68
5.1	Possible parameters to define the Euclidean evaluation space.	111
5.2	Parameters of the Euclidean evaluation space and their weighting factors	112

1 Introduction

1.1 Problem Definition and Motivation

It is of crucial importance to enhance theoretical teaching with practical applications in the education of occupations that require high sensorimotor skills. However, there are many occupations that offer limited practicing chances. The only way to acquire experience in these fields is to carry the real tasks through, depending on the theoretical knowledge and the observations. In some cases, this is a normal procedure to train, not bearing any potential problem. If one is interested in producing earthenware pottery, after getting instructions and observing a skillful pottery producer, nothing more is needed, but baked clay and a turntable. It is highly unlikely that the pottery making student puts any life into danger, or causes any drastic environmental, physical, and/or financial damages.

Nevertheless, this assumption is not valid if the human is training for medical operations, becoming a pilot, or carrying out safety-related tasks in a nuclear plant etc. In such cases, any small mistake may cause fatal results, not mentioning other possible damages. Therefore, it is absolutely essential to offer alternative training possibilities in these areas. One method to satisfy this need is to create so called simulators, which try to replicate the real cases with high-fidelity to the original task. The history of simulators starts with flight simulators, and dates back to the beginning of the 20th century. The first examples, like “Sanders Teacher” and “Eardly-Billing Oscillator” were replicas of complete aircrafts that were mounted on a universal joint and facing into wind that excites pitch, roll, and yaw movements [45]. Today’s simulators exhibit a broad range from system trainers for home computers to the 6 degrees-of-freedom motion simulators, in which the flight equations of motion and realistic aerodynamic forces are implemented in a digital environment [93].

Beyond question, an important field with a need and potential for simulators is medicine. A classical way of surgery training is to observe experienced surgeons during operations. Rembrandt’s famous painting “The Anatomy Lesson of Dr. Nicolaes Tulp” dated 1632, shows how the expert, Dr. Tulp, explaining human anatomy on a cadaver to his students. It is also possible to assume that the scene from the painting shown in Figure 1.1 is an operation conducted on the arm of a patient by an expert while students are observing him. This observation-based method is still valid today and undoubtedly useful for the novices. However, students cannot acquire any hands-on experience only through observations. Especially, if one considers that the word “surgery” derives from Greek “*cheirurgia*”, and means “hand work”, the deficiency of the observation method becomes obvious. Thus, experiments on cadavers or animals, whose roots date back to ancient Egypt, become an important replica for real operations. Nevertheless, strict ethical rules in many countries limit such experiments or even completely forbid them. Additionally, the characteristics of cadaver or animal tissue differ from the characteristics of a living human body, i.e. the reliability of surgery training on cadavers or animals is questionable. Thus, artificial human organs and mock-ups have been considered as alternative training tools. Today, there is a large market for such mock ups varying in quality, reliability and price. The

high quality types are mostly used by experienced surgeons to plan important operations. Allowing students to practice operations with such models is not affordable. Moreover, such mock-ups lack modularity, i.e. different pathological and patient specific cases cannot be demonstrated [99].



Figure 1.1: Rembrandt (1606-1669), *The anatomy lesson of Dr. Nicolaes Tulp*, 1632. Canvas, 169.5 × 216.5 cm, Royal Picture Gallery Mauritshuis, The Hague.

Starting from the 1980's, a new dimension was brought to the medical training: *Virtual Environments* (VEs). Simulating a surgical case and human organs in digital environments opened new doors in the education of medical operations. While the first examples relied only on realistic visualizations, starting from the 90's multi-modal *Virtual Reality* (VR) systems began to appear. At that time, the real breakthrough came with the inclusion of haptic feedback in the VR systems. Therewith, it became possible to interact with the virtual organ models via force feedback devices, and realize a surgical operation in a VE. In the last decade, researchers have intensively been working on creating multi-modal VR systems that exhibit high-fidelity force, visual and acoustic feedback concerning the surgery case in focus. It is still a challenging research area that yields many interesting problems such as modeling and simulating human organ deformations, calculating response forces of organs when they are interacted with, fast and reliable collision detection and haptic rendering techniques etc. The main aim of all these works is to make such simulators as realistic as possible, so that the students can transfer the experience they gain from simulators to real cases. Additionally, experienced surgeons may prepare themselves for extraordinary surgical cases [99].

State of the art in medical simulators and VR training systems is presented in Section 1.2. To this end, it is necessary to clarify the terminology and underline the difference between a simulator and a training system. A **simulator** is a mock-up or digital environment that replicates original tasks as realistically as possible. On the other hand, a **training system** concept includes a simulator, but additionally pays special attention to how to train a novice, and focuses on the issues such as training strategies, skill transfer from an expert and objective evaluation of the student’s skill improvement.

In this sense, a large amount of previous works in the area should rather be called simulators. Considering the potential risks of gaining experience directly on real tasks, it is straight-forward to conclude the necessity and the importance of the practical training on VR simulators before performing the real tasks. In our work, the key question that we raise is whether or not it is sufficient to have a VR system that provides realistic interactions for an adequate training. In other words, does a simulator suffice for a novice to train for real tasks? Our short answer to this question is “no” and the research focus lies behind this answer. Undoubtedly, it is useful to have a simulator for a certain task and experience the interaction awaiting in the real operation. However, in this case, there is little chance to learn how the correct operation should be carried out. Thus, it is necessary to have a tutoring structure to observe, warn, correct and lead the student to perform the task in a correct way. Especially in medicine such a tutoring structure is very important. Consequently, a VR training system must firstly aim at teaching students how to realize the task, and then check the skill improvement via realistic interactions. Thus, we focus on creating a training system, introducing novel training strategies and performance evaluation methods. A special focus in this thesis is on force skill training, which is a barely investigated topic.

Beyond question, an important factor during practical trainings is the trainer. How the trainer himself/herself realizes the task plays a key role, since a well-trained student should be realizing the task on the same level as the trainer. This is an underestimated point, and neglected in currently available training systems. Few works concern multi-user VR training systems, in which the trainer is connected to the same virtual scene as the student to teach him/her [86, 89]. However, none of the research groups has established a detailed investigation of methods to correct and teach the student. In this work, we present a general multi-user VR training concept that offers many different ways of interaction between the student and the trainer. Moreover, we introduce a unique training concept that includes not the trainer himself/herself, but the model of him/her. Previously, there have been investigations on modeling human skill for programming-by-demonstration applications with robots, or for robots to assist and to cooperate with humans [65, 116, 121, 138]. However, no previous work is known to us that integrates trainer skill models into the VR training systems. Moreover, many previous human skill modeling efforts were based on stochastic methods, using *hidden Markov models* (HMMs) [53, 136] or *neural networks* (NNs) [7, 87] to represent human skill. Few deterministic modeling approaches exist in the literature, but they tend to be heuristic [55, 65, 81]. In this work, we present hybrid dynamic modeling of the human skill and its integration into the VR training systems. Although it is not discussed in the frame of this thesis, as an add-on, such a structure has the potential of storing the skill of an extraordinarily skillful trainer to be transferred to future generations.

Inspired by Dr. Tulp’s anatomy lesson, our concept for VR training systems is depicted in Figure 1.2. The students are interacting with the virtual human model via a haptic

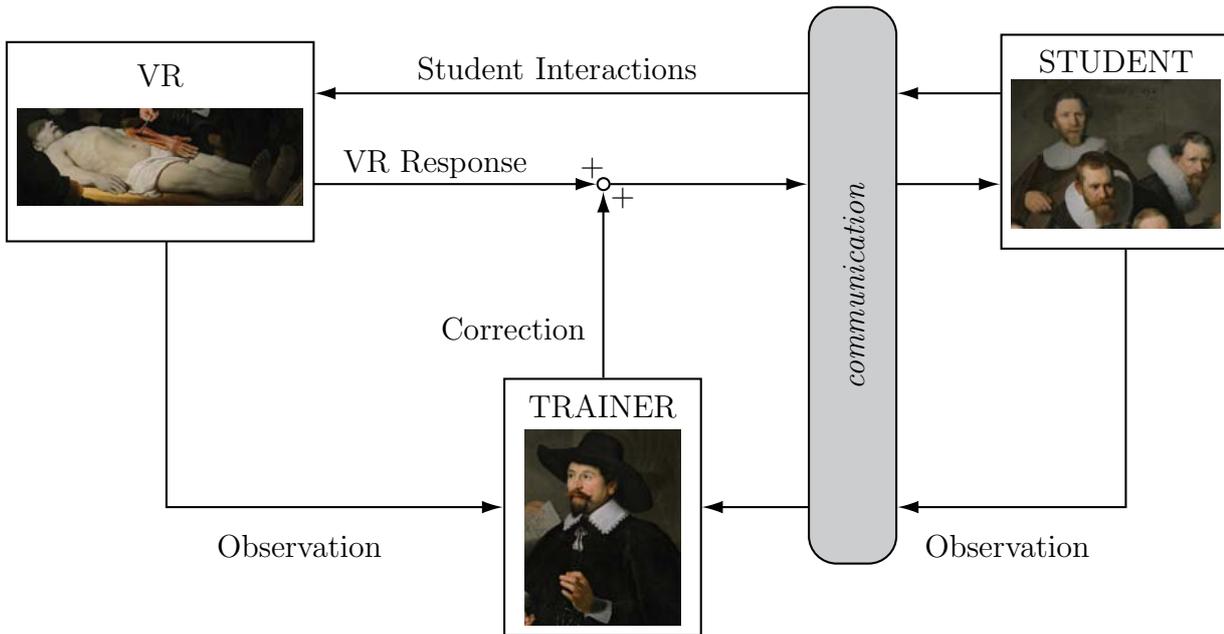


Figure 1.2: Training in VEs concept including the trainer or the model of the trainer, inspired by Dr. Tulp's Anatomy Lesson.

display, which is operated by a student to control the virtual medical tool in the virtual scene. When the tool and the human organ collide, the student starts realizing a surgical task, during which he/she feels the appropriate forces, sees matching visual changes, and hears the suitable sound. In other words, the interaction is multi-modal, and as realistic as possible. At the same time, the trainer, Dr. Tulp himself or his model is integrated into the system. In the first case, a multi-user scheme is constructed so that the trainer connects to the training environment via his individual haptic display. He has a chance to observe and correct the student haptically. If his model is used, the same observation/correction task is then realized by the model.

Note that our training system concept includes, but is not limited to medical training applications. It aims to serve as a basis for any VR training system that incorporates force skill training. Our application examples presented in this thesis are a virtual push-button task teaching and a bone drilling medical training system. The former one allows for testing different modeling methods and training strategies, whereas the latter one, together with verifying the success of modeling and training strategies, offers a setup that allows for assessment of the skill improvement of students by proposed methodologies.

1.2 State of the Art

1.2.1 Medical Simulators with Haptics

Developing a complete VR simulator is a complex process that requires interdisciplinary work. The involved disciplines range from medicine, biomechanics, computer science to mechatronics, control engineering, psychology and psychophysics. Therefore, the development process is a research team work, combining diverse professions.

Over the last decade there has been a growing interest in simulating the surgeries in a VE. As a result, grouping and classifying the previous and on-going research is rather challenging. In a general manner, we divide the simulators into three groups based on the surgery technique that was simulated: Minimally Invasive Surgery (MIS) simulators, Open Surgery (OS) simulators, and cutting / suturing simulators.

There exist a significant amount of active research work to create MIS simulators. Kühnapfel created the 3D graphical simulation program KISMET in 1991 [66]. Using KISMET, Kühnapfel *et al.* developed a VR endoscopic training simulator including deformable tissue simulation [67]. The simulation is applied in gynecology and laparoscopy. Later on, the simulator was further refined by paying the attention to elastodynamically deformable tissue models and adding the pulse palpation feature with a force feedback device [68]. Basdogan *et al.* developed a similar surgical training concept [11]. One of the recent works belongs to Papadopoulos *et al.* and concerns a urological operation training simulator [94]. Since MIS simulators introduce interaction with soft tissues, there are many research works incorporating deformable body simulations [9,16,35,129]. In Chapter 2, we will return to this point and give an insight into deformable object simulation in VEs.

Open Surgery (OS) procedures are much more complex to simulate than MIS procedures. As a result, there are fewer works dealing with OS simulation in comparison with MIS simulation. The exception is in orthopedy that yields many intriguing operations to simulate which are non-invasive. Sourina *et al.* describe a pipeline from a CT scanner, 3D visualization of the VE, collision detection through a Unix-workstation to the personal computer for a virtual orthopedy surgery training system [114]. Riener *et al.* introduce a knee simulator for training by using a bio-mechanical model of the knee [51,52,100]. Agus *et al.* work on bone dissection and they present haptic and visual simulation of the process [3]. They mainly focus on reliable bone-burr interaction modeling. Some other works concerning orthopedical operation simulation are [58,110].

Simulation of the cutting/suturing procedures deserves special attention, because of the topology change in the cut/sutured object. It means that the stiffness matrix of the deformable object model has to be adapted in real-time. Moreover the change in the topology of the object has to be visualized. Two example systems are worth consideration: Webster *et al.* present a prototype haptic suturing simulator [131], whereas Okamura *et al.* discuss a haptic tissue cutting problem using their unique device haptic scissors [133]. Berkley *et al.* present a new real-time methodology based on linear finite element analysis that is appropriate for a wide range of surgical simulation applications, especially for suturing procedures [13]. The main advantages of their methodology are high model resolution, low preprocessing time, unrestricted multi-point surface contact, and adjustable boundary conditions.

Some other works that do not fall into the categories described above are injection simulators [25,122], and the work of Obst *et al.* about a VR training system for baby delivery help with haptic feedback [91].

Above mentioned state of the art should be considered as some chosen examples of the latest medical simulators. There are many others to be found in the literature. In Chapter 2 and 3, we refer to some of these works. A detailed survey on surgical simulation can be found in [74]. Multi-modal VR medical training systems and their main components were discussed in [99]. For information on robotics for surgery and tele-surgery applications, refer to [78,101].

Note that although many of the above mentioned works contain the word “training system” in their title, they do not present detailed investigation about the components of surgical skills and training appropriate to these skills. Thus, following our terminology, they shall be called simulators.

1.2.2 VR Training Systems with Haptics

Although visual feedback has been accepted to be dominant in comparison with haptic feedback, Ernst *et al.* underline that the dominance is defined by the statistical reliability of the available sensory information. This definition suggests that if the visual information is less reliable to realize a task, than haptic cues play a more important role in motor skill learning procedures [29]. Nevertheless, there are very few works concerning the role of haptic feedback on training.

Jiang *et al.* investigate how to improve learning rates by haptic feedback for military and emergency personnel. They indicate that VEs do not need to be entirely realistic for a successful training. As in many other works, Jiang *et al.* also define a performance criterion and compare the training result with and without haptic feedback. They underscore that the personnel who trained with haptic feedback made fewer procedural errors [57].

One of the research efforts towards understanding the perceptual-motor skills and cognitive process involved in a surgical operation is given by Tendick *et al.* [120]. In the context of their training system for laparoscopic surgical skills, they indicate that it is essential to study technical and training aspects simultaneously in order to develop a VE training system. It was emphasized that the realism of the VE plays an important role, but understanding how surgeons perform tasks is no less important to develop training systems. In their work, the surgical skill is divided into two groups: perceptual motor skills and spatial skills. Simple VR scenarios are developed to train these skills. Another important issue they underline is that trying to simulate complete surgical cases realistically is not possible, moreover not necessary. Their idea is to create a training system only for critical steps of a surgical procedure. This work was later on extended for tele-surgery [19]. Schijven *et al.* introduce a basic surgical skill course [109]. Their work aims to train surgeons for particular psychomotor skills of an MIS procedure (laparoscopy). The authors mainly define some criteria to evaluate the level of training. But they neither investigate a human-machine interaction model, nor look for mathematical and generic training algorithms. Another important work that has to be mentioned is from Hayward *et al.*, who give a detailed analysis of haptic signals during surgical cutting. They also discuss measuring and displaying techniques of haptic signals [43].

An important work on the role of haptics in training belongs to Feygin *et al.*, who have introduced the *Haptic Guidance* method, and investigated the role of haptic and visual training comparatively on learning a complex 3D trajectory [38]. They define the haptic guidance paradigm as “the physical guidance of the subject through the ideal motion by the haptic interface, giving the subject a kinesthetic understanding of what is required”. The results indicate that haptic guidance can improve performance, especially for the temporal aspects of the task. However they have not discussed cases, when not only the motion tracking, but force itself is to be trained. Also, they assume that the student does not introduce any force himself/herself and freely allows to be guided by the reference haptic signal. Prada *et al.* also implement haptic guidance, in which the trainees are guided in a soft tissue cutting simulator by virtual fixtures to assist the training with graphic and

haptic feedback [95]. The guidance is implemented by point-attraction type fixtures that pull the avatar point to a special point to guide. As an extension, reactive type fixtures are used as well, in which the fixtures only constrain movements, but there is no guiding. The deficiencies mentioned above also exist in this example. Srimathveeravalli *et al.* introduce a new strategy for providing haptic assistance when training for a motor skill in a VE [115]. Their method is based on haptic record and replay fashion. The first question they ask is whether it is possible to capture the skill of an expert in this fashion and then transfer it to a student. The authors record the haptic profile of an expert writing a letter in a local Indian language called Tamil. Then, while a student writes the same letter, they compare student's haptic profile with teacher's. The error between profiles is scaled and returned to the student. This novel approach lacks some important relevant features: For each new task new expert data needs to be recorded, moreover each expert might perform the task in a different way, i.e. it is not possible to generate a common expert data. Moreover the known problem of not being able to control force and position at the same time is not referred to. It is assumed by the authors that the trainee and the expert have similar haptic profiles, so that the force profile of the expert can be replayed without taking the position trajectory into account. Since the haptic profiles are recorded on time axis, if the student is slower or faster than the trainer in writing, following the trainer would not make sense anymore. Yokokohji *et al.* have outlined several methods to show the correct force to a trainee for a virtual task in a single-user training scenario [139]. They have recorded the force and position information of an expert, and tried to replay them on a trainee using different approaches. However, they did not obtain any clear result, which is probably due to their application example.

All works referenced above investigate training strategies on single-user cases. Recently, there has been an increasing research interest in multi-user virtual training environments. The haptic guidance method has been implemented by a few researchers in a way that the remote trainer guides the student via his/her haptic display. For example, Morris *et al.* suggest a collaborative virtual training environment [86]. Their system allows two users to connect with the surgical scene, where one of the users (the expert one) monitors the performance of the other one (the trainee) and interactively provides feedback. A similar architecture is proposed by Niemeyer *et al.* [89]. By using tele-robotics principles, they enable a teaching surgeon to control a regular training procedure of a novice surgeon. Nevertheless, in both of these works, the authors limit themselves to mentioning the existence or the possibility of tele-haptic guidance. Unfortunately, no information is given on how to realize it. Gunn *et al.* investigate the use of collaborative haptics for surgical training as well [44]. In their work, two users, the teacher and the student, are located on different continents. Their sampling frequency for force is around 400 Hz, and they claim that it is sufficient. However, there is no evidence for this statement. The users had to fill in a questionnaire which indicated that the remote teaching was found very good or excellent. El-Far *et al.* discuss a tele-surgery multi-user training application in a collaborative VE, but only a conceptual overview is given in this work [28]. The expert does not seem to have a chance to affect the trainee haptically. Matsumoto *et al.* discuss network issues and the problems concerning network latency for haptic collaboration in shared VEs [77]. Rodriguez *et al.* discuss the components and the necessity of a collaborative virtual training system that contains haptic feedback [103]. Rossi *et al.* have recently released a pilot study on collaborative haptics [130] that enables haptic interaction of two users through a virtual object, putting the focus on time delay compensation. One of the latest works

of Morris *et al.*, discuss haptic tutoring as a novel training strategy. This is the only paper known to us, which discusses the difficulty of guiding both force and position at the same time, and suggests a solution for this problem [85]. They bind the tool position of the student to that of an instructor's, and via a low-gain spring they added forces to the playback forces. The method is very interesting, however, it is not discussed how to choose the spring coefficient. The cognition of the student has not been validated either. Last but not least, note that the student is passive in all haptic guidance based strategies, which is a perceived drawback. Modeling of trainer skill is required as a part of our VR training concept. Refer to Chapter 3 for the state of the art in human skill modeling.

1.3 Main Contributions and Outline of the Thesis

To bring our VR training concept into life, there are three main tasks to accomplish. First, VR training scenarios have to be generated that incorporate realistic interactions. Secondly, expert interaction with these VEs needs to be identified in order to establish a trainer model. Last, but not least, training strategies, which aim at force skill transfer from an expert or expert model to the trainee, have to be developed and implemented. Figure 1.3 illustrates these main tasks and research focal points. Integration of these points gives rise to the complete training system structure. Consequently, the organization of this thesis is based on these centers of attention.

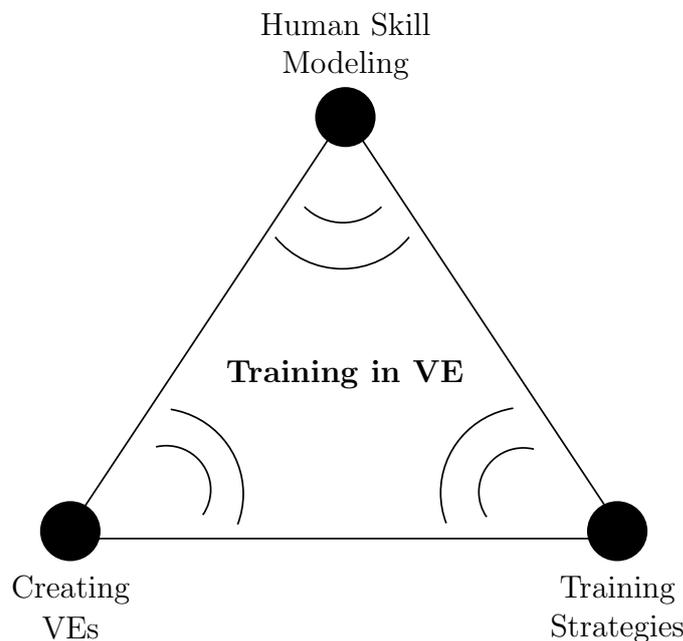


Figure 1.3: The research triangle and the target.

Chapter 2 presents the fundamental software and control engineering works necessary to generate a realistic VE. Some of the discussed issues are ray-based collision detection, principles of physically deformable objects and a multi-user VR system structure to be used for training purposes. Two VR applications, the bone drilling medical training system and a push-button training system, and their key principals are introduced. These examples

represent a test-bed for the developed skill modeling and skill transfer methods, and they are referred to throughout the thesis.

In Chapter 3, a novel representation of trainer skill based on hybrid identification methods is introduced. A competent trainer undoubtedly plays an important role in skill training procedures. Thus, integrating a trainer into VR training systems is a desirable extension. Recently, there have been a few works that investigate these issues in the frame of collaborative VEs as mentioned in Section 1.2. However, it is not always possible to assure the existence of an expert during training - learning activities. Therefore, in this work, we identify a hybrid dynamic model of the trainer, and integrate this model into the training systems, as an alternative to integrating the trainer himself/herself. This novel approach brings in many advantages. Since a multi-user scheme is not necessary anymore, hardware costs are reduced by half. More importantly, the modeled skills of a late expert can be used to train new generations. Likewise, extinct skills for specific tasks can be re-animated. It should be added here that having a possibility to train alone would provide the student with a more relaxed training environment. For hybrid identification of the trainer model, two different methods, one based on weighted K-means data clustering and the other one on stochastically switching principles are explained, compared with each other and applied to developed training systems. The identification methods need a proper regression matrix having position, velocity and/or acceleration as parameters, and an output vector that consists of measured force data. Having this information, the methods define the number of discrete states, impedance model parameters for each state and the switching conditions between the states. Thus, the trainer model is acquired as a set of switching generalized impedances. To speed up the identification process and for having physically interpretable clusters and discrete states, an intensity analysis is introduced. Additionally, for the stochastically switching impedances method, a criterion depending on the minimum model error is presented in order to define the optimal state sequence.

Chapter 4 summarizes broadly investigated force skill training strategies in VEs. Starting with classical VR simulator structure, first it is discussed how to enhance such a simulator in order to create a training system from it. In addition to visual and acoustic enhancement, a novel haptic enhancement principle is presented. This method should be considered as haptic cues that warn the student haptically, if he/she deviates from the force that should be applied. It is especially advantageous to train tasks, which do not have reliable visual information and are realized by mostly relying on haptic sense as in many surgical operations. Following enhanced training methodologies, attention is paid to multi-user training schemes, in which the trainer and the student connect to the same VR and interact with the VR and with each other through their individual haptic displays. Three main training schemes are outlined: Haptic Visual Observation (HVO), Force Demonstrating (FD) and Force/Velocity Leading (FVL). The first one enables an expert to observe the student not only visually, but also haptically, whereas the second one works mainly vice versa, i.e. the student observes the trainer. If one keeps in mind that one of the main deficiencies in classical surgery training is the lack of haptic feeling when observing an expert surgeon in an OP room, the developed strategies are very favorable. The last one, FVL, aims at providing a kind of haptic tutoring to the student. An important advantage of the method is that the student is active, i.e. he/she realizes the task himself/herself. Thus, an important drawback of the haptic guidance based methods is eliminated by the FVL method. Additionally, taking into account the difficulties to show correct force and velocity of a task to a student synchronously, presented methods offer an

interesting and new perspective in VR training. Afterwards, the usage of these schemes is explained when instead of the trainer himself/herself, his/her model is integrated into the training loop. Only few works discuss recording the force applied by an expert and later on replaying them on students [115, 139]. However, there is no other work known to us, which integrates the trainer model into the force skill training procedures. Last but not least, an adaptiveness concept of the training structures depending on the reaction of the student to the training signals is addressed in this chapter.

Chapter 5 is the integration of the concepts presented in Chapters 2, 3 and 4. This integration allows a complete representation of our VR training concept. Following that, organization of the user tests that were performed in order to assess the efficiency of the concept, and their results are presented. An objective evaluation criterion depending on the distance between the student and trainer in n dimensional Euclidean space is defined and used to assess the results of the user tests. This method provides a consistent assessment of the performance improvement of students. The user test results indicate the benefit of using the proposed training methods comparatively.

A conclusion, main results and future work are represented in Chapter 6. In the appendix, other available VR applications and a GUI developed during this thesis work are presented.

2 Virtual Environments for Training

A virtual environment (VE) training system has to stimulate all necessary human sensory modalities as realistically as possible, i.e. it has to be multi-modal. As it is given in Figure 2.1, visual, acoustic, and haptic feedback are essential in many virtual reality (VR) applications. In addition to them, some applications may require olfactory and gustatory modalities to be excited as well. Immersion of the users into the VE is only then possible, when all this information is provided to the users with high-fidelity.

Since there is an inevitable trade-off between the realism of the created VE and the computing time, it is essential to create high-fidelity approximations of real physical behaviors. Especially for force feedback such approximations are necessary because of the high refresh-rate needed (ca. 1 kHz). Simulation of deformable objects, such as human organs for medical simulators, requires simplified calculations, because the high non-linearities and complex continuous models of the organs cannot be processed in real-time by currently available computers.

This chapter aims at presenting general principles to create high-fidelity VEs, putting the focus on collision detection and haptic rendering issues. Following the state of the art on these issues, their principles are summarized in Section 2.2. Section 2.3 gives an overview of our multi-user VE concept, which has crucial importance for collaborative task realizations and multi-user training strategies. In the frame of this work a Bone Drilling Medical Training System (BD-MTS) is developed. Section 2.4 explains the fundamental components of this training system, which is used as the main application scenario throughout this thesis. Another training scenario, a push-button training system, is briefly introduced in Section 2.5.

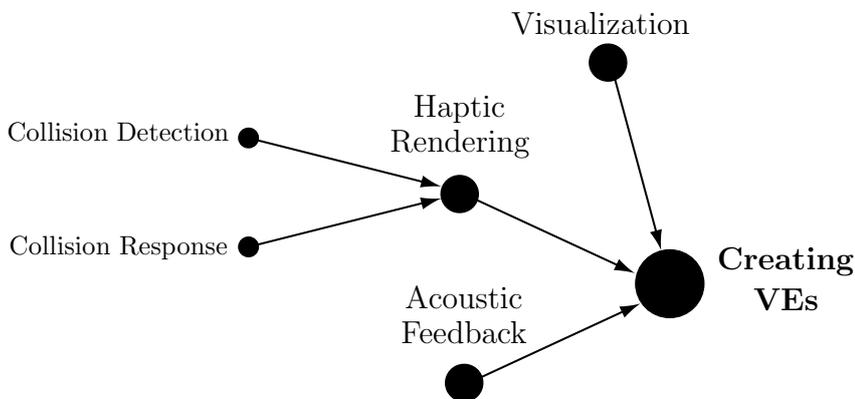


Figure 2.1: Main modalities and rendering issues explored in this work to create a VE.

2.1 State of the Art

2.1.1 Collision Detection

The state of the art in collision detection is considered only with respect to haptic rendering issues. Therefore, computer graphics works from the late 70s concerning the collision of 3D objects are not taken into account.

Since the famous God Object Method [142] was published by Zilles and Salisbury in 1995, collision detection and haptic rendering in a VE have been widely investigated by many researchers. The God Object Method implies a point that represents the user's hand in the VE. In this method, the fact that the avatar is just a point causes some drawbacks (e.g. avatar falls into gaps between polygons of the collided object). Consequently, the God Object is redesigned as a sphere and called *virtual proxy* by Khatib *et al.* in 1997 [106,107]. However, in many real applications, the tool in use is neither a point nor a sphere. It is a shaped object and it has more than one contact point when collided with another object. That introduced the need of *shaped proxies* which were developed by Srinivasan *et al.* [48]. The approach defines the proxy as a point ray and looks for the collision between this ray and the virtual object. The work's contribution to the haptic community is remarkable, especially in the virtual reality minimally invasive surgery simulators. Thanks to the development of ray-based haptic rendering technique, surgical tools and their collision with human organs can be simulated realistically.

There are still many on-going research activities in the field of collision detection. Niemeyer *et al.* have introduced the concept of *dynamic proxies* recently [90]. They discuss zero-, first- and second-order dynamics of a virtual proxy which traditionally had no dynamics. All of the above mentioned methods are discrete. Kheddar brings in a continuous collision detection algorithm [59]. It is based on arbitrary motion interpolation between two successive time steps. In the frame of our work, deformable object principles are applied to the proxy, in order to investigate tool deformations. Kheddar *et al.* contribute to the deformable proxy field as well [1].

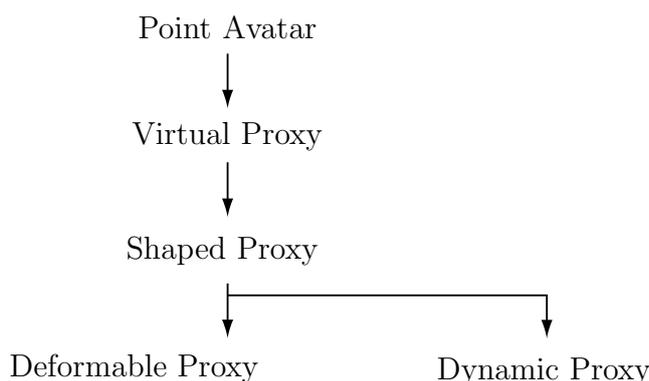


Figure 2.2: Evolution of representing the human hand in VEs.

Collision detection has to be as fast as possible, i.e. computing time must be short. This achieved, there is enough time for haptic rendering and for a reliable, high-fidelity haptic feedback to the user. Especially the collision check between a virtual proxy and an object with many polygons is hectic. An efficient OBB-Tree (oriented bounding box) solution for

this problem was proposed by Gottschalk *et al.* in 1996 [42], which was followed by the AABB-Tree (axis aligned bounding box) method for complex deformable models developed by Van den Bergen in 1998 [127]. Both methods depend on pre-computed hierarchical volume representation of an object with many polygons. The collision check is done between the avatar and the boxes that bound the polygons of the object. Using a binary search tree (BST), the space is divided into smaller regions, which provides eliminating unnecessary collision checks [10]. Some recent works concerning to shortening the computing time for collision detection are related to collision prediction. Kim *et al.* propose that following the trajectory of the user’s hand holding a haptic display in freespace, it is possible to predict the mean tool direction. Thus, the possible collision points between an object in the virtual scene and the virtual avatar can be foreseen [61, 62]. Lombardo *et al.* realize the collision detection with the graphic hardware [75]. The method basically depends on specifying an OpenGL camera that has a viewing volume corresponding to the tool shape, and rendering the main object (e.g. the organ in surgical simulations) relatively to this camera. If nothing is visible, there is no collision.

In our application examples, the virtual avatar was either sphere-like as in the push button scenario, or it was a shaped-proxy as in the bone drilling scenario. For the former one, we have implemented a simple point-triangle collision check, whereas for the latter one a ray-based collision check, using AABB and BST. Both cases are explained in Section 2.2.1 in detail.

2.1.2 Haptic Rendering

Salisbury *et al.* define the rendering as “the process by which desired sensory stimuli are imposed on to the user to convey about a virtual haptic object” [108]. Haptic rendering is the procedure of calculating the force response that results from interaction in a VE and sending the response information to the user.

There have been many works concerning haptic rendering from different perspectives which makes it difficult to classify them. Following their God Object Method, Zilles and Salisbury published the fundamental ideas of haptic rendering in 1995 [142]. Basdogan *et al.* summarized the haptic rendering methods and classified them in 2001 [12]. One of the latest contributions for such a general overview is from 2004 and belongs to Salisbury *et al.* [108]. Most often haptically rendered object is *virtual wall*. The basic idea is to model the wall as a spring which has a large spring coefficient. Modeling a rigid wall introduces the stability problem. When the stiffness of the wall increases, the sampling time has to be decreased to keep the system stable. Therefore, the virtual wall is used in studying haptic stability such as in [2]. Stable and realistic haptic interaction with VEs remains a challenging research problem to solve. Some recent efforts are presented in [76, 98].

A commonly accepted method to calculate collision response is so called penalty-based haptic rendering. After the collision is detected, the penetration depth of the avatar (point, sphere, or ray) into the virtual object, i.e. the distance between haptic interface point (HIP) and God Object, is penalized by a linear spring and a damper. For more interesting impressions, non-linear stiffness can be used [142].

Especially in surgery simulation the realistic interaction with soft tissue has crucial importance. Consequently, numerous researches investigate haptic rendering and visualization of deformable models. Cotin *et al.* investigate modeling the RT elastic deformations for surgery simulations [24]. We will follow the way of Basdogan *et al.* and divide the hap-

tic rendering algorithms for soft tissue interaction into two groups: geometrically-based methods and physically-based methods.

Geometrically-based algorithms (GBA): This group can be further divided into several subgroups, such as vertex based, or spline based approaches. Geometrically-based algorithms are fast, result in smooth visual deformation, but exclude the underlying deformation mechanics. To compute forces, a second model has to be employed [94].

Physically-based algorithms (PBA): Calculated forces and the object deformations as collision response with PBA are more realistic in comparison with GBA, because the physical property of the collided objects and interaction dynamics are taken into account. The PBA can further be divided into three subgroups: Particle-based, mesh-based and meshless methods.

Particle-based models are basically mass-spring networks. The virtual object is divided into vertices and each vertex is represented by a mass. Afterwards, the vertices are connected to each other by spring elements. When there is an external stimuli, using the Newton's 2nd Law, the differential equation of the system dynamics is written. The equation is solved in RT using numerical integration techniques. Mass-spring networks are easy to build and easy to solve in RT. However, they can only approximate the real physics of the object [4,8]. It is difficult to define correct spring coefficients and masses that represent the object characteristics. Interesting deformation effects can be succeeded by torsional and angular springs. Laugier *et al.* suggest using non-linear spring coefficients to model the incompressibility of the deformable objects, but they indicate that the choice of coefficients remains black art [70]. Bourguignon *et al.* deal with this problem and introduce a method that enables controlling the anisotropy in mass-spring networks [14]. In the same work they prove that their model is also able to provide with volume preservation of the deformable object. Another work, which investigates the volume preservation problem in mass-spring networks, is given in [129].

Mesh-based models are obtained by dividing an object into a set of surface or volume elements, and approximating the continuous equilibrium equation over each element [70]. Most widely investigated mesh based modeling method is *Finite Element Modeling* (FEM), which is computationally very expensive method, but physically very realistic. Bro-Nielsen introduced using fast finite element models in 1996 [15,16]. First, he created a volumetric model of an object, then applied condensation techniques and calculated only the deformation of the surface nodes. Later on, James *et al.* used boundary element models (BEMs). They used pre-computed Green's functions for global deformations, and looked for boundary value problem solution, whereas boundary conditions were known [56]. Basdogan uses modal analysis method, which enables to decouple the vibration modes of the object and then choose the most significant ones to compute the deformation in real-time (RT) [9]. In the same work he suggests an alternative method, which is the spectral Lanczos decomposition to obtain the explicit solutions of the finite element equations that govern the dynamics of the deformations. In one of his other works, Basdogan also indicates that FEMs present good results for linear elasticity, but they have less accuracy for non-linear elasticity problems [11]. Another mesh-based method is Long Element Modeling (LEM), which was introduced by Mendoza *et al.* [82]. The model is described by Pascal's principle and the conservation of the volume. The advantages of the method are given by the authors as quasi dynamic behavior, realistic and stable simulation possibilities. Non-homogeneous or even composite materials can be presented by LEM. No pre-calculation or condensation is necessary. The complexity is generally one order of magnitude less than FEM. The

system is solved numerically by using any standard numerical method. LEM is only valid for small deformations.

Kim *et al.* work in the meshless modeling area. They use meshless finite spheres technique [62]. In this technique, when tool-tissue interaction occurs, nodal points are sprinkled around the tool-tissue contact point locally and interpolation is performed by shape functions that are compactly supported on spheres surrounding the nodal points [61].

Apart from the methods above, there are many other deformable object simulation methods. For example, Corso *et al.* use medial axis transform method [23], whereas Radetzky *et al.* model elastic tissues by using a neuro-fuzzy system [97]. Additionally, there are hybrid modeling approaches that combine different haptic rendering methods in order to balance visual fidelity of the object deformation, physical meaning of the haptic feedback and computing time. For example, Kim uses a linear elasticity model for global deformations, and a non-linear one for local deformations. Basdogan uses hybrid modeling to model the virtual organ with FEM, and the medical tool with mass-spring network in his medical training simulator for laparoscopic common bile duct exploration [11]. Another important haptic rendering method is haptic replay [12, 43].

In our application examples, haptic rendering is mostly penalty-based, as in push button scenario. Two example scenarios that introduce deformable object interaction are summarized in Appendix B, one of which is realized by using a mass-spring network, and another one using FEM are developed. In the latter one, we deform the avatar, which is a surgical tool, not the object. Deforming the avatar is not a widely investigated topic, but we think that it has a crucial importance. For example, in bone sawing surgical procedures, the deformation of the saw may lead to irreparable damages. Therefore, it is essential to model the tool deformation to simulate such an operation. Section 2.2.3 summarizes the fundamentals of mass-spring networks and FEM for modeling deformable object interaction. The BD-MTS relates the force to the drilling velocity, and it is discussed in Section 2.4.3.

2.2 General Principles of VR Development

2.2.1 Collision Detection and Haptic Rendering

Haptic interaction with the virtual objects is realized in two steps. At first, collision between the avatar (human hand in VE) and the virtual objects has to be detected. Accordingly, this step is called *collision detection*, which has to be followed by *haptic rendering*, in other words *collision response*. In this step, depending on the collision situation, forces to reflect to the users are calculated. The collision detection methods for the graphics and haptics are the same, but the collision response is different, since no interaction forces are calculated for graphics. The principles of collision detection and response in haptics are explained below.

God Object, Virtual Proxy and Point-Based Collision Detection

The end effector of the haptic display can be represented as a point, HIP, in the VE. This point is used to interact with virtual objects. Initial haptic rendering algorithms use directly the penetration of the HIP, i.e. they one-to-one map the position in space to force. Zilles and Salisbury extended these methods by introducing the “God Object”,

which is a constrained, virtual location of the haptic interface, [142]. In the freespace the HIP and God Object are collocated, but after the collision the God Object stays on the surface, whereas the HIP penetrates into the object. God Object moves on the surface in such a way that the distance to the HIP is minimized. Depending on this distance, a penalty-based haptic rendering strategy is applicable.

In case of using HIP, the collision detection problem is to find the collision of a point and polygonal objects in 3D space. Assume that the object has one triangle, as it is sketched in Figure 2.3. The position \mathbf{p} of the HIP and of triangle corners $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ are known, as well as the normal vector \mathbf{n}_t of the triangle. From the corner points, assume that \mathbf{p}_0 is chosen.

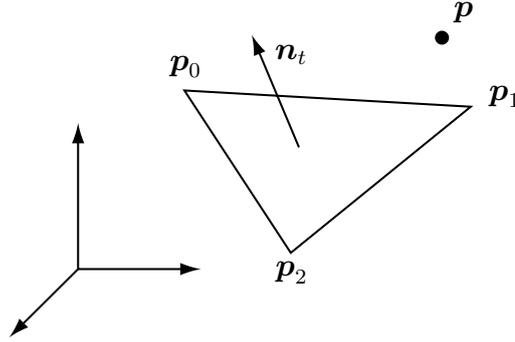


Figure 2.3: Collision detection between a point and a polygonal object in 3D space.

The distance vector \mathbf{r} between the end-effector position \mathbf{p} and corner position \mathbf{p}_0 is

$$\mathbf{r} = \mathbf{p}_0 - \mathbf{p}.$$

If the vectors \mathbf{r} and \mathbf{n}_t are perpendicular to each other, it is concluded that the end effector point and the triangle are on the same plane in 3D space, i.e.

$$\mathbf{r}\mathbf{n}_t = 0. \tag{2.1}$$

The equation (2.1) is the necessary condition for the collision, but it is not sufficient. From the corner points of the triangle, maximum and minimum position limits of the triangle on each axes have to be defined. If the point avatar finds itself in these limits, then the collision is determined.

Because of the previously mentioned drawbacks of representing the human hand as a point in VEs, the point avatar is extended to a sphere, i.e. to a virtual proxy. In this case, the collision detection algorithm needs to check the collision of a sphere and a ray in 3D space. Van den Bergen explains the details of this procedure [128]. Basically, the sphere can be thought of as a point with a margin equal to the radius of the sphere in collision detection procedures.

Ray-Based Collision Detection

In many cases, the avatar is neither a point nor sphere, but a geometric tool. For example, in a virtual bone drilling operation, the avatar is the medical drill. The collision between the drill and bone does not only occur on the drill tip, but also on the other places of the drill bit. Therefore, both virtual object and avatar have many polygons, e.g. triangles.

That means the collision detection problem is to find whether two triangles in 3D space intersect or not. At first, it is needed to detach one edge from one of the triangles, and consider it as a ray. Then, one looks for a ray-triangle intersection, as it is shown in Figure 2.4.

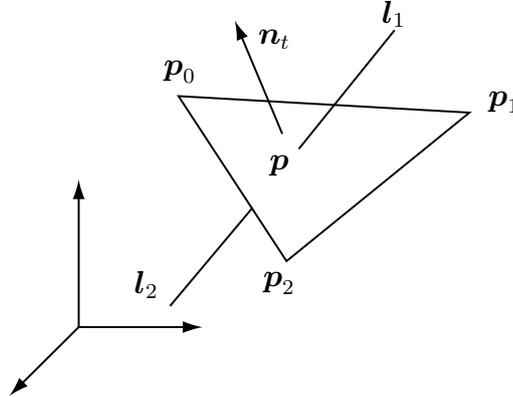


Figure 2.4: Ray-triangle collision detection in 3D space.

The point \mathbf{p} can be defined with respect to edge points of the triangle as below:

$$\mathbf{p} = \mu_0 \mathbf{p}_0 + \mu_1 \mathbf{p}_1 + \mu_2 \mathbf{p}_2, \quad (2.2)$$

where μ_0, μ_1, μ_2 are barycentric coordinates and $\mu_0 + \mu_1 + \mu_2 = 1$, $\mu_i \geq 0$. Replacing μ_0 with $\mu_0 = 1 - \mu_1 - \mu_2$ results in

$$\mathbf{p} = \mathbf{p}_0 + \mu_1 (\mathbf{p}_1 - \mathbf{p}_0) + \mu_2 (\mathbf{p}_2 - \mathbf{p}_0). \quad (2.3)$$

On the other hand, if one takes the ray $\mathbf{l}_1 \mathbf{l}_2$ from Figure 2.4 into account, the point \mathbf{p} can be written as:

$$\mathbf{p} = \mathbf{l}_2 + \rho (\mathbf{l}_1 - \mathbf{l}_2), \quad (2.4)$$

where $0 \leq \rho \leq 1$.

We first define $\mathbf{l} = \mathbf{l}_1 - \mathbf{l}_2$, $\mathbf{d}_1 = \mathbf{p}_1 - \mathbf{p}_0$, and $\mathbf{d}_2 = \mathbf{p}_2 - \mathbf{p}_0$, and then (2.3) and (2.4) are compared with each other. The normal of the triangle is $\mathbf{n}_t = \mathbf{d}_1 \times \mathbf{d}_2$. This analysis gives the following result: If $\mathbf{l} \mathbf{n}_t \neq 0$, $0 \leq \rho \leq 1$ and $\mu_1 + \mu_2 \leq 1$ whereas $(\mu_1, \mu_2) \geq 0$, then the ray intersects the triangle. Note that the case where a ray that lies in the triangle's plane intersects the triangle is ignored. Bergen discusses this theorem in detail and gives an algorithm to program it [128].

Penalty-Based Haptic Rendering

Haptic rendering concerns the collision response, i.e. calculation of the reaction force \mathbf{f}_r . Hooke's Law is widely used in order to appoint this force. After the collision is detected, the penetration depth \mathbf{x}_p of the avatar (point, sphere, or ray) into the virtual object, i.e. the distance between HIP and God Object, is penalized by a linear spring k , so that the force to reflect is $\mathbf{f}_r = k \mathbf{x}_p$.

In many cases, an addition of damping is required to improve the stability of the system and of the perception of rigidity. In Section 2.2.4, this situation is discussed in detail. The stiffness and damping can vary between different objects, or across an object to represent different material characteristics. For more interesting impressions, non-linear stiffness can be used [142].

2.2.2 Bounding Boxes and Binary Search Trees

Since the minimum required refresh rate for high-fidelity force feedback is ca. 1 kHz, the collision between avatar and virtual objects has to be detected as quickly as possible. Checking the collision analytically, as described above, turns to be inapplicable in practice, especially for the VEs consisting of complicated geometrical objects. Therefore, such analytical collision analysis is used only for a limited number of triangles, if the possibility of the avatar-object intersection is very high.

A practical solution for this problem is defining bounding volumes for the virtual objects. Then a hierarchical search tree is created and the collision of the bounding volumes is investigated, not anymore the avatar-object collision.

The bounding volumes can have different shapes, such as AABBs and OBBs. Both cases are shown in Figure 2.5. While it is easy and fast to implement AABB, OBB gives more precise results since the bounding volume fits the model tighter. Both of them are used in a binary search tree (BST) structure for hierarchical search, and they are constructed top-down by recursive subdivision.

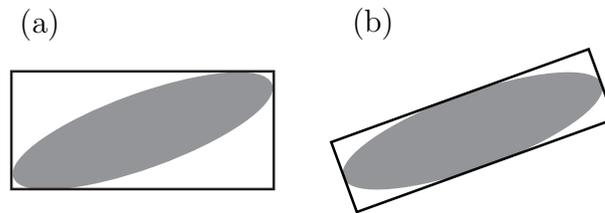


Figure 2.5: Bounding volumes for an elliptic object: (a) AABB (b) OBB.

In Figure 2.6 top-down construction process of an AABB tree is shown. At first, an AABB for the whole object is defined as in Figure 2.6(a). Then, the center of mass of the polygons is found as in (b), and the first child of the tree is created. The first child divides the first AABB into 8 AABBs as in (c) by taking the mass point into account. This procedure goes on until in one AABB there is only one triangle. As it is given in (d), if there are n steps in the procedure, 8^n AABBs are generated. The first collision check is done between the avatar and the most outer AABB. If there is no collision detected, the collision check is finished in one step. Otherwise, one proceeds to the first child for collision check. More detailed information on how to create AABBs, OBBs and hierarchical search trees can be found in [42, 127, 128].

2.2.3 Physically Deformable Objects

Many VR training applications require interaction with the deformable objects, e.g. minimal invasive surgery simulators and training systems. As long as only visual deformation is needed and underlying physics of the deformations is not important, geometrically-based deformation modeling methods, such as vertex or spline manipulations, can be used to simulate the deformations. To obtain more realistic visual deformations and, in addition, to reflect forces related to deformations, physically-based deformation methods are needed. Since complicated continuous and non-linear deformation models are not suitable for real-time interactions, discrete particle or element based approaches are used to approximate the deformations. Working principles of mass spring networks (particle-based)

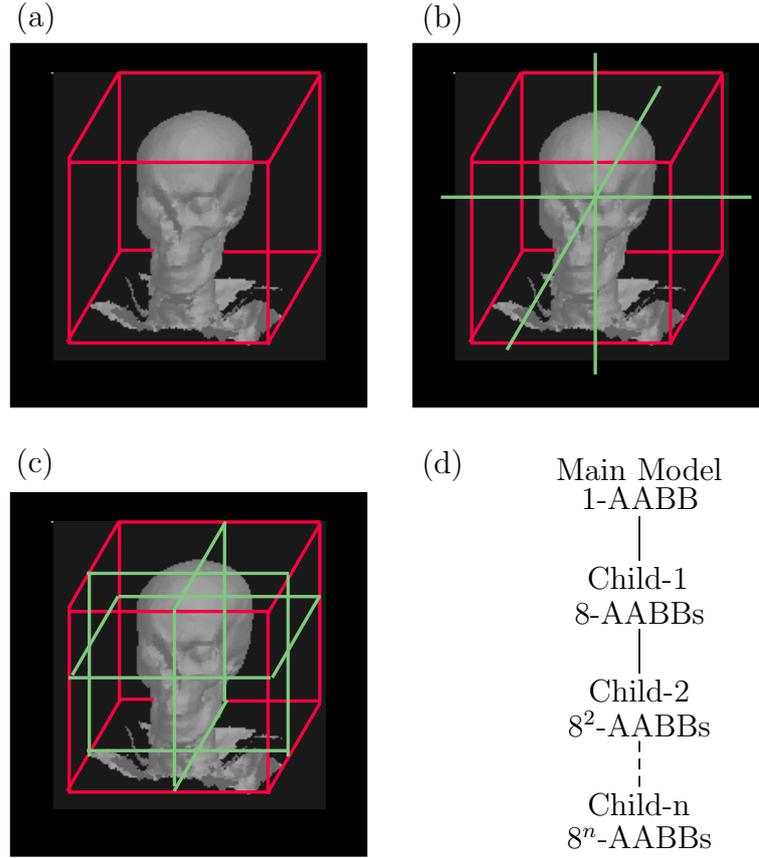


Figure 2.6: Top-down construction of AABBs. (a) The most outer AABB (b) Finding the center of mass of the polygons (c) AABBs in the first child (d) Relation between number of children of a BST and number of AABBs.

and finite element method (element-based) are summarized below. For both approaches an example object deformation is generated in this work, and they are given in Appendix B. Consequently, explanations below show the way that the methods are used in this thesis.

Mass-Spring Networks

The deformable object to be modeled is divided into vertices (nodes) and polygons and each vertex is represented by a mass. Along the edges of the polygons, spring-like elements are defined. Afterwards, the deformation of the object, i.e. displacement of the nodes, is defined via Newton's 2^{nd} Law as follows:

$$m_i \ddot{\mathbf{r}}_i + d_i^u \dot{\mathbf{r}}_i + k_i^u \mathbf{r}_i = \sum_j \mathbf{f}_{ij} + \mathbf{f}_i^e, \quad (2.5)$$

where

- m_i : mass of node i ,
- $\mathbf{r}_i, \dot{\mathbf{r}}_i, \ddot{\mathbf{r}}_i$: position, velocity, and acceleration of node i respectively, and $r_i \in R^3$,
- \mathbf{f}_{ij} : internal force on node i through the spring between node i and j ,
- \mathbf{f}_i^e : external force on node i ,
- k_i^u, d_i^u : constants for spring and damping elements respectively that take place between the original position of node i and its current position after the deformation.

By defining $\mathbf{f}_i^u = -k_i^u \mathbf{r}_i$, (2.5) can be reformed as:

$$m_i \ddot{\mathbf{r}}_i + d_i \dot{\mathbf{r}}_i = \sum_j \mathbf{f}_{ij} + \mathbf{f}_i^e + \mathbf{f}_i^u. \quad (2.6)$$

The equation (2.6) is solved using numerical integration. First of all, the order of the differential equation is reduced by defining $\dot{\mathbf{r}}_i = \frac{\delta \mathbf{r}_i}{\delta t} = \mathbf{v}_i$. Then the explicit one-step Euler integration method is applied to find the nodal displacements when external forces are applied by users.

Mass-spring networks are rather easy to generate, implement and simulate in real-time. However, it is difficult to approximate the desired material characteristics with such a network. The main difficulty is in defining the model parameters. More information concerning mass-spring networks can be found in [8, 14, 70]. An application example is given in Appendix B.

Finite Element Models (FEMs)

FEMs divide the object into several elements and approximate the continuous model of the deformable object by minimizing the potential energy for each element when subjected to external forces. The elements can be surface or volume elements, i.e. 2D or 3D. They may have different shapes as well, e.g. triangle, quadrilateral as surface elements and tetrahedral as volume elements.

Consider a simple geometric object that is divided into four quadrilateral elements as it is given in Figure 2.7. The aim is to calculate the global stiffness matrix \mathbf{K}_g that belongs to the object. First of all, the stiffness matrix of an element \mathbf{K}_e has to be figured out. Therefore, the first element e_1 is separated from the object and investigated closely.

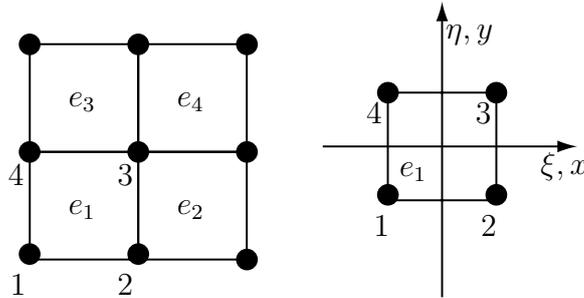


Figure 2.7: A simple geometric object divided into four quadrilaterals.

In the figure (ξ, η) represent the natural (quadrilateral) coordinates that both have values in the range $[-1, +1]$. The Cartesian coordinates are (x, y) . The element is a square and the length of the one edge is a . The material matrix \mathbf{E} for the element is

$$\mathbf{E} = \frac{E_{ym}}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix}, \quad (2.7)$$

where E_{ym} is Young's modulus, and ν is Poisson's ratio. The shape functions of the element are:

$$N_1 = 0.25(1 - \xi)(1 - \eta), \quad (2.8)$$

$$N_2 = 0.25(1 + \xi)(1 - \eta), \quad (2.9)$$

$$N_3 = 0.25(1 + \xi)(1 + \eta), \quad (2.10)$$

$$N_4 = 0.25(1 - \xi)(1 + \eta). \quad (2.11)$$

Shape functions relate the node deformations to the element deformation in Cartesian coordinates. It is necessary to define a Jacobian matrix \mathbf{J} that enables transformation between the differentials of Cartesian coordinates and natural coordinates. This Jacobian is given by

$$\mathbf{J} = \begin{bmatrix} \partial x / \partial \xi & \partial y / \partial \xi \\ \partial x / \partial \eta & \partial y / \partial \eta \end{bmatrix}, \quad (2.12)$$

where $\frac{\partial x}{\partial \xi} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} x_i$ and so on. Therefore the Jacobian \mathbf{J}_{e1} of the element e_1 is:

$$\mathbf{J}_{e1} = 0.5a \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (2.13)$$

The determinant and the inverse of the \mathbf{J}_{e1} are:

$$\det(\mathbf{J}_{e1}) = J_{e1} = 0.25a^2, \quad (2.14)$$

$$\mathbf{J}_{e1}^{-1} = \frac{1}{0.5a} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (2.15)$$

The next step is defining the relation between the strain and the displacement of the object. To achieve this aim, it is necessary to define the derivatives of the shape functions with respect to Cartesian coordinates. This is given in (2.16).

$$\begin{bmatrix} \partial N_i / \partial x \\ \partial N_i / \partial y \end{bmatrix} = \mathbf{J}_{e1}^{-1} \begin{bmatrix} \partial N_i / \partial \xi \\ \partial N_i / \partial \eta \end{bmatrix}. \quad (2.16)$$

Now, it is possible to create the strain-displacement matrix \mathbf{B}_{e1} as a function of natural coordinates as follows:

$$\mathbf{B}_{e1}(\xi, \eta) = \begin{bmatrix} \partial N_1 / \partial x & 0 & \dots & \partial N_4 / \partial x & 0 \\ 0 & \partial N_1 / \partial y & \dots & 0 & \partial N_4 / \partial y \\ \partial N_1 / \partial y & \partial N_1 / \partial x & \dots & \partial N_4 / \partial y & \partial N_4 / \partial x \end{bmatrix} \quad (2.17)$$

Consequently, the matrix \mathbf{B}_{e1} relates the strain vector $\mathbf{e}_{e1} = [e_{xx} \ e_{yy} \ 2e_{xy}]^T$ to the displacement vector $\mathbf{u}_{e1} = [u_{x1} \ u_{y1} \ \dots \ u_{x4} \ u_{y4}]^T$ as $\mathbf{e}_{e1} = \mathbf{B}_{e1} \mathbf{u}_{e1}$.

The element stiffness matrix \mathbf{K}_{e1} is

$$\mathbf{K}_{e1} = \int_{-1}^1 \int_{-1}^1 J_{e1} \mathbf{B}_{e1}^T \mathbf{E} \mathbf{B}_{e1} d\xi d\eta. \quad (2.18)$$

Using Gauss numerical integration method, (2.18) can be re-written as

$$\mathbf{K}_{e1} = \sum_{p=1}^4 w_p J_{e1} \mathbf{B}_{e1}^T \mathbf{E} \mathbf{B}_{e1}(\xi_p, \eta_p). \quad (2.19)$$

Table 2.1: The numerical integration points and weights

point	ξ_p	η_p	w_p
1	$-1 / \sqrt{3}$	$-1 / \sqrt{3}$	1
2	$1 / \sqrt{3}$	$-1 / \sqrt{3}$	1
3	$1 / \sqrt{3}$	$1 / \sqrt{3}$	1
4	$-1 / \sqrt{3}$	$1 / \sqrt{3}$	1

The numerical integration points (ξ_p, η_p) for the element e_1 and their weights w_p are given in Table 2.1.

\mathbf{K}_{e_1} obtained in (2.19) is a 8×8 matrix, relating the deformations of nodes 1, 2, 3 and 4 in x and y directions to the forces applied to these nodes. Since the object is homogeneous, the other element matrices \mathbf{K}_{e_2} , \mathbf{K}_{e_3} and \mathbf{K}_{e_4} are equal to \mathbf{K}_{e_1} . The global stiffness matrix \mathbf{K}_g is the sum of these four element matrices. As a result, the equations of object dynamics are $\mathbf{f}^e = \mathbf{K}_g \mathbf{u}_g$, where $\mathbf{u}_g = [u_{x1} \ u_{y1} \ \cdots \ u_{x9} \ u_{y9}]^T$ is the 18×1 nodal displacement vector. The dimension of \mathbf{K}_g is 18×18 .

After the collision test, using penalty-based methods, the forces \mathbf{f}^e affecting the object can be calculated. Therefore, the object deformation is obtained by $\mathbf{u}_g = \mathbf{K}_g^{-1} \mathbf{f}^e$.

FEMs present more realistic deformations of objects in comparison with mass-spring networks, i.e. they represent the physics behind the deformations better. Main drawback of the FEMs is high-computational time, thus difficult to achieve an acceptable real-time performance. Especially if the deformations of the object are plastic, the topology of the object has to be changed. As a result, the system global matrix has to be updated and then inverted in real-time. There are several works concerning finite element modeling of deformable objects, problems to occur and their solutions. Some of the important works are [9, 15, 16, 70]. As an application example, deformations of a saw blade are modeled using FEM in the frame of this work. Refer to Appendix B for details.

2.2.4 Stability of Interaction with Virtual Environments

The forces to reflect are calculated by haptic rendering, but there are different limitations to display these forces exactly to the user. One reason for such limitations is hardware related, i.e. the mechanics of haptic display devices (force limits, friction, inertia etc.) restrict the applicable forces. On the other hand, the limitations may be caused by the nature of the VEs. Since a VE is a digital medium that works in discrete-time, it is an active environment, i.e. touching a virtual object extracts energy from it [108]. This extra energy may cause instabilities during interactions.

One way to avoid such instabilities is limiting the maximum stiffness of the virtual objects. Colgate *et al.* proved the stability of the interactions if the following holds:

$$b_{md} > \frac{K_{max}T}{2} + |B_{dg}|, \quad (2.20)$$

where K_{max} is the maximum achievable stiffness, b_{md} is the inherited mechanical damping of the haptic display, T is the refresh rate period, and B_{dg} is the added digital damping. It implies that for large stiffness, small T is required. Thus, a method to achieve high stiffness is multi-processing or multi-threading. That means to separate the force calculations (haptic loop) from other VE loops such as visualization or VE dynamics, simplify the force calculations and run the haptic loop in a low refresh rate period. This method is used in many applications successfully, especially for medical simulators. Increasing the inherited damping b_{md} seems to be a solution as well. While this solution helps to increase the maximum impedance to display, it also increases the impedance of the haptic display. Thus, the minimum displayable impedance increases. Since one wishes to demonstrate the dynamics of the VE, not that of the haptic display, increasing b_{md} is not desirable.

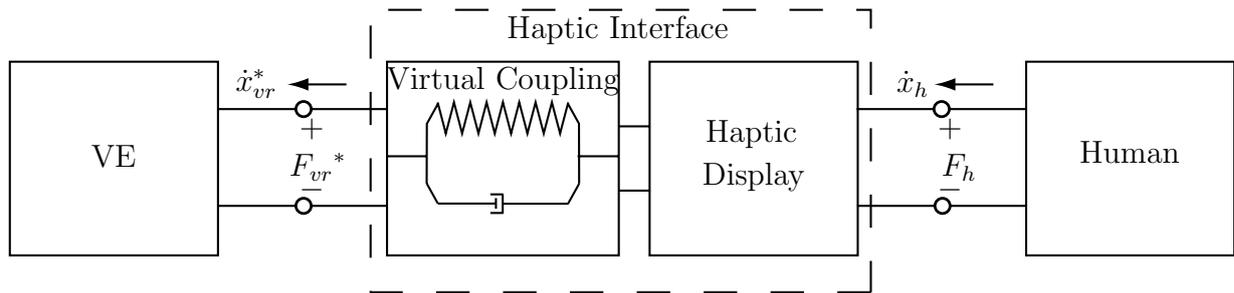


Figure 2.8: Two-port structure of impedance-type human-VE communication via a haptic interface. (*) indicates the discrete variables, index h stands for human and vr for VR data.

Another method to assure the stability of interactions is *virtual coupling*, proposed by Colgate *et al.* [22]. Virtual coupling is a spring-damper connection element, which takes place between the VE and the haptic display as it can be seen in Figure 2.8. Virtual coupling enables to develop VEs without bothering about the stability of the complete system, while by limiting the maximum impedance that the haptic display exhibits, virtual coupling assures the stability. However, it does not help to increase the level of stiffness to display [108]. Adams *et al.* prove the unconditional stability of the two port structure with virtual coupling using Llewellyn’s criteria [2], assuming that human behavior is always passive. Then they conclude that if the VE is also passive, it is possible to find a virtual coupling network that guarantees the unconditional stability of the interaction.

In our work, a combination of both approaches explained above is used. Haptic servo-loop is always separated from other VE loops and a direct virtual coupling is used for force calculation. It means that the force calculations are realized in the haptic interface’s control loop without delay, i.e. in RT.

2.3 Multi-User Virtual Environments

Haptic VEs explained above can be extended for multi-user purposes in such a way that two or more users can explore a haptic virtual object, they may deform-modify the object and even they can interact with each other simultaneously. Depending on the type of the interaction of the users with each other and with the object, Buttolo *et al.* classify the multi-user VR environments under three major groups: Static environments, collaborative

environments and cooperative environments [18]. In our work, the developed scenarios aim at interaction of two users with each other directly or through a virtual object. Thus, following the Buttolo's classification, cooperative environments (CEs), especially for training procedures, are on the focus.

The major problem in CEs is the latency. Due to the latencies in the communication network, delays occur in the haptic loop that calculates the appropriate force information for each user. Since the haptic interaction is a kind of energy exchange between the human and the haptic-device, such delays may cause instabilities. Therefore, a small refresh period for the haptic loop and a small communication delay are essential for stable, high-fidelity multi-user applications. Remember that increasing period or delay would decrease the maximum displayable stiffness.

Bearing this difficulty in mind, a network architecture of three computers to establish CE applications is suggested, as in Figure 2.9. While Computer-I is reserved for the visualization and collision check, in Computer-II and -III haptic loops are run in real-time with 1 kHz sampling rate. The resulting outputs, which can be force or velocity depending on the type of the VE, are fed as reference signals to the robot inner control loops. The key issue is that for one user, the control loop and the haptic loop run in the same computer with the same sampling rate. The users' measured position and force data are sent to the common VR scene and to the corresponding real-time computers. In addition to that, the data is interchanged between two haptic loops in order to minimize the communication delay. In this architecture, maximum measured delay between the real-time computer in one direction is 1 ms.

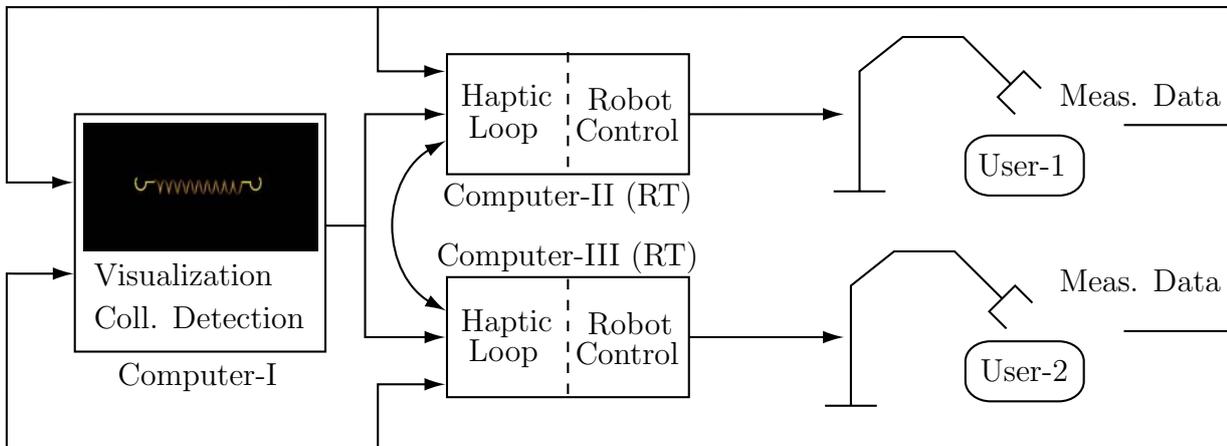


Figure 2.9: Multi-user VR concept: Main elements and the data flow.

The example multi-user VR systems that are developed in the frame of this work can be found in Appendix B. These examples represent the haptic communication of two users through a virtual object. In addition to these, training applications are designed and implemented that allow a trainer and a student to connect to a common VE, each having a separate haptic display. The important issue here is that the expert has a chance to observe the student not only visually and acoustically, but also haptically. Parallel to that, the expert can influence the student and correct him/her in several different ways. These issues are discussed in detail in following chapters in the frame of the BD-MTS and virtual push button training examples. This structure is extended by one more computer

during the training scenarios in order to provide the expert with additional information concerning trainee's performance.

2.4 Bone Drilling Medical Training System

Bone drilling is needed prior to many orthopedical operations, such as pin or screw insertion to the bone and it requires a high surgeon skill. The main problem of a bone drilling operation is heat generation. The friction that occurs during a bone drilling process produces high temperature which may cause irreparable damages to the bone. If the drilling time is long, then the produced temperature will be high, i.e. for an ideal drilling, the feeding velocity has to be as large as possible. The feeding velocity is directly related to the applied thrust force. Large feeding (axial) velocities require large thrust (axial) forces. The other main influencing factors of the axial drilling force are cutting speed and drill tip angle, where the effect of cutting speed is negligibly small. It is possible to state out the core skills for bone drilling procedures as: *Recognizing the drilling end-point; applying constant, sufficient but non-excessive thrust force and feeding velocity.*

Mihoci *et al.* proved experimentally that the effect of the cutting speed on the axial force is negligible in comparison with the effects of the drill tip bit angle and the axial velocity [123]. Their work concludes that if a big drill bit angle (e.g. 2.1 rad) is used, the necessary axial force for a fast drilling process increases dramatically up to 200 N. Malkin *et al.* proved experimentally that the specific cutting energy E_u increases when feeding velocity v of the drill decreases. Applied thrust force f is directly related to v , so that a larger thrust force f results in a larger feeding velocity v . These cases are formulated as

$$E_u = H_1 v^{-h_1}, \quad (2.21)$$

$$v = H_2 f^{h_2}, \quad (2.22)$$

where H_1, h_1, H_2 and h_2 are experimentally determined constants [134].

There are rather limited classical methods for novice surgeons to train bone drilling, and these methods show some deficiencies. For example, artificial bone models are too expensive to use for training purposes, therefore they are only used by experienced surgeons for operation planning. Observing an expert surgeon during drilling is not as efficient as hands-on training methods such as training on cadavers, which in turn brings in ethical problems. Thus, it is necessary to introduce new training possibilities for such surgical cases.

Taking this necessity into account, a BD-MTS is developed in our laboratories. This section gives an overview of the main components of BD-MTS, together with collision detection and haptic rendering principles. Throughout this work, three main versions of BD-MTS have been developed. Sections 2.4.1, 2.4.2 and 2.4.3 explain the technical background of these versions chronologically.

2.4.1 Bone Drilling with a Pedal-Like Haptic Display

In this first version of BD-MTS, a pedal-like haptic display that allows 1D interactions has been used. It is a 1-DoF electromechanical device that contains a PWM amplifier and a DC motor as actuator. As it can be seen in Figure 2.10(a), the display has a lever as end-effector, whose position is measured by a digital encoder (1024 pulses per round). Applied

forces are measured via strain gages. Maximum motor output moment is $\tau_{Mmax} = 0.11$ Nm, gain and efficiency of the gearing system are $k_D = 66$ and $\eta = 0.7$ respectively. The length l of the lever is 0.2 m. With respect to these characteristics, the maximum applicable force is $F_{max} = \tau_{Mmax}k_D\eta/l = 25.4$ N. Experimentally maximum applicable force is measured around 28 N. The working area of the haptic display is limited to ± 1.0 rad.

An appropriate 1-DoF visualization of the drilling has been created as seen in Figure 2.10(b). The visual scene contains simple geometrical objects, and the visualization is programmed in C++ under Linux environment using OpenGL library [30].

Mainly, enhanced training issues and related control problems are addressed in this version. Therefore, haptic and visual enhancement features have been added to the scene. The latter one is a visual indicator that informs the student about the force he/she applies as depicted in Figure 2.10(b). The former one introduces haptic cues that constrain the movement of the haptic display, if improper forces are applied. In Chapter 4, the enhanced training strategies are explained in general.

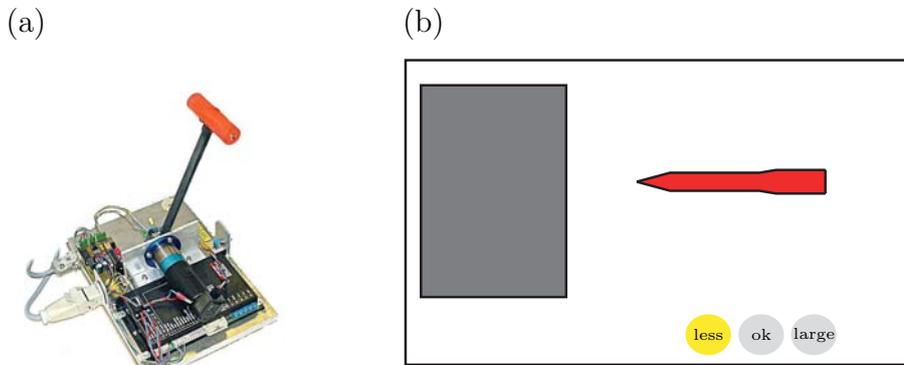


Figure 2.10: Bone drilling with a pedal-like haptic display: (a) The haptic display, (b) visualization.

2.4.2 Bone Drilling with ViSHaRD3

The haptic device ViSHaRD3 has been developed in our laboratories [40]. The main features of this 3-DoF device are given in Table 2.4. The first two links of ViSHaRD3 are designed in SCARA configuration. While these two links provide the movement in $x - y$ directions, the third rotational link changes the end-effector position in z direction. Each joint has a DC electrical motor (Maxon RE40) together with an encoder (Maxon HEDL 5560) and a harmonic drive gear (HFUC-2UH-17). The contact force between the finger of the operator and the manipulator is measured by a force-torque sensor (JR3), which is placed between the last link and the end-effector. For measurement data acquisition and to control the PWM amplifier, an I/O card developed in our laboratories is used. ViSHaRD3 is shown in Figure 2.11(a).

The visual scene contains 3D objects and simulates 3D interactions. Some simple geometric objects representing a bone and a medical drill are used in the VE, as depicted in Figure 2.11(b). When the trainee moves the end-effector of the haptic display, the medical drill model in the GUI moves as well. If there is a collision between the drill model and the box, the trainee feels the reaction force. Again haptic and visual enhancements were

implemented. However, not only a force indicators, but also a velocity bar graph, an acceleration bar graph, and a drilling end-position warning signal were added to the system. The GUIs are programmed in C++ with Open GL in a Linux environment [31–34].

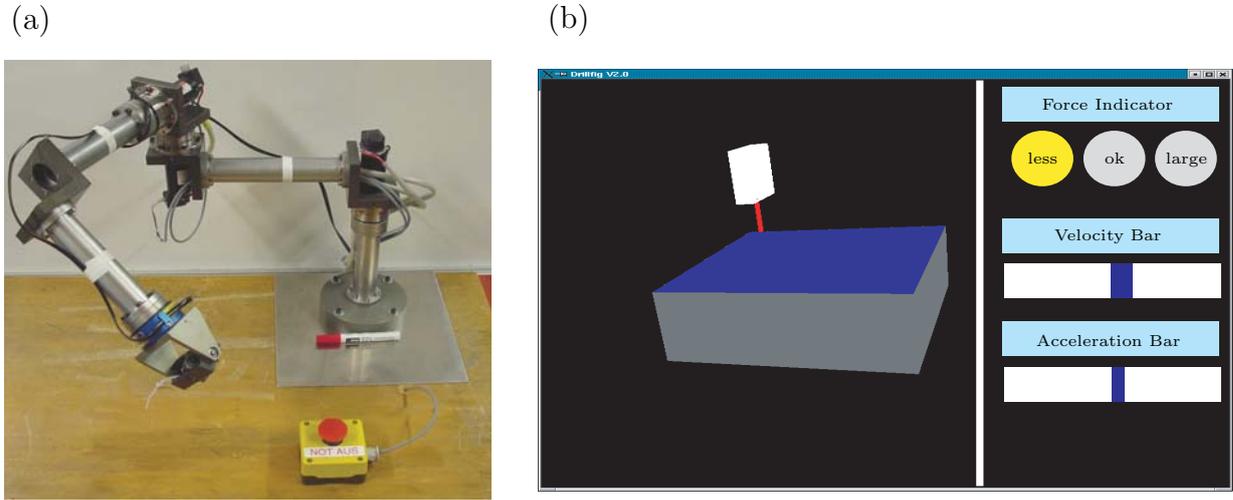


Figure 2.11: Bone drilling with ViSHaRD3: (a) ViSHaRD3, (b) visualization.

In this version, also an acoustic enhancement is enabled. Three different drilling sounds (low freq., middle freq., high freq.) are integrated to the drilling system. The sound files are 2D, which are available in many web pages as sound effects. The sounds are played appropriate to the applied force of the trainee, i.e. the trainee hears either the sound of too small force, drilling force and excessive force. The acoustic feedback is programmed in C++ using FMOD acoustic library. In Figure 2.12, a flow chart of acoustic feedback creation is given.

Table 2.2: Specifications of ViSHaRD3.

Active DOF	3
Passive DOF	3
Workspace (Width X Depth X Height)	$0.6 \times 0.25 \times 0.4$ m
Peak force	86 N
Translational velocity	1 m/s
Maximum payload	6.5 kg
Mass of moving parts	≈ 5.5 kg

2.4.3 Bone Drilling with ViSHaRD10

The concept of the BD-MTS is presented in Figure 2.13. The VE contains a skull and a medical drill model. The student interacts with the VE via the hyper-redundant haptic

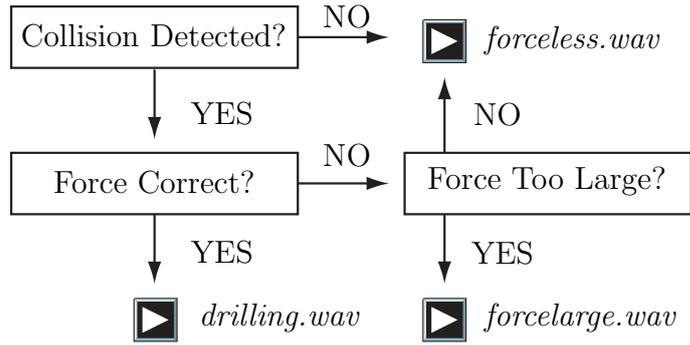


Figure 2.12: Flow chart for acoustic feedback

display ViSHaRD10 [124–126]. Translational and rotational movements of the haptic display and the measured force are transferred to the VE, and the virtual drill model is moved appropriately. If the drill model collides with the skull model, the student feels a resistance in the direction of the collision. From then on, the drilling task is realized with a thrust velocity related to the applied force. An appropriate sound feedback is provided as well. The system is extended by a suitable training strategy during the training sessions. The strategies affect the output of the VE, i.e. they change the desired signal for the human operator.

In general, a network of two computers is used for BD-MTS. One computer is responsible for the visualization and collision detection (2.6 MHz, AMD Athlon XP 2600+), whereas the other one for haptic rendering and the control of the haptic display (RTAI Linux). For multi-user training cases, the network is extended with two more computers to control the second haptic display, and to provide the second user (expert user) with additional information that is not available for the first user.

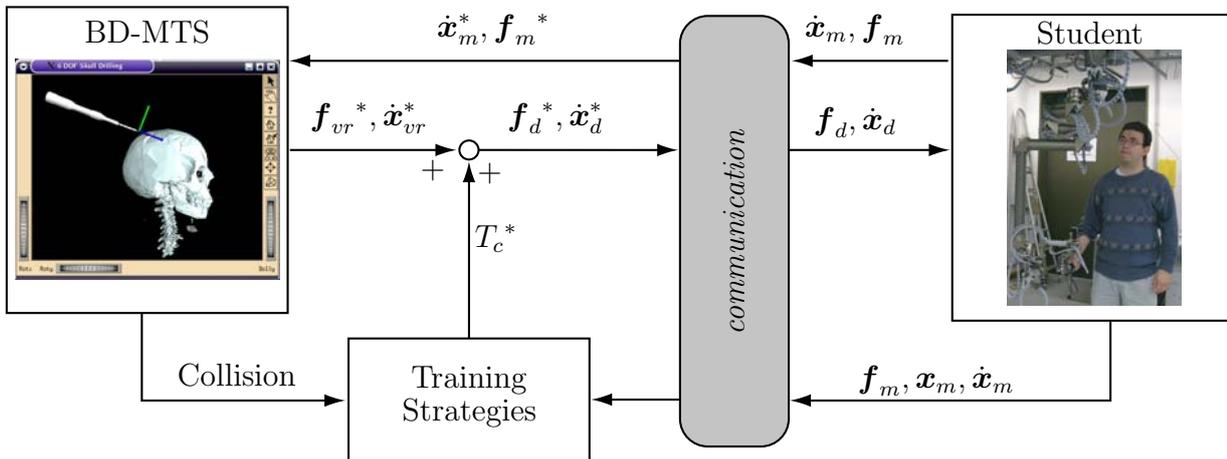


Figure 2.13: The concept of BD-MTS. (*) indicates the discrete variables, index m stands for measured data, d for desired data, and vr for VR output data. In freespace VE is impedance like. After collision, VE is admittance like. T_c^* represents the influence of the trainer model.

Main features of the BD-MTS, such as visualization, acoustic feedback, collision detection, haptic rendering, stability and control issues are summarized in this section.

Visualization

The latest version of the BD-MTS has a skull model in the virtual scene. The CT-data of a skull is triangulated and a surface model is created. The model data can be saved in different formats, e.g. *.wrl, *.iv, *.stl. The saved data is then visualized using OpenInventor library¹ in C++ Linux environment. The scene contains a 1:1 model of a medical drill as well. This model is designed using SolidWorks software. It is possible to add it to the virtual scene by saving it in one of the above mentioned formats. The dimension and number of polygons of the models are given in Table 2.3. The BD-MTS contains visual, haptic and acoustic enhancements as in the previous versions.

Table 2.3: Attributes of skull and drill models

	Drill	Skull
Num. of Polygons	4288	48752
Num. of Vertices	2175	40561
width [cm]	5.74	19.21
depth [cm]	2.68	24.20
height [cm]	25.04	32.25

Collision Detection and Haptic Rendering

As long as the VE is built using simple non-deformable geometric objects like in Figures 2.10(b) and 2.11(b), and a point-based collision detection is satisfying, it is rather straight forward to detect the collision of the virtual objects. In this case, the virtual-drill tip is defined as a virtual proxy, and using the point-based collision detection principles explained in Section 2.2.1, the collision check problem is easily implemented in real-time.

However in the BD-MTS version, whose concept is given Figure 2.13, the virtual scene contains objects made of many polygons. Therefore, a ray-based collision detection algorithm using the AABB and BST techniques explained in Section 2.2.2 is implemented. For this purpose, SOLID C++ collision detection library is used [128]. In SOLID collision detection software, it is only necessary to define the triangle vertices and normals of the models appropriately. An abstract of the programming code is given in Appendix A.

Following (2.22), the collision response of the virtual bone is calculated via following non-linear relation:

$$v = \Lambda f^2 \text{sgn}(f), \quad (2.23)$$

where Λ is a constant, and it is chosen in such a way that when the student applies correct force, the drilling is realized with a correct thrust velocity. The velocity along and about the axes that are perpendicular to the direction of the collision is set to zero. Thus, the student can move the haptic display only in the direction of the collision. The velocity has always the same sign as the force, therefore $\text{sgn}(f)$ term is needed. All the movements in other directions are blocked, independent of the applied force in these directions. As (2.23) suggests, the VE behaves admittance like, i.e. the input of the VE is the force f and the output is the velocity v .

¹<http://oss.sgi.com/projects/inventor/>

Since the aim of the training system is to teach the relation between the force and velocity during drilling operations, the parameters that need to be pre-defined are set in such a way so that the forces are as large as possible and the duration of the process is long enough to check the constancy of the velocity. Thus, the focus is put on this issue, more than creating 1:1 copy of a bone drilling procedure. Consequently, the correct force range is defined as 37 – 44 N and it is aimed that in this force range, the velocity is ca. 1 cm/s. Thus, $\Lambda = 5.67 \cdot 10^{-6}$ is defined, so that the haptic rendering results in the desired velocity, when the applied thrust force is 42 N. In Figure 2.14, the force - velocity relation during the haptic rendering process, and the pre-defined correct ranges are shown.

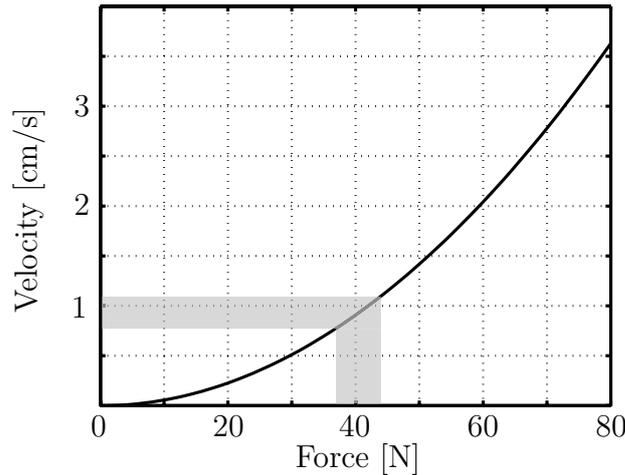


Figure 2.14: The relation between the input force and output velocity of the haptic rendering process and the pre-defined correct ranges.

In this last version of BD-MTS, the hyper-redundant haptic display ViSHaRD10 has been used, which is shown in Figure 2.15(a). In Table 2.4 general specifications of ViSHaRD 10 are given. Main attractive features of the redundant kinematical design in comparison with classical haptic displays are: increased workspace, improved dynamic properties, high-output capability, and possibility of collision avoidance. Technical details of the devices, including the control strategies are widely published by Ueberle *et al.* [124–126]. The visual scene including skull and medical drill models are given in Figure 2.15(b).

Table 2.4: Specifications of ViSHaRD10

Workspace	Cyl. $\phi 1.7\text{m} \times 0.6\text{m} \times 360^\circ$ for each rotation
Peak force	170 N
Peak torque	Pitch, yaw: 13 Nm
Translational velocity	> 1 m/s
Maximum payload	7 kg
Mass of moving parts	≈ 23 kg

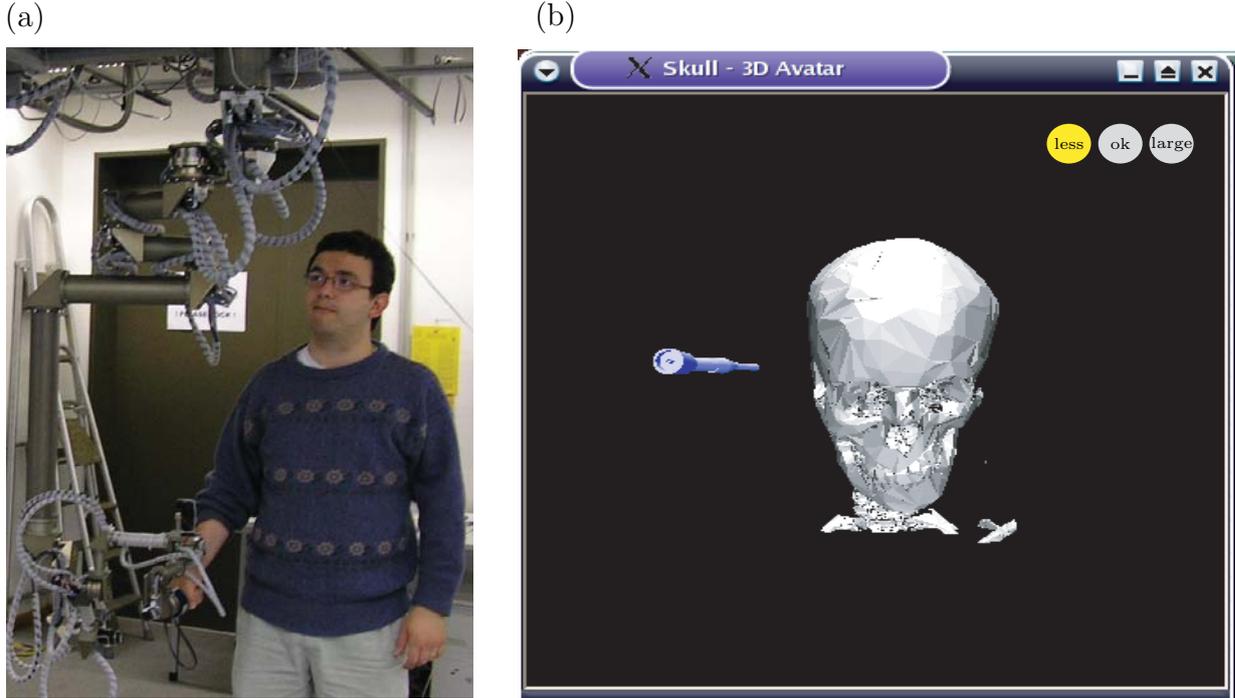


Figure 2.15: Bone drilling with ViSHaRD10. (a) ViSHaRD10 (b) Visualization

Stability of the Non-Linear VE

A strong tool to check the stability of the VEs is the passivity theorem, which says that a system is passive, when the energy flow through a system is more than the net energy stored by the system, i.e. initially stored energy needs to be subtracted from the final stored energy. The theorem is formulated as follows:

$$\int_0^t \mathbf{y}(\tau) \mathbf{u}(\tau) d\tau + V(\mathbf{x}(0)) \geq V(\mathbf{x}(t)), \quad \forall t \geq 0, \quad (2.24)$$

where \mathbf{u}, \mathbf{y} are the input and output signals, V is the energy storage function and \mathbf{x} is the state vector. This definition of the passivity applies both to linear and non-linear behaviors.

In the BD-MTS, whose governing equation for haptic rendering is (2.23), there is no energy storing element and initially stored energy is zero. The VE is a SISO system where the input is force ($u = f$) and the output is velocity ($y = v$). Consequently, (2.24) can be modified for the current problem as follows:

$$\int_0^t v(\tau) f(\tau) d\tau \geq 0, \quad \forall t \geq 0. \quad (2.25)$$

Inserting (2.23) into (2.25) results in:

$$\Lambda \int_0^t f^3(\tau) \text{sgn}(f(\tau)) d\tau \geq 0. \quad (2.26)$$

It is clear that the term in the integral is always positive, since the sign of the force and the sign of the cubic force are always the same. Therefore, for any $\Lambda \geq 0$, the VE for BD-MTS is passive, thus stable.

If the discrete-time character of the VE is taken into account, the integral in the (2.25) has to be replaced with a sum. Thus, the passivity condition for the given problem in discrete time is:

$$\sum_{k=1}^n v(k)f(k) \geq 0. \quad (2.27)$$

Clearly, in the discrete-time domain the VE is passive as well. As human is accepted to behave passive, this result means that it is always possible to assure the unconditional stability for the BD-MTS by an appropriate virtual coupling, whose parameters are related to the mechanical characteristics of the haptic display.

There are several works concerning the analysis of the passivity and stability of haptic interactions with VEs including the sampling, numerical integration and zero-order hold effects [2, 20–22, 83]. Especially the dissertation of Brown represents a very detailed and clear analysis concerning the passive implementation of haptic-multibody simulations [17].

VR Independent Control

A virtual scene can be modeled either as impedance or as admittance. In the impedance model, the VR takes the robot position \mathbf{x}_m and/or velocity \mathbf{v}_m as inputs, and gives reference force \mathbf{f}_d as output, whereas the admittance model works vice versa. For some practical reasons, it might be necessary to switch between these models during a simulation process. For example, it is easy to implement freespace movements in impedance mode, whereas the collision response can be represented in admittance mode more appropriately. The problem is that switching between two models changes the reference signal from velocity \mathbf{v}_d to force \mathbf{f}_d . That results in a need to change the robot's inner control strategy whenever switching occurs.

To prevent that, an adaptation is created. If the impedance model is used, the desired force signal \mathbf{f}_d is converted to the desired position signal \mathbf{x}_d by multiplying the force \mathbf{f}_d with a minimum admittance Y . If the admittance model is used, the VR output velocity signal \mathbf{v}_d is integrated to end up with \mathbf{x}_d . Thanks to this adaptation, robot inner control can be realized independent of the developed VE model.

As it is seen in Figure 2.16, there are three main blocks in the system control structure: control in VR, switching, and robot inner control. Currently, a task space position control is used as robot (ViSHaRD10) inner control structure. The desired Cartesian space position \mathbf{x}_d is transferred to the task space, i.e. to the link angles \mathbf{q}_d , through Inverse Kinematics (IK) equations.

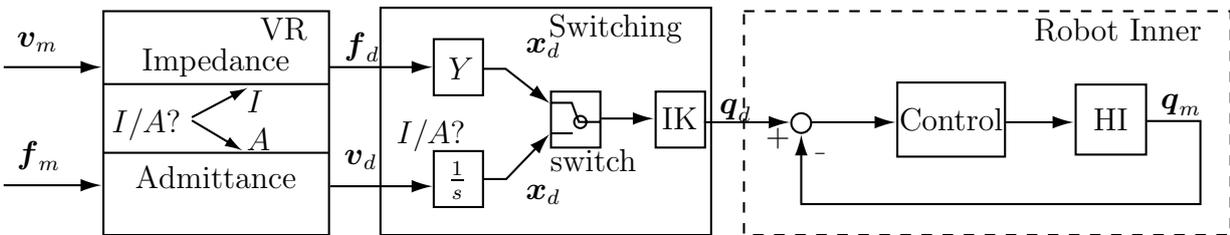


Figure 2.16: Three main blocks of the control structure and the connection between them.

2.4.4 Kinematics Mapping

To be able to control the movements of the virtual medical drill model via haptic display ViSHaRD10, it is necessary to map the kinematics of the ViSHaRD10 end-effector and the virtual drill. Thus, orientation and position of them match with each other. Another issue is the direction fixing after the collision, which needs to be clarified via the haptic display kinematics.

Orientation and Position Matching

From haptic display end-effector to virtual drill, there are four main coordinate systems, as shown in Figure 2.17. On both haptic display and virtual scene parts, there is a world coordinate system (W-CS) and an end-effector coordinate system (EE-CS). The initial rotation matrix ${}^{ev}\mathbf{R}_{eh}$ between the ViSHaRD10 EE-CS and the virtual scene EE-CS is known, since initially both haptic device and virtual drill model have fixed coordinate systems in the space. Using the chain-rule, it can be represented as:

$${}^{ev}\mathbf{R}_{eh} = {}^{ev}\mathbf{R}_{wv} {}^{wv}\mathbf{R}_{wh} {}^{wh}\mathbf{R}_{eh}, \quad (2.28)$$

where the rotation ${}^{ev}\mathbf{R}_{wv}$ is from virtual scene W-CS to EE-CS and its inverse is the rotation matrix that we look for, ${}^{wv}\mathbf{R}_{wh}$ is from haptic device W-CS to virtual scene W-CS and it is known while both coordinate systems are fixed. ${}^{wh}\mathbf{R}_{eh}$ is from haptic device EE-CS to W-CS, and it is also known via the device kinematics. Thus:

$${}^{ev}\mathbf{R}_{wv} = {}^{ev}\mathbf{R}_{eh} \left({}^{wh}\mathbf{R}_{eh}^{-1} \right) \left({}^{wv}\mathbf{R}_{wh}^{-1} \right). \quad (2.29)$$

Since it is necessary to define the rotation of the virtual drill in the W-CS of the virtual scene, the inverse of the matrix given in (2.29) needs to be calculated:

$${}^{wv}\mathbf{R}_{ev} = {}^{wv}\mathbf{R}_{wh} {}^{wh}\mathbf{R}_{eh} {}^{eh}\mathbf{R}_{ev}. \quad (2.30)$$

It must be noted that OpenInventor graphic library needs rotations to be defined in the transpose form to interpret them correctly. Therefore, the rotation of the virtual drill in the virtual scene is defined as follows:

$${}^{wv}\mathbf{R}_{ev}^T = \left({}^{wv}\mathbf{R}_{wh} {}^{wh}\mathbf{R}_{eh} {}^{eh}\mathbf{R}_{ev} \right)^T. \quad (2.31)$$

Velocity Fixing

As it is explained in Section 2.4.3, after the collision between the virtual drill and the skull model occurs, the drilling is possible only in the direction of the collision, i.e. along the z axis of the haptic display's end-effector. Along other axes, the velocity is set to zero. Therefore, using (2.23), Cartesian translational drilling velocity vector on the EE-CS of the haptic display can be written as:

$$\mathbf{v}_{eh} = \left[0 \quad 0 \quad \Lambda f_{z,eh}^2 \text{sgn}(f_{z,eh}) \right], \quad (2.32)$$

where $f_{z,eh}$ shows the force applied along the z axis of the haptic display end-effector. By integrating \mathbf{v}_{eh} , the end-effector position \mathbf{p}_{eh} is defined. Afterwards, using (2.28) the orientation in the virtual scene is fixed.

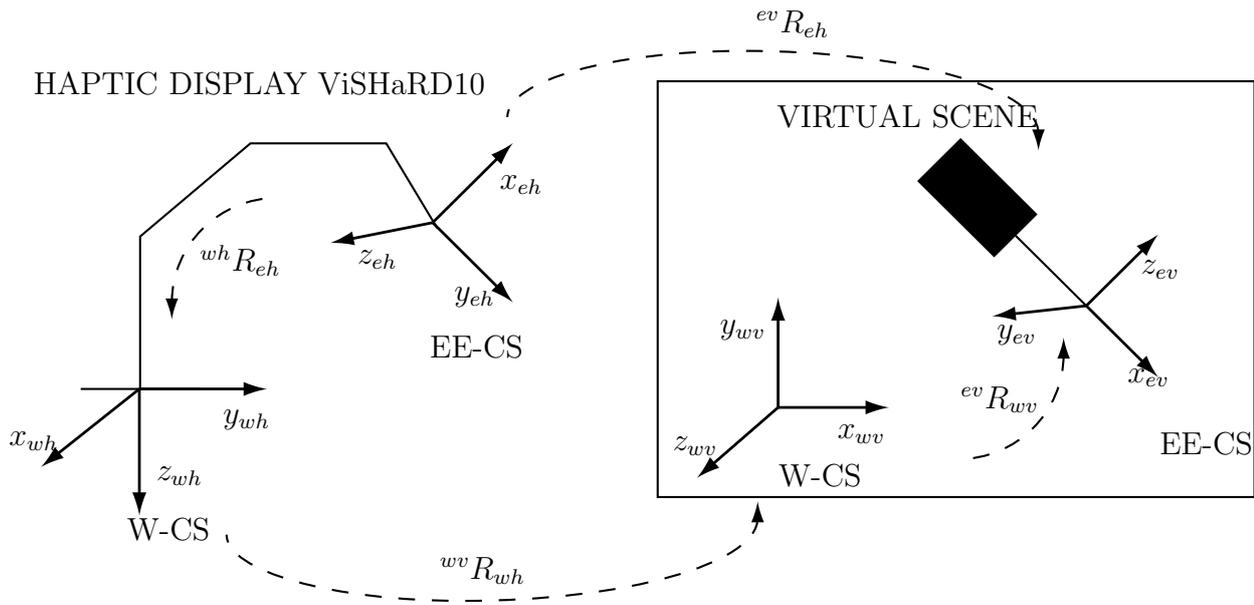


Figure 2.17: Coordinate systems for the haptic display ViSHaRD10 and the virtual scene.

2.5 Push-Button Training System

In order to crosscheck the methods and ideas concerning the VR training systems, another training application for pushing virtual buttons is developed. This application is mainly used as a test-bed to implement and examine the ideas. If successful, they are applied to the BD-MTS and training procedures are performed.

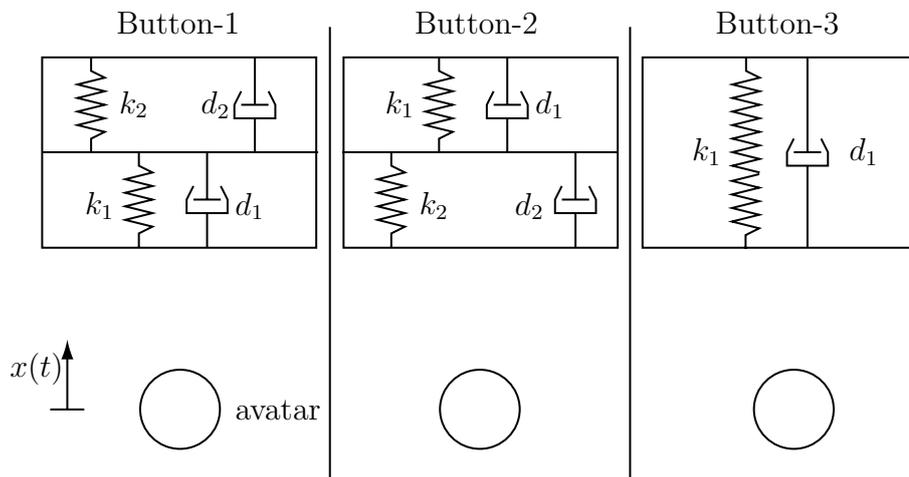


Figure 2.18: Virtual push-button task. Button-1 is used for impedance identification, whereas Button-2 and Button-3 are used to test the identified impedances.

The push-button virtual scenario is also developed in C++ in Linux environment using OpenInventor graphics and SOLID collision detection libraries. To interact with the scene, ViSHaRD10 is used as the haptic interface. The control of haptic display is realized in RTAI Linux using Simulink and Real-Time Workshop Toolboxes of Matlab.

A sketch of the scenario is given in Figure 2.18. The user can move the avatar through a haptic display in 3-DoF. Button-1 and button-2 have two layers, whereas button-3 has only one layer. The collision response of the buttons is defined by virtual springs and dampers, and the buttons are massless. In Table 2.5, the spring and damper coefficients, and corresponding places of the layers are given. The indices $i = 0$ and $i = 3$ represent the freespace and the wall respectively.

Table 2.5: The parameters for virtual push-button application

i	k_i [N/m]	d_i [Ns/m]	interval [cm]
0	-	-	0 – 4
1	500	50	4.1 – 7
2	1000	100	7.1 – 10
3	50000	5000	> 10

Note that the button-1 and button-2 layers have the same coefficients in reversed order. Thanks to that, the results of the identification problem explained in Chapter 3 can be verified. Button-1 is used to identify the motion impedances, whereas Button-2 and Button-3 are used to test the identified impedances. Another important remark is that a blocking mechanism is assumed to exist between layers. Thus, the layers are not coupled. Until the first layer of a button is pushed a certain amount, the second layer does not move. From this certain point on, only second layer is pushed.

2.6 Summary

In this chapter, key collision detection and haptic rendering issues to create high-fidelity VEs were discussed. A multi-user VR concept, which enables a direct exchange of haptic rendering data between two users, was presented. Developed multi-user scenarios let two users interact with each other via a virtual object. The principles of main training applications were introduced. One of the applications is the BD-MTS. After the collision is detected between the virtual skull and medical drill models, the force response of the interaction is calculated via a non-linear relation between the applied thrust force and velocity. The other application is a push-button training system, where simple linear spring-damper elements are used to characterize haptic rendering. While the latter application allows to verify the adequacy of the developed training and identification methods to be explained in the next chapters, the former one enables to create a complete training system that integrates the research work performed.

3 Trainer Skill as a Hybrid Dynamical System

Gaines describes the human being as “a very complex organism whose range of behavior encompasses not only skilled control tasks, but also instinctive and emotional reactions” [41]. Coming up with a model that describes this complex manifold of behavioral patterns is a very challenging and complicated task. A possible solution is to split the behavior into several categories such as conditioning, problem-solving, verbal behavior, perceptual and sensorimotor skills [41]. In this way, the task is narrowed down to a specific behavioral range in question, which then can be modeled in depth. We direct our attention to the last category, sensorimotor skills. It should be noted that the categorization of the human behavior given above neglects possible cross-correlation between categories. Additionally, it is doubtful that each of these categories can be represented by one deterministic function.

This chapter starts with the state of the art, summarizing the methods given in the literature to capture and model human sensorimotor skill. Main methods to model human skill are illustrated in Figure 3.1. Following an overview of the data clustering algorithms, a closer look into the hybrid identification methods that are used in this work is given, and our approach to describe the sensorimotor skill of a human being as switching generalized impedances is introduced. Section 3.4 clarifies the stability conditions for hybrid skill models. Section 3.5 is devoted to two application examples. Since the main goal is to model the trainer skill in order to use it later on in VR training scenarios, interactions of an expert person with BD-MTS and push-button training systems are identified and results are discussed.

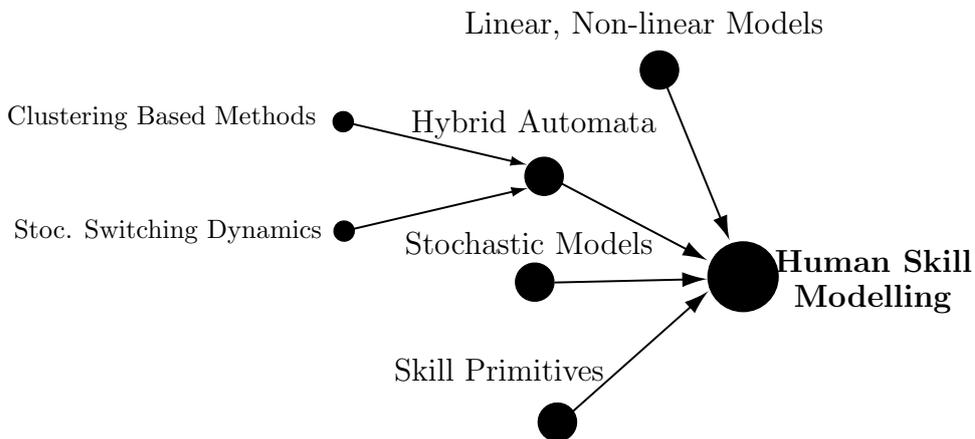


Figure 3.1: Main methods to model human skill.

3.1 State of the Art: Methods to Model Human Skill

Capturing the human sensori-motor skill has attracted many researchers’ attention. Early works, mostly performed by control engineering theoreticians, rely on fundamental linear,

non-linear, continuous or discrete time modeling principles. Later on, robotics researchers have investigated human skill in order to transfer this skill to the robots and realize an extension to the programming by demonstration methods. Nevertheless, trainer skill modeling with the goal to transfer his/her skill to novice users is rather an untouched topic. In this work, we contribute to this issue, using hybrid identification principles. Consequently, a closer look to these techniques is given in section 3.3 individually.

3.1.1 Linear and Non-linear Modeling Methods

Starting from the 40's, there have been several studies to describe human perceptual motor skill as a linear model. An overview of such methods is given by Gaines [41]. The experimental studies generally depend on tracking a reference signal $r(t)$ displayed on a screen through an interface device such as a joystick. Then, a linear transfer function $G(s)$ between the operator's output $m(t)$ and the error signal $e(t)$ is defined. In general, this transfer function contains time delay and phase-lag elements that reflect the dynamics of central nervous system. One of the latest works in this area belongs to Tsuji *et al.* [121], in which human operator's tracking characteristics are defined according to the impedance parameters of the robot that is used as the interface. They measure the force as the operator's output, and extend the linear transfer function with a phase-lead element which is caused by the human arm impedance.

Several other researchers worked on non-linear modeling of the human controller. For example, Smith [113] and Wilde & Wescott [135] described the movements of human limb by a bang-bang controller, which switches between two values of force in order to minimize the position error. Young & Meiry supported the bang-bang controller theory by showing that human movements are time-optimal and minimize the error as quickly as possible [141]. After the introduction of sampled-data systems, some researchers also represented human as a discrete, sample-data model [69]. Another theory set up by Hogan is that the coordination of arm movements is jerk-optimal [39].

The methods concerning linear, non-linear, and optimal modeling of human sensorimotor characteristics are not further described in this work. For more information and detailed overview refer to [26, 41, 92].

3.1.2 Skill Segmentation and Skill Primitives

In programming-by-demonstration (PbD) methods, a human demonstrates a skillful task and a learning algorithm captures this task realization, which is later on used to control a robotic manipulator. Individual tasks can be taught to the robot via human demonstration, or they can be defined as a combination of previously known sub-tasks.

Yang *et al.* save the human demonstration data from a deburring task, and through the acquisition and interpretation of the human linguistic data, the skill is transferred to the robot [138]. In low-level, their control structure contains linear functions that represent the arm posture, and in high-level, there is a decision making algorithm that enables switching between linear models. This high-level controller is derived from human linguistic information. In a previous work, Yang *et al.* attain a human deburring expert's skillful manipulation strategies. They process the sensor signals using pattern recognition techniques in order to define switching conditions between different control signals. Linares *et al.* employ a similar structure. However, their high-level skill acquisition depends on an

image processing algorithm [73]. In another work of the same research group, Zölner *et al.* investigate the fine manipulation [143]. They measure position and force during grasping tasks using a data glove, magnetic trackers and force sensors, and classify the dynamic grasps via support vector machines.

Mayer *et al.* partitions the demonstrated task data using spline feature and 3D chain decoding methods [80]. This way, they create skill primitives, and collect these primitives in a skill database. Each new demonstration of human is first tried to be represented by the primitives that already exist in the database. For the missing parts, a new primitive is defined, so that the database is extended. In a consecutive work, they represent a detailed overview of human skill capturing works in literature for PbD applications [79].

3.1.3 Stochastic Methods

Stochastic models map the system input to the system output depending on probabilistic rules. Most commonly used stochastic modeling method for human skill is based on Hidden Markov Models (HMMs). To fully describe an HMM model λ , one needs to know the state transition probability distribution \mathbf{A} that is related to the switching conditions between discrete states; the observation probability distribution \mathbf{b} that is related to the output of the states; and the initial state distribution $\boldsymbol{\pi}$ that indicates which state is the most probable initial state. As a result, an HMM can be represented in a compact notation as $\lambda = (\mathbf{A}, \mathbf{b}, \boldsymbol{\pi})$.

Xu *et al.* first classify skills as action and re-action skills, and then they model both types via an HMM [136]. Similarly, Hovland *et al.* represent assembly skill using HMMs [53], so do Dong *et al.* [27]. The method is also relevant in medical simulators. For example, Rosen *et al.* define laparoscopic surgery skills via HMMs [104, 105].

Another stochastic representation of human skill is realized via neural networks (NNs). To use in their deburring robot, Asada *et al.* acquire human skill via NNs [7]. Nechyba *et al.* define a balancing inverted pendulum task, and model an expert via NNs. Later on, they use this model to give on-line visual advice to inexperienced users.

The main disadvantage of such stochastic identification processes is that they do not give any physical explanation about the system. Thus the system, in this case human skill, remains a “black box”. It is neither possible to make any interpretation of the human skill, nor to answer the question how humans realize a skillful task. However, Suzuki *et al.* made an important contribution, which eliminates the above mentioned disadvantage [54, 63, 117, 118, 137]. In their HMMs representing human skill, the states are not black boxes anymore, instead they are deterministic models, such as impedance models. Defining the observation probability vector \mathbf{b} as the Gaussian distribution of the model error, the HMM iterations not only result in identification of the switching points, but also identification of the parameters of the impedance models. The principles of this method are used in the frame of our work. Therefore, a detailed discussion is given in section 3.3.4.

3.1.4 Representing Skill as Impedance and Hybrid Automata

A number of researchers focus on representing human skill as dynamic, deterministic functions. A common approach is to find a functional relation between force, position and

position derivatives, i.e. an impedance model. This approach commences a general concept which can be applicable to different task realizations only by identifying appropriate impedance parameters. Other important advantages of impedance models are their being linear and time-invariant.

Asada *et al.* are pioneers in this area. They model human grinding skill as an impedance, and then use it as a reference model for controlling a manipulator arm to replicate the expert motion. The whole process is represented by one impedance model, i.e. by identifying one mass, one damper and one spring coefficient. Some of the similar works are [64, 112, 119, 140].

An extension of this work is to represent human skill as switching impedances. The resulting structure is a hybrid automaton, in which an event observer estimates the state change and switches between dynamic equations appropriately according to the task state. The compliant motions of the human are divided into discrete states and each state is represented by an impedance model. This strategy is mainly applied to peg-in-hole tasks, in which the contact states, i.e. switching conditions are known beforehand [55, 65, 81]. Hirana *et al.* apply the same strategy to the manipulation of deformable objects, namely to hose insertion tasks [47].

Since a hybrid automaton comprises both the human discrete decision making mechanism and the continuous skillful movement dynamics, it is hypothesized that such an automaton is suitable to describe human skill. However, the approaches given above assume that the switching conditions are known, i.e. the applications are heuristic and limited to the individual tasks. Our goal is to be able to describe a human demonstration in a hybrid automaton structure, even though it is not known what the demonstrated task is. In other words, not only the dynamic skill models, but also switching conditions are unknown. The solution for this problem is solved via hybrid identification methods, which are explained in section 3.3. Before that, the idea of data clustering, as well as principles of the main methods used in this work are briefly presented in section 3.2.

3.2 Human Demonstration Data Clustering

Data clustering methods aim to classify the experimental data in an optimal way, i.e. via minimizing a cost function. After introducing K-means clustering, this section discusses the algorithm that is applied in the frame of our work in detail. An overview of several other available clustering methods is given as well.

3.2.1 K-Means Clustering

K-means clustering is based on Radial-Basis-Functions, and it is one of the most common clustering algorithms. The idea is to randomly initialize as many centers as the number of clusters, which is previously known, and then assign the data points from a data set to the closest center point. Let the data set be $\mathbf{X} \in \mathbb{R}^{n \times r}$, and randomly initialized center points for \mathbf{X} be $\hat{\boldsymbol{\mu}}_i$, where $i = 1, 2, \dots, s$ and s is the number of clusters. Each one of the r columns of \mathbf{X} contains a different system parameter and each row represents a data point, where the total number of data points is n .

The K-Means clustering algorithm aims to minimize the following cost function:

$$J_C = \sum_{i=1}^s \sum_{\mathbf{x}_j \in D_i} \|\mathbf{x}_j - \hat{\boldsymbol{\mu}}_i\|^2, \quad (3.1)$$

where \mathbf{x}_j is the j^{th} row of \mathbf{X} , and D_i is the i^{th} cluster. The function J_C collects the data points, whose closest center point is $\hat{\boldsymbol{\mu}}_i$ into the subspace D_i . After that, the center points are renewed so that

$$\bar{\boldsymbol{\mu}}_i = \frac{1}{n_i} \sum_{\mathbf{x}_j \in D_i} \mathbf{x}_j, \quad (3.2)$$

where n_i is the number of data points in the i^{th} cluster. Then, the clustering function is applied again using the new centers. The loop ends when the centers do not change anymore. Because of the fact that K-Means clustering algorithm may finish in a local minimum instead of the global minimum, it is recommended to repeat the K-Means algorithm several times using different initial center points.

Once all clusters D_i , $i = 1, 2, \dots, s$ are defined, unknown parameters of the models representing the dynamics of each cluster can be identified using an optimization technique, e.g. least squares.

Important drawbacks of this method are its high sensitivity to outliers and noise, and the non-scaled (or not normalized) regression matrix. In addition, poor initializations are likely to cause ending up in a local minimum. An improvement of the method, which alleviates the mentioned drawbacks is discussed below.

3.2.2 Weighted K-Means (WKM) Clustering in Parameter Space

The quadratic Euclidean norm given in (3.1) checks the distances without scaling and rotating the axes. It is easy to imagine that this might lead to wrong clustering of the data, if the parameters in the data set \mathbf{X} are heterogeneously scaled, e.g. if the first data column has a range $\pm 10^3$, whereas the second data column range is $\pm 10^{-3}$.

To overcome these drawbacks, it is necessary to weight the Euclidean norm as follows:

$$J_C = \sum_{i=1}^s \sum_{\mathbf{x}_j \in D_i} \|\mathbf{x}_j - \hat{\boldsymbol{\mu}}_i\|_{\boldsymbol{\Sigma}^{-1}}^2, \quad (3.3)$$

where $\boldsymbol{\Sigma}$ is the weighting matrix. If

$$\boldsymbol{\Sigma} = \text{var}(\mathbf{X}) = \text{diag}(\sigma_i^2), \quad (3.4)$$

where σ_i^2 is the variance of the i^{th} column parameters of the \mathbf{X} , then the distance measurement is the standardized Euclidean norm and the data set \mathbf{X} is scaled. If

$$\boldsymbol{\Sigma} = \text{cov}(\mathbf{X}), \quad (3.5)$$

then it is Mahalanobis distance, and the data set is not only scaled, but also rotated.

In this work, following Trecate *et al.*, first it is switched to the parameter space from the data space, and then weighted K-means algorithm using Mahalanobis distance is applied to cluster the data set [36]. This algorithm filters the data noise by switching to the parameter space and enables to use not only one scaling for the whole data set, but different scalings for different data sections. Another advantage is being able to distinguish the clusters that have the same dynamic model, but are spaciouly separated from each other.

The steps of the algorithm are summarized below:

Step-1: Creating Local Models:

Each data point in the data set is associated to a local model, so that a local dataset C_j collects the data point \mathbf{x}_j and its $c - 1$ neighboring points. At the end, one has n local datasets, and each set has c data points. For each local data set C_j , a parameter vector $\hat{\boldsymbol{\theta}}_j$ is estimated. The vector $\hat{\boldsymbol{\theta}}_j$ does not contain any information about the sparsity of the data points of C_j . Because of that, if there are two or more clusters that have the same model parameters, it is not possible to differentiate them. Therefore, mean value $\bar{\mathbf{x}}_j$ of the regressors data points \mathbf{x}_i , where $\mathbf{x}_i \in C_j$ has to be taken into account for each subset C_j :

$$\bar{\mathbf{x}}_j = \frac{1}{c} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i, \quad (3.6)$$

where $j = 1, 2, \dots, n$.

Now, a feature vector $\boldsymbol{\xi}_j$ bearing the information about the local model parameters and spatial localization of the parameters can be written as

$$\boldsymbol{\xi}_j = \begin{bmatrix} \hat{\boldsymbol{\theta}}_j^T \\ \bar{\mathbf{x}}_j \end{bmatrix}. \quad (3.7)$$

If the constant c is too big, then there will be local datasets that contain data from different clusters, which is not desired. If it is too small, then the created local models will have little amount of data points, and as a result, they will be very sensitive to the noise. Trecate *et al.* discuss the importance of the parameter c throughly in [37]. We introduce an intensity analysis, which is partly a remedy to choose c . The details of this analysis is given in section 3.3.3.

Step-2: Weighted Clustering of the Local Models

The local datasets C_j , $j = 1, 2, \dots, n$, are going to be divided into s clusters, using the K-means clustering principle. After defining the initial centers, (3.1) is re-defined using Mahalonobis norm:

$$J_C = \sum_{i=1}^s \sum_{\boldsymbol{\xi}_j \in D_i} \|\boldsymbol{\xi}_j - \hat{\boldsymbol{\mu}}_i\|_{\mathbf{R}_j}^2, \quad (3.8)$$

where matrix \mathbf{R}_j contains the parameter covariance matrix \mathbf{V}_j and the scatter matrix \mathbf{Q}_j that are defined below:

$$\mathbf{V}_j = \text{cov}(\hat{\boldsymbol{\theta}}_j) = \sigma_j^2 (\mathbf{X}_j^T \mathbf{X}_j)^{-1}, \quad (3.9)$$

$$\mathbf{Q}_j = (\mathbf{X}_j - \bar{\mathbf{x}}_j)^T (\mathbf{X}_j - \bar{\mathbf{x}}_j), \quad (3.10)$$

$$\mathbf{R}_j = \begin{bmatrix} \mathbf{V}_j & 0 \\ 0 & \mathbf{Q}_j \end{bmatrix}, \quad (3.11)$$

where \mathbf{X}_j represents the data points matrix belonging to the local model C_j , and σ_j^2 is the variance of noise. While noise cannot be measured, empirically it is possible to calculate its variance using the error signal, so that:

$$\hat{\sigma}_j^2 = \frac{\mathbf{e}_j^T \mathbf{e}_j}{c - n_m + 1}, \quad (3.12)$$

$$\approx \frac{\mathbf{y}_j^T \left(\mathbf{I} - \mathbf{X}_j (\mathbf{X}_j^T \mathbf{X}_j)^{-1} \mathbf{X}_j^T \right) \mathbf{y}_j}{c - n_m + 1}, \quad (3.13)$$

where n_m is the model order. The denominator in (3.12) and (3.13) represents the DoF of residuals [88]. After dividing the data set into the subspaces using (3.8), new centers are calculated by the following formula:

$$\tilde{\boldsymbol{\mu}}_i = \frac{\sum_{j, \boldsymbol{\xi}_j \in D_i} w_j \boldsymbol{\xi}_j}{\sum_{j, \boldsymbol{\xi}_j \in D_i} w_j}, \quad (3.14)$$

where w_j is the weighting factor. The peak of the Gaussian centered in $\boldsymbol{\xi}_j$ and with covariance \mathbf{R}_j is a measure for the weights. This step is repeated by assigning $\hat{\boldsymbol{\mu}}_i = \tilde{\boldsymbol{\mu}}_i$ in (3.8) until $\tilde{\boldsymbol{\mu}}_i$ does not change anymore.

Step-3: Define Final Models

The classification of the data performed in Step-2 allows for collecting data points \mathbf{x}_j under the i^{th} subset D_i , as long as $\boldsymbol{\xi}_j \in D_i$ applies. After that, for each subset D_i , a parameter identification method can be applied in order to define unknown model parameters.

3.2.3 Overview of Other Clustering Methods

In addition to the above given methods, there are several other ones to cluster and classify the experimental data for further investigations. It is neither possible to mention all of these methods here, nor is it the aim of this work. Thus, this section is confined to brief summaries of two further principles. More information on the topic can be obtained in [49, 88].

Principal Component Analysis (PCA)

PCA is a way to identify the pattern in data via transforming the coordinate axis to the optimal axis according to the maximum variance and dimensionality reduction. PCA implies that the direction with the highest data variance is the most important and the direction with the smallest data variance is the least significant [88].

Assume the regression matrix is $\mathbf{X} \in \mathfrak{R}^{n \times r}$ where n is the number of data points and r is the input space dimension. First, it is assured that the data has zero mean by subtracting the mean of the data from each data point. The covariance of $E\{\mathbf{X}\} = \mathbf{X}^T \mathbf{X}$ is calculated and normalized. Eigenvectors of this normalized covariance matrix represent the optimal axis and eigenvalues represent the variances along these optimal axis. Arranging the eigenvalues in a descending order enables one to distinguish “least significant” directions and eliminate them, if needed.

It should be noted that instead of clustering, PCA reduces noise and compresses the data. However, it is a matter of discussion why the input signals having the least variance should have the least importance as well.

Self Organizing Maps (SOMs)

The idea of SOM was introduced by Kohonen in 1989. It is a neural network approach to clustering, using vector quantization principles. The method is highly related to the K-means clustering, where clusters are replaced by neurons and each neuron has weights that are related to the initial centers.

The self-organizing map is a single layer feedforward network. SOMs are mainly used for dimensionality reduction rather than clustering. For more information refer to [88].

3.3 Hybrid Identification of the Trainer Skill

In general, there are four features of a hybrid system that have to be identified in order to obtain a complete representation of the system:

1. Number of discrete states s .
2. Switching conditions between states sw_{ij} . $i, j = 1, 2, \dots, s$
3. Model parameters θ_i , $i = 1, 2, \dots, s$.
4. Model order n_m .

It is expected that a hybrid identification method determines all of these features for given input and output data of the system. Since our aim is to represent the skill of a trainer as switching impedances, the model order n_m is fixed and known.

3.3.1 Skill as Switching Generalized Impedances

Human discrete decision making and continuous task execution mechanism suggest that sensorimotor skills can be represented as a hybrid automata, i.e. as a set of switching dynamic equations. In such a representation, each discrete state can be considered as a skill primitive.

Keeping this in mind, this work aims to capture the human skill as switching generalized impedances. A generalized impedance model relates the position and its derivatives to the force. The concept of the trainer skill as switching impedances is given in Figure 3.2. Explicitly, the i^{th} generalized impedance model is as follows:

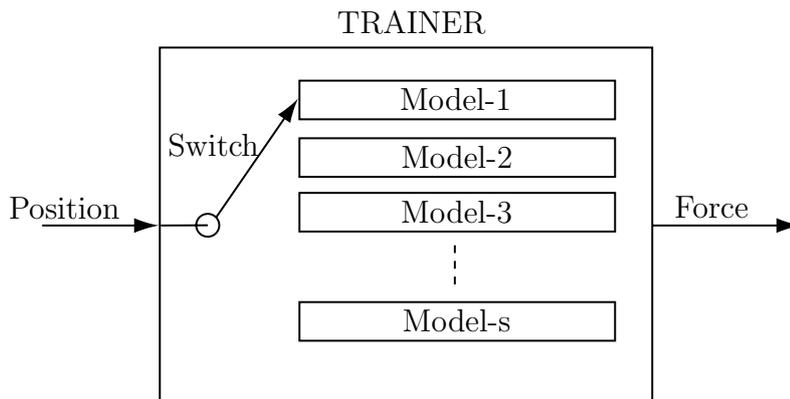


Figure 3.2: Trainer skill as switching impedances.

$$\hat{f}_i(k) = m_i \ddot{x}(k) + d_i \dot{x}(k) + k_i x(k) + f_{i,0}, \quad (3.15)$$

where $\hat{f}_i(k)$ is the model output force, $\ddot{x}(k), \dot{x}(k), x(k)$ are acceleration, velocity and position at time point k , respectively, and $\boldsymbol{\theta}_i = [m_i \ d_i \ k_i \ f_{i,0}]^T$ is the parameter vector of the i^{th} model with the elements mass, damping, stiffness and offset force, respectively. The number of data samples is n , so that $k = 1, 2, \dots, n$. If the number of discrete states, i.e., the number of models is s , then $i = 1, 2, \dots, s$. Thus, the hybrid identification problem (HIDP) to define human skill as switching impedances aims at determining following parameters:

$$\text{HIDP} : \begin{cases} \boldsymbol{\theta}_i = [m_i \ d_i \ k_i \ f_{i,0}]^T & i = 1, 2, \dots, s \\ sw_{ij} & i, j = 1, 2, \dots, s \\ s & \end{cases} \quad (3.16)$$

3.3.2 Least Squares Optimization for Impedance Models

After the data points are clustered using the unsupervised learning algorithm WKM, a supervised learning technique is used to identify $\boldsymbol{\theta}_i$ for each cluster i . The identification of $\boldsymbol{\theta}_i$ is a linear optimization problem. Least squares (LS) is a well-known method to solve such problems.

The regression matrix \mathbf{X} for impedance identification problem is

$$\mathbf{X} = [\ddot{\mathbf{x}} \ \dot{\mathbf{x}} \ \mathbf{x} \ \mathbf{1}], \quad (3.17)$$

where $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}} \in \mathbb{R}^{n \times 1}$, are position, velocity and acceleration column vectors. The last term $\mathbf{1}$ is also a column vector, the elements of which are equal to 1, and it is needed to identify the model offset.

Thus, (3.15) can be re-written as follows:

$$\hat{\mathbf{y}}_i = \mathbf{X}\boldsymbol{\theta}_i, \quad (3.18)$$

where $\hat{\mathbf{y}}_i$ is the model output force. The model error \mathbf{e}_i is

$$\mathbf{e}_i = \mathbf{y}_i - \hat{\mathbf{y}}_i = \mathbf{y}_i - \mathbf{X}\boldsymbol{\theta}_i. \quad (3.19)$$

where \mathbf{y}_i is the measured output, i.e. the measured force vector. The cost function J_C defining the least squares problem is

$$J_{C,LS}(\boldsymbol{\theta}_i) = \frac{1}{2} \mathbf{e}_i^T \mathbf{e}_i. \quad (3.20)$$

To minimize the cost function $J_{C,LS}(\boldsymbol{\theta}_i)$, its gradient with respect to the parameter vector $\boldsymbol{\theta}_i$ is set to zero. As a result, the unknown parameters can be estimated as

$$\boldsymbol{\theta}_i = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}_i. \quad (3.21)$$

Note that in the equations above, the index i is not shown for the regression matrix to simplify the notation.

3.3.3 WKM-Based Approach

The regression matrix \mathbf{X} and the output vector \mathbf{y} are built from the recorded data or from the time derivatives of the recorded data of the system to be modeled as a hybrid dynamic model. The first step is to split the regression space into adequately small pieces, so that each piece can be described as an impedance model. Following the performance of the user, it might be possible to realize this division manually. However, it is aimed to have a generic method, i.e. for any input - output data the division should be done automatically in an optimal way.

This division can be achieved by implementing a WKM-based approach in parameter space. The fundamentals of the method were explained in section 3.2.2. After creating the local data sets and transferring from data space to parameter space, the clustering algorithm is performed using the cost function given in (3.8).

Defining Number of States

The number of clusters s has to be known beforehand in order to perform a WKM clustering. At the same time, it is one of the hybrid system features that needs to be identified. Therefore, to define an optimal “ s ”, the clustering is performed several times in our work, starting from $s = 2$ and then increasing s each time by one. After clustering, LS optimization is applied to each subset, and by the help of 3.21, generalized impedance model parameters are identified. Consequently, the sum of absolute hybrid model errors is calculated and normalized as follows:

$$\bar{e}_s = \frac{\sum_{i=1}^s \sum_{k=1}^{n_i} |y_i(k) - \hat{y}_i(k)|}{n}, \quad (3.22)$$

where n_i is the number of data points in the cluster i , and $\sum_{i=1}^s n_i = n$. If

$$\bar{e}_s < \varepsilon_{ERR} \quad (3.23)$$

is satisfied and \bar{e}_s does not change anymore for increased number of states, then the iterations are stopped. ε_{ERR} is a pre-defined error threshold. Following that, a closer investigation on the hybrid models passing the threshold should be performed in order to deduce physical explanations of defined clusters. Among other benefits, the intensity analysis introduced below provides a possibility of such close investigations.

Define Switching Conditions: Intensity Analysis

The problems concerning the choice of parameter c (number of data points in one local dataset) in WKM clustering algorithm were outlined in section 3.2. To eliminate the undesired effects of the mixed local datasets and define physically interpretable switching conditions, an intensity analysis on the clustered data points is proposed in this work. It is expected that the data points that are close to each other in time belong to the same cluster. Thus, the suggested intensity analysis uses data windows, each with a range of h data points that are consecutive in time, i.e. if the number of total data points is n , a number of n/h windows are generated. In each data window, the distribution of the data points to the clusters $h_{dw,i}$ is checked. The cluster that contains most of the data points,

i.e. most intensive cluster, is then taken as the cluster representing all data points in this data window. The intensity $I_{dw,i}$ of cluster i in data window dw is:

$$I_{dw,i} = \frac{h_{dw,i}}{h}, \quad (3.24)$$

where

$$\sum_{i=1}^s I_{dw,i} = 1. \quad (3.25)$$

Before proceeding to another hybrid identification method, it should be noted that depending on the problem, it might be necessary to define hyperplanes that divide the data space appropriately after performing the clustering. Such a division can be achieved by applying Support Vector Machines (SVM). Detailed information on SVM can be found in [111].

3.3.4 Stochastically Switching Impedances (SS-Imp) Approach

In order to achieve a deterministic description of human skill, stochastic identification principles can be used to identify model parameters of a hybrid system.

Assume that a system is represented by a hidden Markov model (HMM) $\lambda(\mathbf{A}, \mathbf{b}, \boldsymbol{\pi})$ with s hidden states as in Figure 3.3. Following the work of Suzuki *et al.*, it is possible to assign a deterministic model to each discrete state [118]. In our work, each HMM state is represented by a generalized impedance model as in (3.15). Using classical HMM principles, the switching between models, as well as model parameters are to be identified. Consequently, the method presented in this section is called *Stochastically Switching Impedances* (SS-Imp).

The complication that we face is an extension of the 3rd classical problem of HMMs given by Rabiner [96]: Given an initial model λ , find a model $\bar{\lambda}$ iteratively, so that the probability $P(\mathbf{O}|\bar{\lambda})$ is maximized, i.e. the probability of observing the sequence $\mathbf{O} = O_1, O_2, \dots, O_n$ is highest for the model $\bar{\lambda}$. The extension in our case is that the HMM states are impedance models, i.e. deterministic functions. Therefore, the observation probability distribution \mathbf{b} is defined in such a way that it contains model parameters, as given by Suzuki *et al.* [118]. The procedure of developing a SS-Imp model is summarized by following 4 steps.

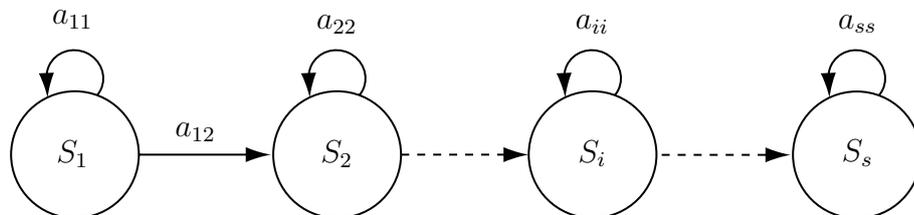


Figure 3.3: A left-to-right HMM, whose states are assigned to dynamic models.

Step-1: Define States as Impedance Models

An impedance model is assigned to each state, so that $\hat{\mathbf{y}}_i = \mathbf{X}\boldsymbol{\theta}_i, i = 1, 2, \dots, s$ as described in Section 3.3.2. Note that the regression matrix is our observation sequence for the HMM. For the i^{th} state, the generalized impedance equation is given in (3.15) explicitly.

Step-2: Initialize the HMM

To start the iterative identification process, the parameters \mathbf{A}, \mathbf{b} , and $\boldsymbol{\pi}$ of an initial HMM $\lambda(\mathbf{A}, \mathbf{b}, \boldsymbol{\pi})$ have to be specified. This can be done randomly for \mathbf{A} and $\boldsymbol{\pi}$, by taking into account that

$$a_{ij} \geq 0; \quad \sum_{j=1}^s a_{ij} = 1, \quad (3.26)$$

$$\pi_j \geq 0; \quad \sum_{j=1}^s \pi_j = 1, \quad (3.27)$$

where $a_{ij} \in \mathbf{A}$ is the transition probability from state i to state j .

Following Suzuki *et al.* [118], the output probability $b_i(O_k)$ of state i at time point k is defined via Gaussian distribution of the equation error:

$$b_i(O_k) = \frac{1}{\sqrt{2\pi}\sigma_i} \cdot \exp \left\{ -\frac{(m_i\ddot{x}(k) + d_i\dot{x}(k) + k_ix(k) + f_{i,0} - f_m(k))^2}{2\sigma_i^2} \right\}, \quad (3.28)$$

where $f_m(k)$ is the measured force at time instant k . In equation (3.28), the elements of the parameter vector $\boldsymbol{\theta}_i = [m_i \ d_i \ k_i \ f_{i,0}]^T, i = 1, 2, \dots, s$ and the variance σ_i^2 are unknown. Therefore, the initial HMM is re-written as $\lambda = (\mathbf{A}, \boldsymbol{\theta}, \sigma^2, \boldsymbol{\pi})$

Step-3: The Forward Backward Procedure

Using the parameters of the initial HMM, it is evaluated how well the model matches the given observation sequence \mathbf{O} . To answer this question in a computationally feasible way, it is necessary to define auxiliary variables that are the forward α and the backward β variables.

The forward variable $\alpha(i, k)$ is the probability of being in state i at time k , and observing the partial sequence O_1, O_2, \dots, O_k . Initializing the variable for $k = 1$ as $\alpha(i, 1) = \pi_i b_i(O_1)$, the rest can be calculated in an inductive way as follows:

$$\alpha(j, k + 1) = \left[\sum_{i=1}^s \alpha(i, k) a_{ij} \right] b_j(O_{k+1}). \quad (3.29)$$

This iteration is terminated by taking the sum of the last forward variables $\alpha(i, n)$, which probabilistically shows how well the model matches the observation sequence:

$$P(\mathbf{O} | \lambda) = \sum_{i=1}^s \alpha(i, n). \quad (3.30)$$

The backward variable β is defined in a similar manner, but it is initiated at time $k = n$ and the induction is backwards until $k = 1$. Refer to [96] or other HMM tutorials for a detailed information on β .

Step-4: Estimation of the HMM Parameters and Maximum Likelihood

Using the classical HMM definitions given by Rabiner [96] and the SS-Imp principles by Suzuki [118], the new HMM parameters $\bar{\mathbf{A}}, \bar{\boldsymbol{\theta}}, \bar{\sigma}^2, \bar{\pi}$ are estimated as follows:

$$\bar{\pi}_i = \frac{\pi_i b_i(O_1) \beta(i, 1)}{\sum_{j=1}^s \pi_j b_j(O_1) \beta(j, 1)}. \quad (3.31)$$

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{n-1} \alpha(i, k) a_{ij} b_j(O_{k+1}) \beta(j, k+1)}{\sum_{l=1}^s \sum_{k=1}^{n-1} \alpha(i, k) a_{il} b_l(O_{k+1}) \beta(l, k+1)}. \quad (3.32)$$

$$\bar{\boldsymbol{\theta}}_i = \left\{ \sum_{k=1}^n \mathbf{x}_k^T \mathbf{x}_k \alpha(i, k) \beta(i, k) \right\}^{-1} \times \left\{ \sum_{k=1}^n \mathbf{x}_k^T y_k \alpha(i, k) \beta(i, k) \right\}. \quad (3.33)$$

$$\bar{\sigma}_i^2 = \frac{\sum_{k=1}^n |\mathbf{x}_k \bar{\boldsymbol{\theta}}_i - y_k|^2 \alpha(i, k) \beta(i, k)}{\sum_{k=1}^n \alpha(i, k) \beta(i, k)}. \quad (3.34)$$

In these equations, \mathbf{x}_k refers to the k^{th} row of \mathbf{X} and y_k is the k^{th} value of the output vector \mathbf{y} that corresponds to \mathbf{f}_m .

It is expected that the new HMM $\bar{\lambda} = (\bar{\mathbf{A}}, \bar{\boldsymbol{\theta}}, \bar{\sigma}^2, \bar{\pi})$ maximizes Baum's auxiliary function, which is

$$Q(\lambda, \bar{\lambda}) = \sum_{\mathbf{U}} P(\mathbf{U} | \mathbf{O}, \lambda) \log[P(\mathbf{O}, \mathbf{U} | \bar{\lambda})] \quad (3.35)$$

where $\mathbf{U} = s(1), s(2), \dots, s(n)$ is the state sequence, n is the number of observations, i.e. the number of data points. Baum *et al.* proved that maximizing \bar{Q} increases the maximum likelihood, i.e.

$$\max[Q(\lambda, \bar{\lambda})] \Rightarrow P(\mathbf{O} | \bar{\lambda}) \geq P(\mathbf{O} | \lambda). \quad (3.36)$$

The initial HMM λ is replaced by the new HMM $\bar{\lambda}$, and the steps 2 – 4 are repeated, until the probability $P(\mathbf{O} | \bar{\lambda})$ does not increase anymore, i.e. maximum likelihood is reached.

Numerical Instabilities, Scaling and Logarithmic Maximum Likelihood

The parameters \mathbf{A} , \mathbf{b} and $\boldsymbol{\pi}$ of an HMM contain elements that have values between 0 and 1. Actually, many of these elements tend to be zero. Considering (3.29) again, it is clear that for each induction step, these elements are multiplied and summed with each other in order to define the forward variable α . Thus, when the number of time points n , i.e. the number of rows in the regression matrix, increases, α and β variables approach zero exponentially. As a result, beyond a certain number n , the precision of any machine is exceeded. In the literature, it is given that even double precision computers cannot process if $n \geq 100$ [96].

To model human skill as switching impedances when he/she interacts with real or virtual objects, the force measurement has to run at a high frequency, approx. 1 kHz. That means 1000 data points in one second at interaction. Since generally the time needed to realize a task comprises several seconds, the data space to identify is very large. Therefore, without scaling, generating an HMM for this problem would inevitably lead to numerical instabilities. For HMMs, scaling is a kind of normalization that is independent of the state i , but depends on time. The first scaling factor $c(1)$ is:

$$c(1) = \frac{1}{\sum_{i=1}^s \alpha(i, 1)}. \quad (3.37)$$

Consecutively, the scaled forward variable for $k = 1$ is:

$$\hat{\alpha}(i, 1) = c(1)\alpha(i, 1). \quad (3.38)$$

In an inductive way, the scaling of all α variables is done as below:

$$\bar{\alpha}(j, k + 1) = \left[\sum_{i=1}^s \hat{\alpha}(i, k) a_{ij} \right] b_j(O_{k+1}), \quad (3.39)$$

$$c(k + 1) = \frac{1}{\sum_{i=1}^s \bar{\alpha}(i, k + 1)}, \quad (3.40)$$

$$\hat{\alpha}(i, k + 1) = c(k + 1)\bar{\alpha}(i, k + 1). \quad (3.41)$$

To scale the backward variable β , the same scaling factor can be used, since the magnitude of α and β is comparable. Thus, the scaled backward variable $\hat{\beta}$ at time k is:

$$\hat{\beta}(i, k) = c(k)\bar{\beta}(i, k). \quad (3.42)$$

In order to obtain numerically stable iterations, in equations (3.31)-(3.34), α and β have to be replaced with $\hat{\alpha}$ and $\hat{\beta}$ respectively.

Note that the probability $P(\mathbf{O} | \lambda)$ cannot be computed using (3.30) anymore, since the forward variables are scaled. Thus, using the definition of the scaling factors, $P(\mathbf{O} | \lambda)$ can be represented as:

$$P(\mathbf{O} | \lambda) = \frac{1}{\prod_{k=1}^n c(k)}. \quad (3.43)$$

However, if one tries to implement equation (3.43) directly, numerical instabilities occur again. To prevent that, it is necessary to move to the logarithmic scale, so that:

$$\log[P(\mathbf{O} | \lambda)] = - \sum_{k=1}^n \log[c(k)]. \quad (3.44)$$

Best State Sequence Depending on the Viterbi Algorithm in Logarithmic Scale

The classical algorithm to find the best state sequence \mathbf{S}_b , i.e. the switching points of an HMM, is the Viterbi algorithm. This algorithm keeps track of the best score (highest probability) $\delta_{i,k}$ along a simple path at time k , which accounts for the first k observations $O_1 O_2 \cdots O_k$ and ends in state S_i [96].

$$\delta_{i,k} = \max_{s_1, s_2, \dots, s_{k-1}} P[s_1 s_2, \dots, s_k = i, O_1 O_2 \cdots O_k | \lambda]. \quad (3.45)$$

Since a scaling had to be applied, we do not have the information $P(\mathbf{O} | \lambda)$, but $\log[P(\mathbf{O} | \lambda)]$. Therefore, the best score $\phi_{i,k}$ has to be defined in logarithmic scale similar to $\delta_{i,k}$ as:

$$\phi_{i,k} = \max_{s_1, s_2, \dots, s_{k-1}} (\log P[s_1 s_2, \dots, s_k = i, O_1 O_2 \cdots O_k | \lambda]). \quad (3.46)$$

In analogy to the calculation of α , the best score can be calculated in a recursive way after initializing it as:

$$\phi_{i,1} = \log(\pi_i) + \log[b_i(O_1)], \quad (3.47)$$

$$\psi_{i,1} = 0, \quad (3.48)$$

where ψ is needed to track the argument that maximizes ϕ . The recursion step is:

$$\phi_{i,k} = \max_{1 \leq i \leq s} [\phi_{i,k-1} + \log(a_{ij})] + \log(b_j(O_k)) \quad (3.49)$$

$$\psi_{i,k} = \arg \max_{1 \leq i \leq s} [\phi_{i,k-1} + \log(a_{ij})] + \log(b_j(O_k)). \quad (3.50)$$

The most probable last state is given by:

$$S_n = \arg \max_{1 \leq i \leq s} [\phi(i, n)]. \quad (3.51)$$

Using S_n , the best state sequence is then determined via backtracking as:

$$S_k = \psi_{i,k+1}(S_{k+1}), \quad (3.52)$$

where $k = n - 1, n - 2, \dots, 1$.

Switching Depending on Minimum Model Error

There is no unique solution to define the best state sequence for a given observation array. Since it is an optimization problem, there can be different solutions depending on the optimality criteria. When Viterbi algorithm is applied, the iterations are continued as long as the probability of observing the sequence \mathbf{O} is bigger in the latest HMM $\bar{\lambda}$ than in the previous one λ , i.e. as long as equation (3.36) holds. In this case, a model that is being better than another one can be expressed only comparatively. Therefore, the risk of local maxima exists.

However, since the discrete states are represented by deterministic models in our HMMs, it is possible to define another criterion, so that the HMM iterations are carried on until a pre-defined minimum error is reached for each data point. If this condition is satisfied, the switching points are determined via keeping at time point k the state i active, which gives the minimum model error in comparison with other states.

The HMM parameter set after the estimation $\bar{\lambda} = (\bar{\mathbf{A}}, \bar{\boldsymbol{\theta}}, \bar{\sigma}^2, \bar{\pi})$ should minimize the model error which is defined as follows:

$$err_k = \min_{1 \leq i \leq s} |y(k) - \hat{y}_i(k)|, \quad (3.53)$$

$$S_k = \arg \min_{1 \leq i \leq s} |y(k) - \hat{y}_i(k)|, \quad (3.54)$$

where $y(k)$ is measured output signal and $\hat{y}_i(k)$ is the model output of the i^{th} state at time point k . The iteration of the HMM parameters is continued until the following condition is satisfied:

$$err_k \leq \varepsilon, \quad \forall k = 1, 2, \dots, n \quad (3.55)$$

where ε is a pre-defined maximum allowed model error.

Since this approach tries to minimize the model error at each data point, it is expected that the best state sequence introduces redundant, noisy switchings. To prevent this, the intensity analysis explained in section 3.3.3 should be applied to filter the unnecessary switchings. One should be careful that this approach may lead to switchings that for the given model are impossible to occur. Another solution for redundant switchings is applying the logarithmic Viterbi algorithm to the latest HMM model, after the equation (3.55) is satisfied. This solution also guarantees switchings to be possible.

It is credible to extend this approach by constraining the normalized squared sum of hybrid model error instead of constraining the error at each data point.

3.3.5 Comparison of the Methods

A hybrid system that consists of switching autoregressive (ARX) models is generated as in the equations (3.56) - (3.58) in order to check the reliability of the proposed methods.

$$\text{Model - 1} : y(k) = 5x(k) + 2 + \nu; \quad x \in [-4, -1] \quad (3.56)$$

$$\text{Model - 2} : y(k) = -2x(k) + \nu; \quad x \in (-1, 2) \quad (3.57)$$

$$\text{Model - 3} : y(k) = x(k) + 7 + \nu; \quad x \in [2, 4], \quad (3.58)$$

where ν is Gaussian noise, and $k = 1, 2, \dots, 50$, i.e. the number of data points is $n = 50$. The elements of $x(k)$ are randomly defined in the range of $[-4, 4]$ and then sorted in the ascendant order.

As illustrated by Figure 3.4, K-means algorithm divides the data into 3 clusters using the equation (3.1). The switching between clusters highly depends on the data distribution, i.e. the algorithm is very sensitive to noise and outliers. Two data points shown in a circle in Figure 3.4 originally belong to the first cluster, but K-means algorithm assigns them to the second cluster. Therefore, the parameters of the second model that were identified by using LS optimization, differ from the original model given in (3.57). All model parameters identified by K-means algorithm are given in Table 3.1.

The same hybrid system is identified again using WKM algorithm. As it can be seen from Figure 3.5, the algorithm fits the 3 clusters to the data better than the normal K-means algorithm. The two encircled points are correctly grouped in cluster 1. Thanks to that, the identified model parameters presented in Table 3.2 are very close to the ones defined by (3.56)-(3.58). The constant c in (3.6) is chosen as 10 to reach this results, and it is obtained through a trial-and error approach.

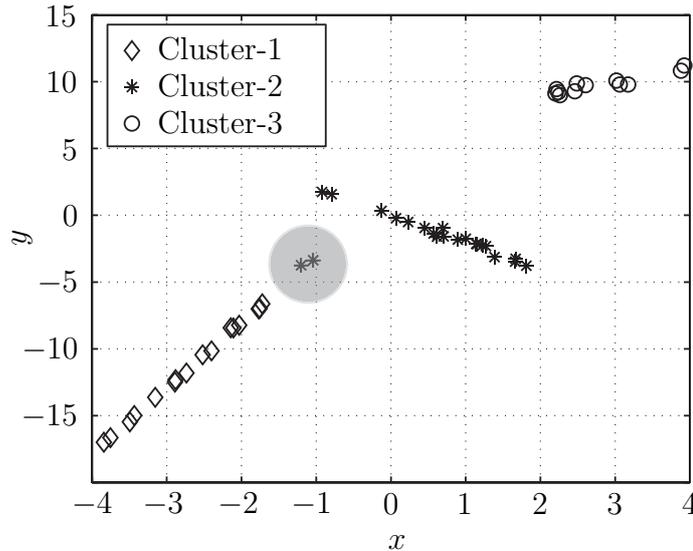


Figure 3.4: The data clustered by the K-means algorithm. Two encircled data points are wrongly clustered.

Table 3.1: Identified models using K-means clustering and LS optimization.

Model-1	$y_{km}(k) = 4.90x(k) + 1.77$
Model-2	$y_{km}(k) = -0.85x(k) - 1.19$
Model-3	$y_{km}(k) = 1.01x(k) + 6.96$

Lastly, the identification of the hybrid system is realized by the SS-Imp method. As the parameters to identify are not impedance parameters, this application here should rather be called SS-ARX as it was done by Suzuki *et al.* [118]. As in the case of using WKM, SS-ARX method clusters the data points correctly. Therefore, Figure 3.5 can be referred to for this method as well. A critical issue in the performance of SS-ARX method is the iteration and switching dependency. If the classical way is used, i.e. maximum likelihood and Viterbi algorithm, then the method highly depends on the initialization of the HMM λ_i , especially it is very sensitive to σ_i and θ_i . Assigning these parameters randomly causes the HMM to be numerically unstable or not to converge to the correct model parameters and switching conditions. However, the method developed in this work, i.e. switching

Table 3.2: Identified models using weighted K-means clustering and LS optimization.

Model-1	$y_{km}(k) = 4.94x(k) + 1.88$
Model-2	$y_{km}(k) = -1.97x(k) + 0.02$
Model-3	$y_{km}(k) = 1.01x(k) + 6.96$

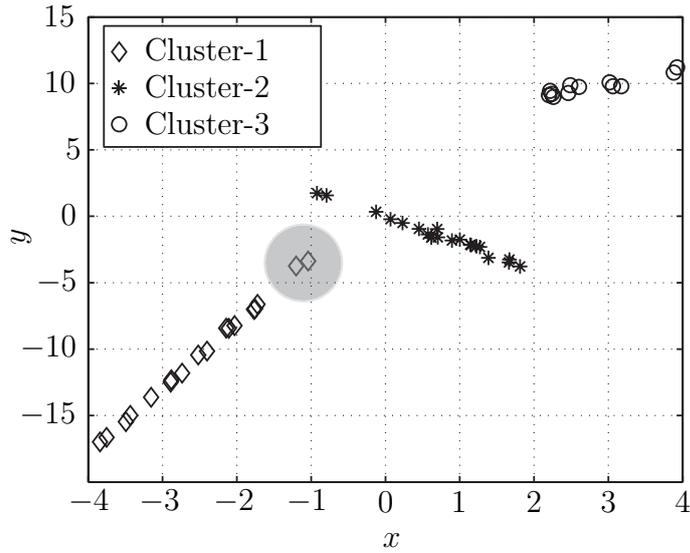


Figure 3.5: The data clustered by the weighted K-means algorithm. Two encircled data points are correctly clustered. The same clustering performance is also obtained by using SS-IMP method.

depending on the model error, is highly robust against initial conditions. By observing the change in the model error after each iteration, it is possible to follow the whether the desired pre-defined error is converged or not. The identified model parameters via SS-ARX principles using equation (3.33) are given in Table 3.3.

Table 3.3: Identified models using HMM.

Model-1	$y_{km}(k) = 4.94x(k) + 1.92$
Model-2	$y_{km}(k) = -1.96x(k) + 0.04$
Model-3	$y_{km}(k) = 1.05x(k) + 6.86$

A comprehensive discussion on the WKM and SS-IMP methods is given in section 3.6.

3.4 Stability of the Hybrid Models

Special attention has to be paid to the stability of the switching (hybrid) systems. That is mainly because of the known fact that a switching system may become unstable, even though all the sub-systems are stable. Stability analysis for different switching cases that occur in our problems is outlined below.

Only Left-to-Right Switching Systems

If the state transitions are in a left-to-right manner as in Figure 3.3, after having switched from state i to $i + 1$, there is no return to state i . In this case, for the switching system’s stability, it is necessary and sufficient that all sub-systems are stable. Assume that the

Lyapunov function for the state i is $V_i(\mathbf{x})$. Then, stability is assured by:

$$\dot{V}_i \leq 0 \quad \forall \mathbf{x}, \quad i = 1, 2, \dots, s. \quad (3.59)$$

Common-Lyapunov Functions

If the switching between states is arbitrary, then the stability condition given in equation (3.59) is not valid anymore. However, if it is possible to define a common Lyapunov function V_c , which is a Lyapunov function for all subsystems, then the necessary and sufficient condition for stability is:

$$\dot{V}_c \leq 0. \quad (3.60)$$

Multiple-Lyapunov Functions

If the switching between states is arbitrary, and it is impossible to define a common Lyapunov function, then the stability analysis depends on the individual Lyapunov function V_i , and its relative value with respect to the time points of switching to the state i .

Assume that the state i is entered at time point t_k . For any later time instant t_l , the relative value of V_i must be smaller. The necessary and sufficient condition for stability is:

$$\begin{aligned} V_i(t_k) - V_i(t_l) &\geq 0, & \forall \quad i, i = 1, 2, \dots, s \\ & & \forall \quad l > k. \end{aligned} \quad (3.61)$$

Refer to [72] for comprehensive information on the stability of switching systems.

3.5 Applications

3.5.1 Hybrid Trainer for a Bone Drilling Medical Training System

Hybrid Identification Problem

Main principles and the technical background of the bone drilling MTS are given in section 2.4. In this section, the hybrid identification of the interaction of an expert with BD-MTS is investigated. The aim is to describe the interaction as switching generalized impedances in the form given by (3.15). Known and unknown elements of the hybrid identification problem are given below:

Known: Model orders.

Unknown: Number of sub-spaces (discrete states) s , parameters $\boldsymbol{\theta}_i$ of the models, switching conditions sw_{ij} .

For this problem, the regression matrix is $\mathbf{X} = [\ddot{\mathbf{x}}_{fil} \quad \dot{\mathbf{x}}_{fil} \quad \mathbf{x} \quad \mathbf{1}]$, where \mathbf{x} is the measured position, and $\dot{\mathbf{x}}_{fil}$ and $\ddot{\mathbf{x}}_{fil}$ are derived and low-pass filtered velocity and acceleration signals respectively. The measured output vector is $\mathbf{y}_{id} = \mathbf{f}_m$.

Thus, the hybrid identification problem (HIDP) for BD-MTS aims at finding the following parameters:

$$\text{HIDP}_{\text{BD-MTS}} : \begin{cases} \boldsymbol{\theta}_i = [m_i & d_i & k_i & f_{i,0}]^T & i = 1, 2, \dots, s \\ sw_{ij} & & & & i, j = 1, 2, \dots, s \\ s & & & & \end{cases} \quad (3.62)$$

The identification problem is first solved using the WKM approach and then using the SS-IMP approach. The results are discussed and compared with each other.

The Model Using WKM-Based Approach

The regression matrix has n rows and 4 columns, i.e. $\mathbf{X} \in \mathbb{R}^{n \times 4}$. By trial-and-error, the c constant for the moving window in the regressors space is chosen as 1000. Thus, n local data groups LD_j $j = 1, 2, \dots, n$ are generated, each having 1000 data points. For each local data group LD_j , LS optimization is applied. As a result, n local models are identified. The clustering is then performed on these local models.

The data range is chosen in such a way that it enables to model the trainee's approach and starting phases of drilling, together with the routine drilling phase. The resulting data has 14000 data points, i.e. $n = 14000$. The weighted K-means clustering algorithm followed by LS optimization for parameter identification is repeated several times by keeping all system parameters constant, but increasing the number of discrete states. After assigning an error tolerance of $\varepsilon_{ERR} = 3.5$ N, the constrained sum of the absolute model error \bar{e}_s criterion given by equations (3.22 - 3.23) is implemented, in order to define s . In Figure 3.6, \bar{e}_s is given for $s = 1, 2, 3, 4, 5, 10, 13$. As can be seen, increasing the number

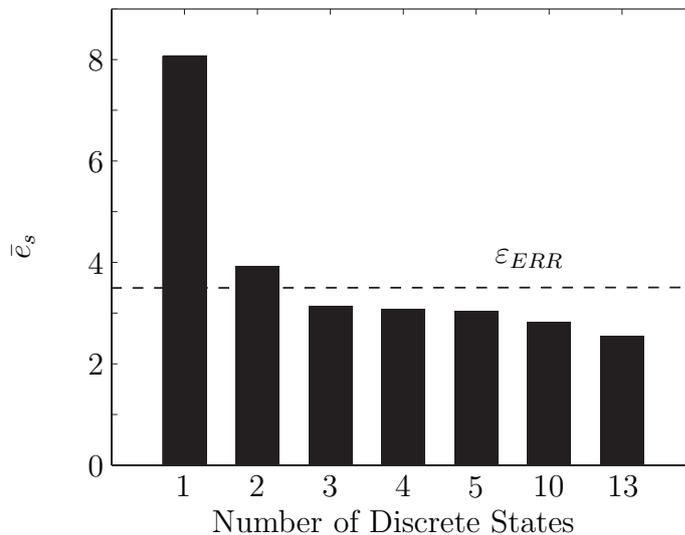


Figure 3.6: Comparison of the normalized sum of the absolute model errors for the increasing number of discrete states.

of discrete states decreases the error index \bar{e}_s . However, an increased number of states implies more complicated switching structure between models and less physical meaning of the clusters. Therefore, a small error index cannot be the only criterion to choose the number of discrete states. Thus, a closer investigation of the hybrid models passing the threshold ε_{ERR} is performed. The current problem allows models with at least 3 discrete states to be qualified for further investigations. The intensity analysis given in section 3.3.3 is used for investigations on $s = 3, 4, 5$ discrete states cases, and the results are depicted in Figure 3.7. The number of data points in one data window is set to $h = 100$. Since the total number of data points is $n = 14000$, there are 140 data windows (dw). For example, the 32nd dw contains data points in the interval 3101 – 3200. At first glance, 3 main intervals are recognizable in Figures 3.7(a),(b) and (c) independent of the total number

of clusters (discrete states). The first interval contains data windows 1 – 13, the second one 14 – 81 and the third one 82 – 140. Since there are 3 physically assumable states in BD-MTS, namely freespace movements, first contact and approaching, and drilling with a correct velocity and force, these three intervals match the assumption. As it is illustrated in the Figure 3.7(b) and (c), while $s = 4$ and $s = 5$, the fourth cluster introduces an extra interval around 80 – 90 dws, i.e. the drilling step is divided into two clusters. However, as can be seen in Figure 3.7(c), the fifth cluster does not bring any extra dw interval, rather some dw points. One should extract from this figure that increasing the number of clusters naturally reduces the number of data points per cluster. Consequently, some clusters contain very few data points and it is very difficult to find physical meaning of these clusters, that is, it is rather hard to associate such clusters with any skill states. In our analysis, we chose the number of states by taking the maximum number of clusters, when all represented by an interval of dw. Thus, for the current problem, the number of discrete states to represent trainer skill is chosen as $s = 4$.

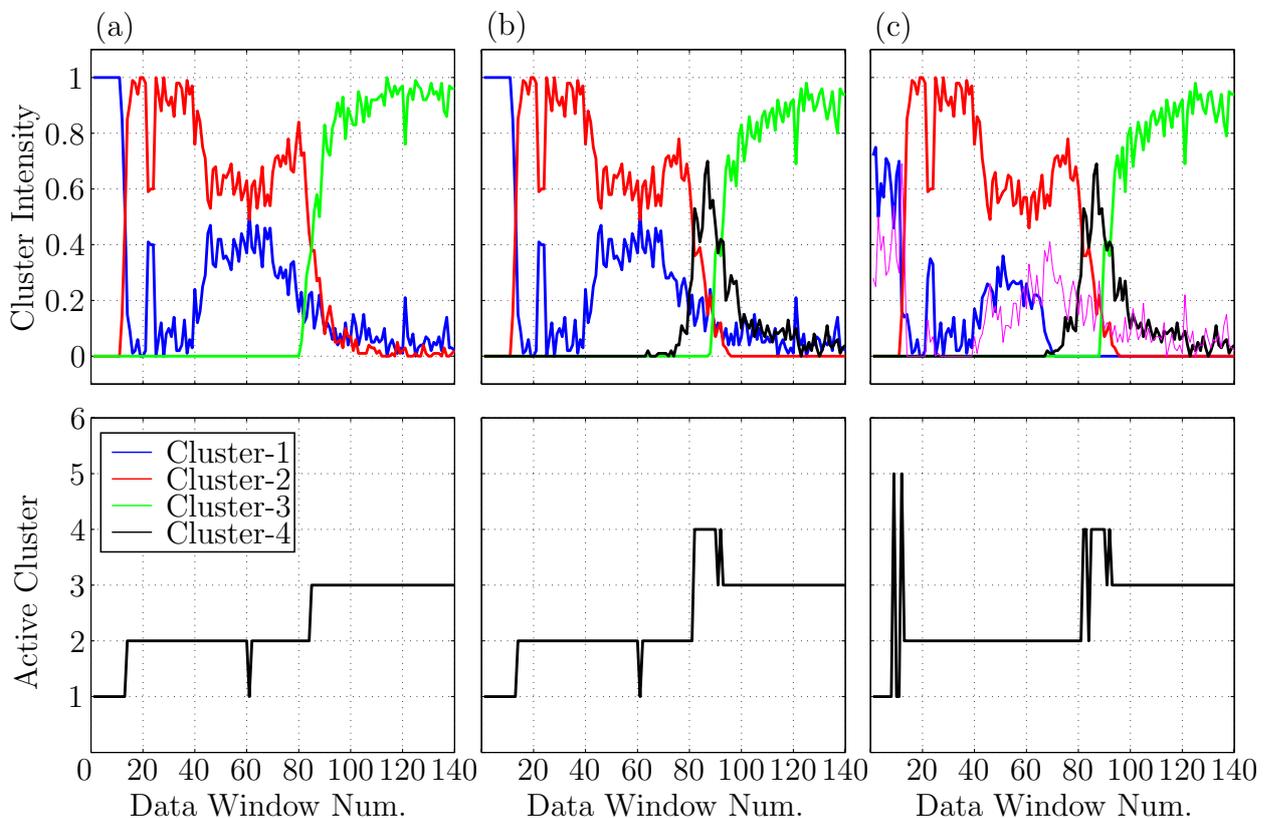


Figure 3.7: Intensity analysis results for (a) 3 clusters, (b) 4 clusters, (c) 5 clusters.

Since the number of states is defined as $s = 4$, different expert performances are divided into four clusters via WKM method. In general, three phases of drilling procedure, “freespace”, “approach” and “drilling”, are very clearly outstanding. Mostly drilling phase, but sometimes freespace phase, is divided into two clusters. Applying the LS optimization for each cluster, four impedance parameter vectors are identified for each expert trial. The parameter vectors for individual expert trials are grouped under the phases mentioned above, and for each phase a characteristic feature represented by impedance parameters is

looked for. The ones which establish common characteristics are put together and depicted in Figure 3.8. It was expected that it would be difficult to determine characteristic behaviors in freespace phase. The reasons for this expectation are that the user is not constricted with an environment and there is no rule to move in freespace. Additionally, in freespace haptic display's eigen dynamics play an important role. However, it is possible to conclude some characteristic features. As it can be seen in Figure 3.8(c), mass m_{fs} is always positive and insignificant in comparison with spring k_{fs} and damping d_{fs} coefficients. The range of k_{fs} is $[110, 450]$ N/m, and d_{fs} limits are $[-140, -450]$ Ns/m, when the only trial (first bar) that does not fit into this schema is neglected.

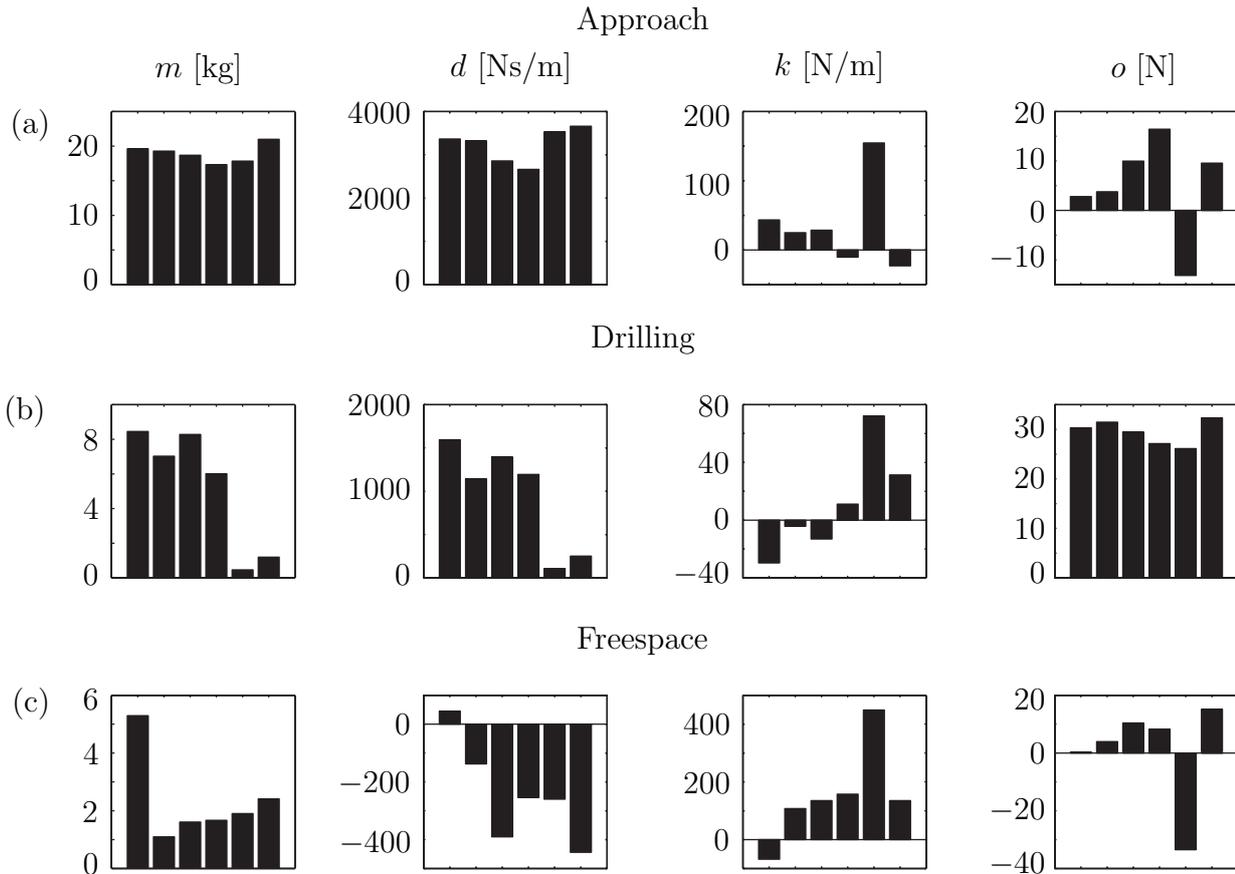


Figure 3.8: Different expert trials and resulting mass m , damping d spring k coefficients and offset o for freespace, approach and drilling phases.

There are distinguishing features for “approach” and “drilling” phases. Note that in both of these phases, the interaction is already defined by the non-linear relation between the velocity and the force. As a result, identified damping coefficient d that relates velocity to the force plays the most important role. In the approach phase, d is identified varying around 3000. The mean value is $\bar{d}_{app} = 3232$ Ns/m. Mass and spring coefficients are much less powerful than damping coefficient. Mass has always a positive value and a mean of $\bar{m}_{app} = 18.9$ kg. Spring coefficient k_{app} changes in a range of $[-20, 150]$, with a mean value $\bar{k}_{app} = 36.4$ N/m.

The drilling phase parameters can be interpreted in a similar way. Again, damping coefficient plays the most important role, but has smaller amplitude than the approach phase

values. Additionally, there is not only one, but two different drilling characteristics that can be deduced from the Figure 3.8(b). The first group (first four bars) has a dominating damping coefficient $d_{dr,1}$ in comparison with mass $m_{dr,1}$ and spring $k_{dr,1}$ coefficient. The mean values in this drilling case are $\bar{d}_{dr,1} = 1332$, $\bar{k}_{dr,1} = -9$ and $\bar{m}_{dr,1} = 7$. The second group (last two bars) has still damping as the largest parameter, but it is not as indicative as first group parameter. Additionally, spring also plays a significant role to define the characteristic of drilling. In this case, the mean values are $\bar{d}_{dr,2} = 180$, $\bar{k}_{dr,2} = 52$ and $\bar{m}_{dr,2} = 1$. Note that the ratio $\bar{k}_{dr,2}/\bar{d}_{dr,2}$ is much higher than the ratio $\bar{k}_{dr,1}/\bar{d}_{dr,1}$. Because of this situation in drilling, it is not improper to consider that there are two types of experts and they are differentiated from each other with their drilling performance.

Since the results of the experts are similar in cluster switching points and identified model parameters are reasonably in a certain range, any expert result can be chosen as the trainer model. In our work, a trainer model is chosen with positive coefficients. This is important for stability issues, which were explained in section 3.4. It is possible to define each discrete state with a second order transfer function, since impedance models are linear 2^{nd} order dynamic equations. Thus, equation (3.15) can be re-written as:

$$G_i(s) = \frac{X_i(s)}{F_i(s)} = \frac{1}{m_i s^2 + d_i s + k_i}. \quad (3.63)$$

Stability of each subsystem individually is assured if the parameters m_i, d_i, k_i are positive, which explains the reason of choosing a trainer model with positive parameters. Note that it is not possible to have all parameters positive for the freespace phase. Therefore, in the case of only left-to-right switching, the complete switching system is stable, if the freespace state is left out.

In addition to the stability, passive behavior of the models is desirable as well. The storage function for each state can be defined as:

$$V_{s,i} = \frac{1}{2} \tilde{k}_i x_{1,i}^2 + \frac{1}{2} x_{2,i}^2, \quad (3.64)$$

where $\tilde{k}_i = k_i/m_i$. It is clear that each discrete state, except the freespace state, is passive regarding velocity $x_{2,i} = \dot{x}_{1,i}$ as the input and force f_i as the output, since they satisfy following condition:

$$\dot{V}_{s,i} \leq x_{2,i} \cdot f_i. \quad (3.65)$$

Consecutively, the resulting chosen parameters are given in Table 3.4. Now, it is interesting and necessary to show how well the identified model represents the expert, and how well the identified switching conditions match the physically assumed switching points of BD-MTS. Figure 3.9 shows the clustered data groups for position, velocity, acceleration and force, whereas Figure 3.10 compares the output force of the hybrid trainer model and the measured force when trainer performs a task in BD-MTS. It is clearly determined that first switching occurs, when the drill and bone model are collided. Sudden decrease in the velocity and constant position are clear indicators for this statement. Thus, WKM clearly identifies the point, where freespace finishes and approach phase starts. Second switching occurs, when velocity increases to the expected drilling velocity and is kept constant. Thus, the transition from approach phase to drilling phase is clear as well. The division of drilling phase into two states and switching between these states are not as clear as the previous switching conditions. However, it is possible to deduce that the first drilling

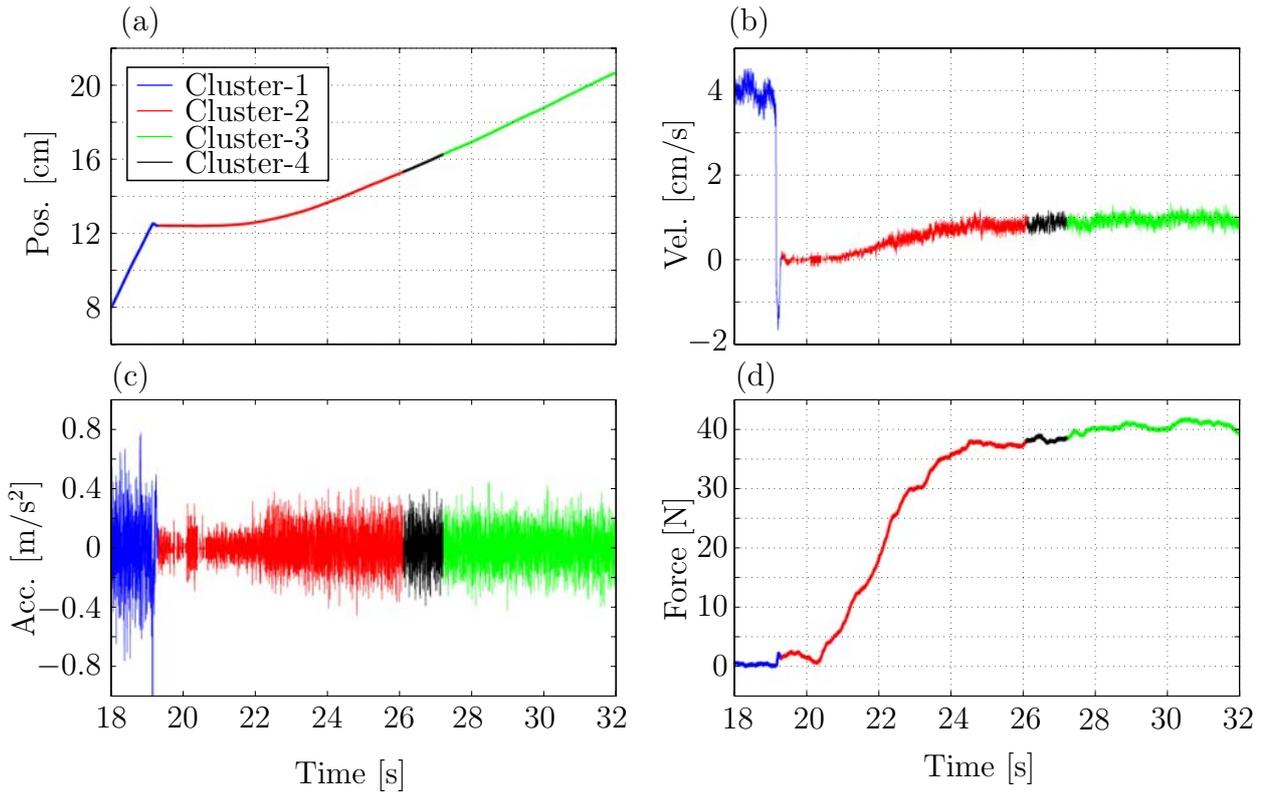


Figure 3.9: Identified clusters using WKM algorithm. (a) Position, (b) velocity, (c) acceleration, (d) force.

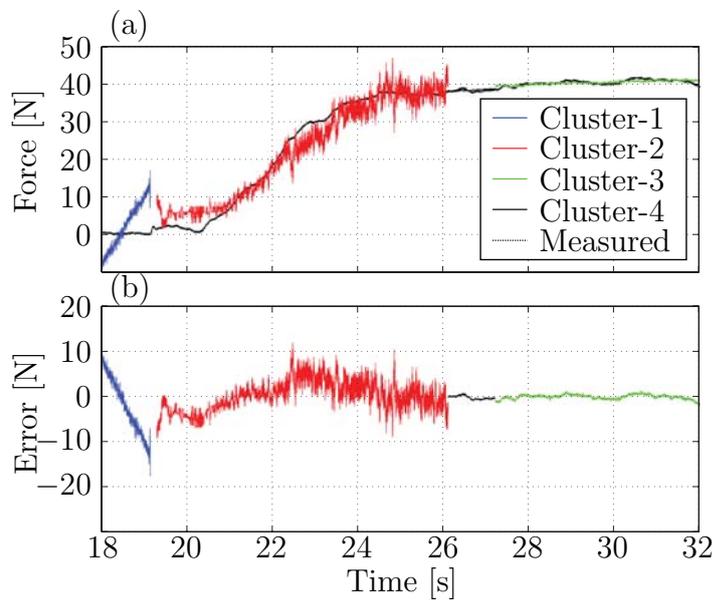


Figure 3.10: (a) WKM model output force vs. measured force. (b) WKM model error.

state concerns the situation when the trainer starts drilling by applying the force and the velocity in the correct range, but still slightly increasing them. The second state occurs, when both are constant.

Note that although the LS optimization is applied to the clusters defined by WKM algorithm to define unknown model parameters, the switching conditions are defined following the intensity analysis. This results in increased model errors, especially in freespace. Since freespace is not considered as a part of trainer model, and normalized error for one data point is in the acceptable range, the model is satisfying. As an alternative, it is also possible to apply LS optimization after intensity analysis, i.e. after switching conditions are clarified.

Table 3.4: Identified models and position based switching conditions, obtained by using WKM clustering with 4 states and LS optimization for BD-MTS.

Parameters	k_i	d_i	m_i	$f_{0,i}$	sw_{ij}
Model-1	449.1	-259.9	1.9	-33.5	init. state
Model-2	154.5	3528.3	17.8	-13.1	sw_{12} : Collision point $x = 12.4\text{cm}$, $v \approx 0 \text{ cm/s}$.
Model-3	31.3	251.5	1.2	32.3	sw_{23} : Drilling-A $x = 15.3\text{cm}$, $v \approx 1 \text{ cm/s}$, $f \approx 37\text{N}$
Model-4	72.1	108.1	0.5	26.1	sw_{34} : Drilling-B $x = 16.2\text{cm}$, $v \approx 1 \text{ cm/s}$, $f \approx 37 - 42\text{N}$

The Model Using SS-Imp Approach and Comparison

Taking the WKM results into consideration, the number of discrete states is accepted to be known as $s = 4$ for SS-Imp approach. Therefore, the HMM is initialized with 4 discrete states, and randomly chosen parameters. While the number-of-data points is quite high ($n = 14000$), the forward and backward variables have to be scaled as explained in section 3.3.4, in order to avoid numerical instabilities. As introduced in the same section, the HMM iterations can be carried on either depending on classical maximum likelihood principle, or minimum absolute model error principle introduced in this work.

The latter approach is implemented by setting the error threshold as $\varepsilon = 4 \text{ N}$. The iterations are continued until the condition given in equation (3.55) is satisfied. Only 7 iterations were needed to satisfy the condition, which was completed in around 10 seconds. The iterations return the switching positions and impedance parameters for 4 discrete states. However, the switching conditions are very noisy as given in Figure 3.11(a), which causes that they are not convenient, and even not actually realizable by the HMM. Thus, it is necessary to apply our intensity analysis to filter the switchings appropriately. Setting the data window (dw) width $h = 100$, 140 dws are generated. Figure 3.11(b) shows the intensity I of the clusters distributed to each dw. Choosing the cluster with maximum intensity as the active cluster, the switching conditions are filtered out, and results are depicted in Figure 3.11(c).

The classical maximum likelihood approach is also implemented, which needed more or less the same time to complete the iterations. Switching conditions are defined via logarithmic Viterbi algorithm and they are depicted in Figure 3.11(d). As one can see, the

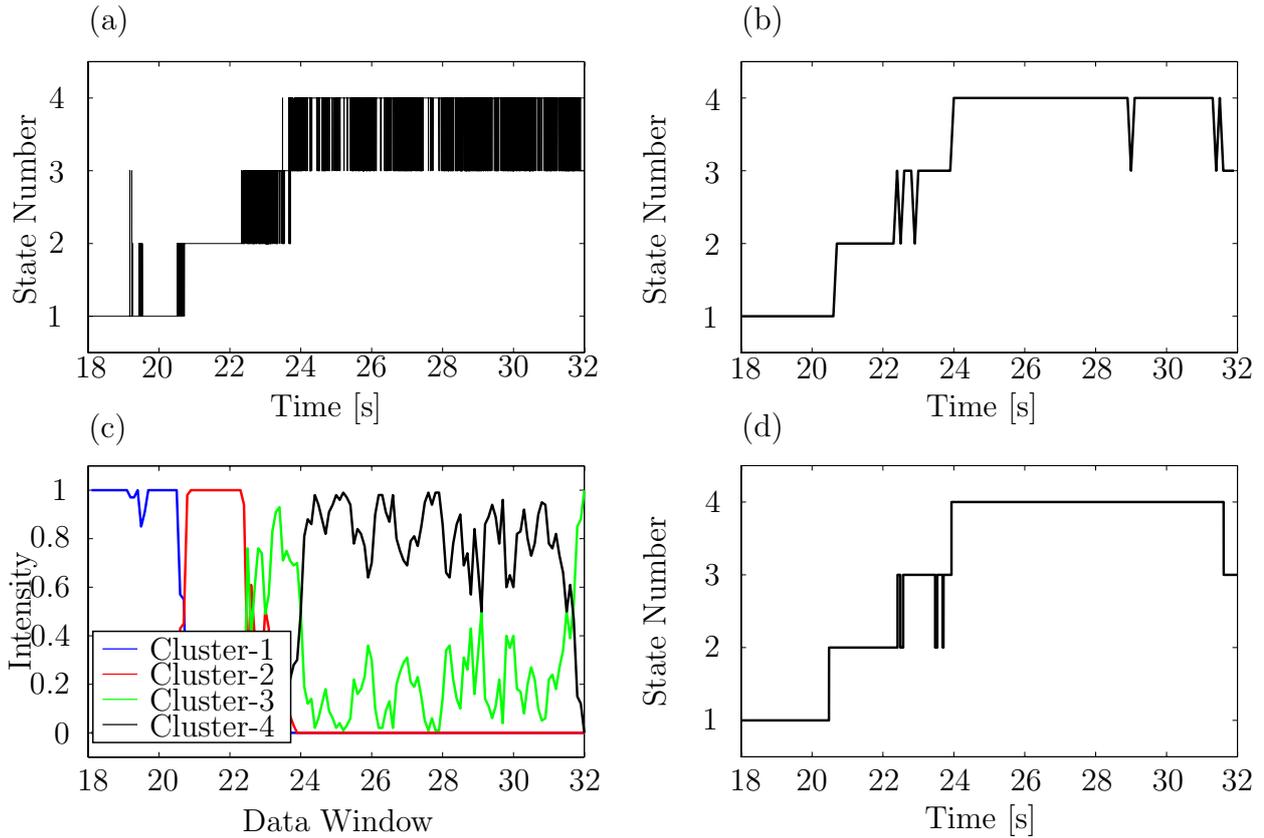


Figure 3.11: Switching condition obtained by using SS-Imp model error in BD-MTS. (a) switchings depending on absolute model error principle (b) switchings after intensity analysis (c) results of intensity analysis (d) switching following viterbi algorithm.

switching conditions obtained via both approaches are very close to each other. But, if the maximum likelihood criteria is used, the identified model parameters highly depend on the initial conditions. That is because the model's fidelity is judged only comparatively after each iteration step. However, model error based approach works until a satisfying model error is reached. Consequently, it is robust against initial conditions. Instead of defining the switchings on each data points, than applying intensity analysis, viterbi algorithm can be used to define the switching conditions as well. This guarantees that switchings are realizable by the HMM.

Identified model parameters are given in Table 3.5. Note that the same expert trial, whose results were discussed for WKM approach, is used to reach these results. The data points that are put to the same discrete states by HMM iterations are shown in Figure 3.12 for position, velocity, acceleration and force. Note that they do not exactly match the clustered data points in WKM, thus physically assumed states. Especially, from freespace to the approach phase the transition occurs later than expected, which results in difference in identified model parameters. Difference between measured force and model output force, i.e. model error, is given in Figure 3.13.

In the following chapters, results obtained from WKM clustering and LS optimization are used to represent the trainer skill. Physical interpretation possibility of the switching conditions, robustness against initial conditions of the hybrid identification process, and

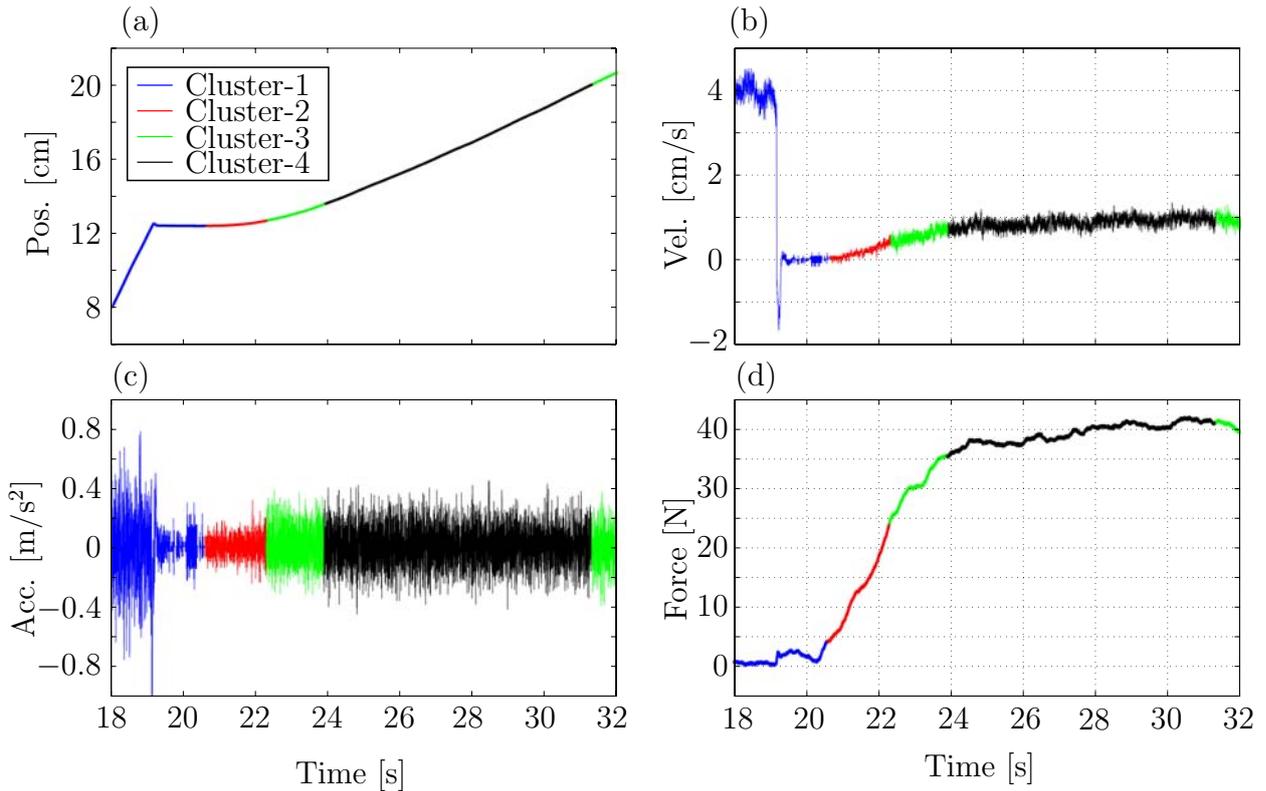


Figure 3.12: Identified clusters using SS-Imp algorithm. (a) Position, (b) velocity, (c) acceleration, (d) force.

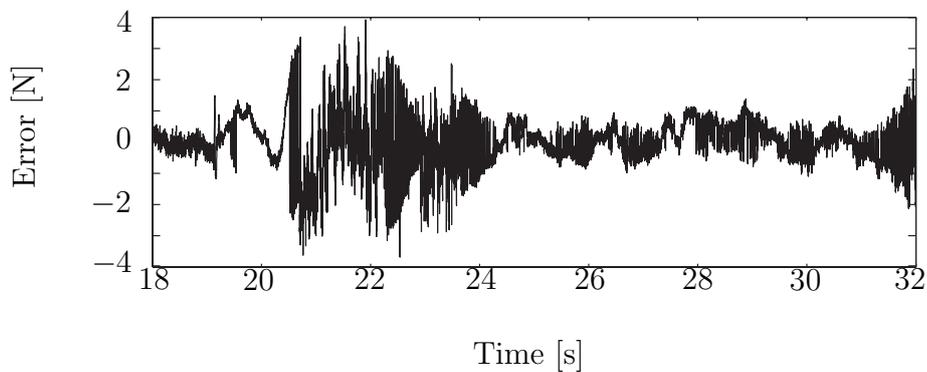


Figure 3.13: SS-Imp model error in BD-MTS

satisfying model accuracy have played important role for the choice. Computing time should be once more underlined as the biggest disadvantage of the method in comparison with SS-Imp algorithm. However, the intensity analysis introduced in this thesis is partly a remedy for this problem. It allows us to sort out the mixed data points in clusters. Thus, it is not necessary to apply trial-and-error approach many times to define the parameter c in WKM method, in order to find a suitable c that minimizes the mixed local data sets.

Table 3.5: Identified model parameters obtained by using SS-Imp with 4 states for BD-MTS.

Parameters	k_i	d_i	m_i	$f_{0,i}$
Model-1	2.3	-21.4	0.5	1.1
Model-2	992.8	3270.3	17.8	-116.7
Model-3	39.5	1749.9	8.8	16.2
Model-4	72.0	169.3	0.9	25.6

3.5.2 Hybrid Trainer for a Push-Button Training System

Hybrid Identification Problem

The push-button training system explained in section 2.5 is used as a second test-bed to verify the WKM and SS-Imp hybrid identification methods. HIDP for push-button is simplified by neglecting the effect of the acceleration and mass. The main reason for this simplification is to reduce the dimension of regression matrix, which comes down to fastening the identification processes. Since the acceleration signals are double derivatives of the measured time, they are noisy, although low-pass filtered. Additionally, the buttons are massless, so it is assumed that mass parameter does not play an important role. Therefore, the parameters vectors θ_i have three elements, i.e. $\theta_i = [d_i \ k_i \ f_{i,0}]^T$, and regression matrix is only built of position and velocity so that $\mathbf{X} = [\dot{\mathbf{x}} \ \mathbf{x} \ \mathbf{1}]$. While buttons have two layers, and before interacting with the buttons there is a freespace part, the number of states s is fixed to 3. In general, known and unknown elements of the identification process are given below:

Known: Number of sub-spaces (discrete states), model orders.

Unknown: Parameters θ of the models, switching conditions sw_{ij} .

Therefore the hybrid identification problem for push-button process is aimed at finding following parameters:

$$\text{HIDP}_{\text{PB}} : \begin{cases} \theta_i = [d_i \ k_i \ f_{i,0}]^T & i = 1, 2, 3 \\ sw_{ij} & i, j = 1, 2, 3 \end{cases} \quad (3.66)$$

The virtual push-button task is realized several times, position \mathbf{x} and force \mathbf{f} data are measured and saved. From the position data velocity $\dot{\mathbf{x}}$ is derived. The noisy velocity data is low pass filtered, the resulting velocity $\dot{\mathbf{x}}_{fil}$ is used in regressors \mathbf{X}_{id} . Thus, the regression matrix for identification is $\mathbf{X}_{id} = [\dot{\mathbf{x}}_{fil} \ \mathbf{x} \ \mathbf{1}]$ and the output vector is $\mathbf{y}_{id} = \mathbf{f}_m$.

The Model Using WKM-Based Approach

Applying equation (3.8) as the cost function to the regression matrix \mathbf{X}_{id} , the data is divided into 3 clusters. It should be kept in mind that the K-means clustering algorithms depend on the initial conditions, and they may end up in local minimums. Therefore, the clustering process has to be repeated several times, and it should be guaranteed that the minimum cost is reached several times. In our work, the clustering algorithm was applied 200 times, and the one that gives the minimum cost is chosen for further investigations.

It is assured that the minimum cost was reached several times as it is depicted in Figure 3.14. The number of data points in local data set, i.e. the parameter c , is set to 250.

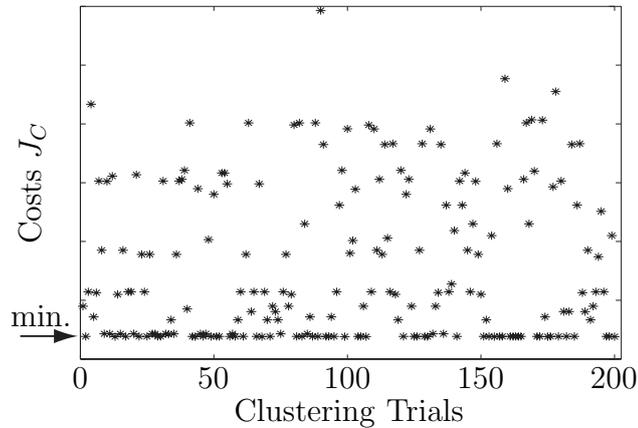


Figure 3.14: Costs of clustering attempts with different initial centers.

As one can imagine, there is no actual expert for a process like pushing buttons. For this reason, it is necessary to define an expert. This can be done by checking the timing of task completion. In other words, the choice of the expert can depend on how fast the pushing is. As a result, the trials that finish between 4.5 and 6 seconds are accepted as the expert performance.

Hence, individual expert trials are identified by WKM method. Identified model parameters for these trials are grouped under the phases “freespace”, “Layer-1” and “Layer-2”, and are shown in Figure 3.15. Note that these phases represent physical states. It is very clear to deduce characteristic features for individual phases from Figure 3.15. Freespace spring coefficient k_{fs} is in the range $[-45, -32]$ N/m, whereas damping coefficient d_{fs} range is $[21, 27]$ Ns/m. Layer-1 parameters vary in a very small interval as well. Namely, k_{l1} is located in $[491, 514]$ N/m and d_{l1} in $[42, 85]$ Ns/m. Similarly, Layer-2 consists of certain impedance model parameters that characterize the interaction in this phase. The narrow band for k_{l2} is $[1012, 1023]$ and for d_{l2} is $[83, 131]$. Offset values increase when it is switched from one state to another. Thus, they have the lowest values in freespace phase and largest values in Layer-2 phase. This is expectable, because when a new state starts, it has implicitly the force offset in itself from the previous state.

If one compares the Figures 3.8 and 3.15, it is understood that in both cases, the hybrid identification via WKM method affirms characteristic impedance features. However, the results for push-button example are more precise than BD-MTS example. The identified parameters for different expert trials are closer to each other and they are non-ambiguous. Most important reason for that lies in the nature of the task. Virtual push-button process is defined via linear spring and damper elements. Thus, the success of modeling the task as switching linear impedance models is an anticipated result. The comparison of the figures yield another interesting result. The sign of the identified freespace spring parameters is always opposite to the sign of damping parameters. While moving in freespace, ideally, it is expected that no resistance force is experienced. In our opinion, the opposite sign is an outcome of this expectation. The spring and damper elements are working against each other to neutralize the force information.

Since all expert models have very similar switching conditions and impedance model parameters, any of them can be chosen as the trainer model to be used for further investi-

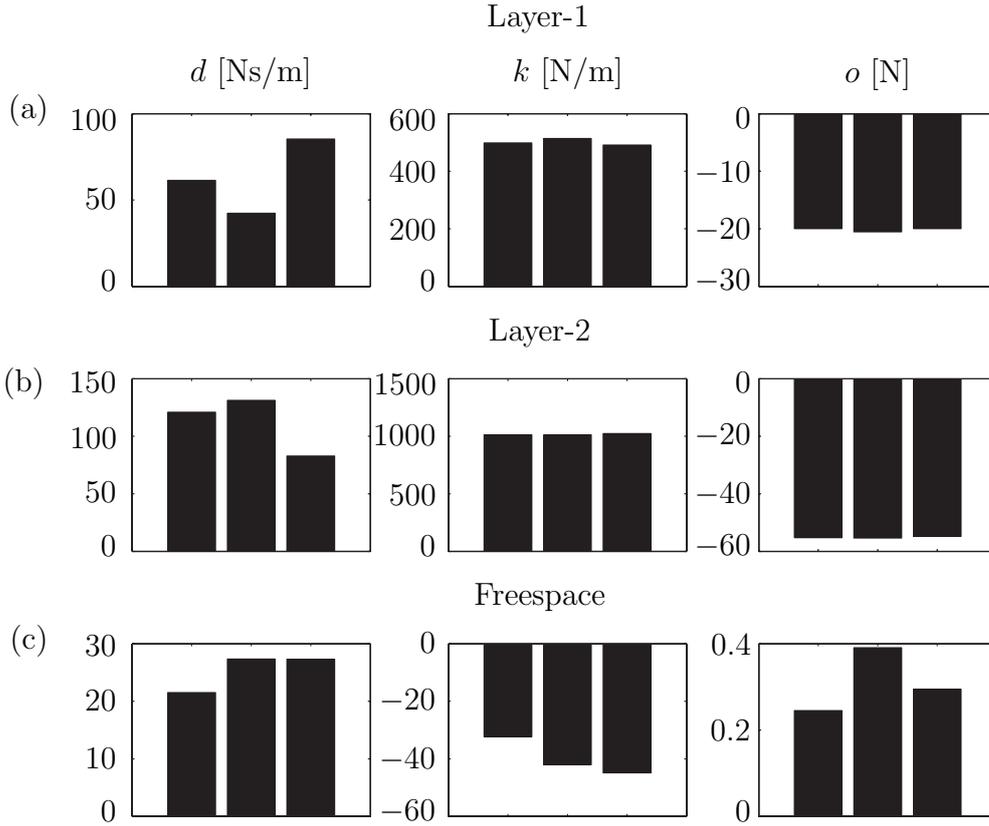


Figure 3.15: Different expert trials and resulting mass damping d , spring k coefficients and offset o for freespace, layer-1 and layer-2 phases.

gations. In Table 3.6., the parameters and the switching conditions for the chosen trainer are given.

Table 3.6: Identified models and switching conditions using weighted K-means clustering and LS optimization.

Parameters	k_i	d_i	$f_{0,i}$	sw_{ij}
Model-1	-32	22	0	Init. model
Model-2	498	61	-20	$sw_{12}, x = 0.037\text{m}$
Model-3	1012	121	-55	$sw_{23}, x = 0.070\text{m}$

Identified data clusters belonging to the chosen trainer are shown in Figure 3.16. The switching between the states most clearly occurs depending on the position signal. On certain positions switching happens, which is expected since the collision with the virtual button occurs on a certain position, and the button’s two layers are separated from each other on a certain position as well. The detected switching conditions match these designed VR positions. Appropriately, measured force signal also indicates clear switching points. A closer look into the velocity signal clarifies that the movements in the first

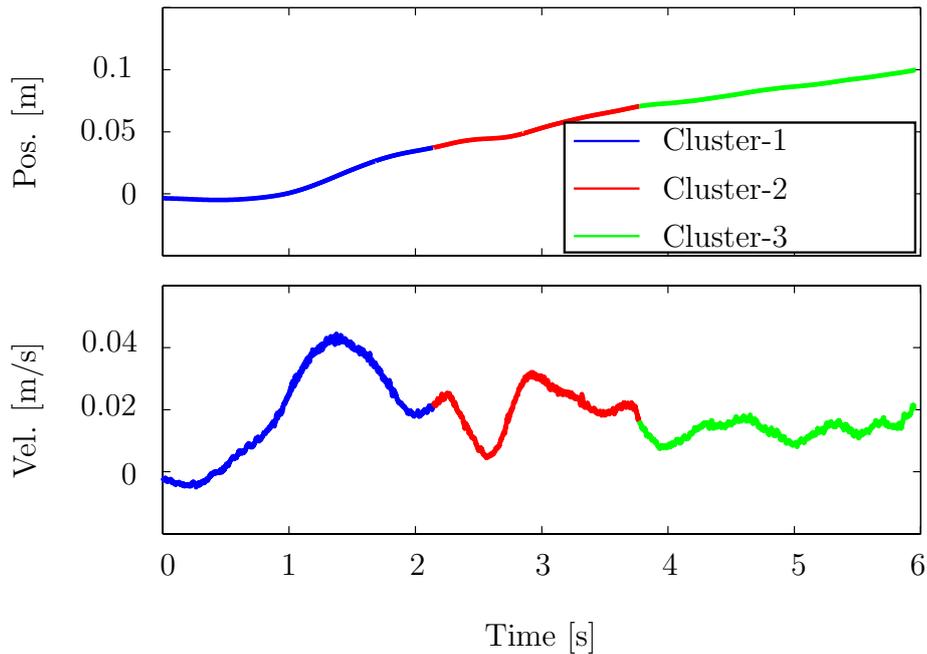


Figure 3.16: Identified clusters using WKM algorithm for push-button system. (a) Position, (b) velocity.

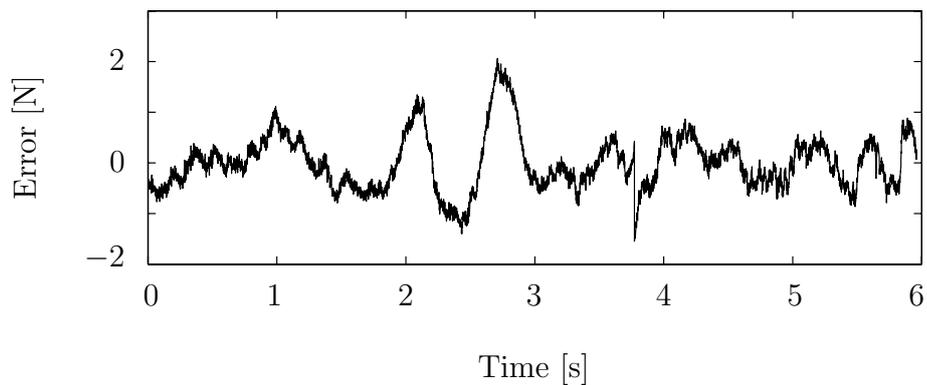


Figure 3.17: WKM model error in push-button system.

cluster related to freespace are fastest, whereas the third cluster related to the button layer with large spring and damping constant slowest. The model error is shown in Figure 3.17.

The Model Using SS-Imp Approach and Comparison

The SS-Imp method is applied to the virtual push-button task. The HMM model is initialized with 3 states by randomly chosen parameters. 4 iterations were necessary for algorithm to converge to the pre-defined absolute error limit $\varepsilon_{ERR} = \pm 3$. The resulting clusters on position and switching conditions are shown in Figure 3.18. The model parameters calculated by using equation (3.33) are given in Table 3.7.

The identified model by using SS-Imp method follows the measured output in an error range of $\pm 2.5\text{N}$ as it is shown in Figure 3.19.

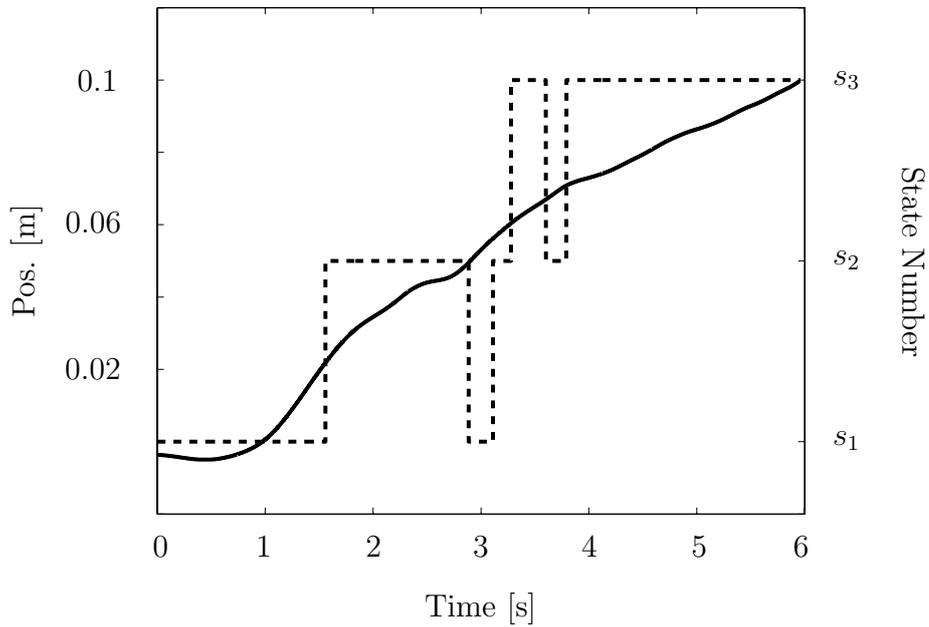


Figure 3.18: Switching on position signal obtained by SS-Imp method for push-button example

Table 3.7: Identified models using SS-Imp

Parameters	k_i	d_i	$f_{0,i}$
Model-1	189	-100	2
Model-2	406	190	-18
Model-3	979	203	-53

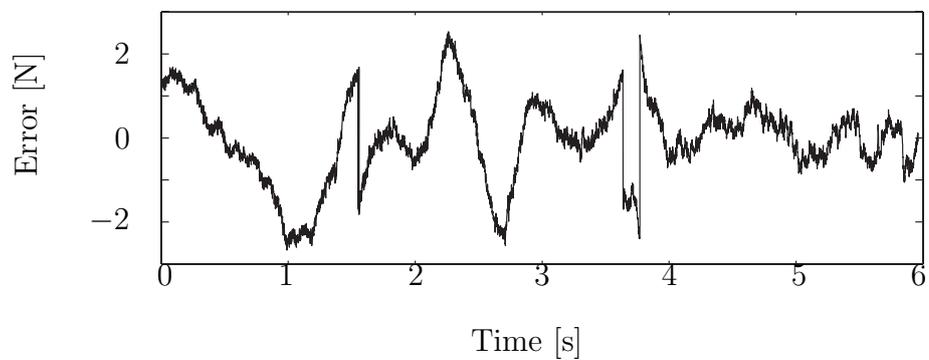


Figure 3.19: SS-Imp model error in push-button example

The problem with SS-Imp is that it is difficult to give physical meaning to the switching conditions. Therefore, it is not possible to use the same identified model for button-2 and button-3. On the contrary, WKM clearly defines 3 states that can be physically interpreted as 'Freespace', 'Button Layer-1' and 'Button Layer -2'. Thus, it was possible to use the same model to represent other virtual buttons which have the same spring and damper coefficients, but in different orders. A big disadvantage of the WKM method is the computational time. Especially, the clustering time increases with the increasing number of data. If the data increase rate is g , the time would increase with g^2 . In the current push-button example, around 2 hours were necessary to create local data sets and clusters, where the number of data was ca. 6000. On the other hand, SS-Imp is a very fast method. For the same problem, around 1 minute was needed to obtain switching conditions and model parameters.

Both methods are sensitive to the initial conditions. But the cost function used in WKM is a good measure to decide on the final model. Repeated clustering approaches should indicate the global minimum which gives the minimum value for the cost function, and is reached several times. Thus, the identification process can be started from any randomly chosen initial conditions, whereas for SS-Imp the initial condition needs to be defined with a pre-knowledge about the system. However, using error based switching method and then applying intensity analysis enables one to chose the initial conditions randomly and eliminate unnecessary switchings. Nevertheless, still the switching conditions obtained via WKM are simpler and physically clearer. If the physics of the interacted system is known beforehand, it would help to have meaningful initial estimates. But this causes the used method to be heuristic.

Taking into account these benefits and drawbacks, the WKM method result is chosen for "Push-Button Hybrid Trainer" model and it is used for the further investigations given in this work.

3.6 Discussion

This chapter starts with an overview of the state-of-the-art methods to capture human skill. As a follow-up, the representation of the skill as a hybrid dynamic system based on hybrid identification techniques has been investigated. On one hand, a weighted K-means (WKM) clustering has been implemented in parameter space, followed by LS optimization. On the other hand, parameters of switching dynamic models and switching conditions are defined stochastically, based on HMMs (SS-Imp).

Conventional K-means methods check the Euclidean distance between data points and they are inadequate to cluster the human demonstration data appropriately, because of being very sensitive to noise, outliers and initial conditions. As it was demonstrated, even in a small data space that contains data points from different first order ARX models, K-means clustering is unsuccessful. To overcome such problems, the distance check for clustering has to be performed after scaling and rotating the data axes in the data space. Thus, a Mahalanobis norm, which weights the data regression matrix with its covariance matrix, is used. A data point j and its $c - 1$ neighboring data points have been assigned to a local data set, and a local parameter vector has been identified for each set using LS optimization to begin with. Following that, the clustering analysis is moved from data space to parameter space. It is a natural consequence that the parameters of the

local data sets that belong to the same mode are closer to each other. Thus, the data points in parameter space are better separated from each other which provides a more suitable basis for clustering. Together with the parameter vectors, information about the spatial position of the parameters is included to form the regression matrix. In order to perform WKM clustering, s has to be known beforehand. We have introduced an error analysis, which enables to define s in a systematic way. The analysis is based on repeating the clustering and parameter identification each time for different s . Following that, the absolute model errors are compared with each other. Defining maximum allowed model error as a threshold allows for determining the minimum s to reach to the threshold. Note that increasing the number of the states, decreases the model error. The error analysis is followed by an intensity analysis. It is expected that the data points that are close to each other on time axis belong to the same cluster. The intensity analysis generates data windows on time axis and finds out the most intensive cluster, i.e. the cluster containing highest amount of data points, in each window individually. Then, all data points in each window are assigned to the most intensive cluster of the window. As a result, convenient clustering of the demonstration data and physical interpretation of the switching conditions between clusters were possible. However, the WKM in parameter space yield two important disadvantages. Firstly, the number of data points c in one local data set is defined in a trial-and-error manner. Higher number of c enables to identify a parameter vector that represents the mode it belongs to properly. However, at the same time, it increases the risk of having mixed data points, i.e. points from different modes, in one local data set. For each new c to try, the whole clustering process has to be re-started. Second hindrance is the computing time. Creating local data sets is very time consuming and related to the number of data points geometrically. If the number of data points increases with a rate g , the time needed for creating local data sets increases with g^2 . The intensity analysis introduced in this chapter is partly a remedy for this problem. It allows to sort out the mixed data points.

Hidden Markov modeling is a stochastic method, matching the input and output signals of a system probabilistically, and leaving the system itself as a black box. However, assigning a deterministic function to each state of an HMM and defining the observation probability distribution \mathbf{b} as a product of the function parameters, provides with an insight look to the system as well. To avoid numerical instabilities, a scaling of HMM forward and backward algorithms is necessary. Note that the higher the number of data points is, the more possible it is to face numerical instabilities. We have introduced an algorithm to define the switching conditions between the discrete states depending on the minimum model error. First, the HMM iterations are carried on, until at least one of the states satisfies the pre-defined, maximum allowable model error for all data points. Then, at time point k , it is switched to the state that yield the minimum error. Following that, in order to filter out unnecessary switchings, the intensity analysis is applied. Alternatively, it is possible to apply logarithmic Viterbi algorithm to define the optimal state sequence after the iterations are completed. It is achieved to converge to the model parameters representing each discrete state with randomly chosen initial conditions. However, it is not possible to reach to the desired model error for any initial setting. Therefore, it might be necessary to repeat the SS-Imp several times, each time with different initial centers. Additionally, a physical interpretation of the switching conditions is not always possible.

Representing human skill as switching impedances is appropriate for the tasks involving human hand (with or without a tool) and object interaction, during which human needs

to vary the force. Tasks like in our application examples: bone drilling and push-button, or others like: peg-in-hole, hose insertion or grinding, have in their nature the potential to relate the force to position and position derivatives. Tasks introducing highly non-linear interactions, as in the interaction with a human organ during a medical operation, are not suitable to be approximated by switching impedances, since for a good approximation the number of the discrete states needs to be very high and switching between them would be stochastic. This would restrain the ability of interpreting the physical meaning of the states. Additionally, each interaction would lead to another model. Thus, neither would it be possible to label any of the models as the model of the expert, nor to use this models to train a novice. Freespace movements of arm, like carrying a glass of water from one point to another, are more likely to be represented as a jerk optimization problem, following the theory of Hogan [39].

4 Training Methods to Transfer Human Skill via Virtual Environments

In the last decade, there have been numerous research works and developments in the field of multi-modal virtual reality (VR) systems that provide human with a better immersion to virtual environments (VEs). Consequently, the number of realistic virtual simulators that aim at training novices has been increasing rapidly. Nevertheless, less attention has been paid to the training strategies and to the function of a trainer in training processes.

Many procedures such as cutting, suturing, drilling in medical operations incorporate following a certain position trajectory by applying a certain force. Mainly, the training strategies given in the literature neglect the force training and concentrate on only position trajectory teaching [38, 80, 132]. One of the main reasons for this situation is the known fact that it is not possible to teach both force and position synchronously. In order to demonstrate the force to the student, the simulation has to be paused. This idea was proposed by Yokokohji for single user record and replay based training structure [139]. In our work, this concept has been extended to multi-user cases, in which the expert has a chance to show the student the correct force to apply. This is achieved by not allowing the student to interact with the VE himself/herself, but just playing the interaction forces of the trainer on him/her. The method is called Force Demonstrating (FD) and explained in detail in Section 4.2.3. The same principle is also implemented by replacing the expert with a hybrid model of the expert. This situation is discussed in Section 4.3.2.

However, this method bears an important disadvantage since it is unnatural to stop simulation during a training session. Thus, haptic guidance based methods offer an interesting alternative. As Feygin *et al.* describe, haptic guidance refers to guiding a student on a trajectory via a haptic display [38]. One disadvantage of this method is that the student is passive, i.e. the task is not realized by him/her. Additionally, at the end, not the force to apply, but the trajectory is learned by the student. Thus, we introduce a method, in which the student realizes the task himself/herself, and is observed by a trainer in a multi-user scheme. If the student deviates from the expected force, the expert warns verbally or corrects haptically. This unique approach allows us to train the force without stopping the simulation. A similar approach was presented by Morris *et al.*, by binding the haptic interface points of the expert and the student via a soft spring in a collaborative VE [85]. However, neither details on how to choose the spring coefficient were given, nor was the efficiency of the training discussed. Their method includes a remote-trainer, i.e. it is a multi-user training strategy. Srimathveeravalli *et al.* define “haptic attributes” that relate a unique haptic profile to tasks performed by using motor skills [115]. Then, while the student is following a trajectory, his/her haptic profile is compared with the expert’s and, depending on the difference, the student’s force is either augmented or decreased. However, the method does not concern the relation between the visual and haptic feedback, i.e. even when the trainee follows a wrong trajectory, he/she keeps on feeling the force that expert would feel. Thus, the authors assume that expert and student have similar haptic profiles, consequently trainee naturally follows a path closer to the expert’s. In addition to that,

their application concerns calligraphy, in which the force to apply is almost orthogonal to the position trajectory. The authors do not discuss, whether it is possible to apply this method to other example where a change in position along the force direction occurs. They did not find any important benefit of using haptic for force skill training. Last but not least, all works of haptic guidance in literature are either in multi-user schema, or related to record and replay fashion. None concerns using a trainer skill model. Therefore, we discuss teaching the force in our Haptic/Visual Observation (HVO) and Force/Velocity Leading (FVL) methods, where the former one can be considered as “Verbal Tutoring” and the latter one as “Haptic Tutoring”. Both methods are implemented in multi-user cases, as well as single-user cases using a hybrid dynamic model of the expert. It should be underscored that the expert or the expert model intervenes the trainee’s performance, only if the student deviates from the expected force. The correction signal is deactivated as soon as the student corrects himself/herself. Please refer to Section 1.2.2 for a more detailed insight into the state of the art in VR training systems.

Since the emphasis is on the force skill training, the trajectory to follow is assumed to be correct or visually available to the user. Thus, we concentrate on training to apply the correct force and realize tasks with correct velocity and timing along these trajectories. In Section 4.4, we discuss the reason of these constraints and the possibility of using the training strategies for scenarios with a broader scope.

This chapter is a classified collection of the training strategies, developed for VR training systems. The classification of the strategies is depicted in Figure 4.1.

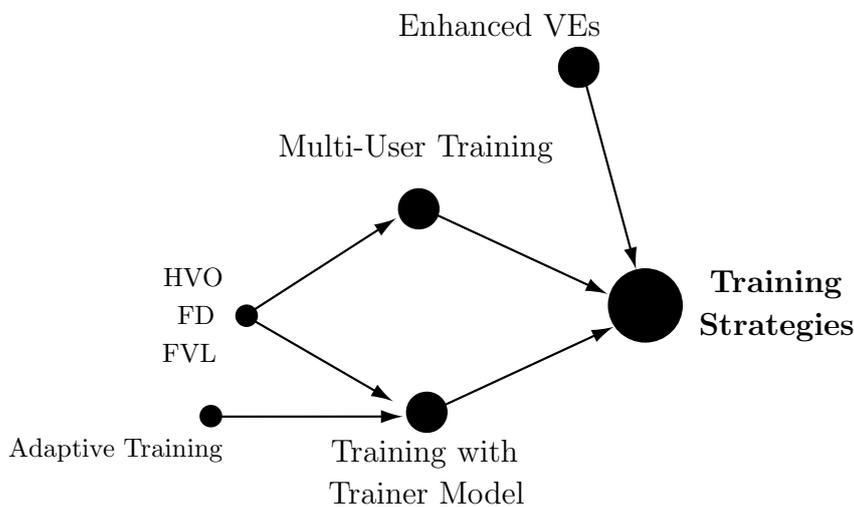


Figure 4.1: Main training strategies applied in this work.

Sections 4.1-4.3 comprehensively explain the developed enhanced, multi-user and with a trainer model training strategies. Simulation and real-time (RT) results of training via a hybrid dynamic model strategies are presented in Section 4.3. Consecutively, the adaptiveness of the training strategies with respect to the student’s psychology is questioned in Section 4.3.5. The chapter ends with a discussion.

4.1 Enhanced Training Methods

As underlined before, the main difference between a simulator and a training system (TS) is that a TS can be enhanced with extra features which do not exist in the real process, whereas a simulator tries to replicate the process as realistically as possible. Such extra features can be in a signal enhancement form that inform the trainee of his/her performance, and lead him/her by stimulating an appropriate modality of the human sensory system.

4.1.1 Visual and Acoustic Enhancement

Conventionally, the stimulation is carried out in a visual form. For tasks, such as trajectory following, it is common to show the path to be followed to the trainee. Virtual fixtures are also useful tools that mark the path to follow and change colors depending on the following performance.

Similar structures can be used to train the student about the force. If the haptic display includes a force sensor, it is rather straight-forward to implement visual indicators about the applied force. Such indicators can be exact numerical displays, or color maps that relate the level of the force.

Assume that there is a path to follow, and during this task the applied force \mathbf{f} should be in a certain range, i.e. $\mathbf{f}_{min} \leq \mathbf{f} \leq \mathbf{f}_{max}$. In this case, the student can be informed by visual force indicators that show whether the applied force is correct, too large or too small, as in Figure 4.2. It is possible to include other indicators for velocity, acceleration etc.

The biggest advantage of visual enhancement is the clarity of the information. Paying attention only to the visual information, the trainee realizes the task very successfully. However, in many cases, such as in surgical operations, there is no visual information about the force available. The operations are realized entirely by relying on the haptic sense. As Ernst *et al.* underscore, haptic cues play more important role in motor skill learning procedures, if the visual information is less reliable to realize a task [29]. Thus, for such cases, a visually enhanced training is degrading the quality and reliability of training.

A similar statement can be made for acoustically enhanced training systems. Instead of visual cues, it is possible to create acoustic cues, which inform the student about the applied force. If you consider the example given above, three different acoustic signal, e.g. one low, another one middle, and the last one having high frequency, can be generated to inform the student, whether the applied force is in the pre-defined range or not.

Thus, for an inadequate training, the signal enhancement of the VR training system should have the same modality of the signal to learn. Consequently, to learn force, we introduce a haptic enhancement technique.

4.1.2 Haptic Enhancement

Following the statement of Ernst *et al.* mentioned above, it can be concluded that to teach a task, which is realized by relying on haptic cues, it is more convenient to provide a haptic enhancement. Although there are several training systems providing with visual

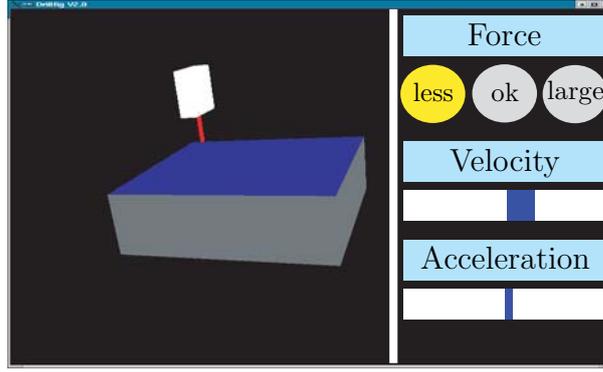


Figure 4.2: A training system for drilling with visual enhancement

enhancement, as a novel approach, haptic enhancement is added to our training systems. It is felt by the user in the form of an additional force depending on his/her performance.

The haptic enhancement we propose has the following structure:

$$\mathbf{v}_{ref} = \begin{cases} \mathbf{v} = f(\mathbf{f}) & \text{if } \mathbf{f}_{min} \leq \mathbf{f} \leq \mathbf{f}_{max} \\ \mathbf{0} & \text{otherwise.} \end{cases} \quad (4.1)$$

The equation above states that the trainee cannot move his/her haptic display, if the applied force is less than \mathbf{f}_{min} or more than \mathbf{f}_{max} , which are pre-defined minimum and maximum forces respectively. This is achieved simply by setting the reference velocity \mathbf{v}_{ref} to $\mathbf{0}$, which enables to fix the reference position to a certain point for the position controller of the haptic display.

A similar idea was presented by Prada *et al.*, who suggest using reactive virtual fixtures with haptic cues together with haptic guidance. However, they inform the student haptically if a trajectory to follow is left by constraining the movements. In other words, not the force, but the position is aimed to be taught to the student.

The strong characteristic of our haptic enhancement is that it is applicable to problems, where there is a relation between force and velocity along the direction of the interaction. Note that it can also be applied if there is a force profile to follow independent of the velocity. Then, instead of defining the reference velocity as a function of the force $\mathbf{v} = f(\mathbf{f})$, it can be set to a constant value $\mathbf{v} = \mathbf{v}'$, and otherwise to zero. A disadvantage of this method is that the student does not know if he/she applies too small or too large force. The only haptic information is that the force is wrong. However, changing the force and finding the point where the haptic display moves, is observed to be efficient in teaching the force. Results of some user tests are discussed in Chapter 5.

4.2 Multi-User Training Methods

Collaborative VEs allow two or more users to realize a task in a VE together. When haptics included, such a collaborative work can be considered as haptic communication of users via a common virtual object. Some example scenarios developed in our laboratories to demonstrate human-virtual object-human haptic communication, such as “tug-of-war” and “load transportation”, are summarized in Appendix B. Below, the usage of collaborative VEs for training is discussed.

The potential of using VEs for training has been long accepted and numerous simulators and training systems have been developed by many researchers. Nevertheless, no strong attention has been paid to haptic influencing of a student by a trainer in a multi-user scheme. Thus, we set our goal as “defining training strategies that enable a trainer to observe the student in all possible sense modalities, and to influence him/her verbally and/or haptically”. The replay methods developed by Yokokohji *et al.* [139] are taken as a base, and they are extended for multi-user training cases in our work. Main difficulty in such methods is that it is not possible to “replay” on the trainee both position/velocity and force/torque information gathered from an expert synchronously. A compromise has to be found between them for an efficient training. Kikuuwe *et al.* tried to solve this problem by developing a new hardware [60]. But the results are not very satisfactory, because the new hardware can give effective results for basic experiments, and reveals some limitations. In Chapter 1, Introduction, a broader overview of the works concerning multi-user training in VEs [28, 44, 77, 86, 89, 103, 130] was given. To briefly repeat the main deficiencies of these works, generally just a conceptual overview is given and/or haptic guidance based techniques were used in which the student is passive and not the force but position is aimed to be trained. Except these few works concerning haptic guidance in a multi-user scheme, there is no classical or modern training system that enables an expert to observe a trainee and intervene in his/her performance haptically. Therefore, it is undoubtedly helpful and necessary to extend currently available virtual training systems in this direction.

4.2.1 Overview

Keeping the difficulties and necessities given above in mind, three main multi-user training methods are developed in this work to teach : Haptic and Visual Observation (HVO), Force Demonstrating (FD) and Force/Velocity Leading (FVL) methods. In all of these methods, there is one common virtual scene that is available both for the expert and for the trainee. They both have a haptic display to interact with the virtual scene, and/or with each other haptically. The first two are associated with Yokokohji’s stop-simulation method, whereas the latter one is related to haptic guidance based methods. The general concept of the multi-user VR training environment is given in Figure 4.3.

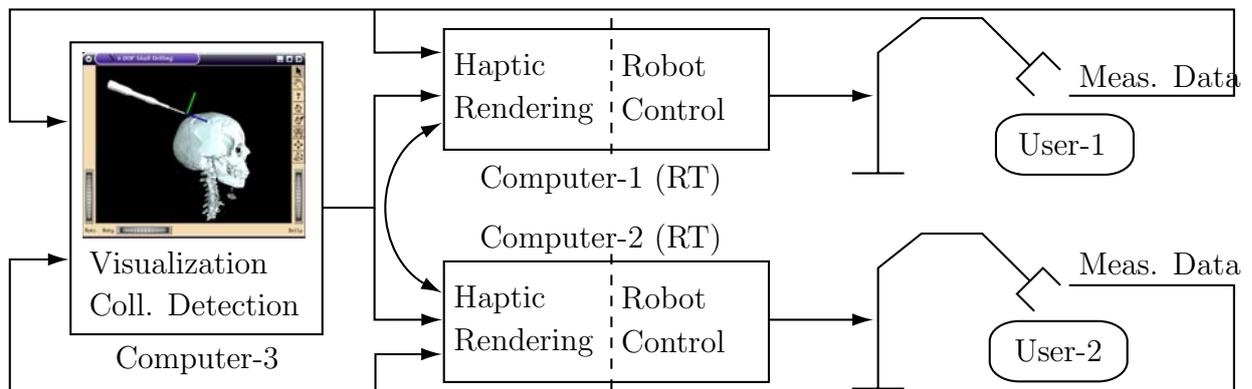


Figure 4.3: Training in a multi-user manner: User-1 represents the student, whereas User-2 represents the expert. During the training sessions, this structure is extended by one more computer which displays the student’s performance on-line for trainer.

4.2.2 Haptic-Visual Observation (HVO), Verbal Influencing

In one of the few works concerning force skill training, Yokokohji underlines that stopping the simulation and replaying on the student previously recorded expert force data can be treated as a method to teach the force. However, the unnatural stop / start structure is degrading the efficiency of the training.

We propose a multi-user approach that allows the trainer to follow on-line the interaction of the student via his/her haptic display. The implementation of the method is quite straightforward: The student interacts with the VE in a conventional way. The visual scene is also available to the remote trainer for visual observations. Additionally, the forces that the student applies are sent to the haptic device of the trainer. Keeping his/her haptic display tight, the trainer has a chance to observe the student also haptically. The expert cannot intervene in the performance of the trainee haptically, but he/she has a chance to do that verbally. In Figure 4.4 the implemented control scheme is shown, when the VR is admittance type. Note that the method can be applied for an impedance type VR as well. The training structure can be outlined as below:

- The student performs a task. The force he/she applies to realize the task is \mathbf{f}_s . The dynamics of the VE is represented by the function $f(\cdot)$.
- Synchronously, the student force is sent to the expert's haptic display. Thus, the expert feels the force $\mathbf{f}_e = \mathbf{f}_s$, as long as he/she does not move his/her haptic display, i.e. keeps the position \mathbf{x}_e and orientation \mathbf{o}_e of the haptic display constant.
- Feeling the force that student applies and observing him/her visually from the virtual scene, the trainer can verbally warn the student to change the force appropriately, or the trainer can confirm that the force is correct.

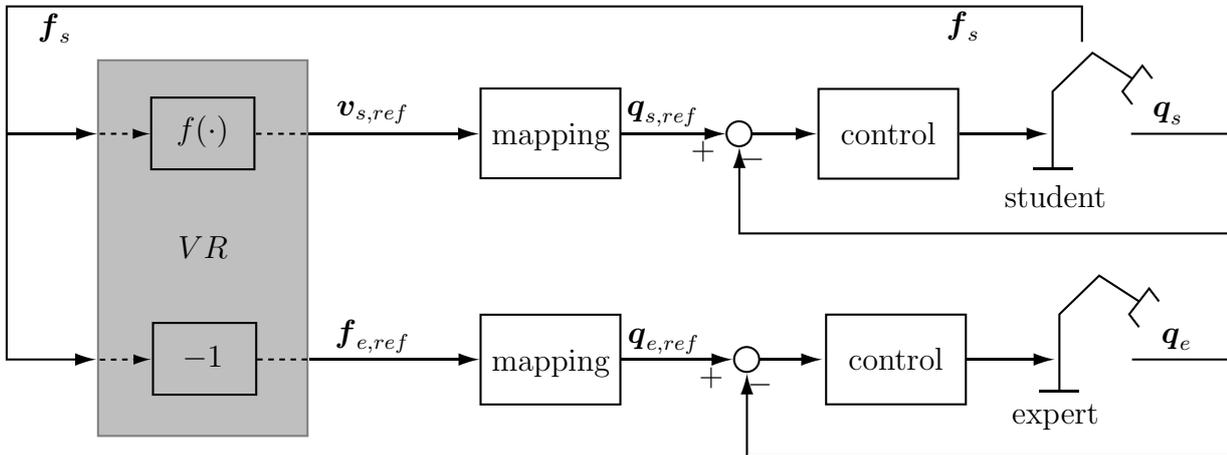


Figure 4.4: HVO method control scheme.

This training structure does not bear the disadvantages of the conventional stop simulation method, since the student continues to realize the task without a break. The verbal warning about the force from the trainer is very clear, and student is very quick to react and correct himself / herself. The efficiency of the method is confirmed by user tests, which is discussed in Chapter 5.

4.2.3 Force Demonstrating (FD) Method

FD-Version 1

As an extension to the HVO method, the FD method aims to give a chance to the expert to show (demonstrate) the applicable force to the trainee. In fact, FD is the inverted implementation of the HVO strategy: The expert performs the virtual task, and the student follows him/her visually and haptically. If one keeps in mind one of the main deficiencies in classical surgery training, that is missing haptic feeling when observing an expert surgeon in an OP room, the developed strategy here is very promising. Since this implementation of FD is very similar to the HVO method, one can refer to Figure 4.4 and the steps given related to this figure to understand how the method functions. The only necessary additional interpretation is to replace the student with expert and vice versa.

FD-Version 2

In this version, the aim is to demonstrate to the student force variations around a reference force. Apart from HVO and FD Version-1, both student and expert needs to constraint the movement of their haptic display along and/or about the axes, which are actively used to apply forces and torques to realize the tasks. The main restriction of the method is that it can only be used maximum 3 dimensional. For example only for the forces along x, y, z axes, or only for the torques along these axes.

To clarify this version of FD implementation, consider the training structure given in Figure 4.5 for 1-D interaction. The trainee needs to balance the reference (initial) force f_0 pressing against him/her, so that his/her haptic display does not move. The expert follows the force applied by the student haptically by restricting the movement of his/her haptic display along the direction of this force. The expert is able to move his/her haptic display in a direction that is perpendicular to the direction of the force pushing against him/her. Theoretically, visual feedback is not necessary for this method, since neither the student nor the trainer performs the task. However, a synchronized visual feedback indicating in which phase of the virtual task the trainee should apply the force he/she currently feels, assists the student in understanding the correlation between visual and haptic feedback.

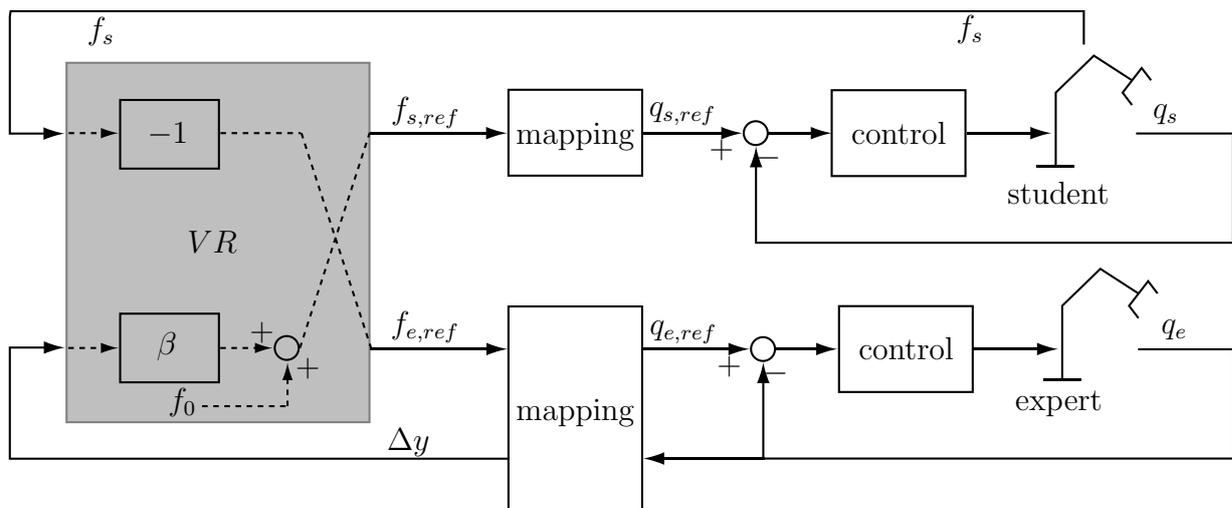


Figure 4.5: FD method control scheme.

The method is summarized below for 1-D interaction case given in Figure 4.5.

- A reference force f_0 presses against the student in x direction, i.e. $f_{s,ref} = f_0$. The student should balance this force by applying $f_s = f_0$. In this case the student's haptic display does not move.
- The expert feels the force f_s pushing against him/her so that $f_e = f_s$.
- The expert can change the reference force f_0 by moving his/her haptic display in a direction, e.g. y direction, that is perpendicular to the direction of the force f_e . The translation Δy of the expert's haptic display in y direction is then multiplied with a scaling factor β .
- New reference force $f_{s,ref} = f_0 + \beta \Delta y$ is felt by the student in x direction. The student should balance this new reference force by applying $f_s = f_{s,ref}$. The student's haptic display does not move.

Main deficiency of the FD methods is student's being passive. The student cannot realize the task himself/herself, just balances the force pushing against him/her. Thus, a method that corrects the student haptically while he/she is actively realizing the task is investigated and discussed below.

4.2.4 Force/Velocity Leading (FVL) Methods

The main impediment of the previously explained methods is that HVO only allows a haptic observation, but not a haptic tutoring. On the other hand, FD methods require the simulation to be stopped for the student, thus he/she is passive during a training session. Therefore, another method is desired, which enables the trainer to observe and influence the student haptically and on-line, i.e. without stopping the simulation.

Our FVL methods are developed to indulge this requirement. The methods basically aim at force training via influencing either force or velocity reference signals for the student. It is necessary to keep in mind that it is not possible to affect force and velocity at the same time. The main principle is actually a combination of HVO and FD Version-2, and relates to the haptic guidance methods given in the literature. However, note that in haptic guidance methods, the student is passive and position is trained rather than force. Thus, our FVL method is one of the few works which aims at teaching force, without stopping the simulation and keeping the student active. The other related works, such as Haptic Tutoring [85] or Haptic Profile [115] have been discussed previously in Section 1.2.2 in detail together with their deficiencies.

If the task in focus suggests that the applied force is related to the velocity, then it is convenient to develop an admittance type VE. Different tasks, such as peg-in-hole, push a box in a labyrinth, medical drilling, cutting operations can be interpreted in this way. In this case, it is more suitable that FVL method enables expert to influence the trainee's performance by changing his/her velocity. The trainer observes the student haptically as in the FD Version-2. Consequently, the observation is limited to 3-DoF, e.g. along x, y, z axes. The rest 3-DoF, e.g. about x, y, z axes is used to send to the student the correction signals. The trainer intervenes in the performance of the student haptically, only if the force that the student applies along the current trajectory is wrong. The haptic

intervention is in the form of changing the velocity of the student that results in change in the resistance that student feels.

For 1-D interaction, the implementation of the method is sketched in Figure 4.6 and the steps to follow are given below:

- The student applies the force f_s , and moves with the velocity $v_{s,ref} = f(f_s)$ in the direction of the force. The expert feels f_s pushing against him, i.e. $f_e = f_s$.
- The expert changes the velocity by moving his/her haptic display along the y direction. The change Δy in the y position of the haptic display is scaled with the factor γ to relate the change in position to the change in velocity v_c , so that $v_c = \gamma\Delta y$.
- The velocity $v_{s,ref} = v_c + f(f_s)$ is applied to the student. The student has to adapt his/her force, until he/she finds back the correct velocity and force.

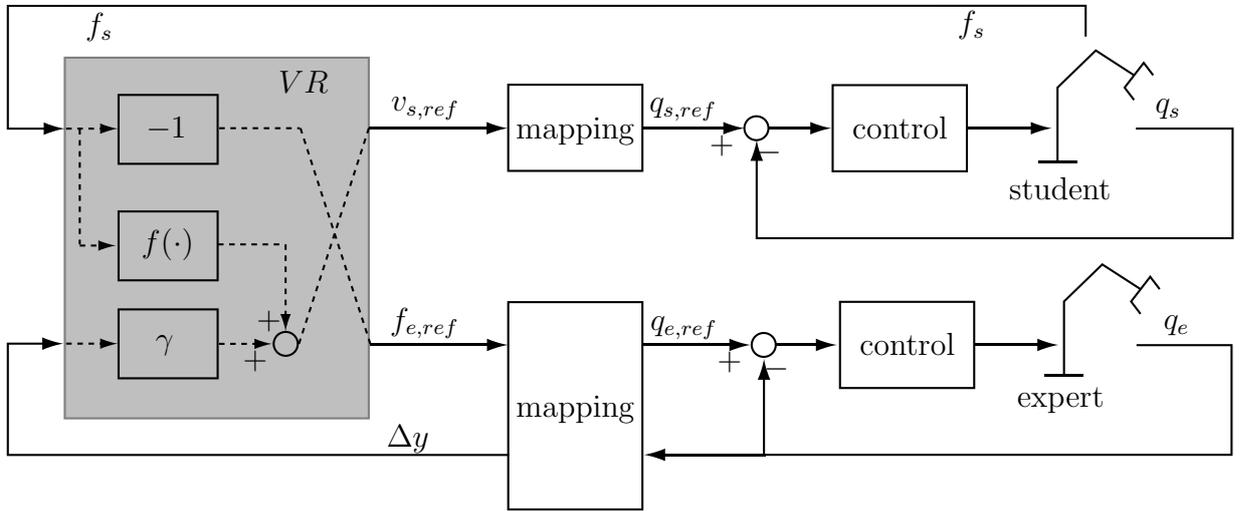


Figure 4.6: FVL method control scheme

It has to be noted here that it is possible to derive several variations of this method depending on the choice of factor γ . For example, if the γ is positive, the trainer lets the student's haptic display move faster, i.e. pulls the student along the trajectory. On the other hand, if the γ is negative, then the trainer slows down the student's haptic display, consequently the force pushing against student increases. In either way, as long as the student finds back the correct force to apply, the trainer lets the student realize the task himself/herself, without interrupting or guiding. Moreover, if an impedance type VR is in use or the force is just a function of time, the method can be implemented in such a way that not the velocity, but the force is enhanced or degraded by the trainer as in our push-button example. However, we confine ourselves to explanations above and presume that the variations can be extracted from these explanations.

Since this method allows an on-line haptic correction of a student's performance, it exhibits a unique and novel training structure. However, it is more complex in comparison with HVO and FD methods, and the training signal is not as clear as in HVO and FD methods. Thus, students need longer time to adapt themselves and understand the way

the training functions. Knowledge about the task beforehand, especially either about the force or velocity to apply, is an advantage to the meaning of the correction signals. In Chapter 5, this issues are investigated via user tests and results are discussed.

4.3 Training via a Trainer Model

It is not always possible to assure the existence of an expert during training - learning activities. However, if a model of the trainer exists, there is a potential that the training procedure is checked / corrected by this model. Such a training structure that involves a trainer model brings in many advantages. Since a multi-user scheme is not anymore necessary, the hardware costs are reduced to half. More importantly, the modeled skills of a late expert can be used to train new generations. Likewise, no longer existing skills for specific tasks can be recreated. It should be added here that having a possibility to train alone would provide the student with a more relaxed training environment.

This section summarizes our contribution to extend VR training systems with a model of the trainer. There is no other work known to us that integrates a trainer skill model into the training systems.

4.3.1 Overview

Trainer skill has been identified as a hybrid dynamic system, which has switching impedances structure, as it was explained in Chapter 3. The knowledge of this modeling and identification work is not desired in order to follow the training strategies explained below. Suffice to assume that a trainer model exists in the form of switching impedances. Combining such a model with HVO, FD, and FVL training strategies introduced above in multi-user training concept establishes unique and novel VR training strategies.

The concept block diagram of “training via a trainer model” is given in Figure 4.7. Reference velocity and position trajectories are defined for the trainer model appropriately to the observed interaction of the trainer with the virtual task. When the training starts, the trainer model calculates the force \mathbf{f}_t that trainer would apply along these trajectories. Then, depending on the training strategy an appropriate correction signal T_c is fed to the student. Main working principles of HVO, FD, and FVL methods with a trainer model are introduced in Section 4.3.2. Sections 4.3.3 and 4.3.4 indicate simulation and RT experiment results of the training models respectively. An important issue concerning the adaptiveness of the trainer model is examined and investigated in Section 4.3.5.

4.3.2 HVO, FD and FVL Methods using the Trainer Model

In the implementation of the HVO method with a trainer model, the observation of the student’s force \mathbf{f}_s is necessary. The method is sketched in Figure 4.8(a). The hybrid dynamic trainer model block is expanded in Figure 4.8(b) to show how the observation and visual/acoustic influence principle functions in order to understand the training structure better. According to the trainer reference velocity, switching impedances hybrid model defines the trainer output force \mathbf{f}_t . Collision information from the VR is used to start the trainer model, i.e. to switch from freespace state to the first state of the task realization. This force is then compared with \mathbf{f}_s . The correction signal generated by the training

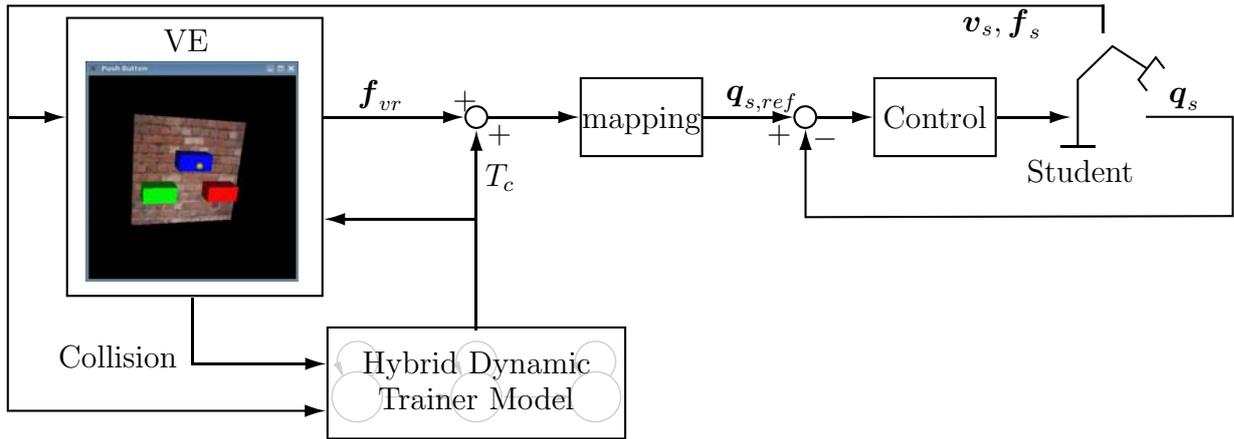
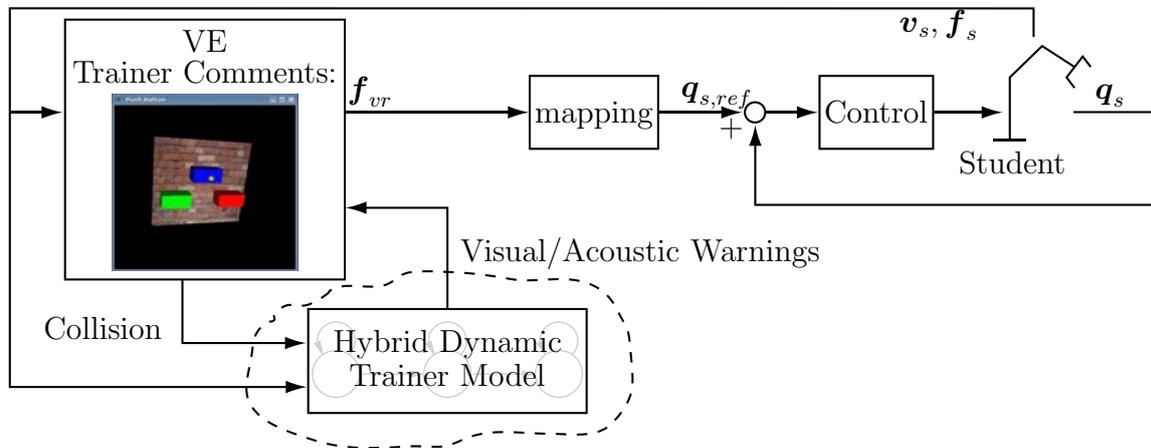


Figure 4.7: The concept of the VR training system including hybrid dynamic trainer model.

system is in the form of a visual and/or acoustic warning. A database is created, which contains different visual or acoustic warning messages. Depending on the magnitude of the force difference $|f_t - f_s|$, a decision is made which warning signal is to be played. If the warning is visual, a trainer comment appears on the visual scene. This implementation is advantageous, since the student is actively realizing the task. However, it bears in its nature the disadvantages of visual and/or acoustic enhanced training: to teach the force, the student is informed using other modalities. This is a hindrance for the tasks, where

(a)



(b)

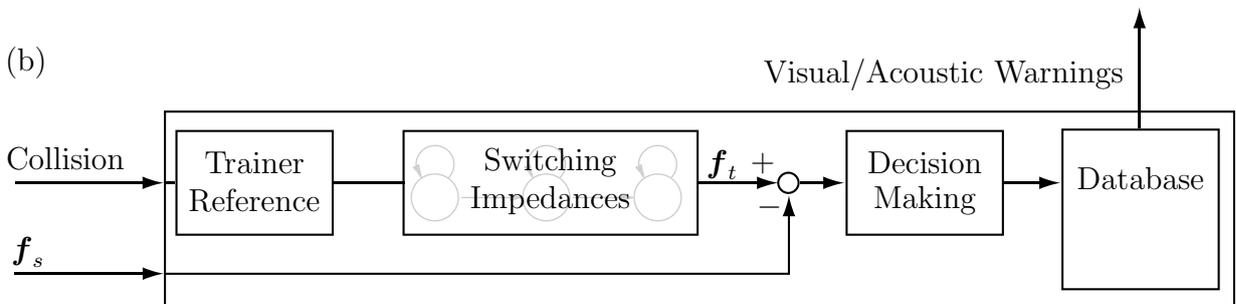


Figure 4.8: (a) HVO training method including the trainer model. (b) Insight of the hybrid dynamic trainer model.

the visual or acoustic information is not reliable, and the task is realized mainly depending on haptic cues. Hence HVO method can also be called “Visual Tutoring”.

Another way of implementation of the trainer model in trainings is to demonstrate to the student complete task using the model (FD approach). For this purpose, it is possible to use the model as a guide, which leads the student on the position trajectory haptically. However, in this case, not the force itself, but the trajectory is the issue to train. Since we aim at force skill training, our implementation requires the student not to move his/her haptic display. The task is realized by the trainer model, a trainer force \mathbf{f}_t is fed to the student. Visualization is synchronized with \mathbf{f}_t by sending the trainer position \mathbf{p}_t and orientation \mathbf{o}_t to the VE, and move the avatar appropriately. Thus, the student has a chance to visually observe the moving avatar and at the same time feel the corresponding force, i.e. haptically observe the trainer. The construction of the FD training loop via a hybrid dynamic trainer model is shown in Figure 4.9. The method enables to student to associate certain phases of the task with appropriate forces and torques. As in multi-user FD cases, student’s being passive is apparently the biggest disadvantage of this method. Additionally, the forces felt by a person are related to the arm posture. In this method, the trainee needs to have a constant arm posture, and feel all the forces in this posture.

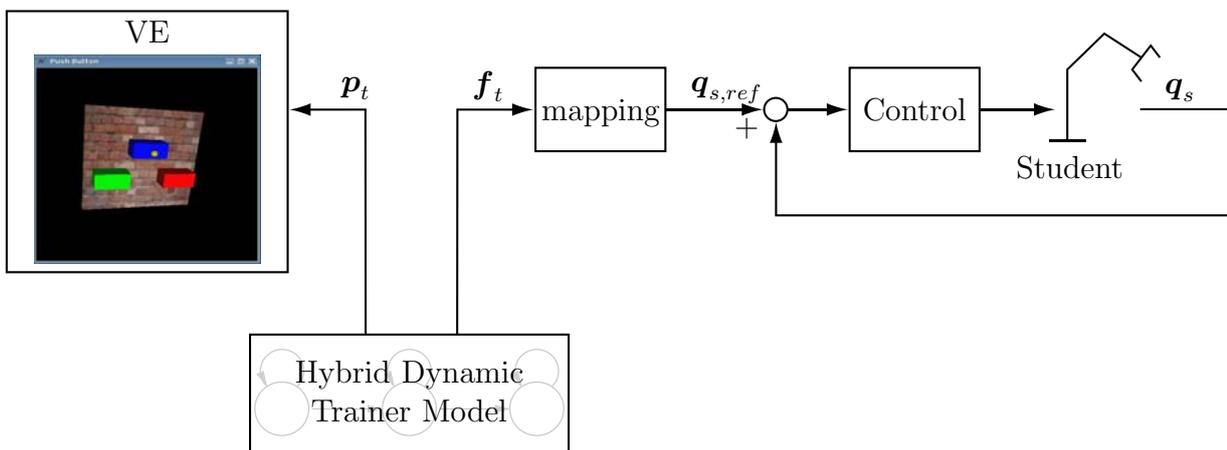


Figure 4.9: FD training method including the dynamic trainer model.

FVL method is the same as HVO method for the observation. But it differs from HVO in the sent correction signal. To overcome the main deficiency of HVO or similar methods, FVL method aims at training the student’s force skill haptically, while the student is actively realizing the task. Thus, it is a kind of “Haptic Tutoring” strategy.

Principally FVL methods allow student to realize the virtual task himself/herself. At the same time, the trainer model simulates the interaction of the trainer with the VE in the background. If there is a deviation between this simulation and the performance of the student, first the trainer model pauses the simulation, and then it modifies the reference signal for the student in such a way that the student is expected to correct himself/herself. If this correction happens, the trainer model goes on with the simulation from the point that was paused, and keeps on checking for deviations. This procedure is repeated, until the task is completed. The time that the trainer model simulation spends in pause, is a good measure on student’s success: The longer the pause, the worse the student’s performance. In Figure 4.10(a) working principle of the FVL method is shown. An insight look into the hybrid dynamic trainer model is given in Figure 4.10(b).

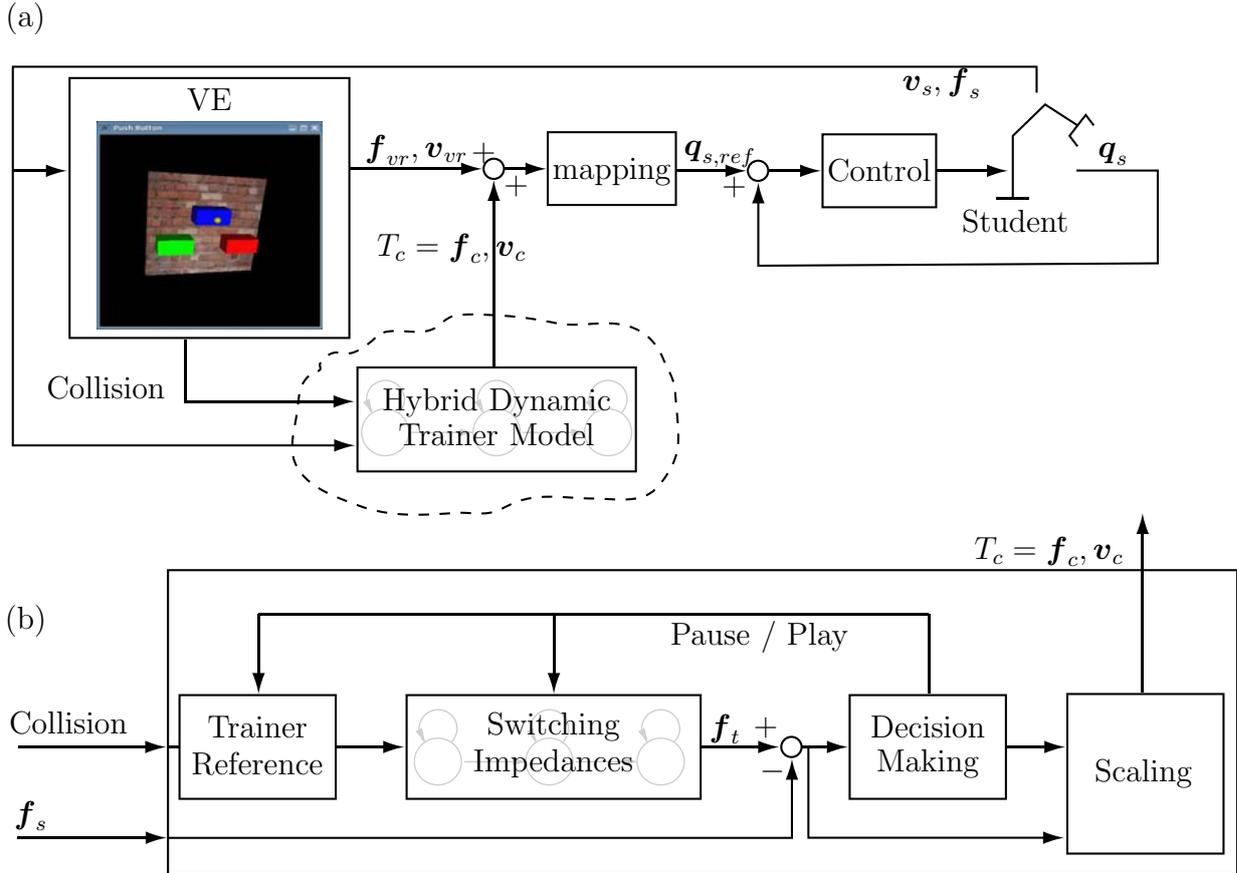


Figure 4.10: (a) FVL training method including the trainer model. (b) Insight of the hybrid dynamic trainer model.

The correction signal T_c of the FVL methods including hybrid trainer can be written as follows, if the interaction is 1D:

$$v_t(t) = \begin{cases} 0 & \text{if } |f_t - f_s| \geq f_{tol} \\ v_{t,ref}(t) & \text{otherwise.} \end{cases} \quad (4.2)$$

$$T_c = \begin{cases} \{\gamma_1, \gamma_2\} \cdot (f_t - f_s) & \text{if } |f_t - f_s| \geq f_{tol} \\ 0 & \text{otherwise,} \end{cases} \quad (4.3)$$

where v_t is trainer reference velocity, f_{tol} is pre-defined maximum allowable force deviation. If the correction signal T_c is force f_c , then the constant γ_1 , if it is velocity v_c , then the constant γ_2 is used to scale the force difference between trainer and student, i.e. $f_t - f_s$.

Before we proceed to the simulation and RT experiment results while using hybrid trainer, a careful discussion on the switching conditions of the trainer model is needed for HVO and FVL methods. As explained above, the switching between states is defined by the trainer reference. It is paused if the student deviates from the expected force, and stays paused until the student correct his/her force. If the interaction force is defined by a spring like element along the trajectory, i.e. force and position is coupled, then correcting the force would automatically suggest correcting the position, like in our push-button example. In scenarios, where the force and position are perpendicular, like in drawing or calligraphy, there is a chance to correct the force with these methods on each position. Thus, defining

the switching positions by trainer model does not bring any problems. If the switchings occur velocity based, not position based, in the direction of the applied force, as in the bone drilling example, then it is acceptable to switch depending on the trainer reference, since we would like the trainee to realize the task with the velocity and force, exactly as the trainer would do. That is why we do not let him go to the next state, if he/she does not follow the trainer's velocity. However, if there is a 3D trajectory to follow and on some certain points of this trajectory switching occurs, it may happen that the student already moves to the next state without correcting himself/herself, and the trainer model still wants to teach the force from the previous state. There can be different approaches to come over such cases. First of all, the switchings of the trainer model can be realized on the basis of student position, not trainer reference. However, we cannot guarantee that in each individual state the student manages to realize the task correctly. On the other hand, it is possible to constraint the student, so that it is not possible to go to the next step, before the applied force is corrected. For this aim, virtual haptic fixtures constraining the movement can be implemented. Another solution is to join FVL method with haptic guidance. In this case first the student is brought to the point where trainer would be via haptic guidance method, keeping the student passive. From then on, FVL method is applied to train the student. In order to stick to the FVL method only, without switching between methods, it is also possible to assume that the student does not introduce abrupt changes, and follows closely the trainer. Thus, the correction is in small range, and followed by the student properly. In other words, the student manages to correct himself/herself before switching to the next state or the switching is not related to the position, only to the velocity. Consequently, switching depending on the trainer reference is accepted to be satisfactory.

4.3.3 Simulation of FVL Method Including Trainer Model

It is a very difficult task to simulate a training process. The VE, student's behavior and his/her interaction with the VE, trainer model, robot's inner dynamics and the control of the robot have to be simulated synchronously. Moreover, the trainer model should observe the student's interaction, and send a correction signal appropriate to the training strategy.

Push-button VE system offers a more suitable platform for simulations than BD-MTS, because of the simple linear spring-damper structure of the VE. Since the trainer interaction is previously observed, the reference velocity v_t followed by trainer is known. Using this velocity as input to the hybrid trainer model, we can simulate the trainer force. To start with, it is assumed that the student realizes the task with a wrong velocity, which is obtained by scaling the reference velocity of the trainer. So that $v_s = \alpha v_t$. This velocity is integrated and noise ν is added to it to define the position of the student as $p_s = \int v_s dt + \nu$. Then, the student's interaction with the VE is simulated by defining this velocity as the input of the VE, and the output of the VE is the force obtained from haptic rendering of the virtual scene. Consequently, we can simulate the student's and trainer's interaction with VE individually as given in Figure 4.11(a)-(c).

Accordingly, it is aimed at simulating how the trainer model corrects the performance of the student. At the beginning, the hybrid trainer model is used in such a way that it does not take into account whether the student is correcting his/her performance or not (feed-forward way). Depending on the difference between forces f_t and f_s , the correction signal T_c is defined by the equation (4.3). The bottleneck of the simulation is the mapping

part. VR output, which is force, plus the correction signal has to define the reference position. However, the position had to be pre-defined as a reference signal for interaction with VR. As a solution for this bottleneck, we let the correction signal affect the position of the student directly, instead of the force. This position signal is then sent to the VE. Consequently, it is observed that the output VR force follows the desired expert force. Note that the mapping is skipped, and the student's force is taken as the VR output force plus noise. The results of this simulation are given in Figure 4.11(d)-(f). Following that, the simulation is repeated while pause/play technique is in use. That is, after the correction signal is sent to the student, the trainer model simulation is paused and suspended until the student corrects his/her force (feedback way). If the difference between the trainer f_t and student f_s forces are more than the pre-defined acceptable force tolerance f_{tol} , then the trainer switches his/her reference velocity to zero as in Figure (h), i.e. stops pushing process and fixes his/her force, and then sends a correction signal to the student. This signal can be velocity or force, depending of the type of the VE. Unless the student corrects his/her performance, and the force difference is again in acceptable range, the trainer model stays fixed in zero velocity. The results are depicted in 4.11(g)-(i).

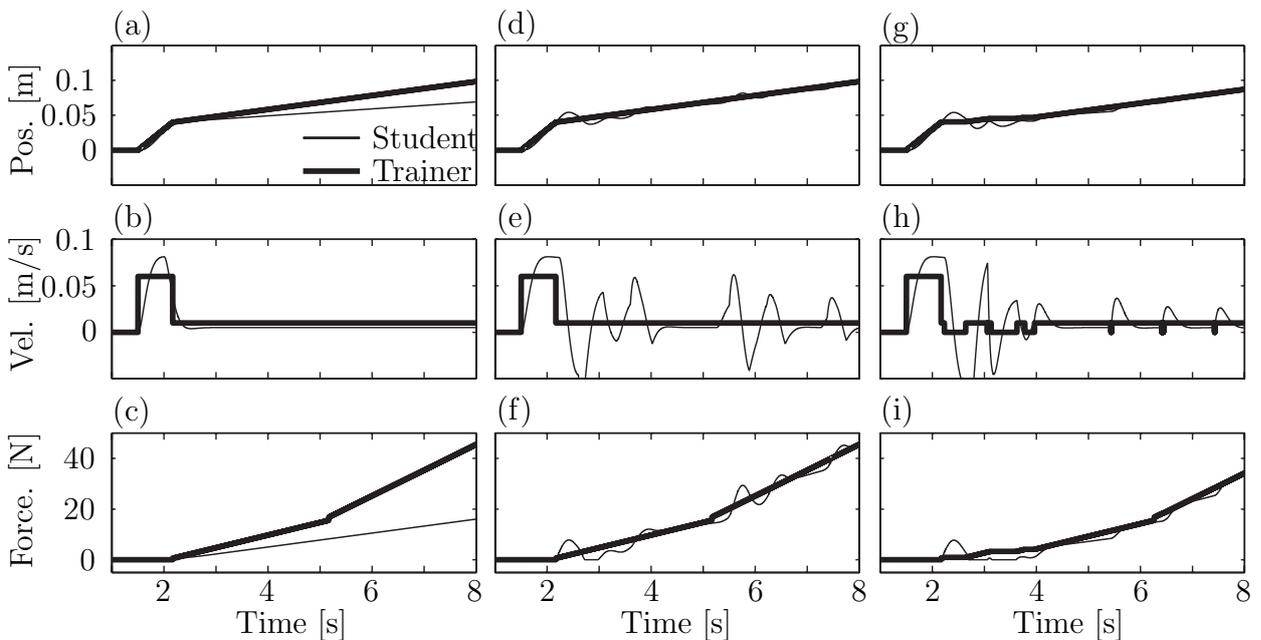


Figure 4.11: (a)-(c) Generated reference position and velocity, and calculated force signals for student and trainer. (d)-(f) Trainer model corrects the student, but does not have pause/play structure (feed-forward way). (g)-(i) Trainer model corrects the student and includes pause/play structure (feedback way).

A closer look to the Figures (f) and (i) is given in Figure 4.12. It can be clearly seen that when the trainer stops and sends the correction signal to the trainee, and waits until the student corrects himself/herself, the squared sum of error between student and trainer is lower. But, in return, the simulation time gets longer. If you again refer to Figure 4.11(f) and (i), at the end of the simulation, using feedback way it is reached to only ca. 35 N, whereas using feed-forward way to ca. 45 N, which is the expected end of the pushing process.

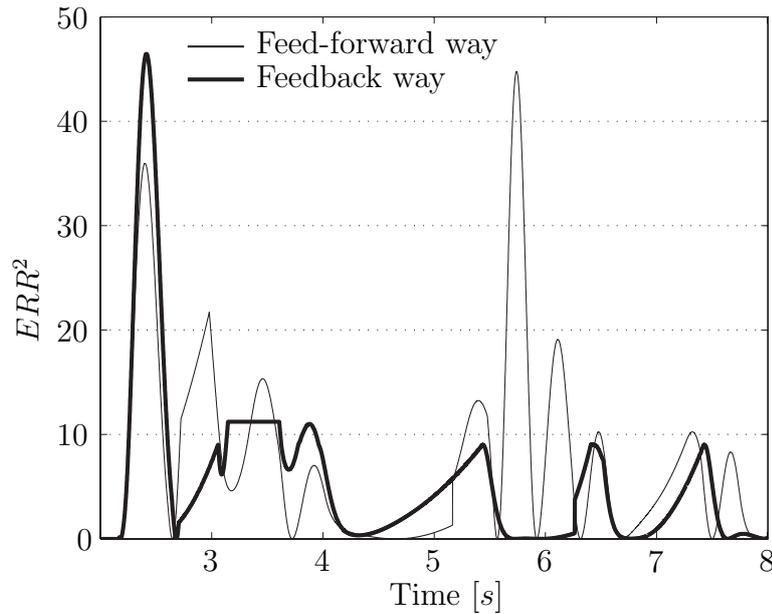


Figure 4.12: Error comparison when the FVL trainer is in feed-forward and feedback way.

As it can be seen, the simulation of such a training structure is very complicated. Many assumptions and simplifications have to be done and the efficiency of the training with a trainer model is restricted to these assumptions. In comparison with the simulation, performance of RT experiments is more straight-forward. Next section summarizes the results of these experiments.

4.3.4 Real-Time Experiment Results of Trainer-in-the-Loop

The training strategies involving a hybrid dynamic trainer model are implemented in RT, both for push-button training system and BD-MTS. Experiments are performed to check how the trainer model functions and influences the student. The results presented in this section account only for these points. The efficiency of the overall training system and the ways of improving the student's skill via proposed methods are investigated in Chapter 5.

Push-Button Trainer in Loop: FD Results

As it is depicted in Figure 4.9, trainer force is sent to the student, who has to balance this force, whereas trainer position is sent to the VE in order to visualize the push-button task.

In Figure 4.13(a), the trainer position is shown. As in the real case, pushing is performed with a constant velocity. The task is finished when the button is pushed 10 cm. The resulting trainer force, that is output of the switching impedance model, is given in Figure 4.13(b), together with the student's measured force. Trainer force shows two different slopes during the pushing process, which is expected since the button has two different layers. Student's force smoothly follows the trainer force. The visible deviations are due to student's moving his/her haptic display, i.e. not keeping it constant. Following the visual scene and feeling the appropriate forces, the trainee can observe the whole process. The final force that is around 45 N is fed to the student until he/she correctly reacts and balances it.

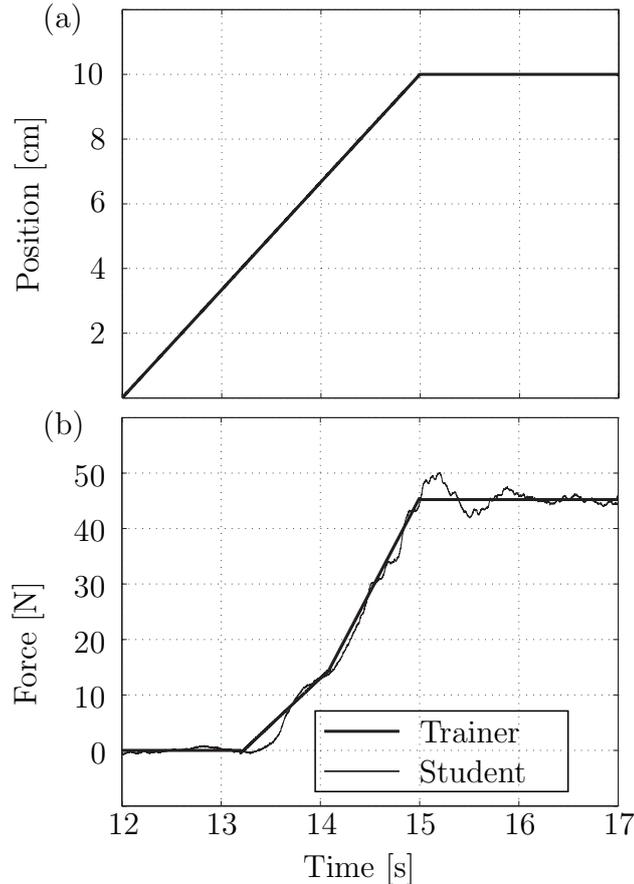


Figure 4.13: FD training via trainer model for push-button task. (a) Trainer position that is sent to the VR. (b) Comparison of the trainer force and student force

Push-Button Trainer in Loop: FVL Results

In Figure 4.14(a) the observation of student’s force by trainer model is given. When the trainer model determines that the deviation of the student force is more than the pre-defined error tolerance, a force correction signal f_c is sent to the student. The trainer model freezes and waits until the student reaches the correct range again. Figure 4.14(b) shows the “pauses” of the trainer reference velocity. It is simply set to zero in case of deviation, so that the position is also kept constant as it can be seen in Figure 4.14(c). Note that the pause signal is also sent to the switching impedances model.

When the FVL method is used, the biggest advantage for the student is that he/she can realize the task himself/herself. Trainer model intrudes only when the student’s task realization deviates from the trainer’s. However, the student has difficulties to differentiate the correction force from his/her interaction force. Thus, there are long “pause” periods that result in longer duration of task realization.

BD-MTS Trainer in Loop: FD Results

Similar to the push-button case, the trainer position given in Figure 4.15(a) is sent to the VR to visualize the virtual task. The student has to balance the trainer force, which is depicted in Figure 4.15(b). Appropriately to the real drilling case, the trainer model starts the drilling as soon as possible after collision, and then keeps the force and velocity constant. Except the small deviations that are caused by student’s not being able to

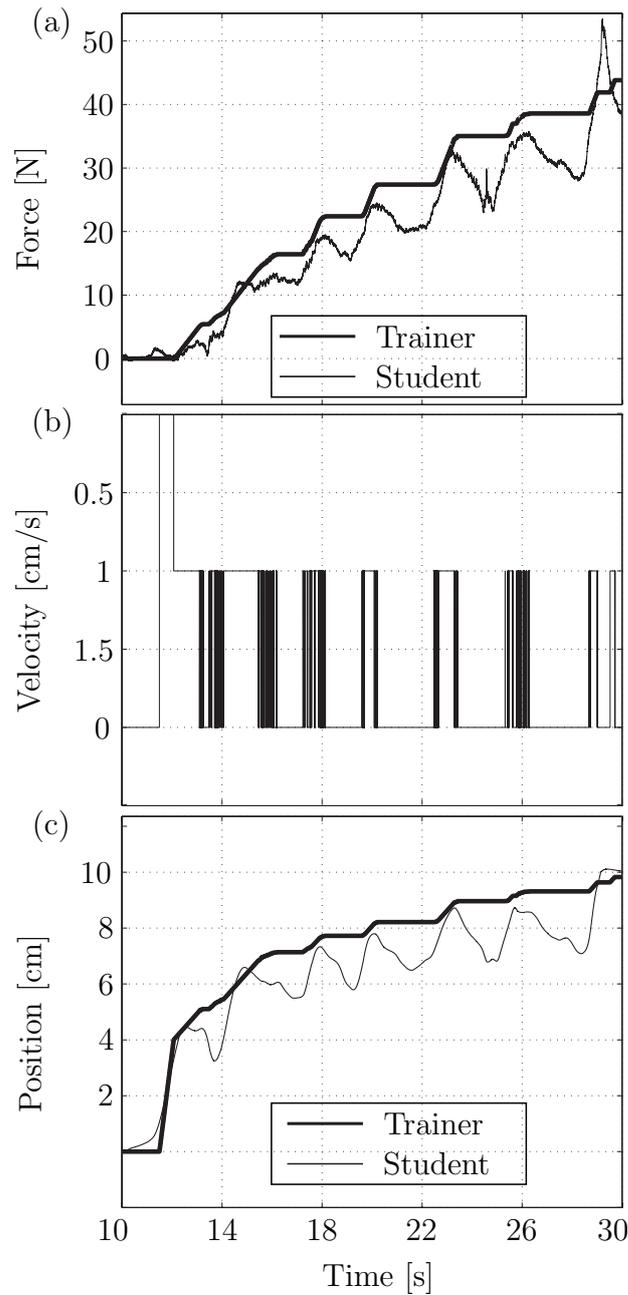


Figure 4.14: FVL training via trainer model for push-button task. (a) Comparison of the trainer force and student force (b) Trainer velocity (c) Comparison of the trainer position and student position

keep his/her haptic display exactly constant, the student can balance the trainer's force smoothly.

BD-MTS Trainer in Loop: FVL Results

Since in BD-MTS the VR is admittance type, the correction signal sent from the trainer is velocity v_c . Following the equation (4.3), v_c is calculated depending on the force difference between the student and the trainer.

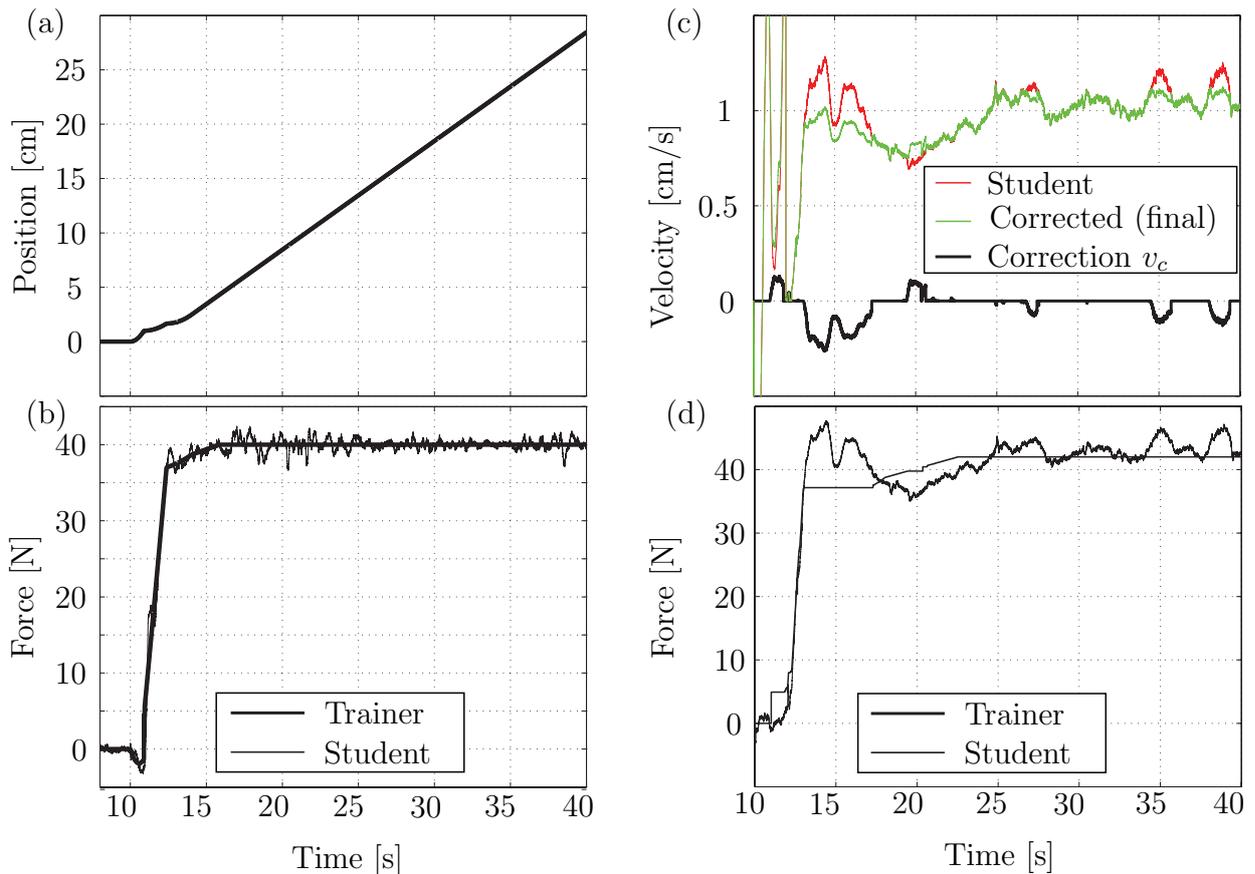


Figure 4.15: FD and FVL training via trainer model for bone drilling medical training system. (a) FD: Trainer position that is sent to the VR. (b) FD: Comparison of the trainer force and student force (c) FVL: Correction of the student's velocity. (d) FVL: Comparison of the trainer force and student force

In Figure 4.15(c), the velocity that the student tends to use, the correction v_c on this velocity and the consequent drilling velocity are given. As it can be seen in Figure 4.15(d), when the force difference is beyond a certain amount, a correction signal is sent to the student. Otherwise v_c is zero.

4.3.5 Including Student Psychology: Adaptive Trainer Model

Human behavior as a whole is very complex to model. To model this complicated organism, it is first needed to split the behavior into smaller categories and then focus on one of these to model. Our identification and modeling effort is focused on sensorimotor skills category. The model, which represents the trainer sensorimotor skill, is then used in VR training systems as an observer and controller of student's performance.

However, as Gaines underlines, the dissection of human behavior does not exclude cross-correlation between categories [41]. This argument causes the following questions for us to arise: "How is the student's progress assessed by the trainer? What is the trainer's cognition on the student's reaction to the teaching signals? How does the trainer change the training strategy depending on his/her cognition?". These questions are not so trivial to answer. However, it is expected that a good trainer adapts himself/herself depending on the student and would find the best way to train each particular student. Although

the cross-correlation effects are not the main focus of this thesis, it is essential to discuss whether they can be taken into account in the proposed training strategies.

The Reaction of the Student to the Action of the Trainer

As it is explained throughout this chapter, the training systems mostly aim to correct the student, when his/her performance deviates from the trainer's performance. The correction stimulates acoustic, visual and/or haptic senses of the student in different forms. It is expected that the reaction S_r of the student to the correction signal T_c differs from one student to another. Assuming the relation between S_r and T_c as a P-controller, the reaction can be formulated as

$$S_r = K_p \cdot T_c, \quad (4.4)$$

where K_p is the controller gain of the student. A possible way to alternate the training strategy is to adapt T_c depending on K_p representing the student.

Below, a summary of the adaptation possibility of the training strategies and, if possible, the realization of the adaptation for individual training strategies are given.

Adaptiveness of HVO Methods

As it was depicted in Figure 4.8(b), the correction signal in HVO training is a visual warning that is chosen from a database depending on the deviation between the student and the trainer. Duration of the correction signals T_c is directly related to the gain K_p . If "Apply Larger Force" visual command is given by the trainer, the student, who finds the correct force range in the shortest time, has the biggest proportional gain. On the contrary, if the warning signal stays for long time on, it indicates a small gain.

Thus, if repeated trials of a student establish a characteristic K_p feature, the database can be adapted. For example, for a student, who presents small K_p gains, "Apply Larger Force" command can be replaced by "Apply Much Larger Force" and so on.

A comparison of two different users, who are trained with HVO method and show different K_p features, is given in Figure 4.16(a). If the difference between student and trainer is more than +3 N "Apply Less Force", if it is less than -3 N "Apply Larger Force" command is given by the training system. In the beginning, both users are applying much smaller force than expected. Consecutively, "Apply Larger Force" command is announced by the trainer model for both. In return, Student-A, finds the correct force within 1 second. However, Student-B needs 4 seconds, although the force difference for A and B was more or less the same before the command. In this case, if Student-B has a gain equal to $K_{P,B}$, the gain of the Student-A is $K_{P,A} = 4K_{P,B}$. If the further performance of the same students are observed, the tendency of Student-A is always to answer with a big gain, i.e. very quick, which results in jumping from "Apply Larger Force" to "Apply Less Force" command as shown in Figure 4.16(b). Thus, for Student-A the training commands can be scaled down, whereas for Student-B it is necessary to scale up the command in order to fasten his/her reaction.

Adaptiveness of FD Methods

In FD methods, the students do not actively realize the virtual task. They only follow the trainer visually and balance the force applying against them. Since it is not possible to check the reaction of the student, FD training methods do not allow an adaptation.

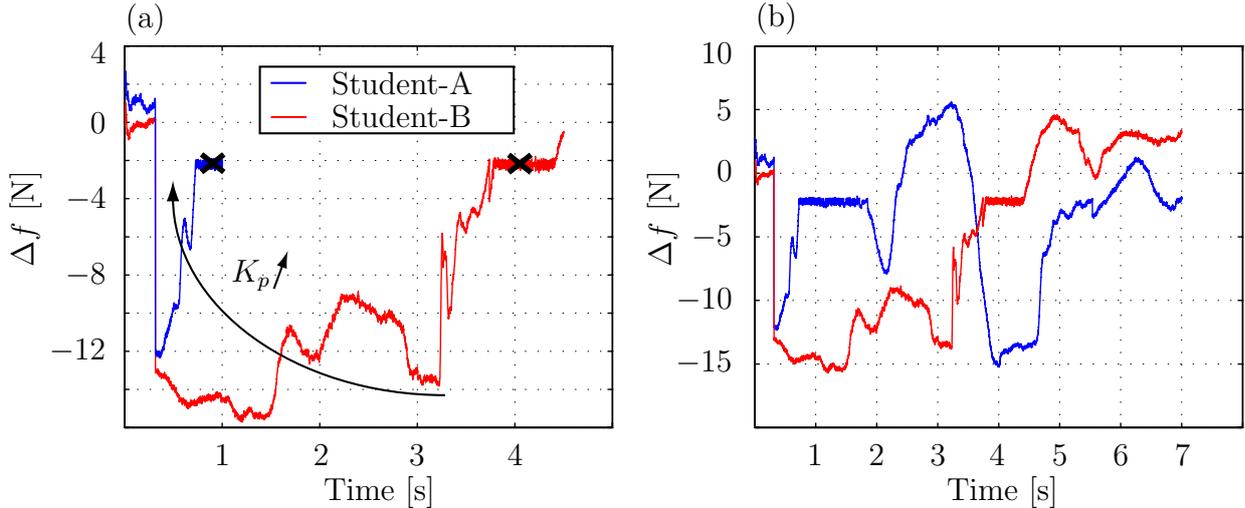


Figure 4.16: Comparison of proportional gains of two different students trained in BD-MTS using HVO method. (a) Reaction of students when both have ca. $\Delta f = -12$ N deviation from the teacher and both get the same correction signal T_c . (b) General progress of students' reaction after finding back the correct force.

Adaptiveness of FVL Methods

In FVL method, the student is realizing the task himself/herself. If the deviation between the trainer and student is more than a pre-defined limit, then the deviation is scaled with a factor γ_1 or γ_2 depending on the VR type, as it was given in the equation 4.3.

If the student's proportional gain is K_p , the factor γ_1 or γ_2 can be adapted as given below:

$$\bar{\gamma}_{1,2} = \frac{K_{p,ref}\gamma_{1,2}}{K_p}, \quad (4.5)$$

where $K_{p,ref}$ is reference gain, which is the gain that belongs to one of the students. For the other students, the gain K_p can be defined relative to $K_{p,ref}$.

In Figure 4.17(a) the force deviation Δf between the trainer and student is sketched for two different students, who were trained in BD-MTS by a FVL method. Student-A reacts on the correction signal quickly and within 1 s he/she is back in acceptable force deviation range. We assign his/her proportional gain as $K_{p,A} = K_{p,ref}$. However, Student-B's reaction is much slower. He/she needs 5 s to correct himself/herself. Respectively, his/her gain is less, i.e. $K_{p,B} = 0.2K_{p,ref}$. The adapted scaling factors are:

$$\bar{\gamma}_{2,A} = \frac{K_{p,nom}\gamma_2}{K_{p,nom}} = \gamma_2, \quad (4.6)$$

$$\bar{\gamma}_{2,B} = \frac{K_{p,nom}\gamma_2}{0.2 \cdot K_{p,nom}} = 5 \cdot \gamma_2. \quad (4.7)$$

Note that the correction signal is velocity, when the FVL method is used in BD-MTS. Therefore, in (4.6) and (4.7) γ_2 is used. In Figure 4.17(b), the correction signals for both students are given. As one can see, after 1 s, there is not anymore a correction signal for Student-A, whereas for Student-B, it continues for 5 s.

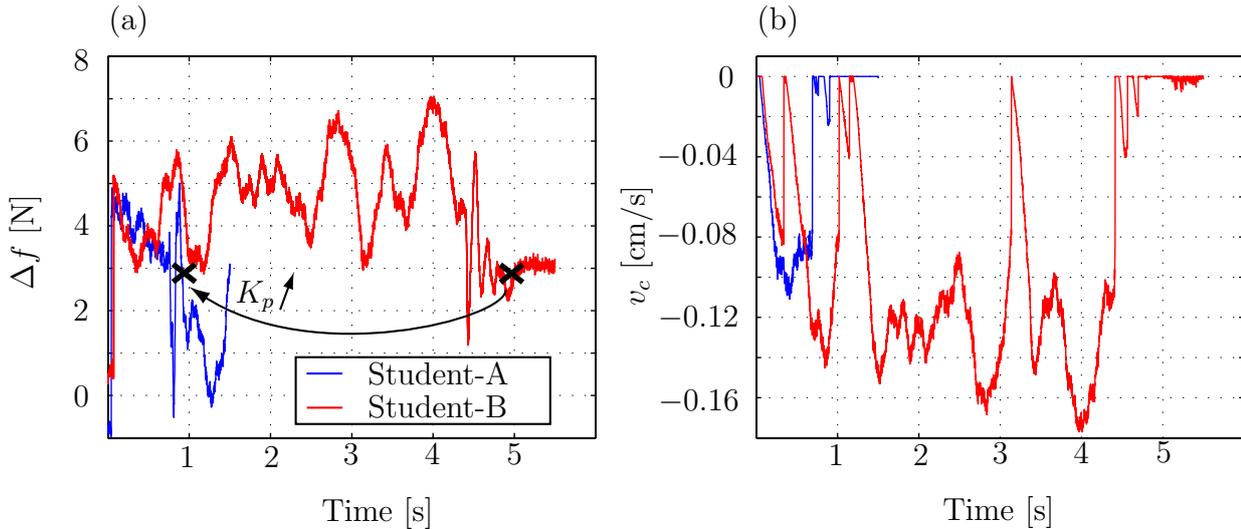


Figure 4.17: Comparison of proportional gains of two different students trained in BD-MTS using FVL method. (a) Reaction of students when both have ca. $\Delta f = 5$ N deviation from the teacher and both get the same correction signal T_c . (b) The correction signal T_c for students A and B.

Beyond P-Controller

The reaction of the student formulated by (4.4) as a P-controller is undoubtedly a gross approximation. As Tsuji *et al.* and many others suggest, the model of human tracking control should include the dead time τ and a phase lag element with time constant T_s . Thus, the reaction can be specified again as follows:

$$S_r(s) = \frac{K_p}{1 + T_s \cdot s} e^{-\tau s} \cdot T_c(s). \quad (4.8)$$

On the other hand, another possible way is to follow the theory of Hogan *et al.* [39] and look for a jerk optimal model of student arm movement with respect to the training signal.

As it was stated out before, these points are beyond the scope of this thesis, but they yield interesting and challenging research issues.

4.4 Discussion

In this chapter, results of a large investigation on training strategies and possibilities of force skill transfer via VEs were presented. While there are numerous training systems aiming at position and 3D trajectory learning, very few works concern force skill training. A perceived reason for this situation is the difficulty to teach position and force synchronously. Another reason is that visual modality had been considered to be more dominant than haptic modality concerning perceptual motor skills [102]. However, as Ernst *et al.* underline, if the visual information is less reliable to realize a task, than haptic cues play more important role in motor skill learning procedures [29]. Many surgical operations, such as cutting, drilling, suturing incorporate such situations and they are realized by surgeons mainly by relying on haptic cues. Moreover, in such cases, applying excessive force causes irreparable damages. Thus, even if the visual information is available, applying correct force on these trajectories plays a crucial role for a successful operation.

In order to achieve this aim, three groups of strategies were proposed: Enhanced training, multi-user training and training via a hybrid dynamic trainer model. Below, we first compare the methods with each other and underline their main features. Then, their possible usage in different application configurations is discussed.

4.4.1 Comparison of the Methods

Enhanced training methods augment the VEs by extra features that are not available in real tasks. In our methods, these extra features are visual, acoustic or haptic warning signals that inform the student about the force he/she applies. The biggest advantage of visual enhancement is the clarity of the information sent to the student. However, in many cases, such as in surgical operations, no visual information about the force is available. The operations are realized entirely by relying on the haptic sense. Thus, for such cases, visual enhancement is degrading the quality and reliability of the training. We concluded that for an inadequate training, the signal enhancement of the VR training system should have the same modality of the signal to learn. Consequently, to learn force, we introduced a haptic enhancement technique, which is in the form of an additional, constraining force that is felt by the user depending on his/her performance. The strong feature of the haptic enhancement is that it is applicable for the problems, where there is a relation between force and velocity along the direction of the interaction, as well as if there is a force profile to follow independent of the velocity. A disadvantage of this method is that the student does not know if he/she applies too small or too large force. The only haptic information is that the force is wrong.

Multi-user methods integrate the trainer into the trainings. Unique training structures arise, when Yokokohji's "stop-simulation to feel the force" principle is inserted into multi-user schemes. On one hand, we can ask the trainer to realize the task and the student to observe (FD method). However, this observation is not only visual, but also haptic, which is achieved by sending the trainer forces to the student. Note that the student has to keep his/her haptic display on a fixed point in space and balance the force that presses against him/her. If one keeps in mind one of the main deficiencies in classical surgery training, that is missing haptic feeling when observing an expert surgeon in an OP room, the developed strategy here is very promising. Main disadvantage of the FD method is student's being passive. The student cannot realize the task himself/herself. Thus, other multi-user training methods were considered, which inversely give the observation duty to the trainer and task realization duty to the student. The trainer can send either verbal (HVO method) or haptic correction signals (FVL method). Implementation of the HVO method is straight-forward. It is a unique method that gives the trainer a chance to feel the forces that student applies and to warn the student verbally to correct them. Since this verbal information is very clear, the student skill training with multi-user HVO method turned out to be very efficient. These issues are discussed again in Chapter 5. Another important feature of this method is that even if the student follows wrong trajectory, he/she can be warned about this by the trainer. In case of FVL method, the trainer augments or decreases the interaction force or velocity of the student in order to correct him/her. This is a kind of haptic tutoring approach, which aims to make student feel exactly what expert would feel on a certain position on the trajectory. However, since it is not possible to control force and position at the same time, the difference between forces is scaled, and presented as an additional resistance to the student. It is expected that the student

finds the correct force and position, because only then the additional resistance would disappear. Although the method is unique, a priori knowledge of the student about the task is essential. Otherwise, the student might have difficulties to interpret the correction signals coming from the trainer.

Training via a hybrid dynamic trainer model functions almost the same as multi-user trainings. The main difference is that instead of the trainer himself/herself, his/her model is integrated into the training structure. The same methodologies, i.e. HVO, FD and FVL trainings were implemented. There is no other work known to us that integrates trainer model in the force skill training procedures. Note that it is not always possible to assure the existence of an expert during training - learning activities. However, if a model of the trainer exists, there is a potential that the training procedure is checked and corrected by this model. Training structure involving a trainer model can be very advantageous. Since a multi-user scheme is not anymore necessary, the hardware costs are reduced. What is more, future generations can make use of the maoeled skills of late experts. In this way, extinct skills for specific tasks can be re-animated. It should be added here that having a possibility to train without the presence of the expert would provide the student with a more relaxed training environment.

4.4.2 Application of the Methods in VR Training Systems

All of the proposed methods are based on modifying or enhancing the VR interaction signals to bring a training effect. Thus, theoretically, the training methods are applicable to any VR training system, such as drilling, sawing, peg-in-hole or grinding that aims at force learning. However, there are different limitations that constraint the range of the applications.

Multi-user FVL method requires the trainer to restrict the movements of haptic display for observations. Consequently, the DoF that the trainer can use to send the correction signals are restricted as well. The trainer can use maximum 3-DoF of the device for observations, so that the remaining 3-DoF can be used for corrections. For example, the trainer can observe the forces along the x, y, z directions, by keeping the robot position fixed in the Cartesian space. The remaining 3-DoF, i.e. rotation along x, y, z axes, are then used to send correction signals. Using the model of the trainer to observe and correct the student eliminates this restriction. Nevertheless, the training relies on the fidelity of the trainer skill model. As it was discussed before, the tasks introducing freespace hand movements and high non-linear objects to be interacted cannot be represented by switching impedances. However, assuming that a perfect trainer model exists, FVL method is a unique solution for the training systems, since it enables the student to experience the correct haptic information without stopping the simulation. Note that for multi-user FVL, hybrid trainer FVL and hybrid trainer HVO methods, it is assumed that the student follows the correct position trajectory, however deviates from the correct velocity or force on this trajectory. An additional assumption for hybrid trainer FVL and HVO methods is that the student does not proceed to the next phase, before correcting himself/herself. To eliminate these assumptions, the methods can be combined with a haptic guidance strategy. So that if the student is on the wrong trajectory, first he/she is brought back to the correct trajectory via haptic guidance. Then it is switched to the FVL or HVO method to train force.

FD methods neither yield such a DoF limitation, nor do they require the trajectory to be known by the student. However, they stop the simulation and keep the student passive. As a consequence, the arm posture of the student is different during the training sessions than while he/she realizes the task himself/herself. Enhanced methods are easily implementable. Their biggest deficiency is that such features do not exist in a real operation. Thus, it is questionable whether the skills learned from the enhanced VE are transferable to the real operations or not. Especially, if the modalities are changed, e.g. if the student was informed about the haptic interactions visually, our experiments underscore very bad performances when the visual enhancement is turned off.

4.4.3 Application of the Methods in Telepresence

The training methods developed in this work are explicitly considered for VR applications. However, the potential of the methods to be used in telepresence applications should be emphasized.

In a telepresence application, the VE is replaced by a remote environment. The telepresence operator controls a remote robotic manipulator (slave robot) via his/her haptic display (master robot) in order to carry through tasks in remote environments. Our HVO and FD multi-user training methods yield appropriate features to train novices for such tele-manipulation tasks. The novice user can be connected to the tele-manipulation environment via his/her own haptic display. The visual feedback that the expert gets from the slave robot can also be made available for the novice user. In the implementation of the FD method, the expert tele-operator realizes the remote task and the force that he/she applies is sent to the haptic display of the novice. By keeping the display tight, the novice has a chance to observe the experienced tele-operator haptically as well. Multi-user HVO method is implemented inversely, so that the novice user actively realizes the tele-manipulation task, whereas expert operator observes and warns the student verbally. Similarly, FVL method can be used to bring the training effect. In this case, the force that the novice applies is degraded or augmented by the expert. The limitations explained above for multi-user FVL applications in VR training systems are also valid here. Remember that VEs are passive, whereas remote environments are active. Hence, the direct modification of the force that student exerts may cause instabilities.

It is known that tele-presence newcomers need training and time to adapt themselves in the specific way of using a tele-presence system. Therefore, integrating a trainer in such a training procedure of a newcomer has the potential of improving the tele-manipulation skills rapidly and reducing the time needed for adaptation. As explained above, using FD techniques, the student can observe the expert tele-operator haptically and visually. Additionally, the newcomer can be led through the trajectory using haptic guidance techniques. Joining these haptic guidance and FD methods can make it possible to demonstrate to the newcomer different phases of the manipulation task.

If a hybrid trainer model exists, it is possible to run a robotic manipulator using the model as a reference generator. Thus, it can be assumed that the robotic manipulator is the expert. In this case, the tele-operator can be trained by the robotic manipulator using FD principles. It is not possible to apply HVO or FVL method in such a configuration.

In addition, the trainer model can be used for intention detection. Assume a box-carrying tele-operation task. The tele-operator first performs “grip the box” task, than “carry the box while keeping the force constant” task and finally “release the box” task.

The model of the trainer would allow the robotic manipulator to estimate the current phase of the task. Consequently, even if there are communication network or tele-operator related disturbances, the manipulator would keep on performing the task correctly.

As the last point, we would like to remind that multi-user VR schemes can also be considered as tele-presence systems that enable two or more persons to communicate with each other haptically via a common VE. The communication may aim at training, but does not have to be limited to it. For example, interactions in applications such as tug-of-war or carrying a table are implemented successfully, and they are summarized in Appendix B.

5 User Tests and Their Objective Evaluation

Training systems are conventionally assessed through subjective evaluation methods. For example training can be observed and graded by a trainer. However, the grading of different trainers on the same task may vary from one another. The students can also be asked to fill in a questionnaire and assign notes to different questions after trainings. Nevertheless, the questions in the questionnaire might be directing the students.

Contrary to the classical training, VR training systems offer a possibility to record various interaction data during the training process. Using this opportunity, quantitative evaluation metrics can be defined.

This chapter is considered as an integration chapter. After discussing the objective evaluation metrics for VR training systems, previously investigated topics from Chapters 2, 3, and 4 are harmonized and a complete virtual training via a hybrid trainer model concept is presented in Section 5.2. This is followed by experimental investigations on the efficiency of different training strategies via user tests. Consequently, the results of the user tests are discussed in Section 5.4.

5.1 Evaluation Methods

If the number of works concerning the development of VR training systems is compared with the number of works referring to objective evaluation of the training systems, it is not improper to say that evaluation is a rather untouched research topic.

A method proposed in the literature is the assessment of the laparoscopic surgical skills using hidden Markov model distances [105]. The method is later on used by others for different applications, e.g Hirana *et al.* implements a deformable hose insertion task based on switching impedance controller, which is evaluated through HMM distances as well [46]. In order to apply this method, a HMM for a trainer λ_t , and another one for student λ_s have to be at hand. The distance $D(\lambda_t, \lambda_s)$ between these two HMMs is a statistical measure showing how close two stochastic models are quantitatively. It was defined by Rabiner as follows:

$$D(\lambda_t, \lambda_s) = \frac{1}{N_s} [\log P(\mathbf{O}_s | \lambda_t) - \log P(\mathbf{O}_s | \lambda_s)], \quad (5.1)$$

which indicates how well model λ_t matches observations O_s generated by model λ_s , relative to how well model λ_t matches observations O_s generated by model λ_s . Number of observations for model λ_s is N_s . This distance definition is not symmetric, i.e. $D(\lambda_t, \lambda_s) \neq D(\lambda_s, \lambda_t)$. Therefore, average of these two distances is used as a measure:

$$\bar{D} = \frac{D(\lambda_t, \lambda_s) + D(\lambda_s, \lambda_t)}{2}. \quad (5.2)$$

Another objective evaluation method is defining individual factors for a specific task, and observe, evaluate and compare improvement in these factors [84]. Similar to this approach, we propose to define some skill levels that are related to the important features of the task, in order to have an objective assessment on skill improvement.

However, in our opinion, the evaluation of the trainings should be based on deterministic functions rather than statistical ones, such as HMM distances. Consequently, we define a weighted Euclidean distance principle to evaluate training procedures.

5.1.1 Using Pre-defined Skill Levels

Our first approach for objective evaluations is defining some skill levels, that are related to measurable or derivable student-VR interaction data. The levels can be characterized depending on the information obtained from literature or directly from experts.

The pre-defined level L_{st} can reflect just one parameter such as position, force or time, as well as a combination of parameters. It is accepted that a student reaches to the “skillful level” L , when

$$L_{st} \geq L, \quad (5.3)$$

where L_{st} is the skill level of the student.

Although this method does not give a detailed insight into the student’s performance, it is quick and reliable to have the first impression. Especially, it is useful to decide whether to continue or finish the training sessions. It is used exactly for this purpose in our applications. In experimental investigations section, this method is referred to again to clarify the evaluation criteria.

5.1.2 Euclidean Distances Between Trainer and Student

Distance Between Measured/Derived Interaction Parameters

We define the trainer skill as a point \mathbf{p}_t in an n dimensional Euclidean space. The dimension of the space is defined by the number of all available (measured or derived) interaction parameters. In other words, the more parameters there are, the larger Euclidean space defining \mathbf{p}_t is.

In this case, a skill level for the student can be assigned via an Euclidean distance. Similar to the trainer, if the student is represented as a point \mathbf{p}_s in n dimensional Euclidean space, then his/her skill level is defined by:

$$D_{ts} = \|\mathbf{p}_t - \mathbf{p}_s\|, \quad (5.4)$$

where D_{ts} is the Euclidean distance in an n dimensional space.

It is desired that the distance D_{ts} gets shorter with increasing number of trainings as it is depicted in Figure 5.1. In other words, it is expected that the trainings should bring the trainee point in Euclidean evaluation space closer to the corresponding trainer point. After final training, it is expected that

$$D_{ts} < \varepsilon, \quad (5.5)$$

where $\varepsilon \in \mathfrak{R}^n$ is the maximum allowable deviation of the student from the trainer.

Explicitly,

$$\mathbf{p}_t = \begin{bmatrix} p_{t,1} & p_{t,2} & \cdots & p_{t,n} \end{bmatrix}, \quad (5.6)$$

$$\mathbf{p}_s = \begin{bmatrix} p_{s,1} & p_{s,2} & \cdots & p_{s,n} \end{bmatrix}, \quad (5.7)$$

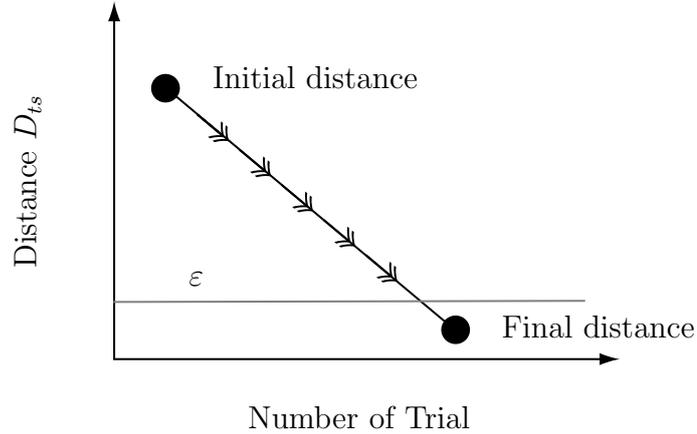


Figure 5.1: Expected change of the distance D_{ts} between the student and the trainer in Euclidean evaluation space.

where indexes $1, 2, \dots, n$ refer to different measured/derived interaction parameters such as position, velocity, time or force. Indexes t and s indicate the trainer and the student respectively. If the initial (first trial) value of the i^{th} parameter for the student is $p_{s,i}(1)$, then the normalized individual distance of this parameter to the trainer's i^{th} parameter $p_{t,i}$ is:

$$\bar{D}_{ts,i}(1) = \frac{|p_{t,i} - p_{s,i}(1)|}{|p_{t,i}|}, \quad (5.8)$$

where $i = 1, 2, \dots, n$. Each individual distance is a measure for the importance of the parameter. Bigger values of $\bar{D}_{ts,i}(1)$ indicate an important parameter for the skill to be improved, whereas the smaller values designate insignificance of the parameter for training. Note that, instead of the very first trial value for one certain student, mean value of the first training period trials and/or mean value of the different student's first trial is more reliable to use for this parameter importance analysis. In addition to that, it should be checked that large values of $\bar{D}_{ts,i}(1)$ decrease while training continues and should reach to a minimum around the last trial.

An important outcome of this approach is that it is possible to weigh the Euclidean distance to define the skill level of the student. For each parameter p_i , a weighting factor w_i is defined as follows:

$$w_i = \frac{\bar{D}_{ts,i}(1)}{\sum_{i=1}^n \bar{D}_{ts,i}(1)}. \quad (5.9)$$

As a result, weighted n dimensional Euclidean distance is an objective assessment of the student skill. It is formulated below:

$$D_{ts} = \sqrt{\sum_{i=1}^n w_i \left| \frac{p_{t,i} - p_{s,i}}{p_{t,i}} \right|^2}. \quad (5.10)$$

Alternatively, the weighted sum of absolute distances between student and trainer can be used to assess the training:

$$D_{ts} = \sum_{i=1}^n w_i \left| \frac{p_{t,i} - p_{s,i}}{p_{t,i}} \right|. \quad (5.11)$$

Distance Between Hybrid Skill Model Parameters

If the student skill is also modeled as switching impedances like the trainer model, there is a possibility to compare the parameters of both models in order to find the distance between two hybrid models. It is expected that throughout the training sessions, the student's parameters go closer to the expert's parameters, and when the training is completed, they are closest.

To check this distance, number of states and the order of the student model have to be the same as for the trainer's model. Assume that the parameters of student model for the j^{th} state are $\theta_{s,j} = [m_{s,j} \ d_{s,j} \ k_{s,j}]$, whereas trainer model parameters are $\theta_{t,j} = [m_{t,j} \ d_{t,j} \ k_{t,j}]$. The Euclidean distance between the model parameters is:

$$D_{ts,statej} = \sqrt{(m_{t,j} - m_{s,j})^2 + (d_{t,j} - d_{s,j})^2 + (k_{t,j} - k_{s,j})^2}. \quad (5.12)$$

5.2 Integration of Virtual Environments, Hybrid Skill Identification and Training Methods

As it was stated before, individual works explained in previous chapters are joined together in this chapter and our VR training system involving different training strategies is accom-

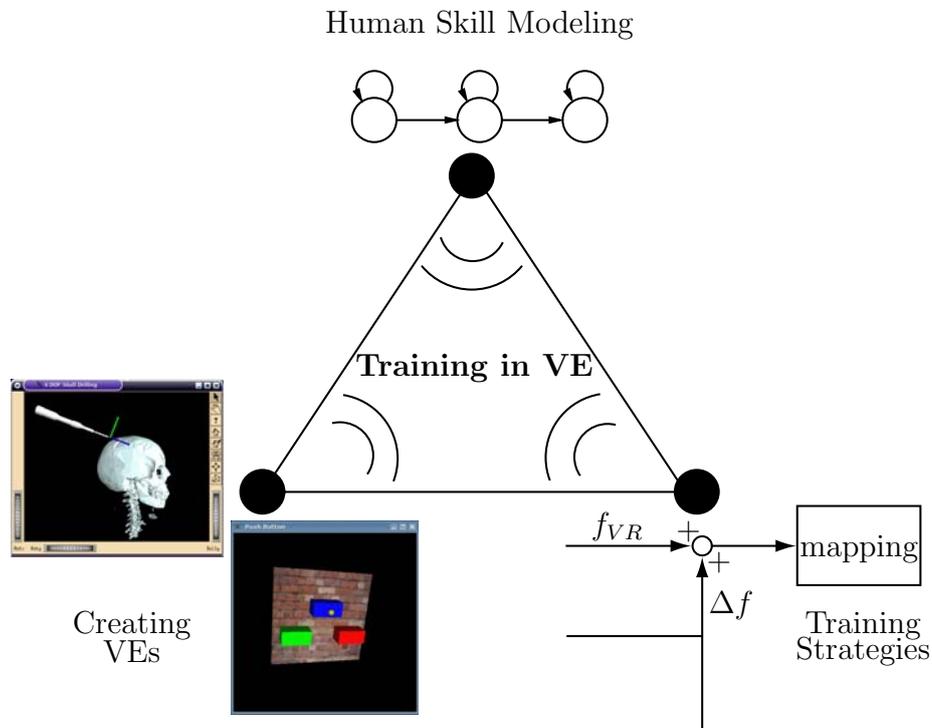


Figure 5.2: Integration of virtual environments, human skill modeling and training strategies into VR training system concept.

plished. The integration is depicted in Figure 5.2. This structure of VR training system offers a broad range of training possibilities from classical ones to unique techniques.

Several user tests are performed to examine the efficiency of this integrated structure. Using the objective measures explained above, effectiveness of each training strategy is indicated quantitatively. However, it is rather impossible to carry out user tests for all kind of combinations of the VR types, skill modeling methods and training strategies.

Consequently, we concluded to achieve and evaluate user tests on one training system, which is our BD-MTS, for all of the different training strategies listed in Chapter 4. By performing these user tests, our aim was to find the effectiveness of the different methods in comparison with each other and establish guidelines for further investigations on these strategies.

5.3 User Tests

Parallel to the developments in BD-MTS, user tests are performed with different intervals. The first group of tests is carried through for enhanced training strategies. This is followed by tests on multi-user methods, and finally on training systems including a hybrid dynamic trainer model. Each of these tests, their organization and results are individually discussed. Before that, some common issues concerning experimental conditions and evaluation criteria are reported.

5.3.1 Experimental Conditions

User Profile

Users are mostly students who study technical subjects. The age group is 20-30. Both male and female users take part in the tests, and gender effect is not investigated.

Appropriate to the training strategies, the participants are divided into several groups. A “control group” always exists, whose members do not acquire any training. Depending on a theoretical basis about the application and following a demonstration of the trainer, they try to realize the drilling task. This group’s results are needed in order to assess the competency of the training methods comparatively.

An expert user is present during the experiments. His/her task is to give theoretical information concerning the BD-MTS task realization, and how the training works before the beginning of the experiments. The expert user also demonstrates a correct drilling on BD-MTS before tests start. Finally, the expert defines and follows the course of the training sessions.

Sensorimotor Coordination Test

In order to prevent an unequal distribution of the naturally skilled or unskilled users into different training groups, applying sensorimotor coordination tests are essential. Thus, the people in each group have on average the same ability regarding sensorimotor skills.

As a test-bed, Vienna Test System from Schuhfried GmbH is used¹. The test person should correct the occurring deviations between the target value and the actual value of a gyroscopic virtual object via a joystick-like input device. This task is easy to explain and the participant’s ability can be only minimally influenced by previous experience.

¹<http://www.schuhfried.at/>

With this test two abilities of the participant are assessed: On one hand, the ability to maneuver an element to a pre-set goal, referring to the sensorimotor coordination ability; on the other hand, the ability to react to an element's unpredictable changes of direction and size, revealing the ability to anticipate movements. The test system is depicted in Figure 5.3.

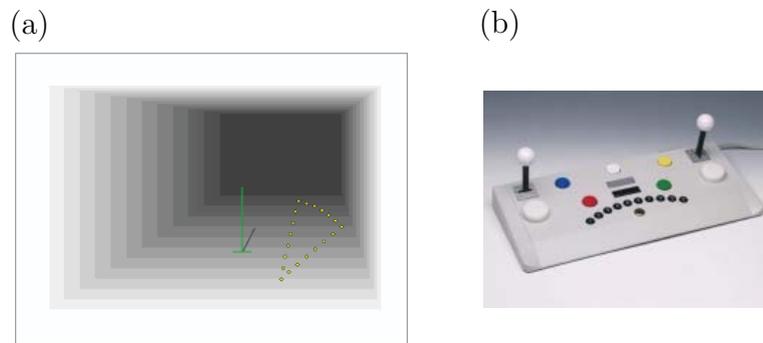


Figure 5.3: (a) Screenshot of the Sensomotor Coordination Test ©Schuhfried. (b) Input device of Vienna Test System ©Schuhfried.

Although Vienna Test System is a clear indicator for sensorimotor skills, it is time consuming to use it. The users have to finish this coordination task first, which takes ca. 20 min, then they start with the drilling training that requires ca. 1 – 1.5 hours. Since the long duration of the experiment causes concentration loss, as an alternative to the Vienna Test System, the drilling task itself is used as an indicator to group the users. The average of the first three experiments following the instructions of the expert, is used to have an idea on the student's skill. Consequently, they are assigned to the training groups.

General Procedure

Although there are slight dissimilarities between different training strategies, the general procedure of user tests can be summarized in 3 steps:

Step 1 - "Beginning Trials": The task is explained to the trainee, then an example drilling procedure is conducted by the expert user, allowing the trainee to visually observe what a correct drilling should look like. After that the trainee is asked to make trials depending on his/her observations.

Step 2 - "Training": One of the training strategies is applied to the user for a certain number of training sessions. Control group students skip this step.

Step 3 - "Checking": The student is asked to perform the drilling task himself/herself, i.e. the training strategy is turned off during these tests. The students repeat the task for a certain amount of times. Then, his/her success is checked with the pre-defined evaluation criteria. If successful, the training procedure is stopped, if not the student is sent back to the step-2.

During the "training" step, as an addition to the GUIs, the trainee is informed about his/her performance by Figure 5.4. It is presented to the trainee with periodic intervals.

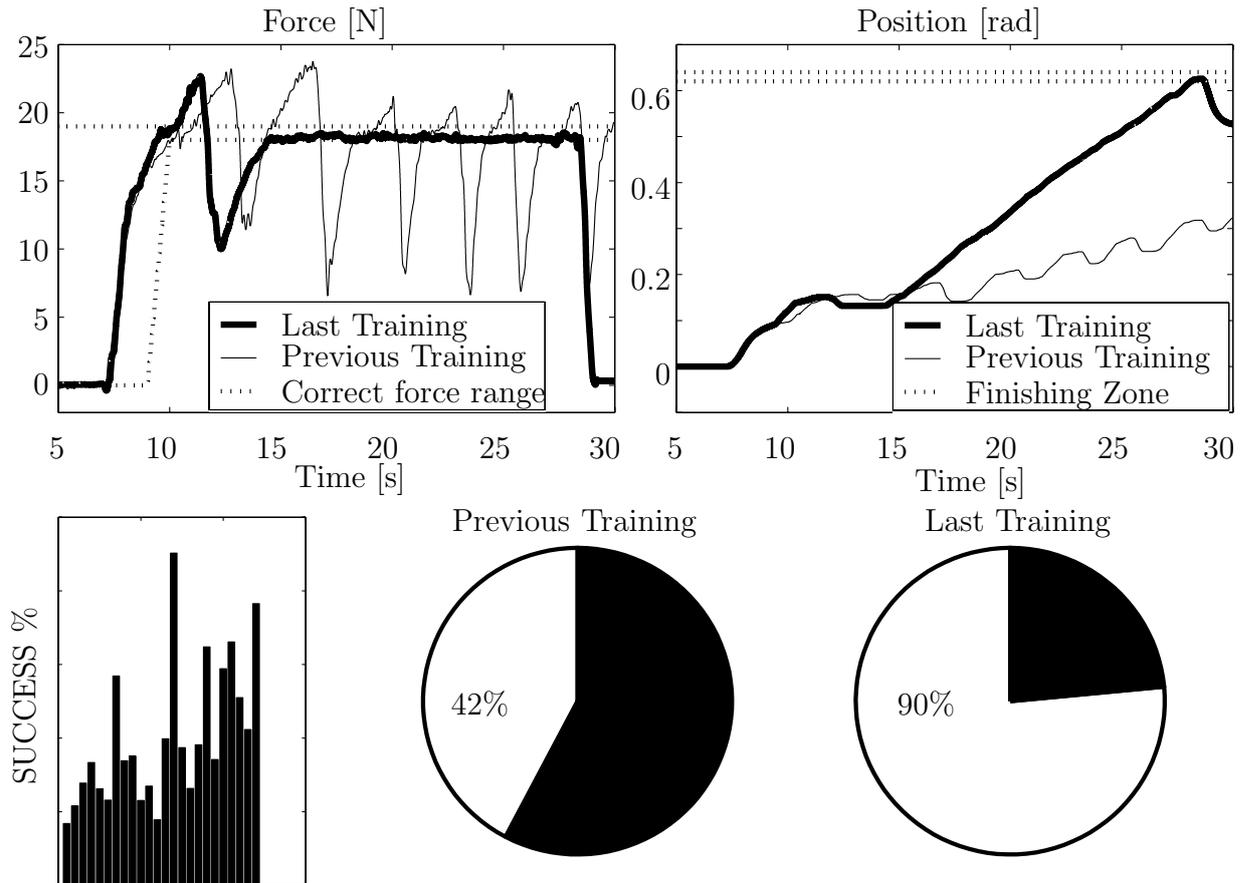


Figure 5.4: Graph presented to the trainees.

From the figure, he/she has a chance to see the comparison of forces from his/her previous and last trials. Resulting drill positions are also presented. A bar chart is used to show the trainee the history of all trials. It indicates the success percentages of the trainee from the first experiment to the last one.

Evaluation Criteria

The objective evaluation of the user tests' results has a crucial importance to determine the efficiency of the training process. Therefore, in this work neither a questionnaire to the users is applied, nor their subjective opinions are taken into account for evaluations. Our evaluation criterion depends on measuring the distance between the expert's performance and student's performance as Euclidean distance in n dimensional space as explained in Section 5.1.2. While the Euclidean distance method enables us to conduct a detailed analysis of the student's progress, a quick indicator is needed to decide whether to finish or continue training process.

Consequently, applied force of the student and the drilling end position during a trial on MTS are used as these quick indicators to decide how successful the training is. The percentage of applied correct force represents how close the performance was to the expert's performance. Applying 90% or more of the simulation time the correct force is accepted as the level of "skillful trainee". Additionally, if a trainee misses the point that he/she has to stop drilling, the trial is counted as a failure, i.e. he/she gets 0% success. The trainee

has limited time to finish the task, i.e. the simulation time is constrained. In terms of equation (5.3), the pre-defined skill level for BD-MTS is $L = 90\%$.

5.3.2 Tests on Enhanced Training Methods

In this section, visually, acoustically and haptically enhanced training applications on BD-MTS and their results are discussed. Note that except the haptic enhancement, the other enhancements are only available during the “training step”. To check the skill improvement, they are turned off, and the student’s performance is observed.

User Tests with Haptic and Visual Enhancements, using 1-DoF Haptic Device

Early tests are carried out by using pedal like haptic display as the input device. Both visual and haptic enhancements are provided to the users. For the experiments, taking the device capabilities into account, applicable force range is set to 17 – 19 N and resulting thrust velocity is set to ca. 1 cm/s. The drilling task is completed when the measured angle of the device is between 0.62 and 0.64 rad.

The “skillful trainee level” $L = 90\%$ is determined by taking the defined narrow force band (17 – 19 N) into account. It is experimentally observed that, e.g. hand tremors of the trainee may cause small oscillations in the force, which may cause the applied force to be out of pre-defined range up to app. 10%. It is known that there are auxiliary systems available that eliminate the hand tremor of a surgeon during a surgery. If a trainee misses the point that he/she has to stop drilling, the trial is counted as a failure, i.e. $L_{st} = 0\%$. The trainee has 25 seconds to finish the bone drilling task. If he/she completes the task successfully earlier, he/she stops applying the force. However, in this case, the results are evaluated as if he/she was applying the correct force during the remaining time.

The experiments are applied to 8 different users. The same skill improvement tendency is observed for all of the trainees. An example skill improvement of a trainee is shown in Figure 5.5. In the first trial the trainee has 17% success. Observations provide the trainee with a slight skill improvement, but he/she is still far from the “skillful trainee” level. During the training period the trainee’s skill improved gradually. After a period of training, the trainee manages to satisfy the requirement given in “*Step-3: Checking*”. In the last three trials, the success is 90%, 90% and 95%. The trainee needed 53 trials on BD-MTS to obtain this result. He/she had additional 22 trials, where the visual force indicator was in use. Because of the necessity of high concentration, whenever the trainee asked, a break was given between trials.

In Figure 5.6(a), applied forces during the first and last trials are compared. An example of the applied force from early training (trial no. 13) and another one from late training (trial no. 46) are also added to the figure. In the first trial, although the trainee increases the force slightly, he/she cannot reach the minimum required force for the drilling process. In the early training, the trainee manages to exert more than the minimum required force, but it is obvious that he/she does not have the correct haptic feeling yet. As a result, there are force oscillations with big amplitudes that continue till the end of the trial. In the late training, the trainee quickly understands if he/she is out of the correct force band. He/she reacts rapidly and returns to the correct band, i.e. keeps on drilling. There are still oscillations in the force, but the amplitude is small and the trainee manages to find back the correct force band. A “skillful trainee” applies sufficient, but non-excessive force

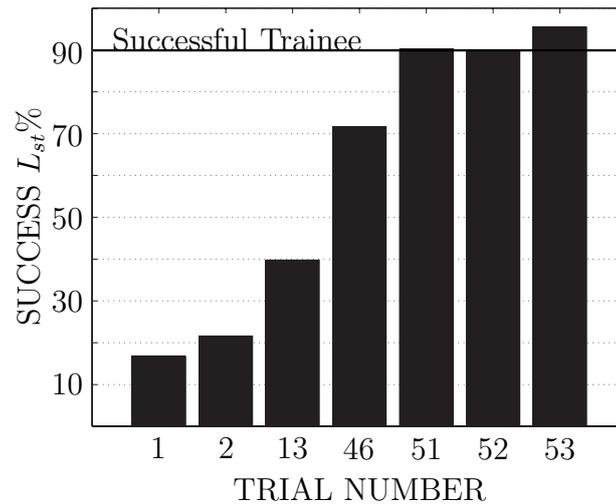


Figure 5.5: Skill improvement of a trainee in a haptically and visually enhanced training environment.

in the “beginning” period of drilling and then he/she keeps the force constant until the end of the drilling process. The last trial (#53) of the trainee is an example of this case.

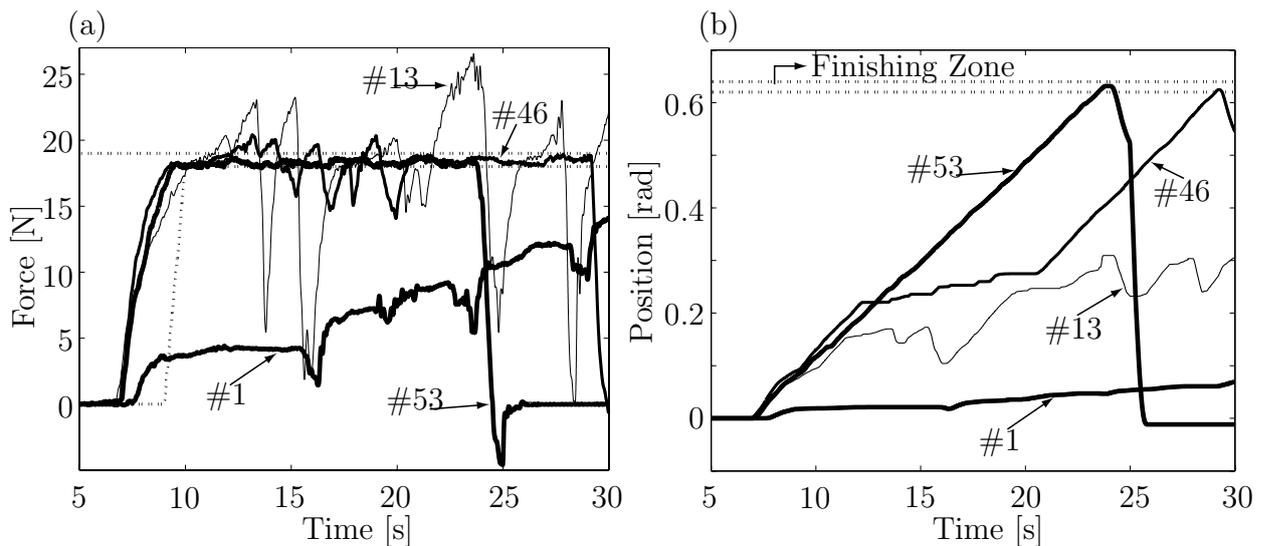


Figure 5.6: Comparison of the (a) applied forces and (b) resulting positions while training session is in progress. The first trial is indicated by #1 and the last one by #53.

The comparison of drill positions with respect to time is given in Figure 5.6(b). Appropriate to his/her applied force, in the first trial, the trainee is almost not able to move the drill into the bone at all. During early training, rarely does the trainee get into the correct force band and manage to drill only halfway. The late training indicates that the trainee is mostly applying the correct force. Therefore, he/she manages to finish drilling process. In the last trial, the trainee starts and completes the drilling task by reaching the drilling end-position successfully in 14.4 seconds. This result is very close to the performance of the expert user.

User Tests with Haptic, Visual and Acoustic Enhancements, using 3-DoF Haptic Device

The next version of BD-MTS uses 3-DoF haptic display ViSHaRD3 as input device, which not only provides more DoF, but also the possibility of larger applicable forces. Consequently, the applicable force range is set to 40 – 45 N. In addition to the haptic and visual enhancements, also acoustic enhancement is included to the training system.

In Figure 5.7(a), success percentages of 3 trainees versus the number of trials are sketched, when acoustic enhancement is turned off. $L = 90\%$ is emphasized by a dashed bold line. At first sight, it is possible to see that the success percentage of trainees increase

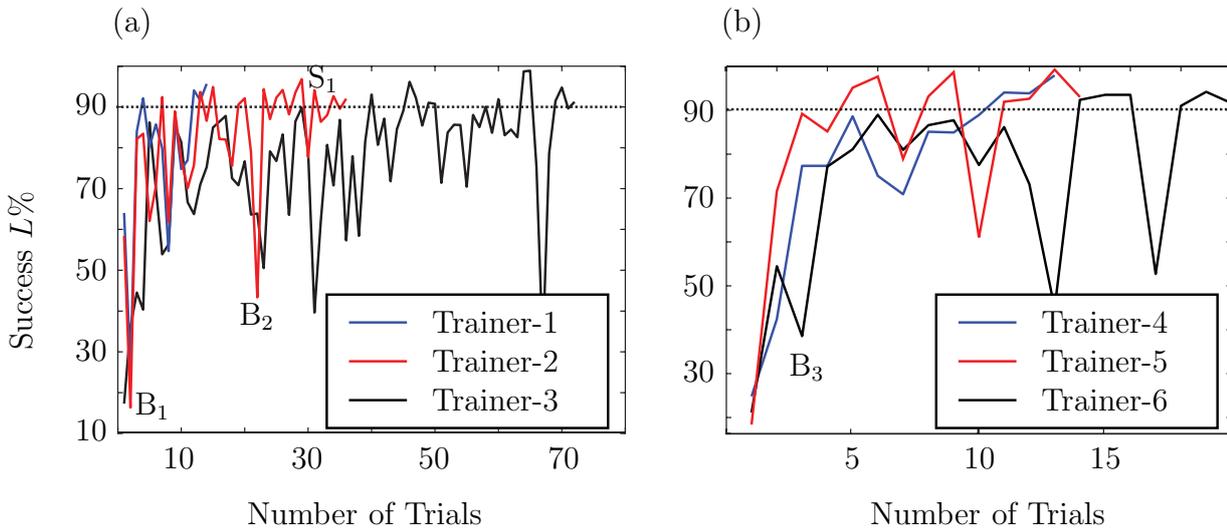


Figure 5.7: Skill improvement using 3-DOF BD-MTS: (a) haptic and visual enhancement, (b) haptic, visual and acoustic enhancement

with increased number of trials until they reach to the L_{st} . Another important information that can be obtained from Figure 5.7(a) is that the number of trials needed to reach the L_{st} varies for each trainee. That is a result of the natural skill of the trainee. It also depends on the concentration of the trainee during a drilling process.

User tests are expanded by adding the acoustic feedback to the MTS. Skill improvement of the trainees is checked when the acoustic feedback is on and results are sketched in Figure 5.7(b). Comparing the Figures 5.7(a) and 5.7(b) gives two important results: Firstly, in less than 20 trials all trainees are able to reach the L_{st} when the acoustic feedback is on. Therefore, it is possible to say that having acoustic feedback speeds up the skill training process. Additionally, the number of trials that are necessary to reach the L_{st} does not vary largely for each trainee. The second important result is that the trainee is able to reach the high success levels very quickly, i.e. in around 5 trials. This effect can be seen more clearly in Figure 5.8(b). A bad trial B_3 when acoustic feedback is on is compared with the bad trial B_1 when the acoustic signal is off. With sound, the trainee reacts much faster. Therefore, applied thrust force has smaller amplitudes around the correct force range.

To see the skill improvement more clearly, experimental results of trainee #2 are focused on and two bad trials of the trainee are compared with each other. The first bad trial B_1 is chosen from the “first trials” period and the second one B_2 from “last trials”. As it is shown in Figure 5.8(a), in trial B_1 , the trainee applies too large force, but cannot notice that. Therefore, first he/she reaches to a force around 100 N which is high above the maximum allowed force. Then the student decreases the force and after

that even exert larger forces. This tendency goes on until the end of the trial. In trial B2, although the trainee again cannot manage to find the correct force range, he/she produces oscillations around the correct range with much smaller amplitudes. It means, he/she knows whether the applied force is too large or too small and by changing it in small amounts, tries to get into the correct force range. A successful drilling trial S1 is also added to the Figure 5.8(a). It shows that in the last trials, the trainee applies correct force with a good timing, i.e. keeps the velocity constant and finishes the drilling in 16 seconds.

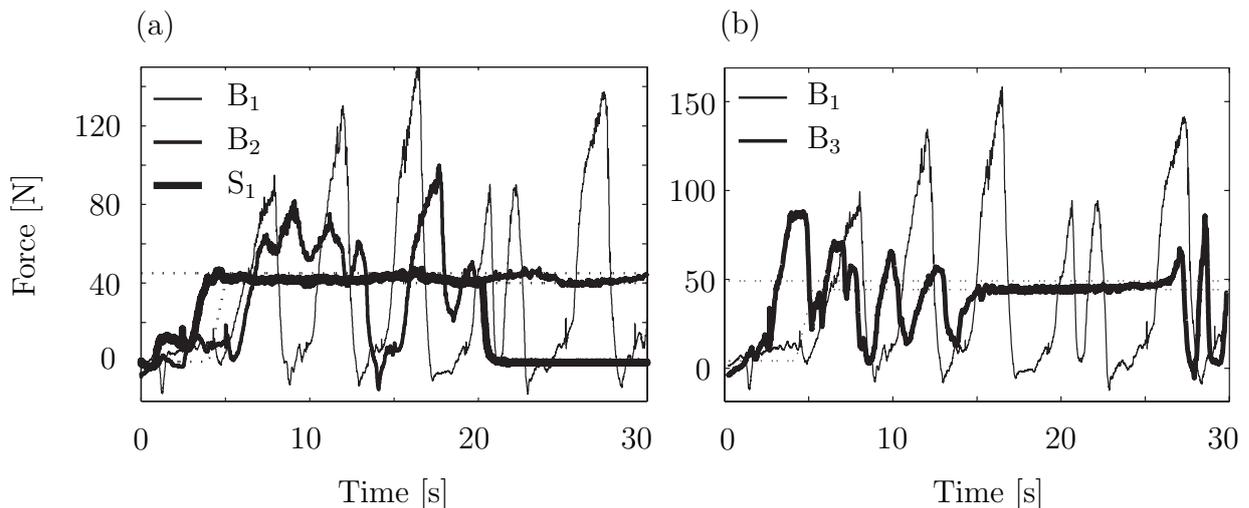


Figure 5.8: Comparison of bad and successful trials from Figure 5.7 (a) points B₁, B₂ and S₁ (b) points B₁ and B₃

Concluding Remarks

These user studies suggest that the proposed enhanced training strategies can be a powerful tool for teaching certain skills to students via VEs. It is indicated that additional visual indicators, i.e. visual enhancements, are helpful for teaching only temporarily. They are good for giving the first idea about the correctness of applied force and velocity, but after a while they cause the trainee to lose his/her concentration on haptic feedback. It is observed that adding acoustic feedback to the created VE increases the performance of users and accelerates the skill training process. A trainee who finished his training with sound feedback can also realize a trial when sound was off with a high success percentage. After trainee #4 reached the L_{st} 3 times sequentially, he/she was asked to go on making trials, but this time the acoustic feedback was turned off. The trainee performed 4 trials and in all of these trials managed to have more than 90% success, i.e. he/she was always above the L_{st} . It means that, in comparison to using visual enhancement, acoustic enhancement does not cause the trainee to lose his/her concentration on haptic feedback. Moreover, it accelerates the learning process of applying a correct and constant thrust force and thrust velocity.

It is observed that, because of the dominance of visual feedback, if trainee uses the visual enhancement more often, it causes his/her loss of concentration on haptic feedback. Therefore, visual enhancement is mostly used at the beginning period to get the first idea about the applied thrust force or in between unsuccessful checking trials, when the trainee wants to be sure about his/her mistakes.

5.3.3 Tests on Multi-User Training Methods

For the following user tests, ViSHaRD10 is used as input device. The applicable force range is set to 37 – 44 N. The thrust drilling velocity in this range is adjusted as ca. 1 cm/s through the equation (2.23).

First, a sensorimotor coordination test is applied to the trainees using Vienna Test System, and according to the results of this test trainees are distributed into 4 groups related to our training strategies: Control (C), Force/Velocity Leading (FVL), Force Demonstrating (FD), and Haptic Visual Observation (HVO) groups. Based on applied thrust force, a skillful trainer level is defined in order to have a quick check on student’s skill improvement.

C-group trainees start performing the drilling task directly by taking the explanations and expert’s demonstration into account. No training is applied. The members of FD, FVL and HVO groups are trained 5 times using the corresponding multi-user training method. Then they are asked to perform the drilling task alone. If a trainee shows a bad performance in drilling consecutively, the trainer repeats the training sessions. If the trainee reaches to the successful trainee level 3 times sequentially, the training experiments are completed. The drilling should be finished in 60 s, otherwise the training is accepted as unsuccessful.

The multi-user training structure depicted in Figure 4.3 is extended with one more computer. On this extra computer, the student’s performance is visualized only for the trainer. It is needed in order to standardize trainer’s influence for each student.

Euclidean Space for Evaluation

The dimension n of the Euclidean evaluation space is defined by the number of important parameters characterizing the interaction with BD-MTS. Force, velocity and time are three main features that are measured during the interactions and play an important role in the success of training. Different parameters can be derived from these three features. In Table 5.1, a list of possible parameters is given.

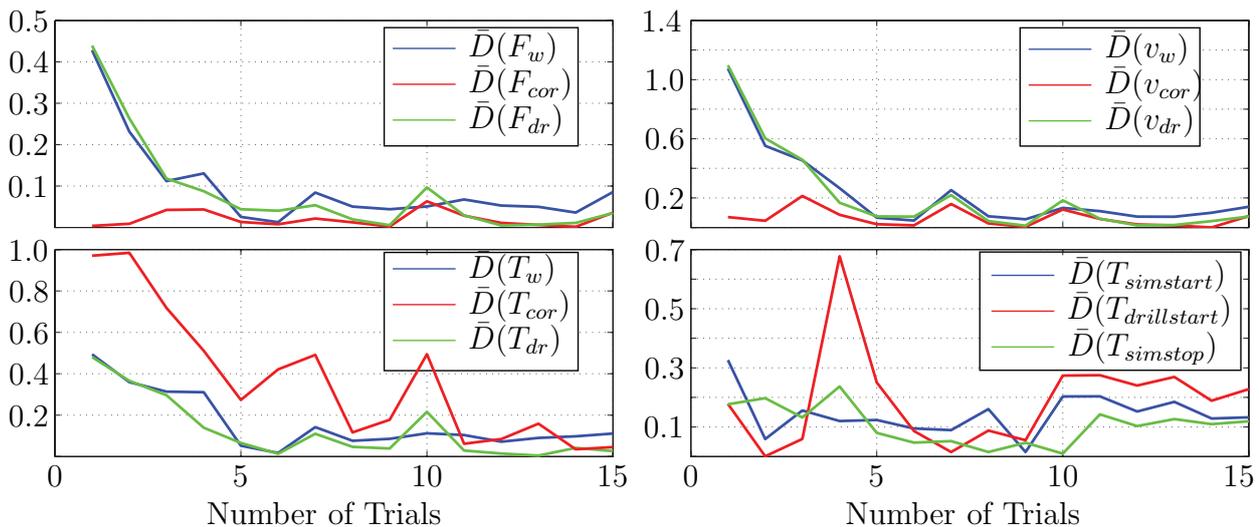


Figure 5.9: Parameter distances between the expert and the student

At first, the values of the parameters in the Table 5.1 are calculated from the trainer’s performance. Average of different experts performances is used to figure out these parameters. Following that, the same parameters are defined for each trial of every students.

Table 5.1: Possible parameters to define the Euclidean evaluation space.

Feature	Parameter	Symbol
FORCE	mean value for whole duration	\bar{F}_w
	mean value for correct forces	\bar{F}_{cor}
	mean value for drilling period	\bar{F}_{dr}
VELOCITY	mean value for whole duration	\bar{v}_w
	mean value for correct forces	\bar{v}_{cor}
	mean value for drilling period	\bar{v}_{dr}
TIME	whole duration	T_w
	duration of correct forces	T_{cor}
	duration of drilling period	T_{dr}
	simulation start time	$T_{simstart}$
	drilling start time	$T_{drillstart}$
	simulation stop time	$T_{simstop}$

Consequently, the change in 1D Euclidean distance for each parameter is observed for individual students. This leads to distinguishing the parameters that play an important role from the others. Figure 5.9 represents obtained results for one of the students. Note that similar results are observed for other students. The distances are calculated by using equation (5.8), not only for the first trial but also for the rest of the trials.

A clear indication for the importance of a parameter is the normalized distance to the expert in the first trial: Higher first trial value of a certain distance indicates a more important parameter, since at this point student is furthest to the trainer. In other words, there is a potential of improvement for this point. On the contrary, low values underline that at this parameter student is from the beginning on close to the trainer, so it should not be expected to see any improvement on this parameter throughout trainings. Another important indicator is the constancy or unexpected changes in a certain distance. This effect also underlines a parameter that does not have a significant role in training process.

Following this analysis, the Euclidean space is defined as $n = 7$ dimensional, and parameter vector is defined as:

$$\mathbf{p} = [\bar{F}_w \quad \bar{F}_{dr} \quad \bar{v}_w \quad \bar{v}_{dr} \quad T_w \quad T_{cor} \quad T_{dr}]. \quad (5.13)$$

Via equation (5.9), the weighting factors for the Euclidean distances are computed as given in Table 5.2.

Evaluation Results

The user tests are evaluated using the weighted n dimensional Euclidean distance calculation given in equation (5.10). Figure 5.10 exhibits distance samples that belong to different training groups. As it can be seen from the Figure 5.10(a), control group students have relatively large initial distances, and there are cases that they cannot manage to achieve

Table 5.2: Parameters of the Euclidean evaluation space and their weighting factors

Parameter	Weight w_i
\bar{F}_w	0.0801
\bar{F}_{dr}	0.0774
\bar{v}_w	0.1669
\bar{v}_{dr}	0.1670
T_w	0.1140
T_{cor}	0.2845
T_{dr}	0.1099

performing a good skill, even when maximum number of trials (40) is reached. Figure 5.10(b) clearly attests that HVO-group members' distance to the expert quickly decreases. Even first trials after the training are not far from expert's performance. FD-training results are presented in Figure 5.10(c). Although not as rapidly as in HVO, the students' skill converge to the trainer's skill after regular training sessions. Last but not least, results obtained from FVL group students are depicted in Figure 5.10(d). Still better than C-group members, the skill improvement for FVL students is slower than for students who are trained via HVO and FD methods. An important case for the FVL methods is that the trainer should improve his/her way of applying the methods as well, should adapt himself/herself. The second example in Figure 5.10(d) is obtained when the trainer is changing the velocity of the student very rapidly. As a result, student is not able to react properly and understand the relation between the force and speed. Thus, this is an example, which should be referred to as trainer mistake, or expressed differently, "trainer's training". Consecutively, results of this test are excluded from further evaluation of the methods.

A general look at the Figure 5.10 as a whole, allows to come to conclusion that similar characteristics of distance change is recognizable. In general, first trial distances are very high, whereas middle of the trainings the distances are on an average level, and finally, towards the end of the trainings, the distance is very low, i.e the student is very close to the trainer.

Note that during the multi-user training, following the expert's demonstration at the beginning, the training sessions start directly. Therefore, the skill improvement curves shown in Figure 5.10 do not include any trials before the training sessions. This appeared to weaken the analysis of the results, because it is not possible to evaluate what the first training session would bring to the student. Another omission was irregular training periods. Only the first training sessions were equal for all trainees (5 trials), but then the course of the training was defined by the trainer, appropriately to the performance of the student. Both deficiencies are corrected during the user tests performed for "training with a hybrid trainer" strategies.

As it can be seen from the Figure 5.11, in the first trial after the first training session, the training group members already perform better than C-group members. Although the students are not able to stay in the correct force range, which is showed by dotted lines,

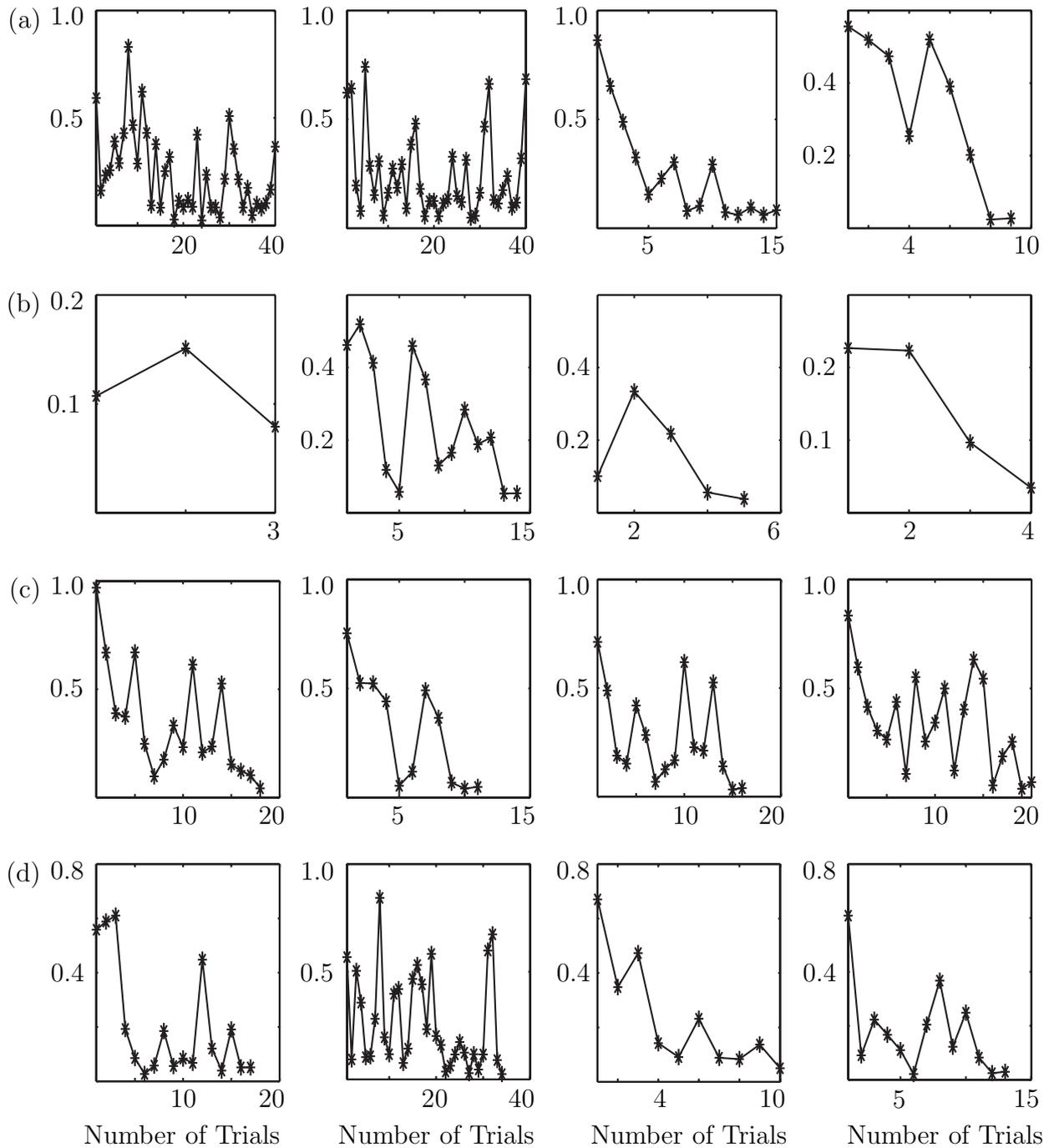


Figure 5.10: Example evaluation results. (a) Control group examples (b) HVO group examples (c) FD group examples (d) FVL group examples.

they are in a close vicinity to it. It is remarkable how HVO group member realizes the task skillfully. This result matches with the results given in Figure 5.10 that is very low first distance values of HVO trainers in comparison with other training group members.

Figure 5.12 shows the sorted success graph of trainees. Trainee #1 is the student, who showed the worst performance for each group. This figure also clearly indicates that all training group members perform better than C-Group, whereas the best performance is obtained after HVO-training. Early experimental results suggest that the multi-user

training methods discussed in this paper improve the learning process. The HVO method is the most promising one, whereas FVL method is quite complex and students need longer time to understand it.

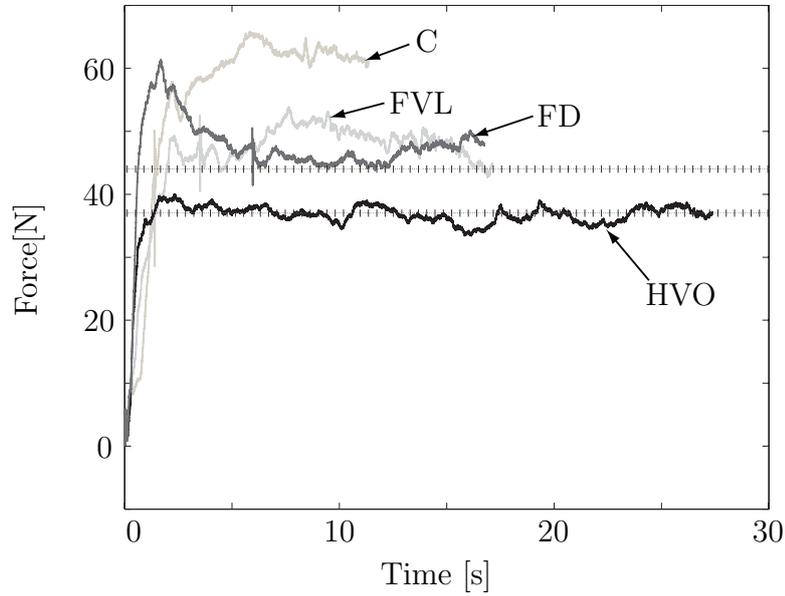


Figure 5.11: Comparison of the first trials after the first training session.

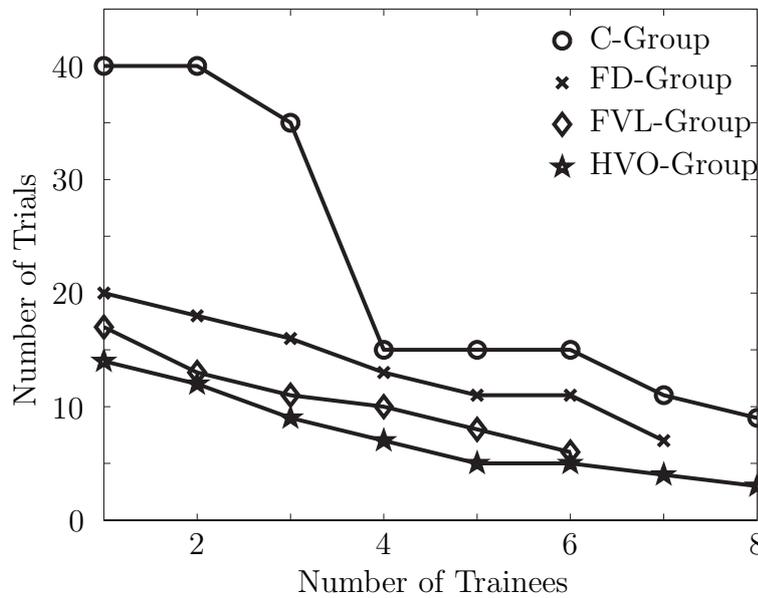


Figure 5.12: Number of trials needed by trainees to succeed in the drilling task.

5.3.4 Tests on Training via a Hybrid Trainer Methods

The experimental conditions that are set for multi-user trainings are untouched. Namely, the applicable force range is 37 – 44 N, for this range drilling velocity is 1 cm/s, and maximum allowed duration of one trial is 60 s.

Instead of Vienna Test System, first trials of the students are used to check the sensorimotor coordination skill. Consequently, the students are divided into 4 groups: C, FVL, FD and HVO groups. Since the C-group here and in multi-user tests is exactly the same, just a few users are assigned to C-group. The rest are distributed to FD, FVL and HVO groups.

The Euclidean evaluation space is also the same as in multi-user case. Thus, it is 7 dimensional and contains the same parameters as in (5.13). For the weighting factors, refer to Table 5.2.

Evaluation Results

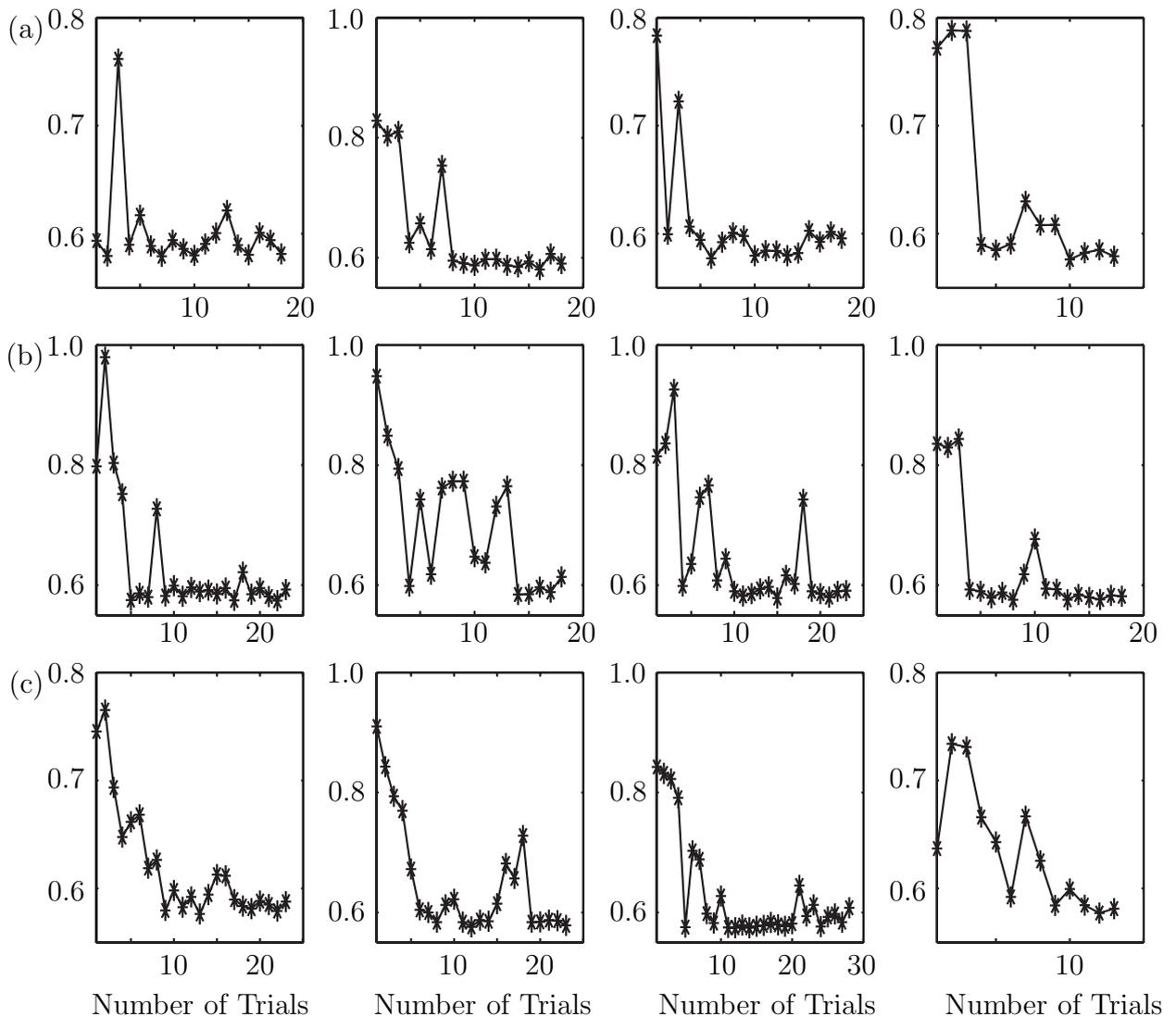


Figure 5.13: Example evaluation results while hybrid trainer model influences students. (a) HVO group examples (b) FD group examples (c) FVL group examples.

For being able to compare the methods with each other, trainings are applied with the same periods. The user tests start with 3 trials without any training, which is followed by 6 training - 6 checking period. From the checking tests, the worst trial is neglected, and the average success rate L_{avg} for the rest 5 is checked. If $L_{avg} = 80\%$ or more, the training is finished after the 27th trial. If not, the period is set as 3 – 6, and the trainings continue until the pre-defined skill level is satisfied. Maximum number of trials is 54, i.e. independent of the student's success, the trainings are finished at the 54th trial.

In Figure 5.13(a), (b) and (c) example student performances, i.e. change in Euclidean distance between the trainer and the student throughout the training sessions, for HVO, FD and FVL training methods is given respectively. General tendency is that the distance D_{ts} gets smaller, when the student proceeds with trainings.

Since the training is applied to the students periodically and with the same period for each, checking the results after certain training sessions allows to interpret the efficiency of the methods comparatively. In Figure 5.14, average skill improvement of the HVO, FD and FVL group members is displayed, and the ranges belonging to before training and after the 1st, 2nd, 3rd and 4th training sessions are highlighted. For each highlighted part, the mean value of the distance for all trials and for all trainees is used. After the 2nd session, if any student reaches the desired L_{avg} , then the experiments are finished for him/her. Thus, the distances D_{ts} for this student after the 3rd and 4th are taken as zero. The best improvement in the skill can be observed after first training session for all cases. FVL keeps on improving the skill steadily, whereas the elevation brought by FD and HVO is rather irregular. The 2nd training session does not provide any improvement as far as FD is concerned. HVO's contribution in this session is less than FVL's. However, FD and HVO students reach to desired L_{avg} faster. Consequentially, in the Figure 5.14, rapid improvements can be seen after the 3rd and 4th sessions.

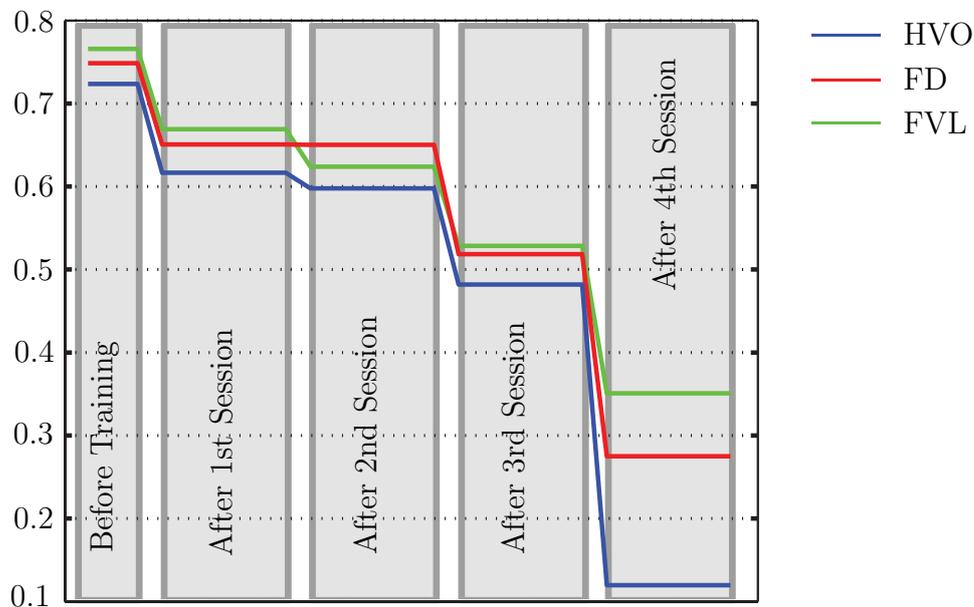


Figure 5.14: Average skill improvement for different training cases

It is also interesting to see how students change the force they apply after trainings. This is illustrated for HVO, FD and FVL in Figures 5.15, 5.16 and 5.17 respectively. In all of these figures, (a) shows the trials before training session, (b) displays results from

trials after beginning training sessions, and (c) exhibits applied forces after late training sessions. Pre-defined correct force range is indicated as black lines.

Figure 5.15(a) clearly shows that before any training, the student is very far from the skillful force range. First HVO trainings bring him/her quickly around the acceptable for range, but he/she cannot repeatedly stay in this range as indicated in Figure 5.15(b). However, after last trainings, the student stays in the correct range in all trials.

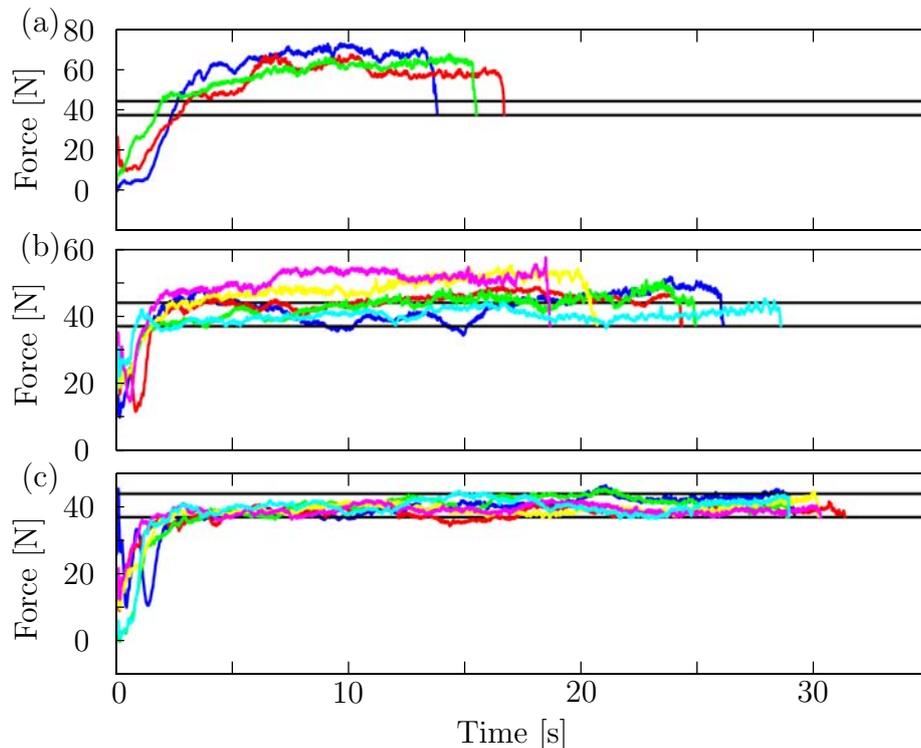


Figure 5.15: Skill improvement via hybrid trainer model and HVO training strategy (a) Before trainings (b) After first training sessions (c) After late training sessions.

The same analysis is done for FD training and the results are depicted in Figure 5.16(a). They yield similar characteristics as in HVO case. Bad performances before training are improved after first trainings (5.16(b)), and finally late trainings enable the student to stay in the correct range. One can see from the comparison of Figures 5.15 and 5.16 that trials of the HVO students are closer to each other, i.e. they are around a mean value with small variances. However, the trials of FD students are more distributed, i.e. have larger variances.

The same examination is conducted on the trials of a FVL student. As Figure 5.17(a) indicates, this student cannot apply the correct force before trainings. However, he/she is closer to the correct range in comparison with HVO and FD students. It is possible to deduce that this student has a better sensorimotor skill than others discussed above. Similarly, after first trainings, the student manages to get into correct force range. Adversely, he/she does not deviate from the correct force range a lot, although still the skillful level is not reached (5.17(b)). Last trainings enable the student to stay in the correct range constantly (5.17(c)).

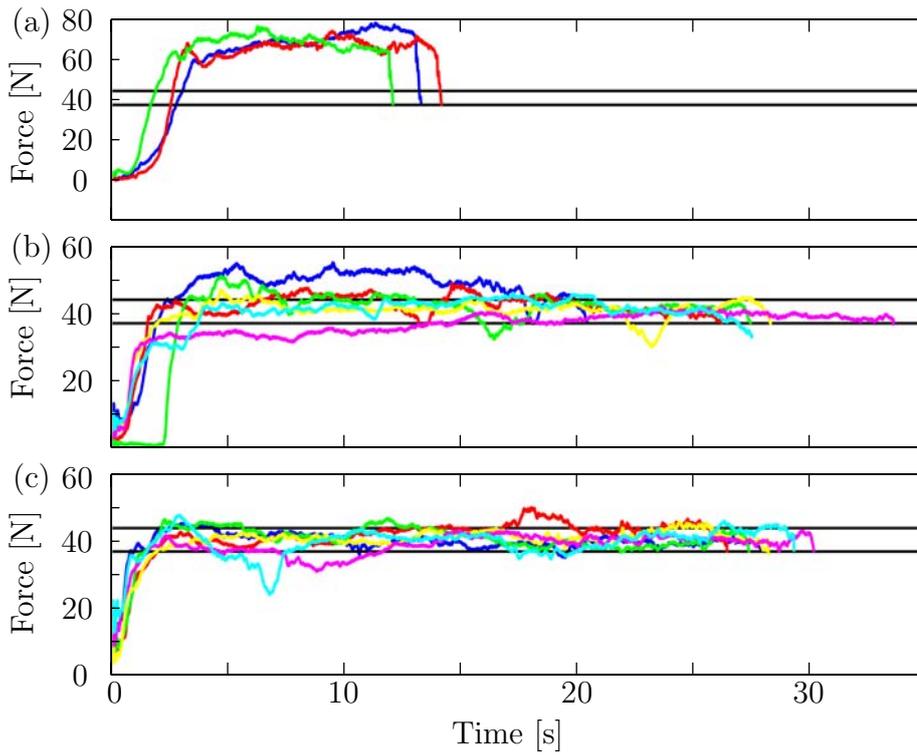


Figure 5.16: Skill improvement via hybrid trainer model and FD training strategy (a) Before trainings (b) After first training sessions (c) After late training sessions.

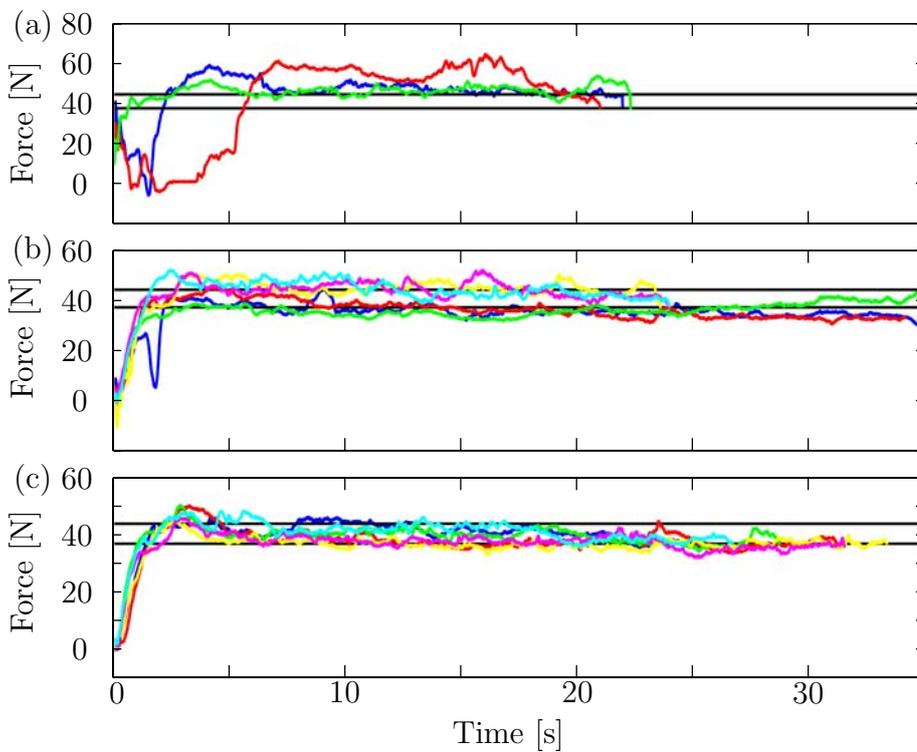


Figure 5.17: Skill improvement via hybrid trainer model and FVL training strategy (a) Before trainings (b) After first training sessions (c) After late training sessions.

Distance Between Impedance Parameters

Another way to assess the training results is to check the progress of the impedance model parameters. It is expected that throughout the training process, the student learns to realize the task like the trainer would do. If the last trials of the student are clustered and modeled as switching impedances, the identified impedance parameters should be similar to the trainer parameters.

However, a distance check between parameters via equation (5.12) introduces some problems. First of all, the first trials of the student are naturally very far from trainer's performance. If such a performance is clustered, resulting clusters and switchings between them are different from expert results. Thus, comparing the impedance parameters for the clusters that do not coincide does not bring any logical conclusion. It is possible that the clusters are different, but model parameters are similar. Undoubtedly, this would lead to a misinterpretation of the student's success, if the distance between model parameters is checked. Another problem is that the impedance parameters in one cluster have different levels of importance. Consequently, they have to be weighted, before the distance is measured.

It is more convenient to identify the model parameters for the students only for their last trials. This does not only provide comparable results, but also saves time. Remember that weighted K-means clustering process and following parameter identification need around one day to be accomplished for one trial. Applying it to a series of trials for only one user needs around one month computing time. Moreover, it is not difficult to presume that this identification process would not lead to any concrete result.

In Figure 5.18 identified damping parameters of the impedance models for three different students who are trained with HVO, FD and FVL methods are shown. Note that the trials that are used for identification are performed by students after the last training session. If the parameters are compared with the expert parameters that were given in Figure 3.8, it is deduced that the trainings bring the impedance parameters of the student interaction closer to the expert interaction. Similar tendency is observed for spring and mass parameters.

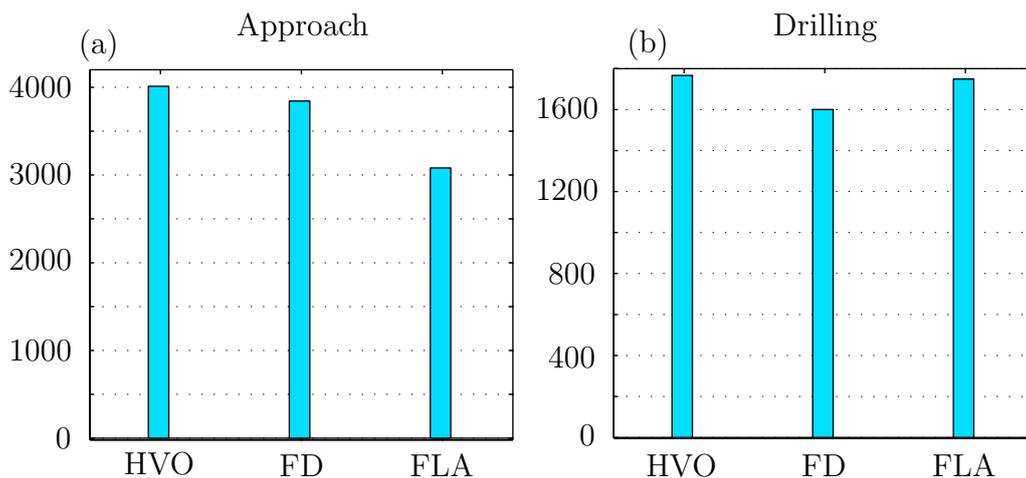


Figure 5.18: Different student trials after last training session and resulting damping d coefficient for (a) approach and (b) drilling phases.

It is also possible not to apply any clustering to the student trials, but define the clusters and switching points as close as possible to the expert cluster and switching points. This

allows us to see the progress in student parameters at a glance. However, such an analysis is not very reliable. The main problem is that in the beginning trials, it is not possible to set switching points like that of the expert, as they simply do not exist. For example, the expert switches from approach case to the drilling case when he/she reaches the correct force and velocity and keeps them constant. Such a performance is not observable in the beginning trials. Without getting into details, it is possible to say that such an analysis indicates much larger spring coefficients in the approach phase during the first trials. Offset values vary largely. Damping is mostly the same, rarely larger values observed. In the drilling phase, the most distinguishing parameter is offset. This is kind of expected as long as the student applies larger or smaller, but constant forces.

5.4 Discussion

This chapter joined VR development principles, trainer skill modeling and training strategies together, in order to accomplish our training system concept. Several user tests were performed in BD-MTS to evaluate the developed concept. An objective evaluation metric based on n dimensional Euclidean distances between the trainer and the student was introduced and used to assess the skill improvement of the students.

Haptic enhancement strategies provide a natural way of informing the students about the force they apply. Our haptic enhancement not only restricts the movements when the student tends to leave the correct trajectory, but also stops the student completely if the applied force is not in the expected range. Unless the student corrects himself/herself, the simulation cannot be continued. The results of the user tests state that when haptic enhancement is combined with visual enhancement, the trainee realizes the task successfully paying attention only to the visual information. It is observed that because of the dominance of visual cues, frequent usage of the visual enhancement causes the loss of concentration on haptic feedback. As a result, when the visual enhancement is turned off, the trainee has difficulties to demonstrate the expected force skill. Therefore, the trainee improves the force skill, when only haptic enhancement is available, because he/she concentrates only on the haptic cues and force to apply. A drawback of the haptic enhancement is that the only haptic information the student receives is that the force is wrong or correct. The quantity of the force is not available. Therefore, visually enhanced trainings should be considered as auxiliary systems to inform students about the task temporarily and give the first idea about the correct force range. Then the trainings should be performed only relying on haptic enhancement as long as the student feels confident. In our work, visual enhancement was mostly used at the beginning period to provide the first idea about the applied thrust force, or in between unsuccessful checking trials, when the trainee wanted to be sure about his/her mistakes. As Ernst *et al.* underscore, if the visual information is less reliable to realize a task, then haptic cues play a more important role in motor skill learning procedures [29]. Depending on this statement and the results of our user tests, we conclude that if the actual task bears mainly haptic cues, then the enhanced VR training should have only extra haptic features, not visual ones. It is observed that adding acoustic feedback increases the performance of the users and accelerates the skill training process. A trainee, who finished the training with sound feedback, could also realize a trial successfully when sound was off. It is discovered that in comparison to using visual enhancement,

acoustic enhancement does not cause the trainee to lose his/her concentration on haptic feedback.

The results of the user tests on multi-user training methods indicate that all methods, i.e. HVO, FD, FVL, speed up the convergence of the skill of the student to that of the trainer. The n dimensional Euclidean assessment was a clear indicator for this statement. It appeared that HVO was the most, and the FVL was the least efficient method. The principal reason for this situation is the clarity of the verbal training signals in HVO method. However, the correction signals in FVL method augment or decrease the resistance that the student feels. Thus, the student is the one to deduce at which points he/she was imprecise and what to correct. Nevertheless, it must be underlined that FVL method presents an important and unique feature, which is the possibility of teaching the velocity and force on a trajectory to the student synchronously using only haptic signals.

The results obtained from trainings via a hybrid dynamic trainer model establish similar characteristics as in multi-user tests. All approaches bring the student in Euclidean evaluation space close to the trainer. However, none of the methods is superior to the others, the results are comparable. FVL method bears the same drawback as mentioned for multi-user case above. HVO method is not as efficient as multi-user HVO. That is because of the fact that the HVO method with trainer model provides visual and discrete warnings about the force that student applies. Consequently, the method inherits the drawbacks of the visually enhanced trainings. Instead of visual warnings, using acoustic warnings might be a solution for this problem, since they can be interpreted as verbal warnings, similar to the warnings in multi-user HVO. The last performances of the students from each group were modeled as switching impedances using WKM clustering method. It was found out that the identified impedance parameters of the students were in the vicinity of the parameters that were identified for expert trials.

A global comparison of the results of the user tests illustrates that the most efficient training method is multi-user HVO. This method eliminates the drawbacks of classical training methodologies. It enables the student to actively realize the task and receive verbal warnings of a trainer, who observes his/her performance not only visually, but also haptically.

As a concluding remark it is worth noting that since numerous training methods were introduced, it has not been feasible to perform detailed and systematic user tests on these methods. The results rather give the preliminary idea on the efficiency and applicability of the training methods. It is undoubtedly useful to extend the application examples and apply systematic user tests with large number of participants.

6 Conclusions and Future Work

6.1 Concluding Remarks

Undoubtedly, the best way to learn a specific task that requires high sensorimotor skills, is to carry through the task itself. However, many tasks, such as flying a plane, making a surgical operation or assuring safety in a nuclear plant, need an adequate training before the real task is carried out. In order to satisfy this need, simulators that replicate the actual tasks and their operational conditions as realistically as possible have been developed and produced since the beginning of the 20th century. Last two decades, following the advances in computer and electronic technologies, virtual reality (VR) simulators that provide high-fidelity visual, acoustic and haptic feedback have been constructed. While many researchers have been working on increasing the realism of such virtual environments (VEs), less attention has been paid to the training procedure itself.

This thesis describes a VR training system concept that not only includes realistic interactions with VEs, but also integrates the trainer or the model of the trainer in the control loop paying special attention to the training strategies. The main goal was to develop generic force skill training methods. In order to bring such a VR training concept into life, three main tasks were accomplished. First of all, fundamental VR engineering problems, such as collision detection and haptic rendering, were investigated and different VR training scenarios were generated. Secondly, the methods to capture sensorimotor skill of a trainer via hybrid identification methods were researched and applied to interaction of experts with developed VR scenarios. On top of that, force skill training strategies in VEs were widely explored. Results of these investigations were joined together in order to create the complete VR training concept. User tests were organized and performed. Using objective metrics, the results of the user tests were assessed. With respect to these research focal points, the main contributions of the dissertation are outlined below.

Creating Realistic Virtual Environments

Several VR applications were developed in the frame of this thesis. These applications are mostly benchmark tests, allowing to examine our training strategies and haptic hardware in our laboratories. While developing the bone drilling medical training system (BD-MTS), some fundamental VR engineering problems were solved, such as ray-based collision detection between polygonal virtual objects, axis-aligned bounding boxes and binary search trees to speed up the collision detection. In this application, haptic rendering relates the force applied by the student to the velocity of the drilling via a non-linear equation, which is given in the biomechanics literature. Depending on the two-port network theory, it was assured that maintenance of the unconditional stability of the interaction via virtual coupling principles was possible.

Special attention was paid to multi-user VEs. A concept that minimizes the communication delay by exchanging the haptic rendering data directly between two real-time (RT) computers was proposed and applied. Maximum delay measured in this structure was 1

ms. Two different scenarios, “tug-of-war” and “load transportation” were developed. They enable two persons to communicate with each other through a virtual object haptically.

A deformable box depending on the mass-spring network was implemented. In another application, FEM was used in order to realize the object deformations. Unconventionally, we aimed at deforming the avatar, i.e. the virtual tool. This work presents some fundamental issues of a bone sawing simulator for further investigations.

Capturing Trainer Skill via Hybrid Identification Methods

In this work, it was investigated to represent the trainer skill as a hybrid dynamical system. The discrete decision making and the continuous task realization of the human being suggest the success of such an approach. In general, there are four unknowns of a hybrid identification problem: Number of discrete states s , switching conditions sw_{ij} , model parameters θ_i and model orders n , where $i, j = 1, 2, \dots, s$. Following the works in the 80s and the 90s concerning human skill modeling for programming by demonstration applications, generalized impedance models were sought for each discrete state in this thesis. Therewith, the model orders were fixed and number of unknowns was reduced to three.

Our target was to have a generic method, which identifies all unknowns for any application deterministically for given input and output data of the system. Two different hybrid identification methods were successfully used for this purpose. The first method (WKM) depends on data clustering by weighted K-means algorithm, whereas the second method (SS-Imp) enables to define deterministic model parameters via stochastic switching principles based on hidden Markov models.

In order to perform WKM clustering, s has to be known beforehand. We have introduced an error analysis, which enables to define s in a systematic way. The clustering is performed several times starting from $s = 2$ and then increasing s each time by one. After clustering, LS optimization is applied for each subset and generalized impedance model parameters are identified. Following that, sum of absolute hybrid model errors is calculated. Increasing the number of the states, decreases the model error. Thus, if the desired model error is defined in advance as a threshold, it is possible to find the minimum necessary s to reach this threshold. The models that satisfy the error threshold should be further investigated by the intensity analysis introduced in this work. It is expected that the data points that are close to each other on time axis belong to the same cluster. Thus, for the suggested intensity analysis data windows are used, each of which has data points that are consecutive in time. In each data window, the distribution of the data points to the clusters is checked. The cluster that contains the most of the data points, i.e. most intensive cluster, is then taken as the cluster representing all data points in this data window. Applying the intensity analysis leads to defining the final number of clusters and physically interpretable switchings between these clusters. With these extra analyses, WKM clustering and LS optimization provide a reliable hybrid identification. However, WKM method yields two important drawbacks. Since it is necessary to divide the data points into local data sets before the clustering, it is necessary to choose the parameter c that is the number of data points in one local data set. Higher number of c enables to identify a parameter vector that represents the mode it belongs to properly. However, at the same time, it increases the risk of having mixed data points, i.e. points from different modes, in one local data set. For each new c to try, the whole clustering process has to be re-started. Second hindrance is the computing time. Creating local data sets is very time consuming and related to the number of data points geometrically. If the number of data points increases with a rate

g , the time needed for creating local data sets increases with g^2 . The intensity analysis introduced in this chapter is also partly a remedy for the problems connected to choice of c , since it allows to sort out the mixed data points.

SS-Imp method allows to assign a deterministic function to each state of a hidden Markov model (HMM) and to define the observation probability distribution \mathbf{b} as a product of the function parameters. Consequently, HMM iteration does not only provide maximum likely switching conditions between the states, but also deterministic model parameters. The biggest advantage of this method is its speed of convergence. To identify a hybrid model for BD-MTS, the WKM method needed around 12 hours, whereas SS-Imp only a few minutes. An important drawback of the method is that if the classical way is used to define the end of iterations and switching conditions, i.e. maximum likelihood and Viterbi algorithm, then the method highly depends on the initialization of the HMM λ , it is especially sensitive to the initial variance σ and model parameters θ . Assigning these parameters randomly causes HMM not to converge to the correct model parameters and switching conditions. Using the fact that deterministic models for the states were identified, we introduced an algorithm to define the number of iterations and switching conditions between the discrete states depending on the minimum model error. First, the HMM iterations are carried on, until at least one of the states satisfies the pre-defined, maximum allowed model error. Then, at time point k , it is switched to the state that exposes the minimum error. Following that, the switchings between states are filtered using the intensity analysis. Via this algorithm, the convergence to the model parameters representing each discrete state with randomly chosen initial conditions is achieved. A disadvantage of this model error-based approach is that it might end up with switching between clusters, which is actually not possible. Therefore, resulting state sequence has to be checked by taking the transition probabilities matrix \mathbf{A}_n into account. Alternatively, it is possible to apply logarithmic Viterbi algorithm to define the optimal state sequence, following the error analysis. The resulting identification method is very fast in comparison with WKM, however the problem with initial parameters has not been completely solved. It is not possible to reach to the desired model error for any initial setting. Therefore, it might be necessary to repeat the SS-Imp several times, each time with different initial centers. Additionally, a physical interpretation of the switching conditions is not always possible. Another problem with SS-Imp is the numerical instabilities. To avoid them, scaling of HMM forward and backward algorithms is necessary. Note that the larger the number of data points is, the more possible it is to face numerical instabilities.

Representing human skill as switching impedances is appropriate for the tasks involving human hand (with or without a tool) and object interaction, during which human needs to vary the force. Tasks like in our application examples: bone drilling and push-button, or others like: peg-in-hole, hose insertion or grinding have in their nature the potential to relate the force to position and position derivatives. Tasks introducing highly non-linear interactions, as in the interaction with a human organ during a medical operation, are not suitable to be approximated by switching impedances, since for a good approximation the number of the discrete states needs to be very high and switching between them would be stochastic. This would hinder the ability of interpreting the physical meaning of the states. Additionally, each interaction would lead to another model. Thus, neither would it be possible to label any of the models as the model of the expert, nor to use this models to train a novice. Freespace movements of arm, like carrying a glass of water from one point to another, are more likely to be represented as a jerk optimization problem,

following Hogan's theory [39].

Training Strategies and Their Efficiency

In this thesis, training strategies and possibilities of force skill transfer from an expert to a novice via VEs were widely investigated. Three groups of methods have been outlined: Enhanced training, multi-user training and training via a hybrid dynamic trainer model.

Enhanced VEs contain some features that do not exist in real procedures. Visual, acoustic and haptic enhancements were proposed and used in our applications. Visual enhancement is in the form of discrete signals that warn the student about his/her performance. Although the student easily interprets the meaning of correction signals, the dominance of such signals degrades haptic understanding of the tasks. This is an important drawback, since we aim at force skill training. Consequently, we have introduced the haptic enhancement that incorporates haptic cues to train the student. Basically, this approach constraints the movements of the student haptically, if the student tends to leave the correct trajectory or applies not desired forces. Note that concerning the applied force, the only haptic information the student receives is that the force is wrong. The quantity of the force is not available. The user studies suggest that the proposed haptically enhanced training strategies can be powerful tools for teaching certain skills to students via VEs. It is indicated that additional visual indicators, i.e. visual enhancements, are helpful for teaching only temporarily. They are good for giving the first idea about the correctness of applied force and velocity, but after a while they cause the trainee to lose his/her concentration on haptic feedback. With reference to the effect of acoustic enhancement, we conducted rather fewer user tests. Their results suggest that acoustic enhancement enables a faster skill improvement.

Multi-user training systems either enable the trainer to observe the student and to correct him/her, or they let the trainer demonstrate the task to the student. Multi-user training methods have been divided into three main groups: Haptic and visual observation (HVO), force demonstrating (FD), and force/velocity leading (FVL) methods. In FD method, the trainer performs the task and the student observes the trainer visually and haptically. This is achieved by sending the interaction forces of the trainer to the haptic display of the student. Note that the student has to keep the haptic display tight, i.e. should not let it move. Thus, the student balances the force pressing against him/her. If one considers the main drawback of the classical surgery training, which is missing haptic observation of the trainer, the proposed strategy is very promising. However, the fact that the student does not realize the task himself/herself has a degrading effect for trainings. Therefore, we propose two more multi-user training structures, in which the task is realized by the student and the trainer observes and corrects the student. The observation is exactly the same as in FD case. HVO method gives the trainer the chance to verbally inform the student depending on the observations. On the other hand, the expert influences the performance of the student by modifying his/her axial force or velocity in FVL method. Performed user tests suggest that all of the proposed multi-user training methods help in improving the student skill. However, the most efficient one is HVO, since the correction commands from the trainer are very clear. For the same reason, FVL method is the least efficient one. Nevertheless, it must be underlined that FVL method presents a very important and unique feature, that is the possibility of teaching the velocity and force to the student synchronously using only haptic signals. Note that for FVL training it is assumed that

the student stays on the correct trajectory, but applies wrong force or velocity on this trajectory.

Training via a hybrid dynamic trainer model functions similarly to the multi-user trainings. The main difference is that instead of the trainer personally, his/her model is integrated into the training structure. The same methodologies, i.e. HVO, FD and FVL trainings were implemented. There is no other work known to us that integrates trainer model in the force skill training procedures. Since the presence of an expert during training - learning activities is not always guaranteed, the existence of a trainer model creates a potential that the training procedure is checked and corrected by this model. Such a training structure that involves a trainer model brings in many advantages. First of all, a multi-user scheme is not anymore necessary, the hardware costs are reduced to half. Furthermore, the modeled skills of a late expert can be used to train new generations. Thus, reconstruction of specific defunct tasks can still be possible. It should be added here that the prospect of training in the absence of the expert would provide the student with a more relaxed training environment. User tests on BD-MTS indicate that the skill of the students gets closer to the skill of the expert after each training session.

Although the strategies summarized in this work have explicitly been considered for VR training systems, there is a potential to use them in tele-presence applications. The main potential is using the FD-based methods to train novices for some tele-manipulation tasks or to fasten the adaptation period of telepresence newcomers. These issues were discussed in detail in Section 4.4.3.

Objective Evaluation Methods for User Tests

Assessing the skill improvement of the students objectively is as important as developing the training system. In this work, the expert is defined as a point in n dimensional Euclidean space. The dimension n is the number of important, measurable or derivable interaction parameters. After each trial, a point in this space can also be defined for the student. The distance between these two points is a very strong indicator of the skill of the student. It has been observed that increasing number of trainings makes the n dimensional Euclidean distance between the expert and the student shorter. Additionally, defining a skill level that is related to one or more interaction parameters enables a quick decision on whether to finish or continue training sessions.

6.2 Future Work

In this dissertation, several training strategies have been proposed for force skill training in VEs. Due to the large number of methods, it has not been feasible to perform comprehensive user tests for all of the methods and for different applications. Therefore, the results of the user tests given in this work rather give the preliminary idea on the efficiency and applicability of the training methods. It is undoubtedly useful to extend the application examples and apply systematic user tests with large number of participants.

The trainer skill has been identified as a set of switching generalized impedances. Although a variety of applications can be considered, such as drilling, sawing, grinding and peg-in-hole, freespace movements and interaction with highly non-linear objects cannot be represented as switching impedances. Therefore, for complicated tasks different identification approaches should be used. For example, the task can be represented as switching

impedances after a collision occurs. Before that, freespace movements can be considered as a jerk optimization problem [50].

The application of the methods presented in this work was limited to VR. However, they can be potentially used in tele-presence scenarios. An example scenario can encompass the expert tele-operator demonstrating a tele-manipulation task, while a tele-presence newcomer observes him/her visually and haptically.

A Defining Polygonal Objects in SOLID

“TriangleCallbackFunction” of OpenInventor C++ visualization library is used in our work to read in the model data and divide them into triangles in the format that can be used with SOLID.

in the main function `mySkull.openFile("skull2-by82000.iv")` *read-in skull data.*
`triCBASkull.apply(mySkull)` *Send the skull data to the triangle callback function.*

in the triangle callback function.

`SoPrimitiveVertex *v1, *v2, *v3` *Vertices for the corners of a triangle.*

`SbVec3f vtx[] = {v1 → getPoint(), v2 → getPoint(), v3 → getPoint() }`

vtx is a 3×3 matrix defining a triangle.

`modelData → vertexPositions[modelData → vertexCounter][x] = vtx[..][x]`

Define the complete model matrix in a form that SOLID can read it.

back to the main function.

`initSOLID()` *call the SOLID initialization function.*

in the SOLID initialization function.

`SolidSkull.verts [n].xyz [i] = drillData → vertexPositions [n] [i]`

Define vertices for SOLID.

`SolidDrill.baseHandle =`

`NewVertexBase(SolidDrill.verts [0].xyz, sizeof(vertexStruct))`

Define SOLID vertex base.

`SolidDrill.shapeHandle = DTNewComplexShape(SolidDrill.baseHandle)`

Define that skull is a complex object made of many triangles. So that SOLID applies AABB and BST methods for collision detection.

back to the main function.

...

B Overview of Other Developed Virtual Reality Applications

B.1 Saw Blade Deformations Using FEM

From the technical point of view, bone sawing presents an interesting and original problem. Although many researchers have worked on modeling and haptic rendering of deformable objects, there are very few works that concentrate on the deformation of the surgical tool. The available works are limited to the deformation of the needle during epidural injection processes [5], or during endodontic surgery simulations [71]. The saw blade, the deformable proxy, can be modeled very realistically using FEM, because of the known material characteristics. There is no up-to-date bone sawing VR simulator or training system known to us.

B.1.1 Saw Blade FE Model

The saw blade is modeled with CATIA, and the model is saved in *.stl and *.wrl formats. The file is read in C++, and the blade is visualized by using OpenInventor. The deformation of the saw blade is simulated in real-time by applying force on one certain node. Quadrilateral elements are chosen for finite element modeling.

B.1.2 Finite Element Deformation of Quadrilaterals

Using the FEM method explained in Section 2.2.3, a deformable saw blade is generated. The haptic rendering is done using virtual coupling principles. In Figure B.1, the modeled saw blade and its deformation when force is applied on its upper right corner is shown.



Figure B.1: Deformation of a saw blade using FEM.

B.2 Deformable Box Using Mass-Spring Network

A deformable box is modeled and programmed in C++ language. For visualization Open-Inventor, for collision detection SOLID 3.6 libraries are used. Three visible surfaces of the box are divided into triangles. The avatar is defined as a sphere that enables to realize the collision detection algorithms based on virtual proxy.

Using appropriate classes of SOLID library, collision between the deformable object and the avatar is checked. If there is a collision, SOLID returns penetration depth $\mathbf{x} \in R^3$ of the avatar. Using virtual coupling, external force \mathbf{f}^e effecting on the deformable object is calculated as $\mathbf{f}^e = k_{vc}\mathbf{x}$, where k_{vc} is a virtual spring between the surface of the deformable object and the centre of the avatar. Depending on this force, the deformation of the box is realized through a mass-spring network as it is explained in section 2.2.3. The resulting object deformations are shown in Figure B.2.

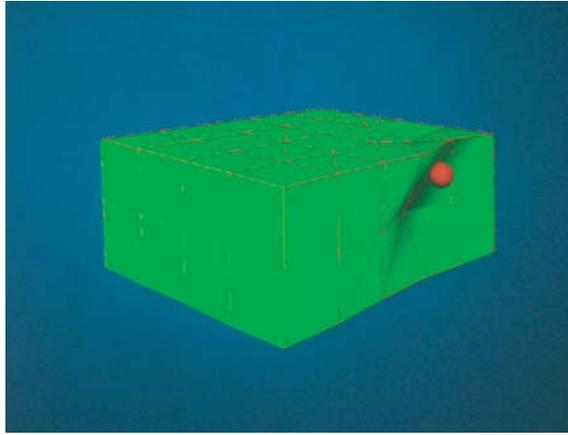


Figure B.2: Deforming a virtual box using mass-spring network.

B.3 Rigid Wall - Virtual Stairs

A VR system is developed to check the haptic interaction with rigid walls and to feel the sharp edges. A virtual stair is created by using simple geometrical objects. The avatar is again a virtual proxy. If the avatar collides with the stairs, then the resulting force is calculated as:

$$\mathbf{f} = k_s \mathbf{x}_s + b_s \mathbf{v}_s, \quad (\text{B.1})$$

where k_s, b_s are spring and damping coefficients of the stairs respectively. The penetration depth of the avatar into the stairs is \mathbf{x}_s and the avatar velocity is \mathbf{v}_s . Addition of the damping element is necessary to manage high stiffness of the stairs. The important point is, when the user changes the direction and wants to leave the stairs, b_s has to be set to zero. Otherwise the avatar sticks to the virtual stairs. The virtual scene of “stairs” is shown in Figure B.3.

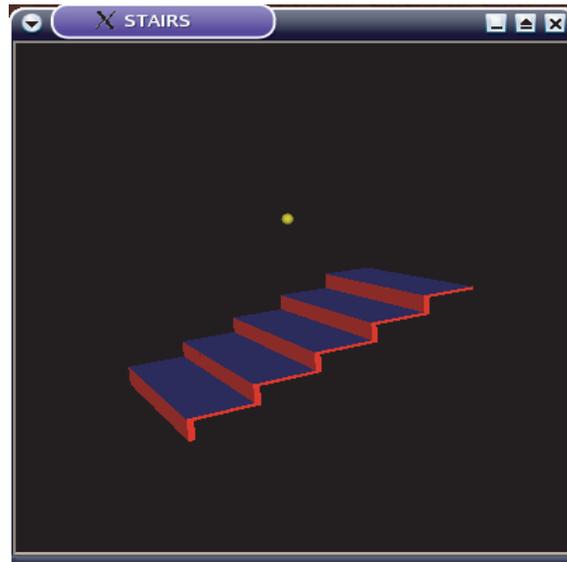


Figure B.3: Virtual stairs.

B.4 Multi-User VR Scenarios

B.4.1 Pull a Spring

The virtual scene consists of a spring model with hooks at each end as it is shown in Figure B.4. The spring is pulled by two users through two ViSHaRD10 haptic interfaces against each other. The force caused by the deformation of the spring is felt by the users appropriately. The calculation of the force f corresponds to Hooke's law, so that $f = k(\Delta x_1 - \Delta x_2)$, where $\Delta x_1, \Delta x_2$ are distances from original positions to the actual positions of the end-effector of the first and second haptic displays respectively. The spring coefficient is k .

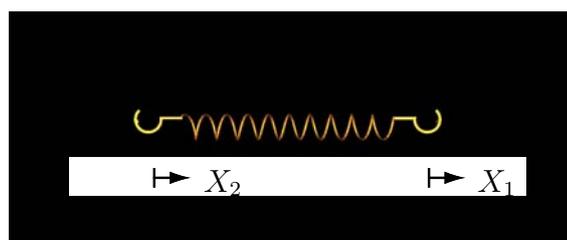


Figure B.4: A virtual task for two users: pulling a spring.

B.4.2 Load Transportation

The “pull a spring” application enables only 1-DoF interaction between users. Therefore another scenario is developed, in which the users can move a load in a 5-DoF space and they both feel the weight of the load, appropriately to its position, as a three dimensional force vector. The VR is modeled as admittance. In order to get a proper physical model of carrying a load, the forces in every state of the load have to be calculated.

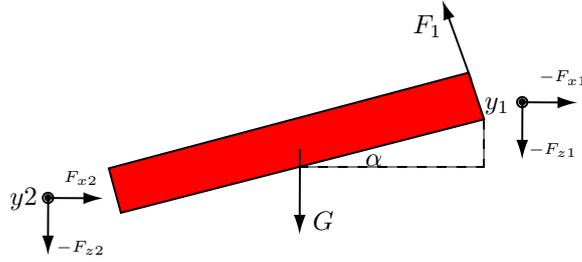


Figure B.5: Collaborative load transportation. Load when tilted along y axes with angle α .

In Figure B.5, the force F_1 acting to the load and reaction forces when the table is tilted around y axes with an angle α are shown, together with the coordinate systems. If the system is frozen on this position, the reaction force are calculated as follows:

$$F_{x1} = F_{x2} = -\frac{1}{4}G \sin(2\alpha),$$

$$F_{z1} = \frac{1}{2}G(1 - \sin^2(\alpha)), \quad F_{z2} = \frac{1}{2}G(1 + \sin^2(\alpha)).$$

While the load is free movable, the sum of the forces in x direction is $F_x = F_x^{u1} + F_{x2}^{u2} + F_{x1} + F_{x2}$, where F_x^{u1} and F_{x2}^{u2} are the forces applied in x direction by user-1 and user-2 respectively. Then velocity v_x in x direction is calculated as a linear function of the force F_x , so that $v_x = \mu F_x$, where μ is a pre-defined constant. The velocity v_x is applied to both robots as $v_{x1} = v_{x2} = v_x$, so that they both move in the same direction with the same velocity. The modeling of the movements in y and z directions are done accordingly. If one of the users keep the haptic display tight, while the other one is moving, then the load visually rotates. The rotations are possible only around y and z axes.

To limit the workspace, invisible virtual walls are constructed. Although traditionally rigid walls are modeled using an impedance scheme [126], making use of the force sensor of our haptic display, an admittance scheme is used in this work. Thus, when the robot's end-effector is pressed into the virtual wall, the velocity of the robot in the direction of the applied force is set to zero.

B.5 A Graphical User Interface for VR Applications

All VR applications are collected under a graphical user interface (GUI). The GUI allows the user to choose one of the available VR applications, run the application, initialize and run the haptic interface, parameterize the chosen VR scenario etc. The GUI is created using Qt library in Linux. It is possible to extend and add new VR scenarios to the VR-GUI. Figure B.6 shows the main window. For detailed information and user's manual please refer to [6].

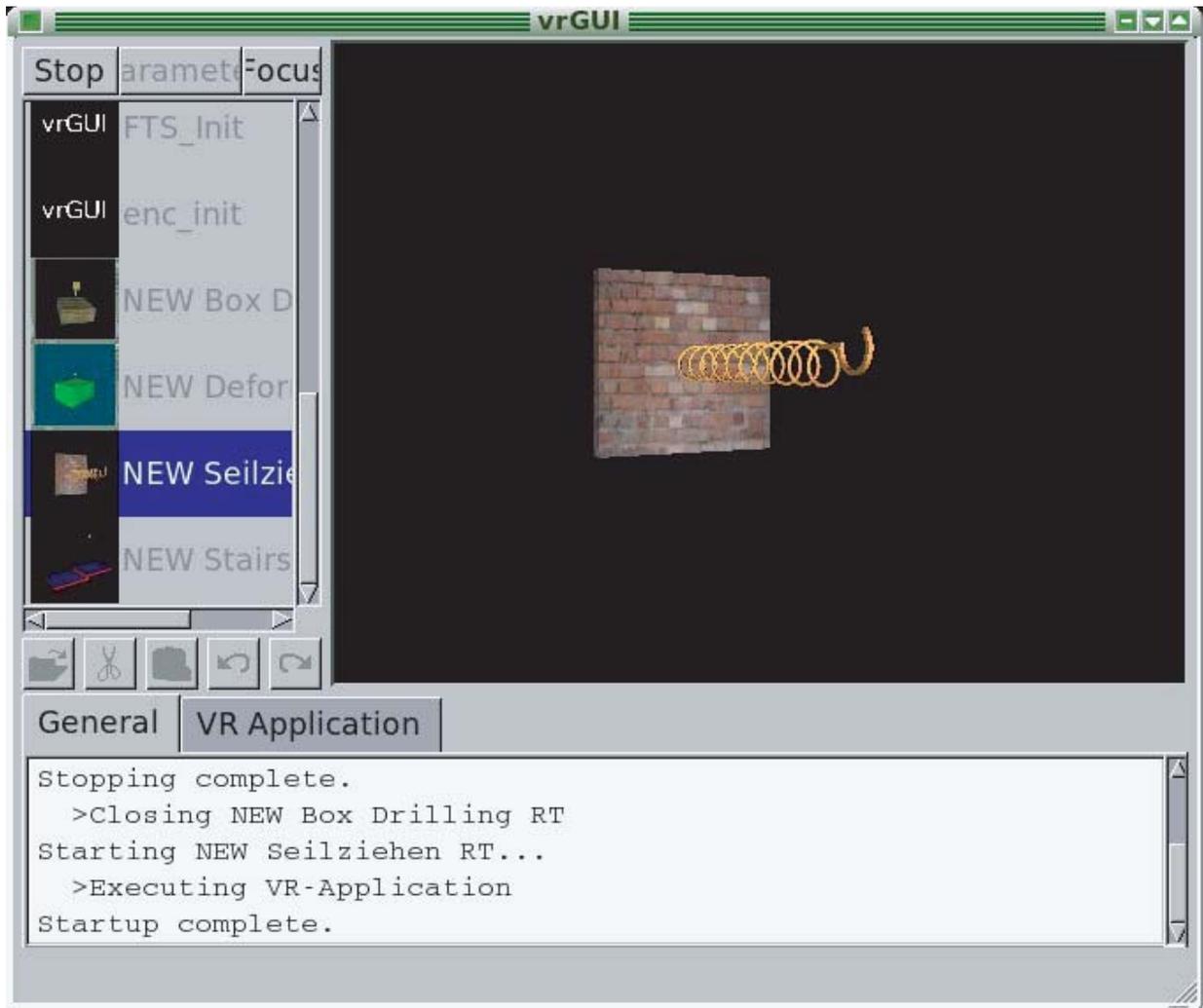


Figure B.6: Graphical user interface for VR applications.

Bibliography

- [1] Deliverable no: D5.8. Technical report, EU Project Touch - Hapsys IST-2001-38040, 2005.
- [2] R.J. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):465 – 474, 1999.
- [3] M. Agus, A. Giachetti, E. Gobbetti, G. Zanetti, and A. Zorcolo. Realtime haptic and visual simulation of bone dissection. *Presence*, 12(1):110 – 124, 2002.
- [4] A. Al-khalifah and D. Roberts. Survey of modelling approaches for medical simulators. In *5th International Conference on Dissability, Virtual Reality & Assoc. Tech.*, United Kingdom, 2004.
- [5] R. Alterovitz, K. Goldberg, and A. Okamura. Planning for steerable bevel-tip needle insertion through 2D soft tissue with obstacles. In *Proceedings of the International Conference on Robotics and Automation ICRA*, Barcelona, Spain, 2005.
- [6] J. Andrasch, T. Bui, and Q. Mühlbauer. Entwicklung einer GUI für VR Anwendungen am LSR. Technical report, Lehrstuhl Steuerungs- und Regelungstechnik (LSR), Technische Universität München, 2005. Projektpraktikum, Betreuer: Hasan Esen.
- [7] H. Asada and S. Liu. Transfer of human skills to neural net robot controllers. In *Proceedings of the International Conference on Robotics and Automation ICRA*, volume 3, pages 2442 – 2448, Sacramento, CA, 1991.
- [8] C. Basdogan. Force reflecting deformable objects for virtual environments. Technical report, Research Laboratory of Electronics - MIT, 1999. Course in SIGGRAPH'99 Conference.
- [9] C. Basdogan. Real-time simulation of dynamically deformable finite element models using modal analysis and spectral Lanczos decomposition methods. In *Proceedings of the Medicine Meets Virtual Reality (MMVR) Conference*, Irvine, CA - USA, 2001.
- [10] C. Basdogan and C. Ho. Principles of haptic rendering for virtual environments. Technical report, Laboratory for Human and Machine Haptics. MIT.
- [11] C. Basdogan, C. Ho, and A. Srinivasan. Virtual environments for medical training: Graphical and haptic simulation of laparoscopic common bile duct exploration. *IEEE/ASME Transactions on Mechatronics*, 6:269 – 285, 2001.
- [12] C. Basdogan and A. M. Srinivasan. *Haptic Rendering in Virtual Environments*. Lawrence Erlbaum Associates, Inc., 2001. from Virtual Environments Handbook, Chapter: 6, pp. 117-134.

- [13] J. Berkley, G. Turkiyyah, D. Berg, M. Ganter, and S. Weghorst. Real-time finite element modeling for surgery simulation: An application to virtual suturing. *IEEE Transactions on Visualization and Computer Graphics*, 10:314 – 325, 2004.
- [14] D. Bourguignon and M.P Cani. Controlling anisotropy in mass-spring systems. In *Computer Animation and Simulation*, Springer Computer Science, pages 113–123, 2000.
- [15] M. Bro-Nielsen. Surgery simulation using fast finite elements. In *Proceedings of the 4th International Conference on Visualization in Biomedical Computing (VBC)*, pages 529–534, Hamburg, Germany, 1996.
- [16] M. Bro-Nielsen and S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In *Comp. Graphics Forum 15(3)*, pages 57–66, 1996.
- [17] J.M. Brown. *Passive Implementation of Multibody Simulations for Haptic Display*. PhD thesis, Northwestern University, 1998.
- [18] P. Buttolo, R. Oboe, and B. Hannaford. Architectures for shared haptic virtual environments. *Computers and Graphics*, 21(4):421–429, 1997.
- [19] M.C. Cavusoglu, W. Williams, F. Tendick, and S.S. Sastry. Robotics for telesurgery: Second generation berkeley/ucsf laparoscopic telesurgical workstation and looking towards the future applications. In *39th Allerton Conference on Communication, Control and Computing*, Monticello, USA, 2001.
- [20] J.E. Colgate, P.E. Grafing, M.C. Stanley, and G. Schenkel. Implementation of stiff virtual walls in force-reflecting interfaces. In *IEEE Virtual Reality Annual International Symposium*, pages 202–208, Seattle, WA, USA, 1993.
- [21] J.E. Colgate and G. Schenkel. Passivity of a class of sampled-data systems: application to haptic interfaces. In *American Control Conference*, volume 3, pages 3236–3240, Maryland, USA, 1994.
- [22] J.E. Colgate, M.C. Stanley, and J.M. Brown. Issues in haptic display of tool use. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction and Cooperative Robots*, pages 140–145, Pittsburgh, PA, USA, 1995.
- [23] J.J. Corso, J. Chhugani, and A.M. Okamura. Interactive haptic rendering of deformable surfaces based on the medial axis transform. In *In Proceedings of Euro-haptics*, pages 92–98, Edinburgh, UK, 2002.
- [24] S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. Technical report, INRIA, France, 1998.
- [25] T. Dang, T. M. Annaswamy, and M. A. Srinivasan. Development and evaluation of an epidural injection simulator with force feedback for medical training. In *Medicine Meets Virtual Reality*, Newport Beach California, USA, 2001.

-
- [26] P.R. Davidson and D.M. Wolpert. Motor learning and prediction in a variable environment. *Current Opinion in Neurobiology*, (13):1–6, 2003.
- [27] S. Dong and F. Naghdy. Manipulation skills acquisition and classification from haptic rendered virtual environment. In *WorldHaptics Conference*, Pisa, Italy, 2005.
- [28] N.M. El-Far, S. Nourian, J. Zhou, A. Hamam, X. Shen, and N.D. Georganas. A cataract tele-surgery training application in a haptic-visual collaborative environment running over the canarie photonic network. In *IEEE International Workshop on Haptic Audio Visual Environments and Their Applications, HAVE*, Ottawa, Ontario, Canada, 2005.
- [29] M. Ernst and M. Banks. Does vision always dominate haptics? In *Conference on Touch in Virtual Environments*, University of South California, 2001.
- [30] H. Esen, K. Yano, and M. Buss. A control algorithm and preliminary user studies for a bone drilling medical training system. In *12th IEEE Workshop Robot and Human Interactive Communication, ROMAN*, San Francisco, USA, 2003.
- [31] H. Esen, K. Yano, and M. Buss. Bone drilling skill training in a 3D virtual environment with haptic. In *VDI - Berichte 1841, ROBOTIK*, Munich - Germany, 2004.
- [32] H. Esen, K. Yano, and M. Buss. Preliminary user studies of a virtual reality medical training system for bone drilling procedures. In *Proceedings of the International Mediterranean Modelling Multiconference I3M - IMAACA Vol.2*, pages 203–208, Bergeggi, Italy, 2004.
- [33] H. Esen, K. Yano, and M. Buss. Teaching bone drilling: 3D graphical and haptic simulation of a bone drilling operation. In *Proceedings of the Scientific Workshop on Medical Robotics and Navigation, MRNV*, Remagen - Germany, 2004.
- [34] H. Esen, K. Yano, and M. Buss. A virtual environment medical training system for bone drilling with 3 dof force feedback. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, Sendai - Japan, 2004.
- [35] H. Esen, K. Yano, and M. Buss. Interaction with virtual deformable objects for surgery simulation using a hyper-redundant haptic display. In *Proceedings of the EuroHaptics International Conference*, pages 519–522, Paris, France, 2006.
- [36] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. In *Hybrid Systems Computation and Control (HSCC) 4th International Workshop*, pages 218–231, Rome, Italy, 2001.
- [37] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. Identification of Piecewise Affine and Hybrid Systems. In *American Control Conference*, pages 3521–3526, Arlington (VA), US, 2001.
- [38] D. Feygin, M. Keehner, and F. Tendick. Haptic guidance: Experimental evaluation of a haptic training method for a perceptual motor skill. In *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS*, Orlando, USA, 2002.

- [39] T. Flash and N. Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5:1688–1703, 1985.
- [40] M. Fritschi. Aufbau eines Trainingssimulators für die digitale rektale Untersuchung von Prostatakarzinomen. Master’s thesis, Institut für Flugmechanik und Flugregelung, Universität Stuttgart, 2002.
- [41] B. R. Gaines. Linear and nonlinear models of the human controller. *International Journal of Man-Machine Studies*, (1):333–360, 1969.
- [42] S. Gottschalk, M. Lin, and D. Manocha. OBB-tree: A hierarchical structure for rapid interference detection. In *ACM Proceedings of SIGGRAPH*, pages 171–180, New Orleans, USA, 1996.
- [43] S. Greenish, V. Hayward, V. Chial, A. Okamura, and T. Steffen. Measurement, analysis, and display of haptic signal during surgical cutting. *Presence*, 11(6):626–651, 2002.
- [44] C. Gunn, M. Hutchins, D. Stevenson, and M. Adcock. Using collaborative haptics in remote surgical training. In *WorldHaptics Conference*, Pisa, Italy, 2005.
- [45] D.M. Haward. The sanders teacher. *Flight*, 2(50):1006–1007, 1910.
- [46] K. Hirana, T. Nozaki, T. Suzuki, S. Okuma, K. Itabashi, and F. Fujiwara. Quantitative evaluation for skill controller based on comparison with human demonstration. *IEEE Transactions on Control Systems Technology*, 12:609–619, 2004.
- [47] K. Hirana, T. Suzuki, S. Okuma, K. Itabashi, and F. Fujiwara. Realization of skill controllers for manipulation of deformable objects based on hybrid automata. In *IEEE International Conference on Robotics and Automation ICRA*, pages 2674–2679, Seoul, Korea, 2001.
- [48] C. Ho, C. Basdogan, and A. Srinivasan. Ray-based haptic rendering: Force and torque interactions between a line probe and 3d objects in virtual environments. *International Journal of Robotic Research*, 19:668–683, 2000.
- [49] I. Hoffmann. *Identifikation hybrider dynamischer Systeme*. PhD thesis, Universität Dortmund, 1999.
- [50] N. Hogan. Controlling impedance at the man/machine interface. In *IEEE International Conference on Robotics and Automation, ICRA*, pages 1626–1631, Scottsdale, AZ.
- [51] J. Hoogen, M. Ponikvar, R. Riener, and G. Schmidt. A robotic haptic interface for kinesthetic knee joint simulation. In *Conference on Robotics in Alpe-Adria-Danube Region (RAAD)*, pages 125–130, Balatonfred, Hungary, 2002.
- [52] J. Hoogen, R. Riener, and G. Schmidt. Control concepts for an industrial robot used as kinesthetic knee joint simulator. In *3rd International Workshop on Robot Motion and Control*, 2002.

-
- [53] G.E. Hovland, P. Sikka, and B.J McCarragher. Skill acquisition from human demonstration using a hidden markov model. In *IEEE International Conference on Robotics and Automation*, pages 2706–2711, Minnesota, USA, 1996.
- [54] K. Itabashi, K. Hirana, T. Suzuki, S. Okuma, and F. Fujiwara. Modelling and realization of the peg-in-hole task based on hidden markov model. In *IEEE International Conference on Robotics & Automation (ICRA)*, pages 1142–1147, Leuven, Belgium, 1998.
- [55] K. Itabashi, K. Hirana, T. Suzuki, S. Okuma, and F. Fujiwara. Realization of the human skill in the peg-in-hole task using hybrid architecture. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 995–1000, Victoria, BC, 1998.
- [56] D. L. James and D. K. Pai. A unified treatment of elastostatic contact simulation for real time haptics. *Electronical Journal Haptics-e* <http://www.haptics-e.org>, 2(1), 2001.
- [57] L. Jiang, R. Girotra, M. R. Cutkosky, and C. Ullrich. Reducing error rates with low-cost haptic feedback in virtual reality - based training applications. In *WorldHaptics Conference*, Pisa, Italy, 2005.
- [58] E. Keeve, S. Girod, and B. Girod. Craniofacial surgery simulation. In *Proceedings of the 4th International Conference on Visualization in Biomedical Computing (VBC '96)*, pages 541–546, Hamburg, Germany, 1996.
- [59] A. Kheddar. Problems in designing inclusive haptic devices. In *International Conference on Intelligent Robots and Systems, IROS*, Sendai, Japan, 2004.
- [60] R. Kikuuwe and T. Yoshikawa. Haptic display device with fingertip presser for motion/force teaching to human. In *IEEE International Conference on Robotics and Automation ICRA*, volume 1, pages 868–873, Seoul, Korea, 2001.
- [61] J. Kim. Virtual environments for medical training: Graphical and haptic simulation of tool-tissue interactions. Technical report, Department of Computer Sciences at MIT, 2002.
- [62] J. Kim, S.W. De, and M. A. Srinivasan. Computationally efficient techniques for real time surgical simulation with force feedback. In *10th Symposium on Haptic Interfaces For Virtual Environments & Teleoperator Systems (HAPTICS)*, 2002.
- [63] J.H. Kim, S. Hayakawa, T. Suzuki, K. Hayashi, S. Okuma, N. Tsuchida, M. Shimizu, and S. Kido. Modeling of driver's collision avoidance maneuver based on controller switching model. *IEEE Transactions on System, Man, and Cybernetics*, 35:1131–1143, 2005.
- [64] R. Koeppe, A. Breidenbach, and G. Hirzinger. Skill representation and acquisition of compliant motions using a teach device. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 897–904, Osaka, Japan, 1996.

- [65] K. Kosuge, T. Fukuda, and H. Asada. Acquisition of human skills for robotic systems. In *Proceedings of the IEEE International Symposium on Intelligent Control*, pages 469–474, Arlington, VA, 1991.
- [66] U. Kühnapfel. *Grafische Realzeitunterstützung für Fernhandhabungsvorgänge in komplexen Arbeitsumgebungen im Rahmen eines Systems zur Steuerung, Simulation, Off-Line Programmierung*. PhD thesis, University Karlsruhe, 1991.
- [67] U. Kühnapfel, H.K. Cakmak, and H. Maass. Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computer & Graphics*, (24):671–682, 2000.
- [68] U. Kühnapfel, H.K. Cakmak, and H. Maass. Virtual endoscopic surgery training (vest): Development of surgical interactions and virtual patient models for laparoscopic applications. In *Medicine Meets Virtual Reality (MMVR) 11, Workshop on Surgical Simulators*, pages 113–124, Newport-Beach, California-USA, 2003.
- [69] G.W. Lange. *Syntheses of a Model of the Human Operator Engaged in a Tracking Task*. PhD thesis, University of London, 1965.
- [70] C. Laugier, C. Mendoza, and K. Sundaraj. Towards a realistic medical simulator using virtual environments and haptic interaction. In *International Symposium in Research Robotics*, Sydney, 2001.
- [71] M. Li and Y.H. Liu. Modeling interactions of pulpal tissue with deformable tools in endodontic simulation. In *Proceedings of the International Conference on Robotics and Automation ICRA*, Barcelona, Spain, 2005.
- [72] D. Liberzon. *Switching in Systems and Control*. Birkhäuser, 1973.
- [73] J.G. Linares, N. Guil, P. Perez, M. Ehrenmann, and R. Dillmann. An efficient image processing algorithm for high-level skill acquisition. In *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, pages 262–267, Porto, Portugal, 1999.
- [74] A. Liu, F. Tendick, K. Cleary, and C. Kaufmann. A survey of surgical simulation: Applications, technology, and education. *Presence*, 12(6), 2003.
- [75] J.C. Lombardo, M.P. Cani, and F. Neyret. Real-time collision detection for virtual surgery. In *Proceedings of the Computer Animation Conference*, pages 82–91, 1999.
- [76] S. Mahapatra and M. Zefran. Stable haptic interaction with switched virtual environments. In *IEEE International Conference on Robotics and Automation, ICRA*, Taipei, Taiwan, 2003.
- [77] S. Matsumoto, I. Fukuda, H. Mrino, K. Hikichi, K. Sezaki, and Y. Yasuda. The influence of network issues on haptic collaboration in shared virtual environments. In *Proc. of Phantom Users Group*, 2000.
- [78] Y. Matsuoka and R.D. Howe. Robotics for surgery. *Annu. Rev. Biomed. Eng.*, 01:211–240, 1999.

-
- [79] H. Mayer, I. Nagy, and A. Knoll. Application of skill transfer in robotic surgery. Technical report, Technische Universität München, 2003.
- [80] H. Mayer, I. Nagy, and A. Knoll. Skill transfer and learning by demonstration in a realistic scenario for laparoscopic surgery. In *IEEE International Conference on Humanoids*, Munich, Germany, 2003.
- [81] B.J McCarragher. Force sensing from human demonstration using a hybrid dynamical model and qualitative reasoning. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 557–563, 1994.
- [82] C. Mendoza, K. Sundaraj, and C. Laugier. Faithfull haptic feedback in medical simulators. In *8th International Symposium on Experimental Robotics, ISER*, 2002.
- [83] B.E. Miller, J.E. Colgate, and R.A. Freeman. Guaranteed stability of haptic systems with non-linear virtual environments. *IEEE Transactions on Robotics and Automation*, 16(6):712–719, 2000.
- [84] L. Moody, C. Baber, T. N. Arvanitis, and M. Elliot. Objective metrics for the evaluation of simple surgical skills in real and virtual domains. *Presence*, 12(2):207–221, 2003.
- [85] D. Morris, C. Sewell, F. Barbagli, N. Blevins, S. Girod, and K. Salisbury. Visuohaptic simulation of bone surgery for training and evaluation. *IEEE Transactions on Computer Graphics and Applications*, November/December:48–57, 2006.
- [86] D. Morris, C. Sewell, N. Blevins, F. Barbagli, and K. Salisbury. A collaborative virtual environment for the simulation of temporal bone surgery. In *International Conference on Medical Imaging Computing and Computer Assisted Intervention MICCAI*, St. Malo, France, 2004.
- [87] M.C. Nechyba and Y. Xu. Human skill transfer: neural networks as learners and teachers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction and Cooperative Robots*, volume 3, pages 314–319, Pittsburgh, PA, USA, 1995.
- [88] O. Nelles. *Nonlinear System Identification*. Springer, Berlin, Germany, 2001.
- [89] G. Niemeyer, K.J. Kuchenbecker, R. Bonneau, A.M. Mitra, P. Reid, J. Fiene, and G. Weldon. Thump: An immersive haptic console for surgical simulation and testing. In *Medicine Meets Virtual Reality (MMVR)*, pages 272–274, 2004.
- [90] G. Niemeyer and P. Mitra. Dynamic proxies and haptic constraints. In *IEEE International Conference on Robotics and Automation, ICRA*, New Orleans, USA, 2004.
- [91] T. Obst, R. Burgkart, R. Riener, M. Frey, and E. Ruckhäberle. Prototypischer multimodaler trainingssimulator für die geburtshilfe. In *Automed*, 2003.
- [92] Y. Ou and Y. Xu. Modeling human skills to control dynamic systems. In *Proceedings of the IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, volume 1, pages 382–387, Changsha, China, 2003.

- [93] R.L. Page. Brief history of flight simulation. In *SIMTECT Conference*, Sydney, Australia, 2000.
- [94] E. Papadopoulos, A. Tsamis, and K. Vlachos. A real-time graphics environment for a urological operation training simulator. In *IEEE International Conference on Robotics and Automation, ICRA*, New Orleans, USA, 2004.
- [95] R. Prada and S. Payandeh. A study on design and analysis of virtual fixtures for cutting in training environments. In *WorldHaptics Conference*, Pisa, Italy, 2005.
- [96] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77:257–286, 1989.
- [97] A. Radetzky, A. Nürnberger, and D.P. Pretschner. the simulation of elastic tissues in virtual medicine using neuro-fuzzy systems. In *Medical Imaging:98, Image Display, Provedings of SPIE*, volume 3335, pages 399–409, San Diego, USA, 1998.
- [98] M. Renz, C. Preusche, M. Pötke, H. P. Kriegel, and G. Hirzinger. Stable haptic interaction with virtual environments using an adapted voxmap-pointshell algorithm. In *Proceedings of the Eurohaptics Conference*, Birmingham, UK, 2001.
- [99] R. Riener. Techniken der multimodalen virtuellen realität in der medizin, Juli 2002. Habilitationsschrift der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München.
- [100] R. Riener, J. Hoogen, M. Ponikvar, R. Burgkart, M. Frey, and G. Schmidt. Orthopädischer trainingssimulator mit haptischem feedback. *Automatisierungstechnik at 50(6)*, pages 296 – 303, 2002.
- [101] G. Riva and L. Gamberini. *Communications Through Virtual Technology: Identity, Community and Technology in Internet Age*, chapter 7, Virtual Reality in Telemedicine, pages 101–116. IOS Press, 2001.
- [102] I. Rock and J. Victor. Vision and touch: An experimentally created conflict between the senses. *Science*, 143:594–596, 1964.
- [103] M.A.F. Rodrigues, R.R.C. Chaves, and W.B. Silva. Collaborative virtual training using force feedback devices. In *17th Brazilian Symposium on Computer Graphics and Image Processing*, pages 332–339, Brasil, 2004.
- [104] J. Rosen, B. Hannaford, C.G. Richards, and M. Sinanan. Markov modeling of minimally invasive surgery based on tool/tissue interaction and force/torque signatures for evaluating surgical skills. *IEEE Transactions on Biomedical Engineering*, 48(5):579–591, 2001.
- [105] J. Rosen, M. Solazzo, B. Hannaford, and M. Sinanan. Objective laparoscopic skills assessments of surgical residents using hidden markov models based on haptic information and tool/tissue interactions. In *Studies in Health Technologies and Informatics, Medicine Meets Virtual Reality*, Newport Beach, CA, USA, 2001.

-
- [106] D.C. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. In *Computer Graphics Proceedings, Annual Conference Series*, 1997.
- [107] D.C. Ruspini, K. Kolarov, and O. Khatib. Haptic interaction in virtual environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, Grenoble, France, 1997.
- [108] K. Salisbury, F. Conti, and F. Barbagli. Haptic rendering: Introductory concepts. *IEEE Transactions on Computer Graphics and Applications*, 24:24–32, 2004.
- [109] M. Schijven, R. Klaassen, J. Jakimowicz, and O. T. Terpstra. The intercollegiate basic surgical skills course. *Surgical Endoscopy*, pages 1978–1984, 2003.
- [110] R. Schmidt. Spinal tap: An architecture for real-time vertebrae drilling simulation. Technical report, Department of Computer Sciences at the University of Calgary, 2002.
- [111] J. Shawe-Taylor and N. Cristianini. *Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
- [112] P. Sikka and B.J. McCarragher. Stiffness-based understanding and modeling of contact tasks by human demonstration. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, volume 1, pages 464–470, 1997.
- [113] O.J. Smith. Nonlinear computations in the human controller. *IRE Transactions*, BME(9):128–128, 1962.
- [114] O. Sourina, A. Sourin, and H.T. Sen. Virtual orthopedic surgery training on personal computer. *International Journal of Information Technology*, 6(1):16–28, 2000.
- [115] G. Srimathveeravalli and K. Thenkurussi. Motor skill training assistance using haptic attributes. In *WorldHaptics Conference*, Pisa, Italy, 2005.
- [116] S. Suzuki, N. Tomomatsu, F. Harashima, and K. Furuta. Skill evaluation based on state-transition model for human adaptive mechatronics (HAM). In *30th Annual Conference of the IEEE Industrial Electronics Society*, pages 641–646, Busan, Korea, 2004.
- [117] T. Suzuki, S. Inagaki, and N. Yamada. Human skill modeling based on stochastic switching dynamics. In *2nd IFAC Conference on Analysis and Design of Hybrid Systems*, pages 64–70, Alghero, Italy, 2006.
- [118] T. Suzuki, S. Sekizawa, S. Inagaki, S. Hayakawa, N. Tsuchida, T. Tsuda, and H. Fujinami. Modeling of recognition of human driving behavior based on stochastic switched ARX model. In *Proceedings of the 44th IEEE International Conference on Decision and Control, and the European Control Conference*, pages 5095–5100, Seville, Spain, 2005.

- [119] Y. Takeda, Tanaka Y., and T. Tsuji. A virtual training sports system for measuring human hand impedance. In *Proceedings of the. 2003 IEEE / ASME. International Conference on. Advanced Intelligent Mechatronics (AIM)*, volume 2, pages 914–919, 2003.
- [120] F. Tendick, M. Downes, T. Goktekin, M.C. Cavusoglu, D. Feygin, X. Wu, R. Eyal, Hogarty M., and L.W. Way. A virtual environment testbed for training laparoscopic surgical skills. *Presence*, 9(3):236–255, 2000.
- [121] T. Tsuji and T. Yoshikuji. Tracking control properties of human - robotic systems based on impedance control. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 35:523–535, 2005.
- [122] C.S. Tzafestas, Y. Koumpouros, and K. Birbas. Paracentesis modeling and VR-based simulation with haptic display for clinical skill training and assessment. In *Proceedings of the International Mediterranean Modelling Multiconference I3M - IMAACA Vol.2*, pages 196–203, Bergeggi, Italy, 2004.
- [123] T. Udiljak, D. Ciglar, and K. Mihoci. Influencing parameters in bone drilling. In *9th International Scientific Conference on Production Engineering CIM*, pages 133–142, Lumbarda, 2003.
- [124] M Ueberle and M. Buss. Control of kinesthetic haptic interfaces. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, Workshop on Touch and Haptics*, Sendai, Japan, 2004.
- [125] M. Ueberle, N. Mock, and M. Buss. ViSHaRD10, a novel hyper-redundant haptic interface. In *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 58–65, Chicago, USA, 2004.
- [126] M. Ueberle, N. Mock, A. Peer, C. Michas, and M. Buss. Design and control: Concepts of a hyper redundant haptic interface for interaction with virtual environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, Workshop on Touch and Haptics*, pages 1–14, Sendai, Japan, 2004.
- [127] G. Van-den Bergen. Efficient collision detection of complex deformable models using aabb trees. *J. Graphic Tools*, 2(4):1–13, 1997.
- [128] G. Van-den Bergen. *Collision Detection in Interactive 3D Environments*. Elsevier, Inc., 2004.
- [129] T. Vassilev and B Spanlang. A mass-spring model for real time deformable solids. In *In Proceedings of East-West Vision*, pages 149–154, Graz, Austria, 2002.
- [130] D. Wang, M. Rossi, J. Shu, and K. Tuer. Collaborative tele-haptics - a pilot study evaluation. In *Laval Virtual, 8th International Conference on Virtual Reality*, Laval, France, 2005.
- [131] R.W. Webster, D.I. Zimmerman, B.J. Mohler, M.G. Melkonian, and R.S. Haluck. A prototype haptic suturing simulator. In *Proceedings of the Medicine Meets Virtual Reality (MMVR) Conference*, pages 567–569, Irvine, CA - USA, 2001.

-
- [132] Y. Wei, J. Patton, P. Bajaj, and R. Scheidt. A real-time haptic/graphic demonstration of how error augmentation can enhance learning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, Barcelona, Spain, 2005.
- [133] D. Weiss and A.M. Okamura. Haptic rendering of tissue cutting with scissors. In *12th Medicine Meets Virtual Reality (MMVR) Conference*, Newport Beach CA, USA, 2004.
- [134] K.L. Wiggins and S. Malkin. Drilling of bone. *J. Biomechanics*, 9:553–559, 1976.
- [135] R.W. Wilde and J. H. Wescott. The characteristics of the human operator engaged in a tracking task. *Automatica*, 1(5), 1962.
- [136] Y. Xu and J. Yang. Towards human-robot coordination: skill modeling and transferring via hidden markov model. In *Proceedings of the International Conference on Robotics and Automation ICRA*, volume 2, pages 1906–1911, Nagoya, Japan, 1995.
- [137] N. Yamada, S. Inagaki, T. Suzuki, H. Okuda, S. Hayakawa, and N. Tsuchida. Behavior modeling in man-machine cooperative system based on stochastic switching dynamics. In *Proceedings of the International Conference on Robotics and Automation ICRA*, pages 3618–3623, Orlando, Florida, 2006.
- [138] B.H. Yang and H. Asada. Hybrid linguistic/numeric control of deburring robots based on human skills. In *Proceedings of the International Conference on Robotics and Automation ICRA*, pages 1467–1474, Nice, France, 1992.
- [139] Y. Yokokohji, R. L. Hollis, K. Kanade, T. Henmi, and T. Yoshikawa. Toward machine mediated training of motor skills, skill transfer from human to human via virtual environment. In *IEEE International Workshop on Robot and Human Communication, ROMAN*, pages 32–37, Japan, 1996.
- [140] T. Yoshikawa and Y. Ichinoo. Impedance identification of human fingers using virtual task environment. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 3094–3099, Las Vegas, USA, 2003.
- [141] L.R. Young and J.L. Meiry. Biological control systems - a critical review and evaluation. Technical report, NASA CR-190, 1965.
- [142] C. B. Zilles and J. K. Salisbury. A constraint-based God-Object method for haptic display. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction and Cooperative Robots*, volume 3, pages 146–151, Pittsburgh, PA, USA, 1995.
- [143] R. Zollner, O. Rogalla, R. Dillmann, and M. Zollner. Understanding users intention: programming fine manipulation tasks by demonstration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, volume 2, pages 1114–1119, 2002.