POLITECNICO DI MILANO
Department of Electronics, Information and Bioengineering
Master of Science in Biomedical Engineering

# Evaluation of Haptic Virtual Fixtures in Psychomotor Skill Development for Robotic Surgical Training

Supervisor: DE MOMI Elena, PhD.
Assistant Supervisor: LUCIANO Cristian, PhD.
Tutor: BUZZI Jacopo, PhD. Student

Master Thesis of
Cecilia GATTI
Student ID: 850353

Academic Year 2016-2017

*Osa diventare ció che sei. E non disarmarti facilmente. Ci sono meravigliose opportunitá in ogni essere. Persuaditi della tua forza e della tua gioventú. Continua a ripetere incessantemente: “Non spetta che a me”.*

*- André Gide*

# Sommario

Recentemente, l'uso della chirurgia robotica teleoperata é cresciuto notevolmente grazie a vantaggi considerevoli rispetto alla tradizionale laparoscopia. In chirurgia mini invasiva teleoperata, i chirurghi operano tramite un *master device*, ovvero un manipolatore in grado di controllare un braccio robotico che si interfaccia direttamente con il paziente. Questo sistema permette di ottenere performance piú elevate grazie alla possibilitá di ridurre e filtrare i movimenti della mano del chirurgo, portando quindi a maggior precisione e minor invasivitá. Le sostanziali differenze nella cinematica, cinetica e percezioni sensoriali introdotte dall'uso di un controllore a distanza (il *master device*), ha portato nuove sfide per i chirurghi, a partire dall'apprendimento di abilitá psicomotorie. Mentre feedback visivi, sonori e tattili (*haptici*) sono stati ampiamente usati in sistemi di allenamento tradizionali, l'assenza di feedback haptici nel piú usato sistema di chirurgia teleoperata, il da Vinci Surgical System, impedisce l'uso di questi ultimi. Lo sviluppo di piú avanzati sistemi di controllo teleoperati ha portato all'integrazione di questo tipo di feedback nell'allenamento su sistemi di realtá virtuale aumentata. Questi sistemi sono stati applicati con successo in riabilitazione: l'uso della tecnica di error augmentation basata su feedback haptici ha portato a miglioramenti nel ri-sviluppo di capacitá motorie in pazienti colpiti da ictus. Tuttavia, gli effetti di questo approccio in un contesto di allenamento per chirurgia teleoperata sono ancora da approfondire.

Questo lavoro di tesi analizza l'uso della teoria di error augmentation per lo sviluppo di capacitá psicomotorie per chirurgia robotica. Abbiamo sviluppato un set up di realtá virtuale e feedback haptici per valutare gli effetti nell'apprendimento di un task di inseguimento di traiettoria. Tre diversi gruppi di allenamento sono stati considerati: con solo feedback visivo, con approccio Limit-Push, un'applicazione di feedback di forza usata attualmente in riabilitazione, e con l'approccio qui introdotto per la prima volta, il Limit-Trench, basato sull'allenamento con instabilitá. I miglioramenti nell'esecuzione dell'esercizio proposto sono stati analizzati tra i tre gruppi tenendo in considerazione cinque diversi indici di performance. Da questo studio é risultato che allenarsi senza feedback di forza porta ad una migliore esecuzione del task. Inoltre, abbiamo riscontrato che l'allenamento condotto con forze che portano ad instabilitá nell'esecuzione é addirittura controproducente per l'apprendimento di capacitá psicomotorie, delineando quindi l'esistenza di condizioni limite per un allenamento adeguato basato sull'error augmentation.

# Summary

In recent years, the impact of robotic teleoperated surgery has grown significantly thanks to the considerable advantages with respect to traditional laparoscopy. In teleoperated Robotic Minimally Invasive Surgery, surgeons use a master device: a manipulator that controls a slave robot that directly interacts with the patient. This system allows to obtain high performances thanks to the possibility of downscaling and filtering the surgeon hand movements, achieving higher accuracy and lower invasiveness. The substantial differences in the kinematic, kinetic and sensory perceptions introduced by the interaction with the master devices gave rise to a series of new challenges for surgeons, starting from skill acquisition. While visual, auditory, and touch senses have been widely used in traditional training, the absence of inherent haptic feedback in the currently most used surgical teleoperation system, i.e., the da Vinci Surgical System, prevented the use of tactile and kinesthetic information. The development of more advanced master devices enabled the possibility of providing augmented feedback in the form of tactile or kinesthetic information while training on VR simulators. These systems have been successfully applied in rehabilitation: the use of haptics-based error augmentation has been able to swiftly improve dysfunctional psychomotor skills in stroke patients. However, its effects on teleoperation training and expertise skills development are yet to be fully understood.

This thesis work further explore the use of error augmentation theory for psychomotor skill training for robotic surgery. We developed a virtual reality and haptic test bed to evaluate learning effect in a trajectory-following task. Three different training conditions have been taken into consideration: with visual feedback only, with Limit-Push condition, which is an error-augmented force feedback application currently used in rehabilitation, and with the novel Limit-Trench approach, based on training with instability conditions. We evaluated improvements of the groups of subjects among five different performance indexes. Our results show that training without any force feedback leads to better task execution. Furthermore, we found out that instability is even detrimental for development of psychomotor skills, thus delineating the existance of boundary conditions of positive training effects.

# Acknowledgements

I would like to thank all NearLab members, PhDs, and professors, for all the support, help and opportunities I had in these three years spent with them. In particular, I would like to thank my advisor, Prof. Elena De Momi, for her constant and fundamental assistance.

Eventually, I would like to deeply thank Jacopo and Francesca: you taught and inspired me much more than what you imagine.

# Contents

**Sommario** **iii**

**Summary** **v**

**Acknowledgements** **vii**

**1 Introduction** **1**

1.1 Physical Simulation . . . . . . . . . . . . . . . . . . . . . . . . . 2

1.2 Virtual Reality Environment . . . . . . . . . . . . . . . . . . . . 2

1.3 Surgical Simulators . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.3.1 da Vinci Surgical Skill Simulator® . . . . . . . . . . . 5

1.3.2 dV-Trainer® . . . . . . . . . . . . . . . . . . . . . . . . . 5

1.3.3 Robotic Surgical Simulator™ . . . . . . . . . . . . . . . 6

**2 Background** **9**

2.1 Motor Learning . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

2.1.1 Products of Motor Learning . . . . . . . . . . . . . . . . 11

2.1.2 Mechanisms of Motor Learning . . . . . . . . . . . . . . 11

2.1.3 Stages of Motor Learning . . . . . . . . . . . . . . . . . . 12

2.2 Inherent and Augmented Feedback . . . . . . . . . . . . . . . . 13

2.2.1 Visual feedback . . . . . . . . . . . . . . . . . . . . . . . 15

2.2.2 Haptic feedback . . . . . . . . . . . . . . . . . . . . . . . 17

2.2.3 Multimodal Feedback . . . . . . . . . . . . . . . . . . . . 20

**3 Methods** **23**

3.1 LACE Implementation . . . . . . . . . . . . . . . . . . . . . . . 24

3.2 Bezier Curve . . . 30
3.2.1 Graphic Rendering . . . 33
3.3 Force Fields . . . 35
3.4 Implementation . . . 35
3.4.1 Force Rendering . . . 37
3.5 Experimental Setup . . . 42
3.5.1 Choice of Trajectories . . . 42
3.5.2 Choice of Force Field . . . 43
3.5.3 Task Execution . . . 43

**4 Experimental Evaluation 46**
4.1 Experimental Setup . . . 46
4.2 Subjects Involved . . . 47
4.3 Experimental design . . . 49
4.4 Force Fields Parameters . . . 50
4.5 Performance Metrics . . . 50
4.6 Statistical Analysis . . . 52
4.6.1 **Baseline** . . . 52
4.6.2 **Training effects** . . . 52

**5 Results 54**
5.0.1 Baseline Analysis . . . 54
5.0.2 Performance Analysis . . . 54

**6 Discussion 57**

**7 Conclusion 60**

**Bibliography 62**

# List of Figures

1.1 Example of a Phantom-based model. . . . . . . . . . . . . . . . 3

1.2 Most used surgical simulators. In A the dVSS is displayed, the red circle highlights the actual simulator, which is a sort of backpack attached to the same exact system used during procedures; in B the RoSS system [41]; in C it is represented the dv-Trainer (*Image courtesy of Mimic Simulation*). . . . . . 4

1.3 RobotiX Mentor surgical simulator: it is provided with an external screen, too, so that the instructor can easily give feedback to trainers. . . . . . . . . . . . . . . . . . . . . . . . . . . 7

2.1 Visual Feedback Modalities. In A the performance is summarized and given to the user as a graph [16]; in B the feedback is represented as a curve (*©2007 IEEE*), while in C there is an example of superposition: the arm of the user is hidden by a screen where will be displayed the fake movement. (*©2011 IEEE*) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

2.2 Haptic Constraints Modalities. In A it is represented the rigid position control concept: the user is forced to strictly follow the trajectory, in B and C there are represented a convergent and divergent force field respectively. *©2013 IEEE.* . . . . . . . 18

2.3 Hybrid Approach Test Results. In both immediate and delayed retention tests the hybrid approach does not show any evidence of better performance. *©2014 IEEE.* . . . . . . . . . . 19

2.4 Visuohaptic feedback example. A haptic device is used along with a screen, so that visual and haptic information are assess together and combined by the user to improve trajectory learning under the form of letters. ©*2008 Bluteau et al.* . . . 20

2.5 Limit-Push (A) and Limit-Trench (B) concept representation. The thick line represents the ideal trajectory, arrows represent the force field direction, while the dashed lines define the distance threshold used to apply or switch force. . . . . . . . . 22

3.1 Touch 3D Stylus. . . . . . . . . . . . . . . . . . . . . . . . . . . 24

3.2 LACE logic . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

3.3 LACE Classes . . . . . . . . . . . . . . . . . . . . . . . . . . 27

3.4 LACE Object organization . . . . . . . . . . . . . . . . . . . . 28

3.5 LACE Transform . . . . . . . . . . . . . . . . . . . . . . . . . 29

3.6 Bezier Curves Construction of a quadratic (left), cubic (center) and quartic (right) Bezier curve. . . . . . . . . . . . . . . . . . . 32

3.7 Composite Bezier Curve . . . . . . . . . . . . . . . . . . . . . 33

3.8 Example of an extrusion from a circular shape along a Bezier curve. Start and end points are marked with spheres. . . . . . 34

3.9 General scheme of an active constraint implementation . . . . 35

3.10 Psychomotor Skill scheme . . . . . . . . . . . . . . . . . . . . 37

3.11 Cursor and Trajectory representation . . . . . . . . . . . . . . 37

3.12 Representation of the trajectory's change of color depending on cursor distance, i.e. the error made by the user. The more the cursor is away from the ideal curve, the more the trajectory will move toward the red color. . . . . . . . . . . . . . . . . 38

3.13 Force Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . 39

3.14 Geometry representation of a distance from a linear segment. Using the scalar product among first and last point of the line segment, and the external point, the distance is computed. . . 40

3.15 Qualitative representation of forces as linear functions of the distance . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41

3.16 Experimental Setup Workflow. First the user decided the trajectory to use, then force field is chosen and eventually the actual task can be carried out, editing the force function parameters to optimally tune them. . . . . . . . . . . . . . . . . 43

3.17 Force arrows visualization examples. In the upper trajectory, it is displayed a convergent force field, while a divergent one is represented in the lower curve.The arrow elongates or shortens proportional to the force magnitude, pointing accordingly to the direction. . . . . . . . . . . . . . . . . . . . . . . . . . . 44

4.1 Geomagic Touch. . . . . . . . . . . . . . . . . . . . . . . . . 46

4.2 Experimental setup of the virtual reality and haptic test bed for error augmentation. A zoom on the cross section of the trajectory represents limit-push and limit-trench concept application. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

4.3 Trajectories used for the training protocol. They are built as two interconnected cubic Bezier curves. . . . . . . . . . . . . . 49

4.4 Translational Path Error. Red line represents the end effector movement, while the black is the target path. Considering the actual cursor position and closest point on curve ($X_n$, $C_n$ respectively), and previous cursor position and closest point on curve ($X_{n-1}$, $C_{n-1}$ respectively), $TPE_i$ is the rectangular area pictured by current closest distance from curve, and distance between current and previous closest point on curve. The overall TPE is the sum of every $TPE_i$ computed, thus approximating the surface difference between the two paths. . 52

5.1 Median, first and third interquartile distance as well as minimal and maximal values are presented as box-plots of the four performance metrics that showed statistical differences. Statistical difference is quantitatively represented with asterisks (* $p \leq 0.05$, ** $p \leq 0.01$, *** $p \leq 0.001$). . . . . . . . . . . . 56

# List of Tables

2.1 AUGMENTED FEEDBACK. . . . . . . . . . . . . . . . . . . . 14

3.1 TECHNICAL SPECIFICATION TOUCH 3D STYLUS. . . . 24

4.1 TECHNICAL SPECIFICATION GEOMAGIC TOUCH. . . . 47

5.1 MEDIAN AND INTERQUARTILE RANGE OF $\Delta_{BE}$ FOR EACH GROUP. . . . . . . . . . . . . . . . . . . . . . . . . . . 55

5.2 P-VALUES FROM THE NON-PARAMETRIC ANALYSIS IN R; ASTERISKS INDICATE STATISTICAL DIFFERENCES BETWEEN GROUPS. . . . . . . . . . . . . . . . . . . . . . . . 55

# List of Abbreviations

| | |
|---|---|
| **AS** | Ascension Library |
| **DoF** | Degree of Freedom |
| **dVSSS** | daVinci Surgical Skill Simulator |
| **MIS** | Minimally Invasive Surgery |
| **OR** | Operating Room |
| **QH** | QuickHaptics |
| **RAS** | Robot Assisted Surgery |
| **RMS** | Root Mean Square |
| **ROS** | Robotic Surgical Simulator |
| **VL** | Visualization Library |
| **VR** | Virtual Reality |
| **VRE** | Virtual Reality Environment |
| **WK** | Wykoby Library |

# Chapter 1

# Introduction[1]

Psycho-motor skills for surgery are hard to develop: usually, they are very technical movements that have to be carried out precisely and in a very restricted workspace (minimally invasive surgery).

The traditional surgical training phase is based on the apprenticeship model, developed by Halsted, of “learning by doing” [14]: along their studies, medical students and residents are able to deal with a great variety of clinical conditions. However, as real patients are involved, learning is always subordinate to the needs and safety of the patients [38] and experience possibilities are restricted to what the hospital “can offer”, in terms of diseases and procedures [50].

Furthermore, bureaucratic and financial issues have more and more limited training hours [55], and only attendants assessment, i.e. a subjective evaluation, is used to verify residents skill level.

For these reasons, current surgical training hallmark is the exposure of a great volume of general knowledge, rather than specific ones [55]. Thus, key challenges for training are:

- to provide effective learning without compromising patient safety;
- to be able to adapt to different skill levels: residency is 5 years, therefore, $5^{th}$ year residents will be definitely more capable than $1^{st}$ year ones;

[1] This chapter has been taken from the work of the author at [22].

- to quantitatively assess skill levels.

Simulation, i.e. the act of mimicking a real object, could be considered an optimal solution for surgical training [27]:

- it proposes a safe, error-free environment, so that the learners can fully understand to what consequences such a failure brings [38];
- it is attendant-independent, so that the learners can focus on entire procedures or just on his/her weaknesses.

Two classes of simulation can be identified: physical simulation and computer-based simulation, also known as Virtual Reality Environments (VREs).

## 1.1 Physical Simulation

It is traditionally present as wet and dry laboratories, where cadaver or animal models are used.

Cadaver models are very precise in terms of anatomy with respect to animal models, but they are definitely more expensive and of limited availability.

On the other hand, animal models are much easier to find, but an ethic committee is necessary to approve and validate their use. In addition, a large number of animals is required [6] [17].

Phantom-based models are another type of physical simulation: they provide the opportunity to practice surgical skills with realistic-pulsatile-bleeding tissues, teaching also how to deal with stressful and emergency situations [42]; however, they can be considered as static, i.e. they are not able to change depending on the users performance or ability [74].

## 1.2 Virtual Reality Environment

It overcomes many possible drawbacks of physical simulation: computer-based training is safe, can be independent from the presence of an attendant

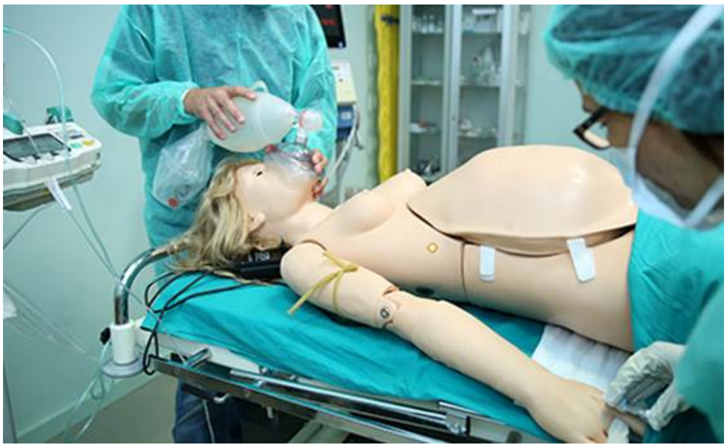


*Figure 1.1: Example of a Phantom-based model.*

(integrating guided tutorials), and, above all, it allows rehearsing in different procedures independently on actual patients diseases. Moreover, it is able to provide different scenarios complexities and allows a real time feedback and quantitative evaluation of the trainee [50] [55] [63].

On the other hand, it has not been fully demonstrated the efficacy of computer-based training on traditional surgery: high fidelity of the scene and implementation of procedures are current open problems [55] [7].

However, in the last decade, the advent of robotic surgery transforms teaching and training fields, due to different task execution and lack of haptic feedback in the system [48]. As result, the use of surgery simulators has been triggered, bringing to a fast and rapid growth of different systems.

Currently, the most used teleoperation system is the da Vinci® Surgical System (Intuitive Surgical® Inc., Sunnyvale, California): it is composed by a slave part, 4 robotic arms, and a master controller. In addition, it is equipped with stereoscopic view for depth perception, and pedals to switch among the instrumentations assembled on the robotic arms. The manipulators are not equipped with haptic feedback, so that the entire reliability of the procedure is based on visual feedback only [43].

Intuitive offers its own simulator; however, during the last few years, different companies have proposed other training systems: virtual reality simulators can give access to different simple (psychomotor) or complex (entire procedure) tasks using an off-site location available at any time, thus helping residents to fully understand and practice the gestures needed [42].

In robotic surgery field, a simulator can be considered almost essential because the trainee is not able to have the same point of view and approach as the operative attendants; moreover, in case of error from the trainee during a procedure, the attendant intervention can be delayed due to switching of the actual controller at the work station [5].

For these reasons, robotic surgical training is also even more hindered than traditional one.

## 1.3 Surgical Simulators

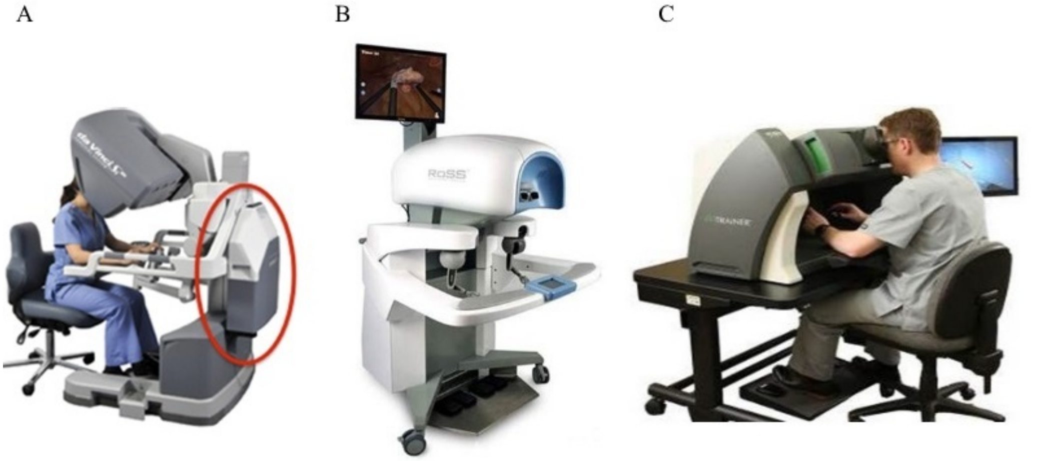


*Figure 1.2: Most used surgical simulators. In A the dVSS is displayed, the red circle highlights the actual simulator, which is a sort of backpack attached to the same exact system used during procedures; in B the RoSS system [41]; in C it is represented the dv-Trainer (Image courtesy of Mimic Simulation).*

Currently, five different surgical simulators, displayed in 1.2, are commercially available. The most used and studied simulators are the da Vinci Surgical Skill Simulator$^{®}$ (dVSSS; Intuitive Surgical$^{®}$ Inc., Sunnyvale, California), the dv -Trainer$^{®}$ (Mimic Technologies$^{®}$, Inc, Seattle, Washington) and the Robotic Surgical Simulator$^{TM}$ (RoSS; Simulated Surgical System LLC$^{TM}$, Buffalo, NY) evaluated on some main criteria connected to the concept of *validation* [44], i.e. "*the process of determining the degree to which a model or simulation is an accurate representation of the real world from the perspective of the intended uses of the model or simulation*" [4]:

- Face validation: linked to the accuracy of the simulator;
- Content validation: the utility of the simulator as a training system;
- Construct validation: ability of the simulator to recognize the user performance level, i.e. classify the user as expert or novel;
- Concurrent validation: quantitative correlation with the gold standard;
- Predictive validation: ability to predict forthcoming performance.

Face and content validities are subjective validation, because related to the own meaning of realism and coherence about training task of the trainees, while the others are able to provide an objective evaluation of the simulators.

### 1.3.1 da Vinci Surgical Skill Simulator®

The dVSS is a sort of backpack device mounted behind the da Vinci system itself. In this way, the trainee practices on the actual master controller that he/she will use in real procedures. However, this could be a limitation because the simulation modality is available only when the robot is not used in actual surgeries [17].

Training on this simulator has brought positive results compared to traditional training [37] [34].

### 1.3.2 dV-Trainer®

The dV-Trainer is composed by 2 haptic devices, so that force, tactile and pose feedback are provided at each hand, along with visual ones. 3D vision is realized using a simulator stereo eyepiece. Also this simulator has brought positive results as a training tool in robotic surgery [53], except for predictive validity, which has not been proved, yet; however, the dv-Trainer has the big advantage to provide a realistic workspace as the one of the daVinci Surgical System, since it is based on Mimic's simulation Technology (MSim$^{\text{TM}}$) [2].

### 1.3.3 Robotic Surgical Simulator$^{\text{TM}}$

The RoSS$^{\text{TM}}$ is one of the most recent surgical simulator approved, therefore, not so many validations have been carried out. It is modeled on the dVSS master controls: it is provided with two 6 DoF input devices, and stereo visualization for depth perception. Haptic feedback is possible thanks to the use of two haptic devices as master controllers. Only face, content and construct validity have been studied [61][62][68] with positive results.

During the last year, a new simulator has been assessed with face, content and construct validity as training surgical simulator: the RobotiX Mentor$^{\text{TM}}$, by 3D Systems$^{\text{TM}}$ (Simbionix Products, Cleveland, OH) is composed by stereoscopic vision and adjustable headset and foot pedals, but as opposed to the other simulators, it is equipped with non-fixed hand controls, as shown in 1.3 [73]. The main innovations brought by this simulator referred to the possibility to train different techniques, simulate complications and injuries, and train surgeons on visual cues only, enable them to rely on their vision only, as in the actual procedure. Therefore, no haptic feedback are provided.

Summing up, only two simulators effectively provide haptic feedback as training tool, thus not considering them as requisites for the correct and good execution of the surgery procedure. However, haptic information are fundamental for tissue characterization and palpation, and also for all that tasks involving tissue-tool interaction, where it is necessary to avoid tissue damages or internal bleeding.

Haptic feedback are tactile (cutaneous) and kinesthetic (forces) information, characterized by the unique bidirectional property: the haptic sense is able to provide information of the environment around us, but also to sense these interactions [45]. They can be implemented not only as a simple feedback, but also as an intelligent assistance (virtual fixtures), that could lead to improvement in task precision and execution [48].

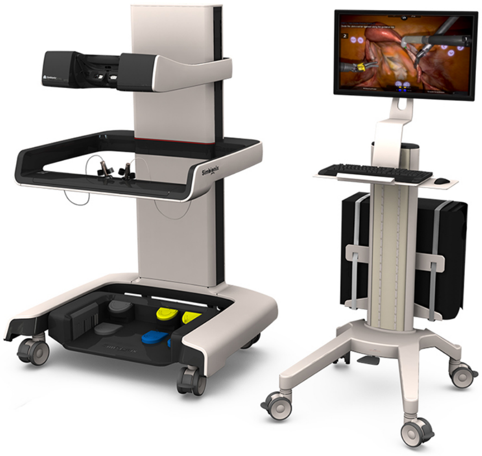


*Figure 1.3: RobotiX Mentor surgical simulator: it is provided with an external screen, too, so that the instructor can easily give feedback to trainers.*

Nevertheless, the absence of this feature in the actual robot teleoperation system is controversial in terms of procedure efficiency and accuracy: just getting used to visual clues, provided by high definition displays, surgeons are able to counterbalance the absence of haptic feedback during procedures [28]. However, in terms of motor learning, haptics is the fundamental sense used by infants to understand and learn the physical world around them, without visual control [59], and also growing up, the loss of the sense of touch leads to deformed and incorrect skill actions [56]. In addition, haptic feedback showed positive results in the rehabilitation field: especially virtual fixtures have been successfully implemented to accelerate re-learning of daily tasks [33].

Based on these results, this study aims to analyze and try assessing the possibility of benefits of haptic feedback in motor learning (and its effectiveness in surgical training). The major point of this validation is to assess motor learning using feedback that will not be used in the real task implementation: in different fields of study, augmented feedback enhance motor learning; the main issue is how to provide these feedback in order to be effective [35].

In this way, we do not want to influence the well established techniques and systems of minimally invasive surgery (MIS) procedures, i.e. teleoperation and laparoscopy, but we want to understand if haptic feedback can be actually learned and used to improve training of robotic surgery. This objective will be analyzed studying trajectory learning in a virtual environment, applying both haptic and visual feedback.

Since one of the main issues is *how* to provide visual and, especially, haptic feedback for an optimal learning [33], we firstly focused on understanding and developing the optimal experimental setup. Eventually, an actual training protocol will be tested.

The whole work has been divided into two thesis works, conducted by the same author in two different places. The first part took into consideration the development and first validation of the experimental setup; this work has been conducted at the University of Illinois at Chicago, IL, USA [22]. In the second one (the one here proposed), a training protocol is tested to evaluate and quantitatively assess possible benefits of haptic error augmentation in psychomotor skill learning. This second part has been developed at Politecnico di Milano, Italy.

# Chapter 2

# Background[1]

During teleoperated robotic surgery, the user controls a slave robot via a master controller that detects the pose, i.e. position and orientation, of his/her hands. In comparison with traditional MIS, the mechanical characteristics of teleoperation tools allow a natural wrist mobility, thus giving a better overall experience to the surgeon. Indeed, Robot-Assisted Surgery (RAS) increases precision and reduces hand tremors thanks to these features, too [48].

However, these differences bring to various tasks execution, differentiating motor control and skill learning from MIS, thus constraining surgeons to undergo a long and intensive training phase [31]. Moreover, as already said in Chapter 1, the lack of haptic feedback in the system can lead to complications in some very delicate and precise tasks that surgeons have to do in their normal procedure, such as suturing [48]. On the other hand, the majority of surgeons affirm that just getting used to visual clues, provided by high definition displays, they are able to counterbalance the absence of haptic feedback during procedures [28].

Thus, our purpose is to understand if motor skill learning, in an augmented virtual reality environment, can be improved if haptic feedback under the form of force fields are used.

[1] This chapter has been taken from the work of the author at [22].

## 2.1 Motor Learning

Richard Schmidt, one of the most important intellectual leader in motor learning and control, defined motor learning as "*a set of processes associated with practice or experience leading to a relatively permanent changes in the capability for responding*" [60].

This definition includes the main features of motor learning:

- "*Motor Learning is a set of processes*", thus, it leads to some changes, states or products, of acquiring the capability of responding; moreover, unlike other kind of activities, the processes here involved are hidden, i.e. not directly observable; therefore, they are inferred by changes in motor behavior. For these reasons, motor learning experiments have to be outlined so that differences in motor behavior reflect some associated changes in the internal states.

- Motor Learning is "*associated with practice or experience*"; the goal is to strengthen or increase the products of motor learning, maximizing the related skill capabilities.

- Motor Learning leads to "*relatively permanent changes*": with continuous and persistent practice, something is then embedded in the learner, so that he/she will interpret the learned activities differently than before. Thus motor learning is able to change the learner itself quite permanently.

Indeed, motor learning affects the brain, producing both structural and functional changes that go along with learning of optimal muscular patterns. These differences represent the *products* of motor learning, i.e. what is learned; the ways how we learn them, i.e. the *mechanisms* of motor learning, could be different depending on the type of training underlying the motor learning itself [32].

### 2.1.1 Products of Motor Learning

They can be divided into two main types of product: neural representation of trajectory and neural representation of transformation.

Neural representation of trajectory, i.e. trajectory learning, refers to the ability to target a reference trajectory with the end-effector, resulting in learning the trajectory itself. This process involves both spatial (static) and temporal (dynamic) features, which are controlled separately by our brain [26].

On the other hand, neural representation of transformation, also called adaptation, refers to the trajectory of the end-effector, thus, it is related to the joints chain that allows the end effector to move in a particular way. In this case, we can differentiate between dynamic (forces into movements) and kinematic (movements into movements) transformation. Combining the output of the transformations, an internal model is built: this model could be a forward or inverse model, depending whether motor movements or motor commands needed to achieve that movements are predicted [32].

### 2.1.2 Mechanisms of Motor Learning

Different mechanisms can give rise to products as explained above. The most known mechanisms are observational learning, use-dependent learning, reward-based learning and error-based learning [32].

Observational learning is based on skill demonstration so that the learner can see directly what he/she has to do; skill demonstration could be based on models, but also on videotapes or photos explanation [60]. However, observational learning can consider even proprioceptive of movement information, i.e. skin receptors, muscles or joints information [32]. This kind of training starts attracting researchers after mirror neurons theory took places in motor learning with positive results [29]; this is true also with unnatural movement, like the ones carried out by robots [24].

Use-dependent learning has the same underline concept of mirror neurons theory, but implies the user active role. It is basically behavioral changes induced by repetition of movements: indeed, according to Hebb theory, repeated movements increase synaptic efficacy, and also, due to plasticity of the nervous system, increase the creation of new synapses and deletion of those not used anymore [32] [30].

Reward-based learning and error-based learning both rely on the use of feedback, since it has been demonstrated that learning improves when qualitative ("right" or "wrong" movement) or, even better, quantitative (actual movement error) feedback are given to the user. Thus, reward-based learning depends on the use of feedback to let the user know about positive performances, while error-based learning is centered on giving feedback to the amount of error [32].

The relationships among products and mechanisms of motor learning are really complexes, so that no one-to-one connection has been established so far.

### 2.1.3 Stages of Motor Learning

Motor skills learning usually consists of performance of several training session until that skill is mastered. When this happens, it is memorized by the user, i.e. it causes a permanent change in the subject (see Section 2.1). During their training, learners seem to pass through different phases or stages, which are not strictly defined [60]: the two main variation are related to two or three stages.

The two stage process considers a first phase called *adaptation*, which involves understanding and acquisition of the neuromuscular pattern necessary to perform the task, and the *facilitation* stage, which is related to the improvement phase of the task [67] [25].

However, the most important and recent studies referred motor learning as based on three different stages: cognitive, fixation and autonomous [60] [21] [8].

- *Cognitive stage.* Since the trainee is new to the presented task, he/she has to figure out what he/she needs to do, in order to achieve the goal set; indeed, a large cognitive effort is required to determine the best and optimal strategy. For this reason, the task motor program is built in this stage.

- *Fixation stage.* It begins when the general neuromuscular pattern is then settled, and thus, the learner can focus on how to improve the skill, i.e. the motor program is refined in its details and error-detection mechanisms are enhanced. Thus, from the first stage to this second one, the learner switches from *which movement pattern is better* to *how to improve the pattern itself.*

- *Autonomous stage.* This is the last and long-term stage, which will come along the user after months or years of practice: the skill is basically automatic, i.e. the mental effort requested to the learner is approximately null, and the performance is generally high level, even including ongoing simultaneous activities that can be considered as interferences.

## 2.2 Inherent and Augmented Feedback

Feedback are fundamentals for the learner in order to improve performances and be able to understand if his/her training is efficient and effective.

Some of these movement-related-information can be directly bounded to learner sensory mechanisms. This kind of feedback is defined *inherent.*

Inherent feedback information can be connected to both performance evaluation, i.e. feedback after the action has been computed, but also to the action itself: sensors systems information could be enough to understand the immediate outcome of the task, thus giving a sort of run time feedback based also on experience. However, not all aspects of inherent feedback are so easy understandable: sometimes the trainee has to understand and actually learn feedback meaning and occurrence [60].

Opposite to inherent feedback, augmented feedback contain more information than inherent ones, or even augment the information content.

A very clear and intuitive definition of augmented feedback is presented in [65], where it is formalized as "*information that cannot be elaborated without* *an external source*". Different kind of feedback are possible, as summarized in 2.1.

*Table 2.1: AUGMENTED FEEDBACK.*

*Table comparison among the different types of augmented feedback. KR stands for knowledge of results; KP stands for knowledge of performance.*

| | | | |
|---|---|---|---|
| **Concurrent** | Given during the performance. | **Terminal** | Given when the performance is completed. |
| **Immediate** | Given runtime, i.e. as soon as it is processed. | **Delayed** | Given after a certain amount of time. |
| **Verbal** | Given under a spoken form. | **Non-verbal** | Feedback under a form not containing actual readable characteristics. |
| **Accumulated** | Given as summary of past events. | **Distinct** | Given distinctively for each instant of the performance. |
| **KR** | Given as information about the outcome of the performance in terms of movement. | **KP** | Given as information about the movement pattern and its nature. |

In different fields of study, augmented feedback can be effectively used to enhance motor learning: first of all, they can present information to the user, but avoiding the reward-punishment issue when assessing the performance. This means that the errors of the previous task execution are presented to the learners as hint to improve. Furthermore, it subsequently follows motivation function: since the trainee is pushed to understand and learn what was wrong in the execution, he/she is challenged to improve him/her -self [60].

However, the main issue related to augmented feedback is how to provide them in order to be effective [35]. Usually the learner is subjected to a large variety of different inherent feedback: they are physiologically given at any time to the body. For this reason, it is necessary to discriminate which of them are actually used and how.

In order to have a clearer experimental setup, researchers often alter the task environment, taking advantage of artificial augmented feedback, and limiting intrinsic one. In this way, the study is driven depending on fixed and well-defined variables [60].

Despite of the previous feedback classification, another easier and much more intuitive characterization of inherent and augmented feedback is based on the sensory system taken into consideration: visual, auditory and tactile. In addition to unimodal feedback, multimodal, i.e. combination of different sensory system information, are possible, and are used by people everyday, even much more efficiently than unimodal ones [65].

The last part of the chapter will focus on the main studies and their conclusions, focusing on this last division; as already said, we can differentiate among visual, auditory, haptic or multimodal feedback. However, even if auditory feedback could be powerful because they allow to stay focus on the patient while operating, and have shown positive applications in surgery field [49], in the Operating Room (OR) there are already a large amount of sounds mainly used to monitor the patient vital signal; thus stating this feedback modality as meaningless in surgery [10].

For this reason, only visual, haptic and multimodal feedback will be considered. A brief introduction on these feedback modalities and the main associated studies and research evidences will be carried out.

### 2.2.1 Visual feedback

Visual information are considered as the most important perception information in daily life [65]; indeed, it is the sensory modality more investigated in motor learning. The main mechanisms of motor learning applied are observational and use-dependent learning: using informational videos or miming actions, the user can easily learn tasks, just by following tips and movements from experts.

Visual augmented feedback, instead, is a much more open field. Different kind of augmented feedback are possible (see also 2.1): curves, graphs, plots, or simple scores [46] [16], or even *avatar*, under the form of superposition or side-by-side 3D perspective of a reference on the users part, can be employed [72]. For this last type of feedback, the efficiency is related on the amount of body part superimposed and on the perspective given to the user (first/third person visualization).

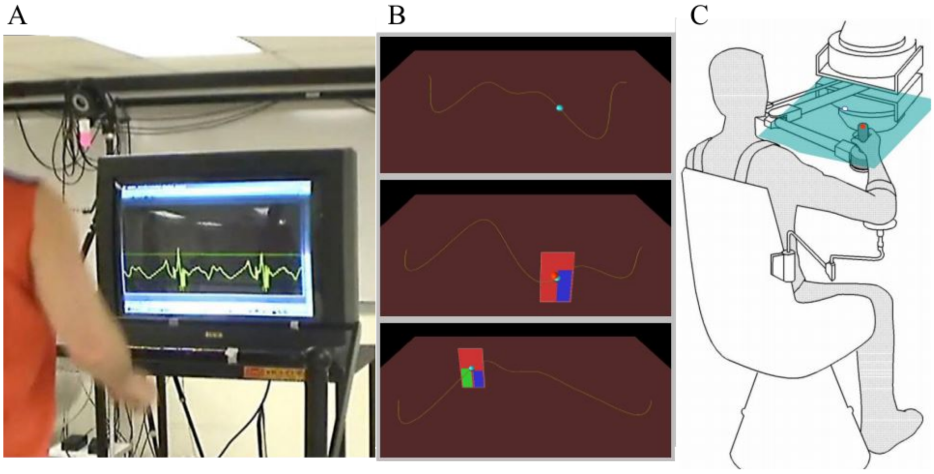


*Figure 2.1: Visual Feedback Modalities. In A the performance is summarized and given to the user as a graph [16]; in B the feedback is represented as a curve (©2007 IEEE), while in C there is an example of superposition: the arm of the user is hidden by a screen where will be displayed the fake movement. (©2011 IEEE)*

The effectiveness of visual feedback is mainly related to task complexity: a task is defined complex if it "*cannot be mastered in a single session, has several degrees of freedom, and perhaps tend to be ecologically valid*" [75]. From the great and large variety of studies based on visual feedback applied to motor learning, it has been shown that visual augmented feedback effect is determined by task complexity and user level of experience: concurrent visual feedback seem much more adequate when used along with complex tasks than with simple tasks [65].

Regarding surgical field, visual feedback are nowadays fundamental during surgery, since no other kind of feedback are applicable. In particular, surgeons rely on tissue deformation, so that, increasing visualization accuracy, the chance of error decreases. Indeed, 3D video-camera or the use of the so called *visual haptics*, i.e. replacement of haptic feedback with visual representation of forces exerted by the user, have shown to be useful and adequate substitutes of actual haptic feedback [10] [71].

During training phase instead, a large variety of visual cues is used: graphs and scores are currently implemented in all the major surgical simulators to give quantitative performance feedback [66]. They are also equipped with videos and bullet lists, which help to understand the entire work flow of the procedure, and even Hands-on instruction, where the trainee has to follow a target avatar [34] [61].

### 2.2.2 Haptic feedback

As already explained in Chapter 1, haptic feedback refers to both force and tactile feedback; however, motor learning studies apply it only as a force one, since tactile information are still difficult to correcrtly reproduce [65].

Different haptic augmented feedback can be used to improve and facilitate motor learning and training. Usually, they are used as *active constraints* in trajectory learning. Active constraints are cooperative control strategies applied to manipulation tasks to assist or enhance difficulty of the regulating motion, i.e. the hand movement detected by a proper haptic device. They can be classified as (see also 2.2):

- *Rigid position control*: the trainee movement is basically intended as an external disturb that has to be counteracted; it consist of a force application that oblige the user to follow a default movement.

- *Guidance constraints*: the trainee is pushed by convergent forces toward the target point or trajectory. This modality is also called convergent force field application, as the name better and easily explain the aim of the constraint.

- *Repulsive constraints*: the trainee is pushed by divergent forces away from the target point or trajectory. This modality is also called divergent force field application.

Studies based on rigid path control have shown that such a strict approach can be even detrimental in motor learning: completely avoiding the possibility of mistakes, the trainee will not focus on the task and thus, the performance would not improve; these results also prove that human beings learn by making mistakes [52].

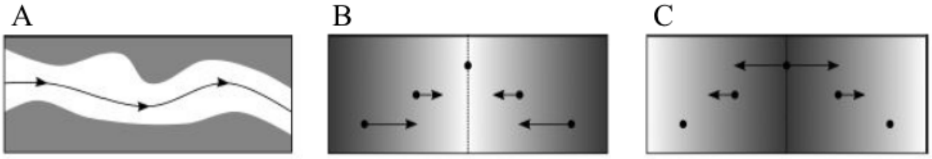


*Figure 2.2: Haptic Constraints Modalities. In A it is represented the rigid position control concept: the user is forced to strictly follow the trajectory, in B and C there are represented a convergent and divergent force field respectively. ©2013 IEEE.*

Convergent force fields represent a very similar concept of position control; however, the user is much more free to explore the environment around the target, but, in the same time, is literally guided along the ideal path. This modality has been largely studied, since it has been expected that having an active assistant that guides you along the correct path could improve learning. Nevertheless, different studies proved that even convergent guidance, i.e. convergent force field application on trajectory learning, does not improve trainee performances [40]. This failure seems to lie on the *guidance hypothesis* concept: during training, the trainee tends to develop a dependency on the haptic feedback, so he/she is then prevented in acquiring the new skill [57].

On the other hand, the opposite approach, i.e. divergent force field application, seems promising, challenging more the trainee and following the concept of learning-by-making-mistakes that drives the human motor learning. Indeed, this modality is inspired by the error augmentation idea, where visual or haptic cues related to errors are magnified, displaying a distorted result. This approach has shown positive results in rehabilitation [72] [52], but also some good conclusions are related to skill learning: in [40] Lee, J. et al. studied the application of force feedback disturbance and noise-like disturbance in learning a 2D trajectory task. They compared these results with haptic guidance and a control group, who executed the training without any force feedback; they concluded that, as already known, haptic guidance is ineffective for motor learning, while haptic disturbance, and in particular, noise-like forces, benefits motor learning.

Summing up, divergent force fields could be beneficial or inefficient for motor learning: inadequate haptic disturbance could lead to useless training procedures, showing that the motor learning method should be fitted on the nature of the motor task itself.

Following the foregoing researches, also a hybrid approach has been developed: in [39], haptic guidance is exploited in the early stages of motor learning in order to improve understanding of the task, while haptic disturbance takes place when user performance increases, creating a motor skill training setup based on user learning itself. However, this approach has not shown any significant outcome compared to no assistance, haptic guidance only or haptic disturbance only (2.3).

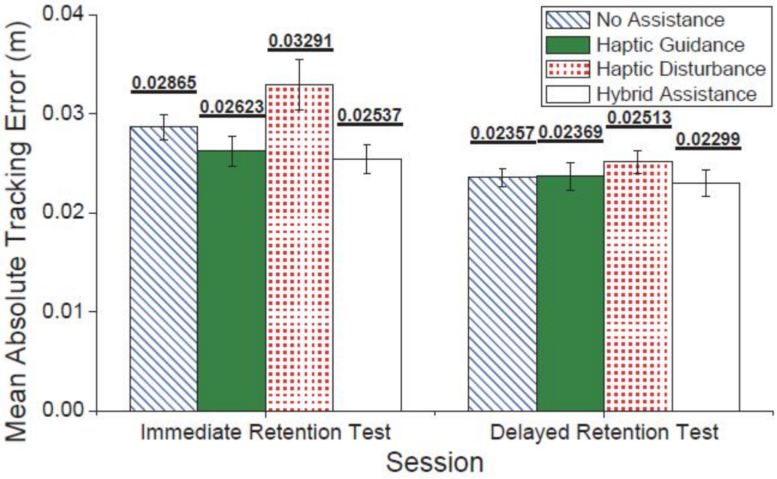


*Figure 2.3: Hybrid Approach Test Results. In both immediate and delayed retention tests the hybrid approach does not show any evidence of better performance. ©2014 IEEE.*

Regarding surgical training, haptic feedback has been applied in different VR simulators with promising outcomes, even if there is not an absolute consensus on how much it could be eventually effective: tissue consistency information is fundamental to avoid tissue damages; this information can only be delivered as haptic feedback. The main issue is which is the best ”force quantization” necessary to produce a meaningful and useful feedback [70].

### 2.2.3 Multimodal Feedback

Up to now, it has been proved the efficacy of unimodal feedback on motor learning in different type of implementation and with different modalities. However, people are usually subject to different stimuli at a time, and they process them together, thus being able to catch and receive different combined information. Indeed, multimodal perceptions are usually faster and more precise than unimodal ones, due to decrease of cognitive load. Moreover, different aspects of motor learning are supported simultaneously. All these advantages are possible thanks to the distribution of information processing. hHwever, few studies have been validated on the effectiveness of multimodal feedback in motor learning [65].

Nevertheless, visuohaptic feedback in trajectory learning proved to be more effective than visual information only, reducing spatial errors [11]. Also in rehabilitation, the use of modulated haptic feedback along with visual ones, has been effective in terms of assist-as-needed application and patient-cooperation control strategies [19].

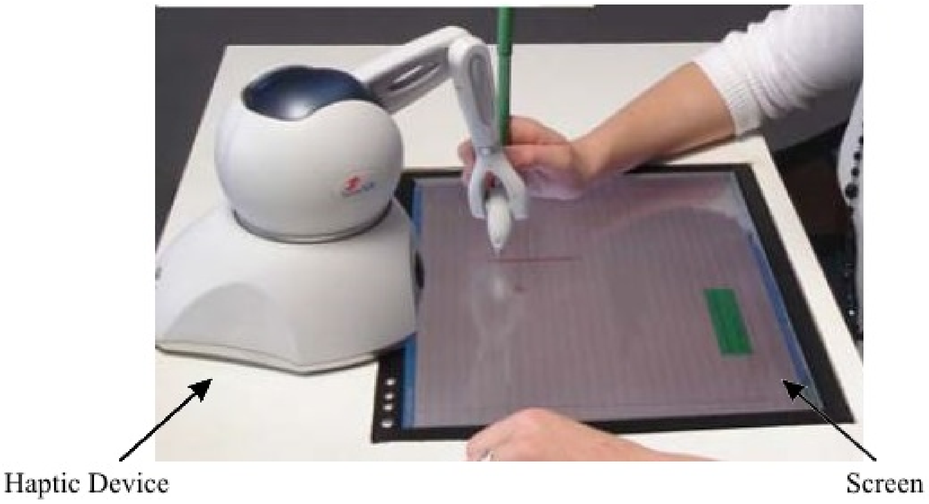


*Figure 2.4: Visuohaptic feedback example. A haptic device is used along with a screen, so that visual and haptic information are assess together and combined by the user to improve trajectory learning under the form of letters. ©2008 Bluteau et al.*

In surgical training, the combination of visual and haptic cues has been tested without significant results: this type of feedback could combine the stated use of visual information along with new and much more detailed tissue-related data, improving capabilities of residents and even expert surgeons faster, when learning teleoperation [70]. However, the main limitation is still the lack of knowledge present in haptic implementation in surgery where adequate and really precise force information are needed.

In conclusion, augmented feedback, both visual and haptic, can be effective in motor learning. However, lots of problems and issues still persist in trying to understand their actual optimal implementation and use. Currently, some new approaches are rising from rehabilitation fields: the so called limit-push condition is inspired by the error augmentation concept. The purpose is to bring the trainee to learn the boundaries of a safe area, otherwise, his/her hand would be pushed away by a divergent force field. This study showed that trainees were able to re-shape motion distribution according to the safe volume of movement [64].

This implementation further brings our attention towards the concept of learning by making mistakes: since the limit-push condition has been studied only in rehabilitation, we would implement it in a trajectory learning experiment. In addition, we would like to test another force field implementation that combines a divergent and a convergent force field, called *Limit-Trench* condition.

This idea comes from the fact that making errors improve user concentration on the task, but could also lead to frustration and discouragement, that are instead counter-productive for motor learning [32]. Therefore, we will exploit two different force fields, whose application is function of the distance from the target trajectory: the force feedback will push the user away from the trajectory up to a certain distance threshold, if the performance is good, while it will drive him/her toward the divergent area if he/she is too far from the trajectory. The concept is better explain in 2.5, and it will be further analyzed in the next chapter.

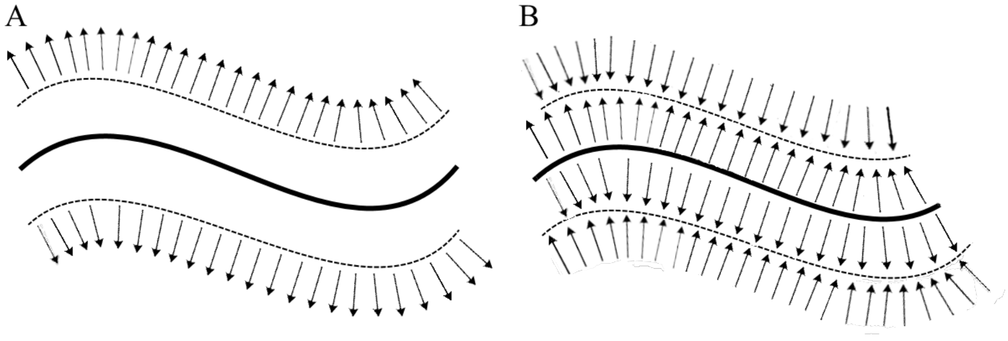


*Figure 2.5: Limit-Push (A) and Limit-Trench (B) concept representation. The thick line represents the ideal trajectory, arrows represent the force field direction, while the dashed lines define the distance threshold used to apply or switch force.*

# Chapter 3

# Methods[1]

This work aims to develop and assess an innovative experimental setup for psychomotor skill training. The whole work has been divided into two main steps:

- creation of the experimental setup in a flexible virtual reality environment;

- validation of haptic error augmentation in a following-the-trajectory task.

In order to achieve these main objectives, the experimental setup has to consider both graphic and haptic elements. For this reason, a high definition stereoscopic display, able to provide 3D visualization, along with a haptic device have been used.

The haptic device is a Touch$^{\mathrm{TM}}$ 3D Stylus (3.1), provided by 3D Systems$^{®}$ (Rock Hill, SC, USA), whose characteristics are summarized in Table 3.1. It is an impedance control device, i.e. it senses a position and commands a force. From a software point of view, the Touch 3D stylus interfaces with a haptic library, which is, however, limited in graphics rendering. For this reason, an external graphic library has been implemented to supply this lackness.

In addition, a computational geometry library have been used to speed up forces and complex curve computation. Indeed, in order to challenge enough

[1] This chapter has been taken from the work of the author at [22].

the trainee, and effectively allow training, splines curves have been employed.

This work has been settled merging these components in the so-called *LACE library* [20] [54] [69].

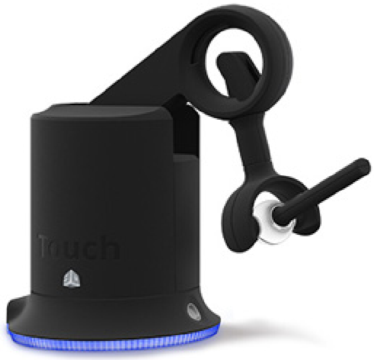


Figure 3.1: Touch 3D Stylus.

Table 3.1: TECHNICAL SPECIFICATION TOUCH 3D STYLUS.

| | |
|---|---:|
| Type | Touch 3D Stylus |
| Positional Feedback | 6 (complete pose) |
| Force Feedback DoF | 3 (position only) |
| Force Feedback workspace (WxHxD) | 265 x 241 x 89 mm |
| Maximum Force | 3.4 N |
| Nominal position resolution | 0.084 mm |

## 3.1 LACE Implementation

The implementation of the libraries has resulted in the creation of a platform among four different software-hardware environments:

- *QuickHaptics*$^{TM}$ (QH): the MicroAPI built upon Geomagic® OpenHaptics$^{\text{TM}}$, which is the software development toolkit provided for the haptic devices of 3D Systems; it allows to easily create a graphic and haptic application [3].

- *Visualization Library* (VL): the external graphic library used to allow a better trajectory visualization, and to generally improve the standard graphic rendering brought by QH [12].

- *Wykobi Computational Geometry Library*© (WK): a 2D-3D computational geometry library very useful to improve math calculation [51].

- *Ascension* (AS): the library required to integrate the 3D Guidance electromagnetic tracking unit (composed of both transmitters and 6 DoF sensors) into the application (*Ascension* Technology Corporation©, Shelburne, VT, USA) [1].

In order to be able to synchronize graphics, haptics and computational rendering, and the electromagnetic tracking system, it has been necessary to re-organize and merge the work flow of all the components in a new structure composed by both software and hardware parts, as represented in 3.2.

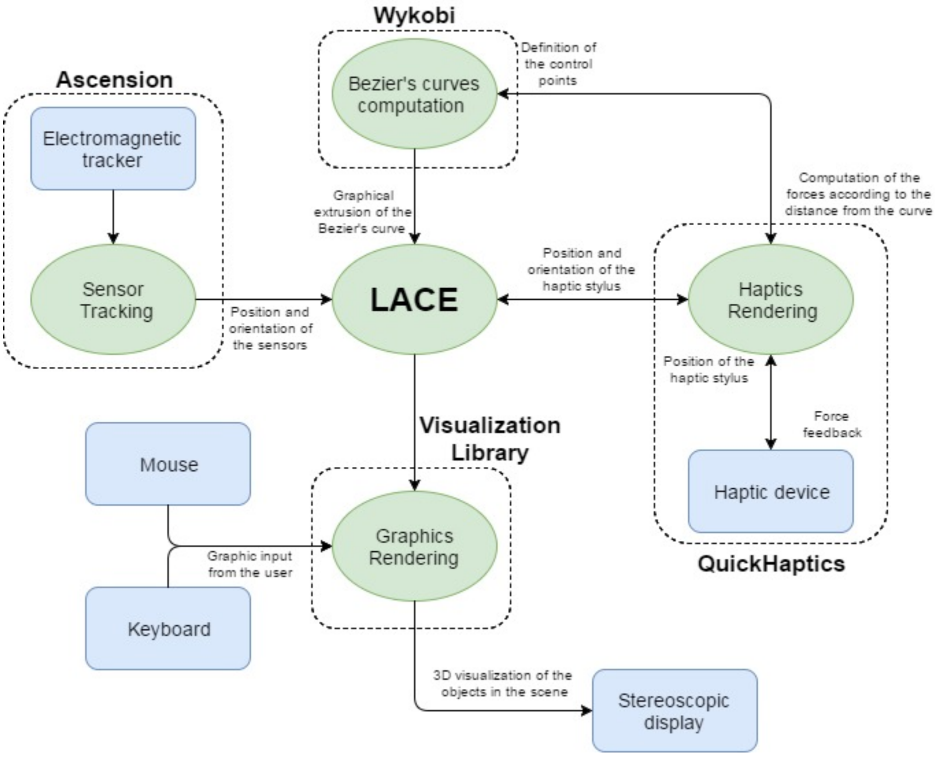

Wykobi
Bezier's curves computation
Definition of the control points
Ascension
Electromagnetic tracker
Sensor Tracking
Position and orientation of the sensors
Graphical extrusion of the Bezier's curve
Computation of the forces according to the distance from the curve
LACE
Position and orientation of the haptic stylus
Haptics Rendering
Position of the haptic stylus
Force feedback
Haptic device
QuickHaptics
Visualization Library
Mouse
Graphic input from the user
Keyboard
Graphics Rendering
3D visualization of the objects in the scene
Stereoscopic display

*Figure 3.2: LACE logic*

LACE is organized on two different renderings that communicate thanks to LACE itself. As already said, the graphics rendering is handled by VL: in this way there are increased the user possibilities in terms of visualization, because VL allows stereoscopic view, creating full 3D environments, but also in terms of interaction, since keyboard and mouse inputs can be easily implemented.

On the other hand, the haptics rendering is handled by QH: here forces are managed differently, depending on the goal the programmer has to achieve. The interconnections between QH and VL allow to map the haptic device in the graphic environment, thus creating interactions between the renderings.

As an useful extra features, WK library boosts performances in terms of geometry computations, thus making the haptics rendering and the user-interaction faster and smoother, i.e.without delays between graphics and haptics.

In addition, the Ascension system inserts a tracking item, i.e. the electromagnetic sensors, so that an exhaustive and complete platform is built for a large variety of augmented virtual reality applications.

The communication between the four libraries is done thanks to LACE library classes. Based on their functions, they can be divided in four groups, as also represented in 3.3:

- *Renderable Objects Classes*

  These are the classes for the creation of haptically and visually renderable objects. All of them are derived from the base class (LACE Object) that allows the communication between QH and VL. Different types of objects are available in LACE, from simple geometries, such as spheres, cones and cylinders, to more complex ones, like meshes, volumes and extrusions.

- *Tracking System Classes*

  This group includes the LACE classes that are responsible for the communication with AS.

- *Special Forces Classes*

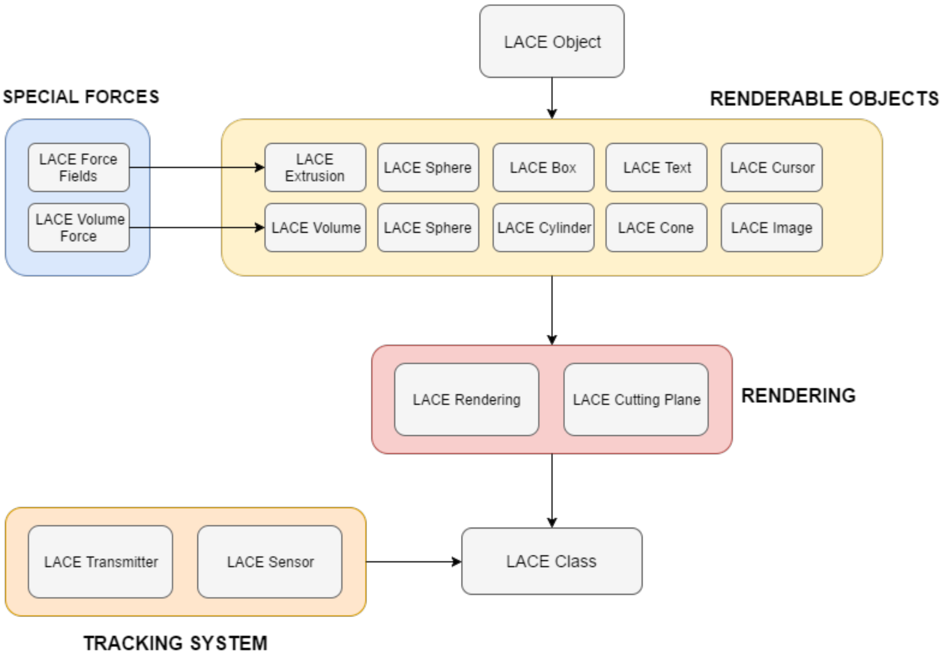


*Figure 3.3: LACE Classes*

LACE VolumeForce and LACE ForceField are two classes that can be associated respectively to a LACE Volume and a LACE Extrusion in order to define variables or functions used to render force fields.

- *Rendering Classes*

  LACE library allows the creation of multiple renderings in the scene. Each rendering is defined as a LACE Rendering class instance, which contains both a list of the objects to be rendered and all the rendering parameters, i.e. camera and viewport parameters. In this group it can be included also the class LACE CuttingPlane that implements the possibility to cut one or more LACE Object instance.

The class LACE Class is not included in this division because it is a *singleton* class that contains pointers to all the other classes, handling their initializations and updates.

All the objects that can be graphically created in LACE inherit from the same base class: LACE Object. A scheme of LACE Object class organization and interaction with QH and VL is reported in 3.4.

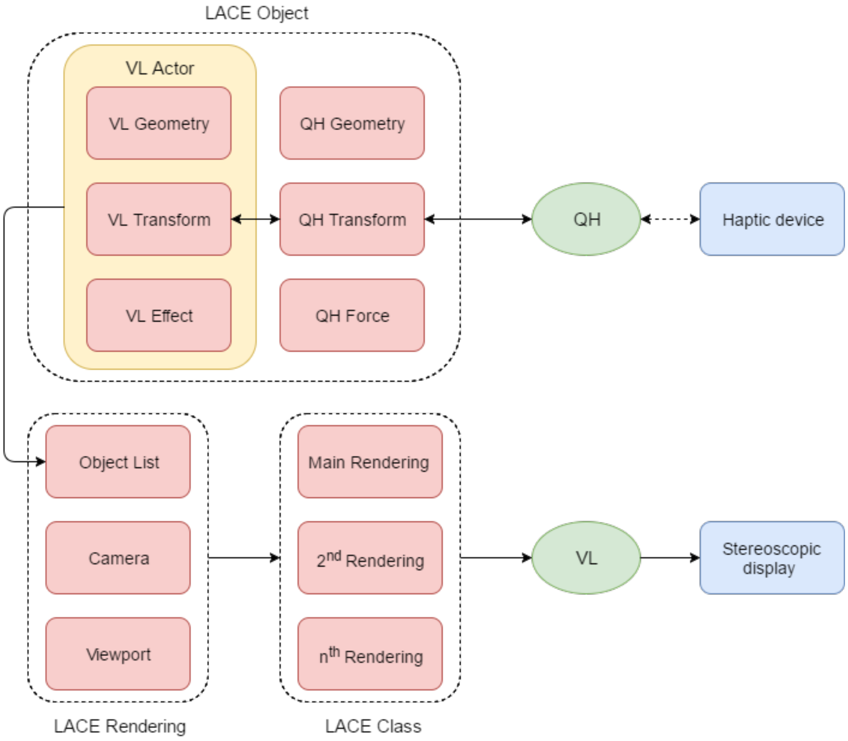


*Figure 3.4: LACE Object organization*

Each LACE Object contains three main elements:

- an Actor, linked to a geometry, i.e. the specific created shape;
- a Transform, that define the pose of the object;
- an Effect, to be able to tune and customize the object (changing color, visual effects of texture and materials).

Moreover, LACE Object class contains the pointer to the corresponding QH shape, so that also the properties of the shape in QH can be accessed and tuned.

The presence of a common base class among all the objects, containing shape-specific variables and parameters, is needed to properly render the shape in the scene and to allow the communication between VL and QH in a standardized way. The graphics rendering of all the objects is obtained by adding all the created objects to the main rendering. On the other hand, the haptic properties of each object can be defined using QH functions, called passing through the QH shape pointer.

The LACE Object class contains the two main members that allow to control VL and QH exchange of information: VL Transform and QH Transform. They are the transformation matrices that define the rotation, translation and scale (i.e. the pose) of each object in VL and QH respectively. To make the two environments consistent, they always need to be equal, therefore a connection mechanism reflecting their variations is needed.

There are two main circumstances where an object transform can change (see 3.5). The first one is when an object is created for the first time in the scene: in this case the transform takes into account an object-dependent transformation, defined by default just for some specific objects for rendering purposes, and an eventual user-defined initial transformation. Then, the communication process defines VL Transform first and sets QH Transform accordingly, if QH is used. However, the transformation matrix can change also as consequence of one particular event while the application is running; this second scenario is mainly related to the graphic cursor, which could be either the haptic device or the electromagnetic tracker: at each graphic frame, either the QH Transform or the AS Transform are automatically updated by QH or AS respectively, according to the typology of the defined cursor. The VL Transform is the last one to be updated.

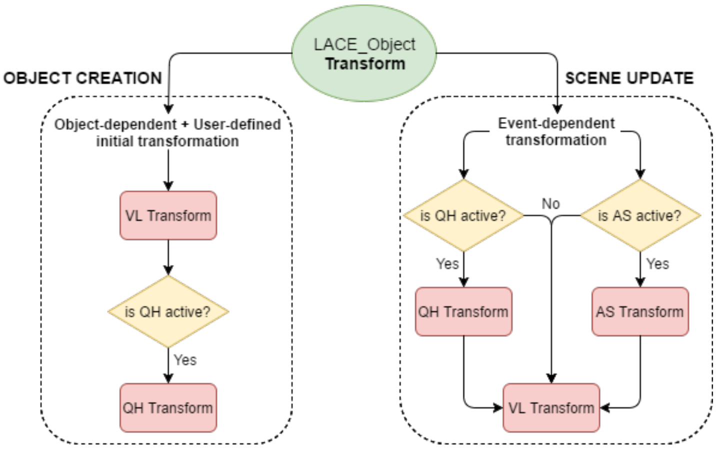


*Figure 3.5: LACE Transform*

Since the proposed approach is based on determining run-time the error computed by the user as distance between the tip representation of the haptic device and the ideal trajectory, the curves have to be defined computationally low-cost, but in the same time, they have to be challenging enough for the user, thus providing articulate trajectories and reliable force feedback simoultaneously.

For these reasons, spline curves have been implemented, specifically, *Bezier curves*, which are a particular type of parametric curve used in computer graphics, designed by using few control points. Each of these points is defined by a position on the 3-dimensional work space and delineates the path direction [36].

On the other hand, forces are based on force fields application, in particular, linear force fields.

## 3.2 Bezier Curve

Bezier Curves are B-splines curves, i.e. basis spline; a B-spline is defined as a linear combination of $n$ control points $\mathbf{P}_i$ and B-spline basis functions $N_{i,k}(t)$:

$$\mathbf{C}(t) = \sum_{i=0}^{n} \mathbf{P}_i N_{i,k}(t) \tag{3.1}$$

Control points are points of a set used to define the spline itself [58], while the *i-th* B-spline basis function of degree $k$ is defined recursively using the *Cox-de Boor recursion formula* as:

$$\begin{gathered} N_{i,0}(t) = \begin{cases} 1, & \text{if } t_i \leq u \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,k}(t) = \frac{t - t_i}{u_{i+k} - t_i} N_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) \end{gathered} \tag{3.2}$$

This formulation basically relates each basis between two consecutive control points: basis function $N_{i,0}(t)$ is 1 if $t$ is in the *i-th* span $[u_i, u_{i+1})$ [9].

Bezier curves can be represented using this notation, and their basis functions are Bernstein Polynomials of general $n$ degree:

$$b_{i,k}(t) = N_{i,k}(t) = \binom{k}{i} t^i (1-t)^{k-i} \tag{3.3}$$

Therefore, 3.1 becomes:

$$\mathbf{B}(t) = \sum_{i=0}^{n} \mathbf{P}_i b_{i,k}(t) \tag{3.4}$$

Depending on the number of control points, we can differentiate among linear, quadratic, cubic or higher order Bezier curves; using the *de Casteljau* algorithm [9], we can easily define them as:

$$\begin{gathered}
\mathbf{B}(t) = (1-t)\mathbf{P}_0 + t\mathbf{P}_1, \qquad 0 \le t \le 1 \\
\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2, \qquad 0 \le t \le 1 \\
\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3, \qquad 0 \le t \le 1
\end{gathered} \tag{3.5}$$

The de Casteljau algorithm defines how to split a Bezier curve $P_{[t_0,t_2]}$ into two segments $P_{[t_0,t_1]}$ and $P_{[t_1,t_2]}$ whose union is equivalent to $P_{[t_0,t_2]}$. This is a very important and useful property of Bezier curves, which allows to easily compute different curves just using a recursive algorithm.

In 3.6 there are displayed three basic examples. From this figure, it is also easy to understand how the de Casteljau algorithm works. For example, taking into consideration the quadratic Bezier curve, for each segment created connecting two consecutive control points, we consider a point $Q_n$ in the interval $[1-t, t]$. On the consecutive segment, we have to find the point $Q_{n+1}$ at the same ratio. Then, connecting these two points, the point belonging to $B(t)$ is a point of this last segment and at the same ratio of the other two. This is the point on the curve at the specified value of $t$.

Bezier curves are characterized by some interesting and convenient properties:

- the curve begins at $\mathbf{P}_0$ and ends at $\mathbf{P}_n$;
- if and only if all the control points are collinear, the curve is a straight

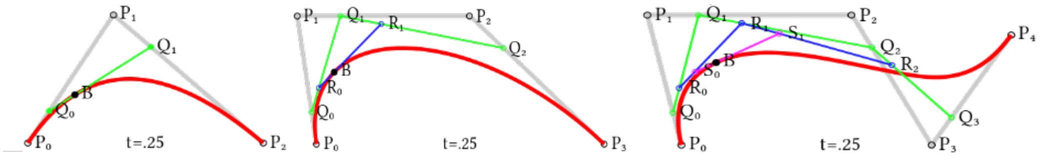


*Figure 3.6: Bezier Curves Construction of a quadratic (left), cubic (center) and quartic (right) Bezier curve.*

line;

- a curve can be split at any point into two subcurves, or into arbitrarily many subcurves, each of which is also a Bezier curve;

- every degree $n$ Bezier curve is also a degree $m$ curve for any $m > n$.

Since composite Bezier can give rise to a large variety of different curves, and in order to challenge adequately the user, the trajectory implemented would be based on two consecutive and connected Bezier geometries. In particular, task trajectories are built as two connected cubic Bezier curves, guaranteeing continuity up to the second order.

Having two Bezier curves of the same order: $S(t), Q(t)$ defined respectively by $(S_0, S_1, ..., S_n)$ and $(Q_0, Q_1, ..., Q_n)$, $C^0$-continuity, i.e. continuity of position, is achieved by setting last control point of the first curve coincident with the first one of the second curve [18]:

$$\mathbf{S}(1) = \mathbf{Q}(0) \Rightarrow \mathbf{S}_n = \mathbf{Q}_0 \tag{3.6}$$

In order to achieve $C^1$-continuity (tangent continuity), also the first derivatives at the two points $S(t), Q(t)$ has to be guaranteed. Specifying to a cubic Bezier case:

$$\mathbf{B'}(t) = -3(1-t)^2\mathbf{P}_0 - 6(1-t)t\mathbf{P}_1 + 3(1-t)^2\mathbf{P}_1 - 3t^2\mathbf{P}_2 + 6(1-t)t\mathbf{P}_2 + 3t^2\mathbf{P}_3 \tag{3.7}$$

From 3.7 it is possible to compute $C^1$-continuity condition:

$$\mathbf{S'}(1) = \mathbf{Q'}(0) \Rightarrow \mathbf{S}_3 - \mathbf{S}_2 = \mathbf{Q}_1 - \mathbf{Q}_0 \tag{3.8}$$

Thus, combining 3.6 and 3.8, $\mathbf{Q}_1$ is set in terms of the first curve as:

$$\mathbf{Q}_1 = 2\mathbf{S}_3 - \mathbf{S}_2 \tag{3.9}$$

Eventually, following the same logic, $C^2$-continuity, i.e. curvature continuity, is guaranteed by $C^1$-continuity and equivalence of second derivative at the end of first curve and start of the second one:

$$\mathbf{B}\text{''}(t) = 6(1-t)\mathbf{P}_0 + 6t\mathbf{P}_1 - 12(1-t)\mathbf{P}_1 - 12t\mathbf{P}_2 = 6(1-t)\mathbf{P}_2 + 6t\mathbf{P}_3 \tag{3.10}$$

$$\mathbf{S}\text{''}(1) = \mathbf{Q}\text{''}(0) \Rightarrow \mathbf{S}_3 - 2\mathbf{S}_2 + \mathbf{S}_1 = \mathbf{Q}_0 - 2\mathbf{Q}_1 + \mathbf{Q}_2 \tag{3.11}$$

and combining 3.6, 3.8 and 3.11, it is possible to define $\mathbf{Q}_2$ coordinates as function of $\mathbf{S}(t)$:

$$\mathbf{Q}_2 = 4\mathbf{S}_3 - 4\mathbf{S}_2 + \mathbf{S}_1 \tag{3.12}$$

3.7 represents an example of two cubic Bezier curve composition.

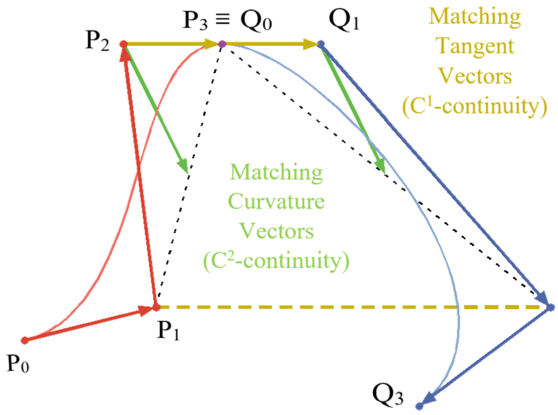


*Figure 3.7: Composite Bezier Curve. In yellow there is the graphical interpretation of tangent continuity, while in green the curvature one.*

### 3.2.1 Graphic Rendering

Bezier curves are the mathematical tool used to create the trajectories, however, a graphic element is necessary in order to display them.

Exploiting VL, the curves are rendered by extruding a circular shape along the mathematical function (see 3.8); in order to clearly identify start and end of the trajectory, two spheres are used.

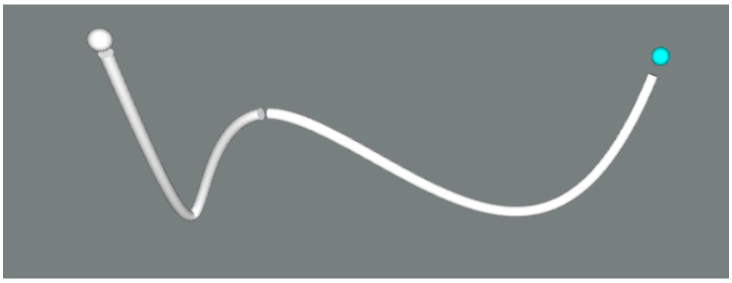


*Figure 3.8: Example of an extrusion from a circular shape along a Bezier curve. Start and end points are marked with spheres.*

Extrusion is a process used to create objects of a fixed cross-sectional profile [47]. In addition, exploiting a stereoscopic display, the trajectory visualization is even increased, since depth perception allows a better interpretation of a larger number of trajectories.

For this reason, an *Extrusion class* has been implemented in LACE Library. From this point of view, VL improves the implementation consistently: since in QH only simple shapes can be built, it would be really heavy computational trying to display such curves just by lines, cylinders or series of points, thus slowing the overall program. However, in order to apply forces consistently with the trajectory itself, it has been necessary to synchronize the two libraries.

LACE Extrusion is able to compute the Bezier needed just using the control points as input. In particular, LACE implements automatically continuous Bezier, if needed, applying continuity up to the second order. In this way, the variety of possible trajectory increases consistently, since they will be longer (one, two or more extrusions), thus, also increasing the overall task complexity.

The LACE Extrusion class structure enables both the creation of the extrusion itself, but also insert graphics and haptics rendering of the control points by using spheres. In this way the user can interact and have a better general view on how the control points have to be placed to obtain a predetermined path.

## 3.3 Force Fields

Force fields are haptic applications belonging to the category of active constraints, or *virtual fixtures* [48]. Generally, active constraint implementations are based on three main processes, that, put together, give rise to the framework in 3.9 [13].

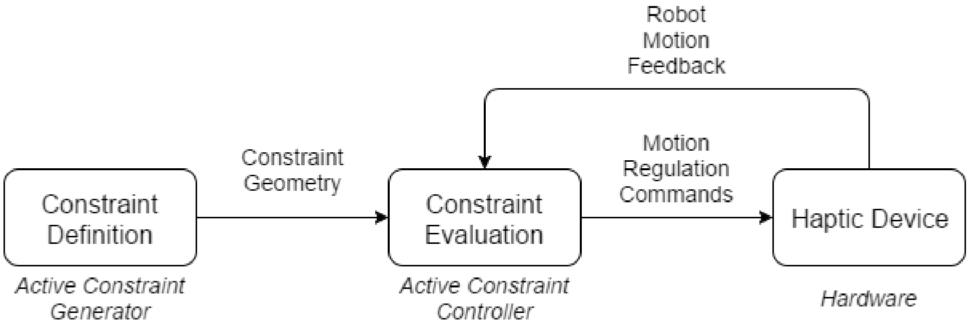


*Figure 3.9: General scheme of an active constraint implementation*

The first process is the definition of the constraint itself. It creates the computational representation, i.e. a geometry, of the constraint as output. The second stage allows the haptic device to evaluate the constraint. Eventually, the regulation can be carried out as third stage: here the software implementation is translated in hardware commands, so that the motion of the trainee could be regulated and modified.

## 3.4 Implementation

Exploiting the great number of LACE functionalities, the psychomotor skill learning application has been based on force field applied to spline curve trajectories. Since the user will receive run-time haptic and visual feedback, WK library has been used, helping to avoid delays. In this way, force computation is always fast and precise, and feelings among the two main sensory modalities here taken into account, i.e. visual and tactile, are well matched.

The logic behind the overall implementation is showed in 3.10: the application runs on a Windows platform where the graphic and the haptic threads run concurrently, but at different speeds (60 Hz for graphics and 1000 Hz for haptics).

The main elements are the trajectory (i.e. the extrusion) and the cursor, both displayed visually and haptically. The graphics rendering enables the visualization of the correct meshes and texture that create the scene. A stereoscopic display has been used in the preliminary phase of testbed implementation. The haptics rendering is focused on the Touch 3D Stylus device: the current cursor position is read via QH library. Considering the extrusion function, the distance between the trajectory and the current position is computed. Depending on this operation, a force feedback is given run-time to the haptic device. In the same time, also a visual feedback is given, as change of color of the trajectory as function of the trainee's performance.

Since distance from the trajectory also represents the trainee's error, this values is stored everytime, and it will be then used for data analysis.

Furthermore, in a first testbed implementation, some predefined keyboard input are used to easily tune force field parameters and study the effect of different applications.

In order to facilitate haptic dexterity, the mesh applied to the cursor has been chosen to help the self-evaluation of the task: a hollow sphere, with slightly bigger dimensions than the extrusion shape, has been placed so to match its center with the tip of the haptic default cursor. In this way, the user will carry out the task by having a clear visual feedback on his/her performance, since a good task would be equal to have overlapped sphere and trajectory, as displayed also in 3.11.

An additional visual feedback has been implemented to further help the user during task execution (3.12): depending on the error made, the trajectory will change color from green, i.e. low or null error, to red, i.e. high error.

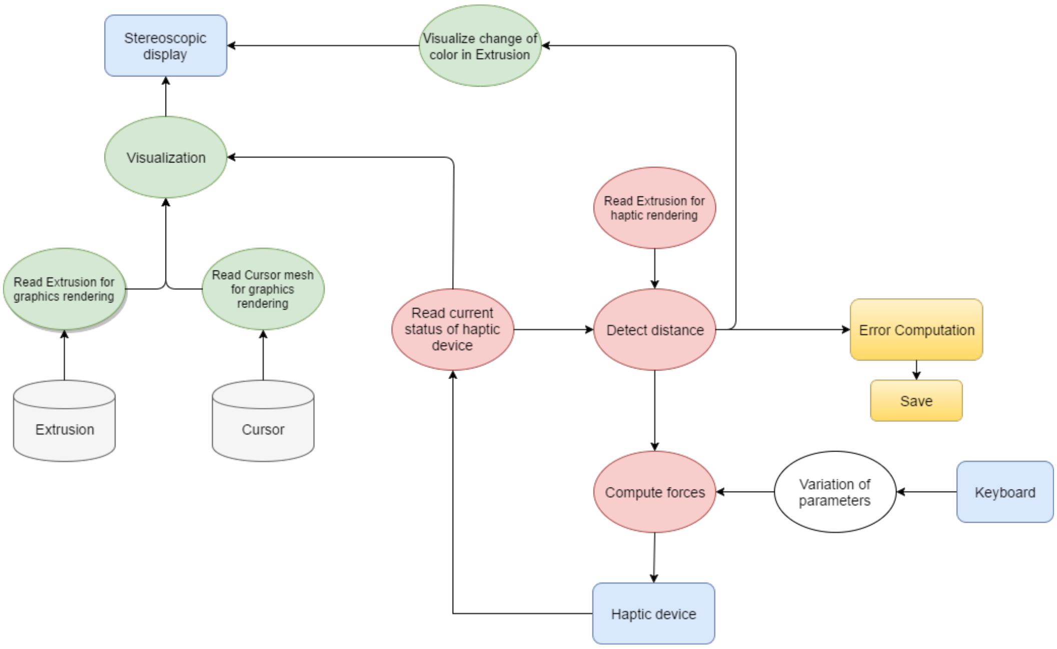

Stereoscopic display
Visualize change of color in Extrusion
Visualization
Read Extrusion for haptic rendering
Read Extrusion for graphics rendering
Read Cursor mesh for graphics rendering
Read current status of haptic device
Detect distance
Error Computation
Save
Extrusion
Cursor
Compute forces
Variation of parameters
Keyboard
Haptic device

*Figure 3.10: Psychomotor Skill scheme*

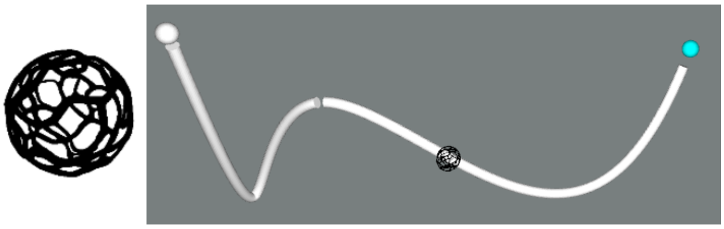


*Figure 3.11: Cursor and Trajectory representation*

### 3.4.1 Force Rendering

In this study, a linear 3-dimensional force field has been applied as function of the error computed. In order to create these forces, it has been necessary to exploit the QH force callback, along with WK library, so that the computation would be faster. Indeed, force callback is executed at 1kHz, thus, a fast and reliable implementation is needed. The algorithm used is represented in Figure 3.13.

When the "start sphere" is touched, the force callback starts. At this point all variables used to evaluate the performance are reset. As soon as the callback begins, the actual cursor position is saved and used by WK

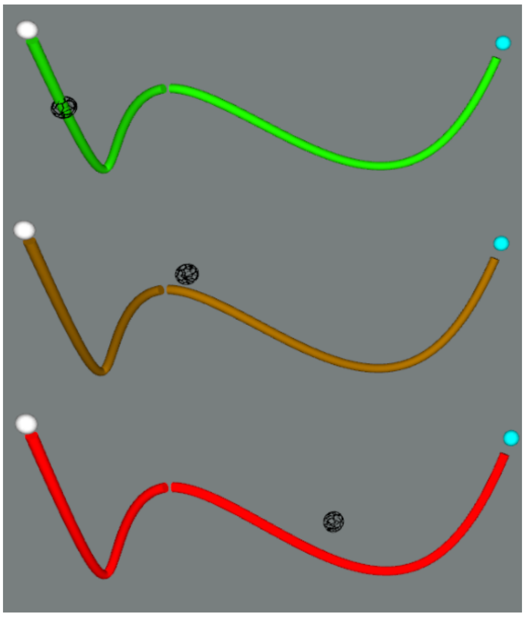


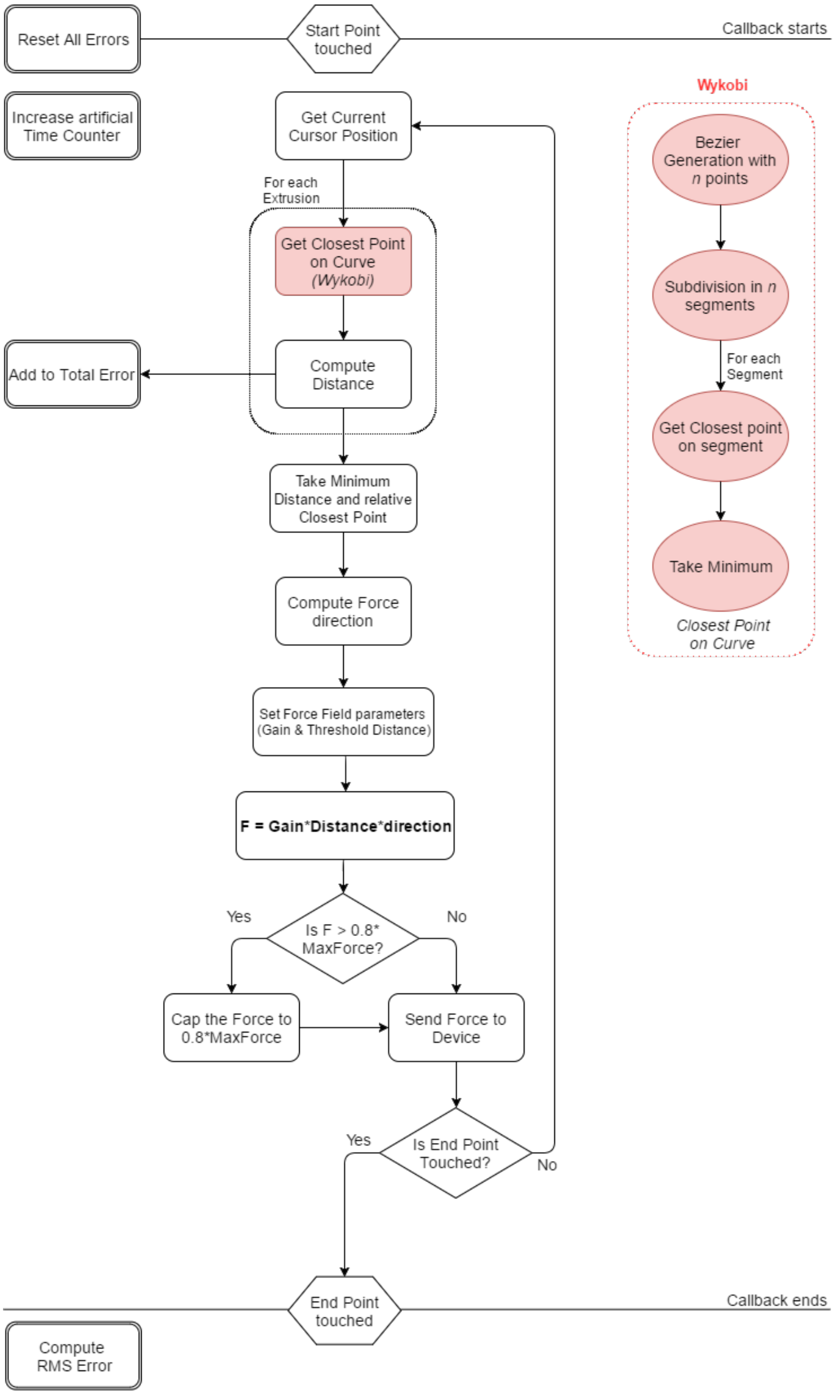
*Figure 3.12: Representation of the trajectory's change of color depending on cursor distance, i.e. the error made by the user. The more the cursor is away from the ideal curve, the more the trajectory will move toward the red color.*

library to compute the closest point on curve from an external point in three dimension.

Since WK algorithm works only with single cubic Bezier, it is necessary to compute this operation for all the extrusions that create the actual path. Once the function is called, WK automatically generates the cubic curve, given the control points. Then it is divided into $n$ segments (1000 if not specified) and for each segment finds the closest point to the external point given. This operation is computed by knowing that a point on a line segment can be parameterized by

$$\mathbf{P}(t) = \mathbf{P}_0 + t(\mathbf{P}_1 - \mathbf{P}_0), \quad t \in [0, 1] \tag{3.13}$$

so that $\mathbf{P}_0$ and $\mathbf{P}_1$ are first and last points belonging to the segment itself (see also 3.14). Thus, $\mathbf{P}_0\vec{\mathbf{P}}(b)$, i.e. the distance from the segment, is the projection of $\mathbf{P}_0\vec{\mathbf{P}}$ onto $\mathbf{P}_0\vec{\mathbf{P}}_1$.



Reset All Errors
Start Point touched
Callback starts
Increase artificial Time Counter
Get Current Cursor Position
For each Extrusion
Get Closest Point on Curve (Wykobi)
Compute Distance
Add to Total Error
Take Minimum Distance and relative Closest Point
Compute Force direction
Set Force Field parameters (Gain & Threshold Distance)
F = Gain*Distance*direction
Yes
Is F > 0.8* MaxForce?
No
Cap the Force to 0.8*MaxForce
Send Force to Device
Yes
Is End Point Touched?
No
End Point touched
Callback ends
Compute RMS Error
Wykobi
Bezier Generation with n points
Subdivision in n segments
For each Segment
Get Closest point on segment
Take Minimum
Closest Point on Curve

*Figure 3.13: Force Algorithm*

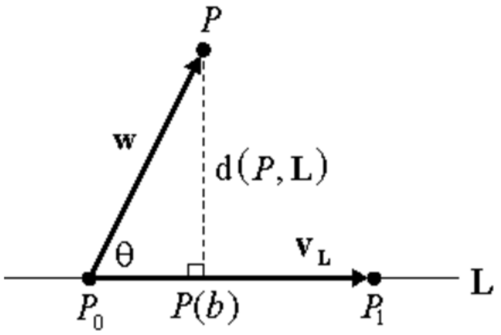


*Figure 3.14: Geometry representation of a distance from a linear segment. Using the scalar product among first and last point of the line segment, and the external point, the distance is computed.*

In this way, it is possible to compute $b$ as

$$b = \frac{d(\mathbf{P}_0, \mathbf{P}(b))}{d(\mathbf{P}_0, \mathbf{P}_1)} = \frac{\vec{\mathbf{w}} \bullet \vec{\mathbf{v}_L}}{\vec{\mathbf{v}_L}} \tag{3.14}$$

where $\vec{\mathbf{w}} = \mathbf{P} - \mathbf{P}(b)$ and $\vec{\mathbf{v}_L} = \mathbf{P}_1 - \mathbf{P}_0$.

Thus, as result,

$$d(\mathbf{P}, \mathbf{P}_1\bar{\mathbf{P}}_0) = \|\mathbf{P} - \mathbf{P}(b)\| = \|\vec{\mathbf{w}} - (\vec{\mathbf{w}} \bullet \vec{\mathbf{v}_L})\| \tag{3.15}$$

Finding out the minimum distance among all the segments, the algorithm also reaches the nearest point to the curve itself.

Once this point has been computed, the distance between it and the cursor position is calculated and the force vector is built. This is characterized by magnitude equal to the distance itself and direction equal to the one connecting the two points.

Both force fields linearly vary according to the distance from cursor position to the closest point on curve:

$$\vec{\mathbf{F}} = G * \Delta d * \vec{\mathbf{v}} \tag{3.16}$$

where $G$ is a gain factor, $\Delta d$ is the difference between the euclidean distance between cursor position and closest point on curve and distance threshold, and $\vec{\mathbf{v}}$ is the unit vector indicating the force direction.

Trends for both the Limit-Push and Limit-Trench condition are presented in Figure 3.15; here the aim is to give only a qualitative representation of the forces trends.

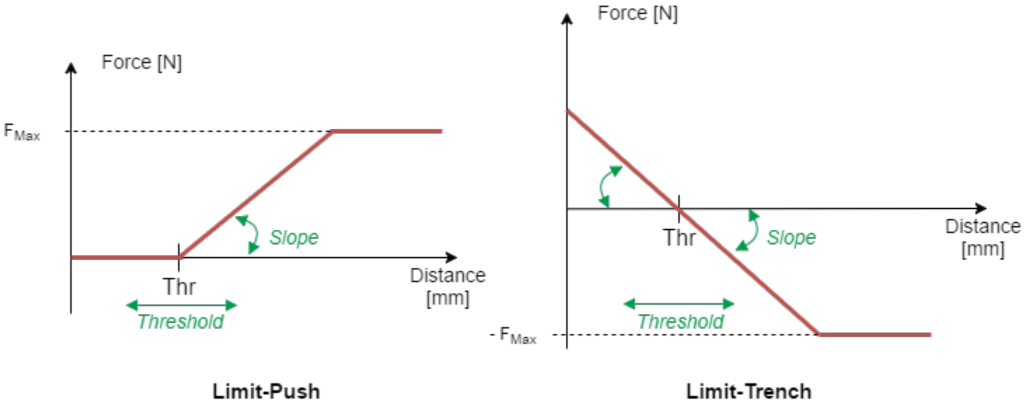


*Figure 3.15: Qualitative representation of forces as linear functions of the distance. Divergent forces are characterized by positive values, while convergent forces by negative ones. Force magnitude is capped at 70% ($2.38N$) of the device Nominal Maximum Force ($3.4N$). In green are also showed the parameters that can vary force field action on the trainee.*

- **Limit-Push**

  is currently used in rehabilitation, as already said in Section 2. In current state of the art, Limit-Push condition is based on the application of a step function. Here, it has been replaced with a ramp function, so that increasing the error, also the force increases, proportionally to a certain slope. However, following the traditional concept, forces are applied only behind a certain distance threshold (Thr in Figure).

- **Limit-Trench**

  introduced in this work, focuses on the *instability* to which trainees are usually exposed while dealing with error-augmentation. A convergent force towards the ideal trajectory is applied when the trainees' error exceeds certain threshold to help them to stay focus on the task. Once the error gets reduced (i.e., the distance between cursor and trajectory is lower than the threshold), a divergent force is generated by the haptic device similar to the one applied in the limit-push method. The idea behind this approach is to both motivate and challenge the trainees during the learning process with the goal of improving their overall performance.

As already said in Chapter 2, because different parameters, i.e. distance threshold and slope of the straight line, can affect the overall experiment, and both of them are relevant in terms of haptic feedback, which is the one we want to study as primary feedback, a preliminary study has been prepared, creating a *Test-bed version*.

This first application has been implemented so as to try different threshold and gain (slope) values. In this way, analyzing these preliminary data, the optimal parameters will be then used in the real experimental phase.

## 3.5 Experimental Setup

The experimental setup is structured in different sections, as displayed in 3.16, and it is enriched with extra-features in order to be as flexible as possible in terms of trajectory creation and slope and distance threshold values settings.

### 3.5.1 Choice of Trajectories

As soon as the application starts, the user can decide to train him/herself on default trajectories or change them creating new ones. For this first implementation, it has been decided to use a set of five different trajectories, allowing the same user to try different kinds of curves in the trial.

Trajectory editing is carried out by directly moving the control points with the haptic device. Since Bezier curves are splines, thus it would be difficult to exactly know its shape until it is displayed, the curve variations are visible run-time. Of course, if multiple Bezier are connected together, only

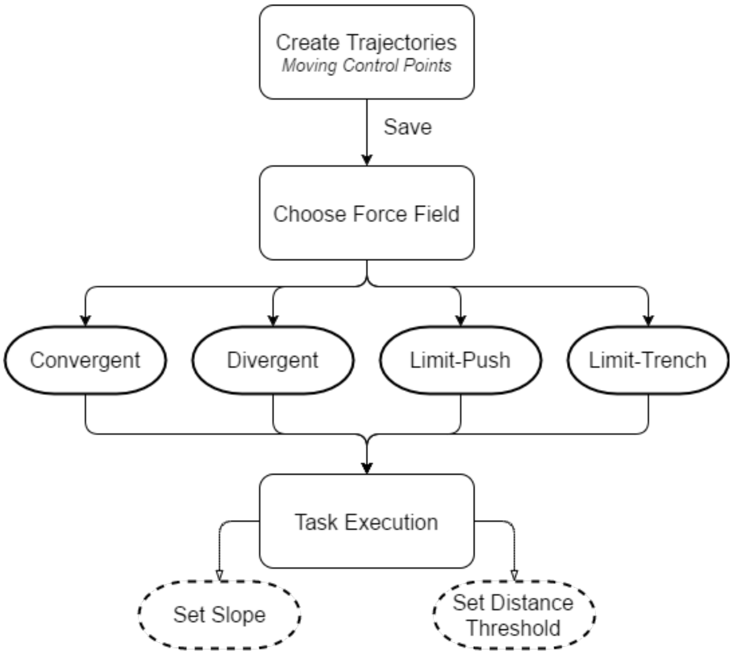


*Figure 3.16: Experimental Setup Workflow. First the user decided the trajectory to use, then force field is chosen and eventually the actual task can be carried out, editing the force function parameters to optimally tune them.*

some control points will be visible and usable, due to second order continuity assessment. This task is automatically provided by the application.

### 3.5.2 Choice of Force Field

Once the trajectories are decided, the user can choose which kind of force fields he/she wants applied. For testing purposes, also simple convergent and divergent force fields can be used. The values of the different parameters are uploaded as default based on heuristic a priori values, but they are modifiable during task execution.

### 3.5.3 Task Execution

Task execution is based on the algorithm shown in 3.13. When the "start sphere" is touched, its rapid change of color indicates the user that forces are on. When the end point is reached, the RMS Error is displayed and the curve will be uploaded to the next available, if it exists. Otherwise the task is completed.

Two different visuo-haptic feedback can be used to check and evaluate force strength, or to just visually inform the user about his/her performance:

- *Change of color in the trajectory*: as explained in Section 3.4, depending on the run time error computed by the user, i.e. the distance from the trajectory, the curve changes color accordingly.
- *Arrow Force visualization*: this feature is not displayed by default, but it is possible to turn it on or off via keyboard interaction. It enables the visualization of a qualitative information of force under the form of an arrow. The magnitude is represented by changes in dimension of the arrow, while direction as direction of the arrow itself (3.17). This tool is also useful to debug the application and verify the correct force computation.

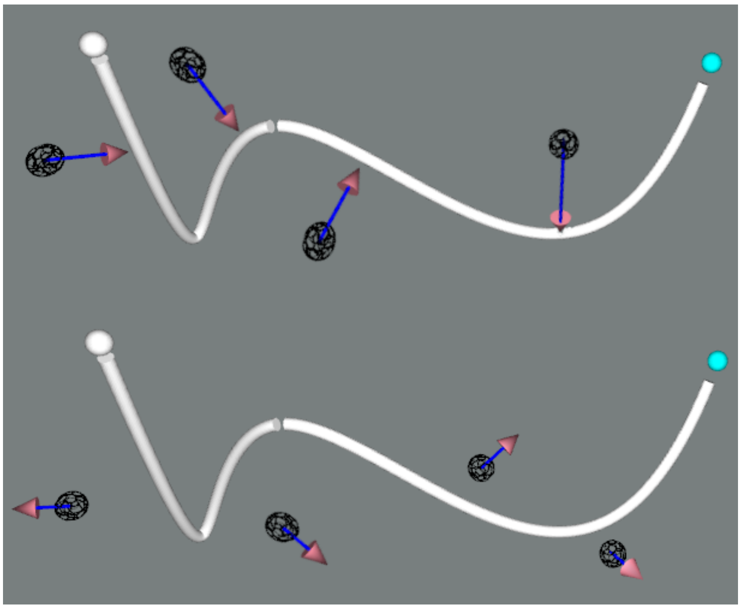


*Figure 3.17: Force arrows visualization examples. In the upper trajectory, it is displayed a convergent force field, while a divergent one is represented in the lower curve.The arrow elongates or shortens proportional to the force magnitude, pointing accordingly to the direction.*

Also force parameters are changed via keyboard interactions:

- Slope setting can be changed by using up and down keyboard input. Editing this parameter, the force could be stronger or weaker, therefore, it is necessary taking into account that also the capping will be performed earlier, i.e. nearer to the ideal path.

- Distance threshold is set by using left and right keyboard input. In this way, both Limit-Push and Limit-Trench conditions can vary in terms of precision and overall ability required during task execution. However, the effects on trainee performances could be different: in a Limit-Push set up, a lower threshold means higher error possibilities, thus the user need to be more careful while performing. On the other hand, a lower threshold in a Limit-Trench set up does not necessarily mean that the user has more error possibilities due to lower divergent force application on the ideal trajectory and due to nearer null force point application, i.e. nearer threshold.

# Chapter 4

# Experimental Evaluation

Training protocol acquisitions have been conducted at the Neuroengineering and medical robotics Laboratory (NearLab) in Politecnico di Milano. We used a high quality display, along with a Geomagic®Touch™ haptic device, by 3D Systems® (see Figure 4.1). Technical specifications are reported in Table 4.1.

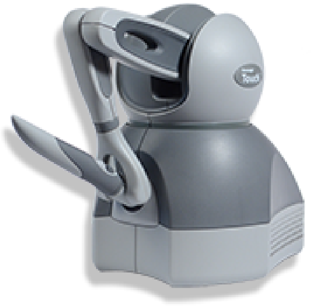


*Figure 4.1: Geomagic Touch.*

## 4.1 Experimental Setup

Using a haptic interface manipulator, trainees had to control a spheric end-effector along multiple trajectories. The experimental setup, shown in Fig.4.2, consists of a high definition display along with a Geomagic®Touch™ device for force feedback. While doing the task, objects can compenetrate, thus

*Table 4.1: TECHNICAL SPECIFICATION GEOMAGIC TOUCH.*

| | |
|---|---:|
| Type | Touch |
| Positional Feedback | 6 (complete pose) |
| Force Feedback DoF | 3 (position only) |
| Force Feedback workspace (WxHxD) | 160 x 120 x 70 mm |
| Maximum Force | 3.3 N |
| Nominal position resolution | 0.055 mm |

leaving all haptic feedback to force fields only. To partially restore the 3D perception, and to give a real-time feedback of the user performance, the trajectory color changes from green to red with respect to the distance between current cursor position and closest point on curve.

The application runs on a Windows platform where the graphic and the haptic threads run concurrently, but at different frequency (60 Hz for graphics and 1 KHz for haptics). The computer in use was provided with an Intel® Core™ i7-6800K CPU at 3.40GHz and a TITAN Xp graphic card by NVIDIA.

## 4.2 Subjects Involved

Eighteen right-handed volunteers (13 males and 5 females; aged 28 ± 7.8) participated in the study. Most of them had a slight experience using the master device, while none of them experienced 3D teleoperation on these tasks before. Any subject without arm or hand problems can participate in the study.

They were randomly divided into three groups of six:

- *Control group* (C): no forces are applied along the entire experiment.
- *Limit-Push group* (LP): forces are applied following Limit-Push condition during training phase.
- *Limit-Trench group* (LT): forces are applied following Limit-Trench condition during training phase.

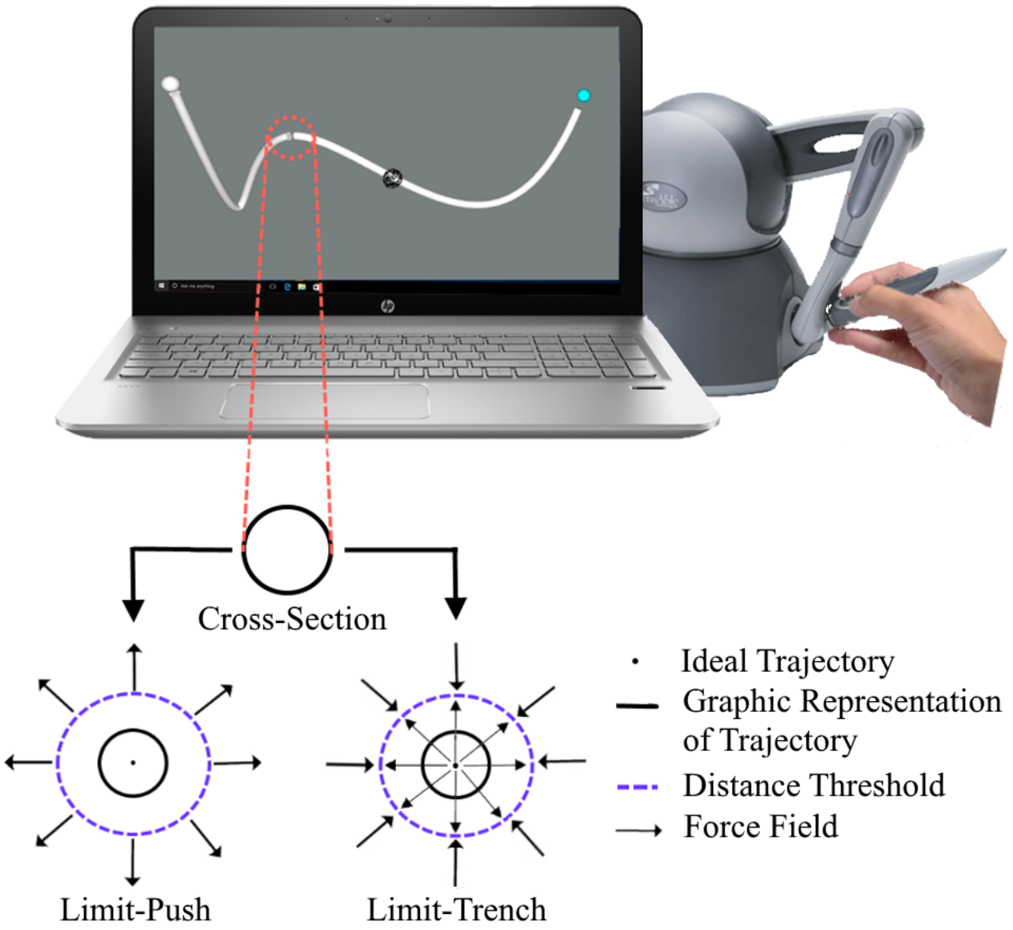


*Figure 4.2: Experimental setup of the virtual reality and haptic test bed for error augmentation. A zoom on the cross section of the trajectory represents limit-push and limit-trench concept application.*

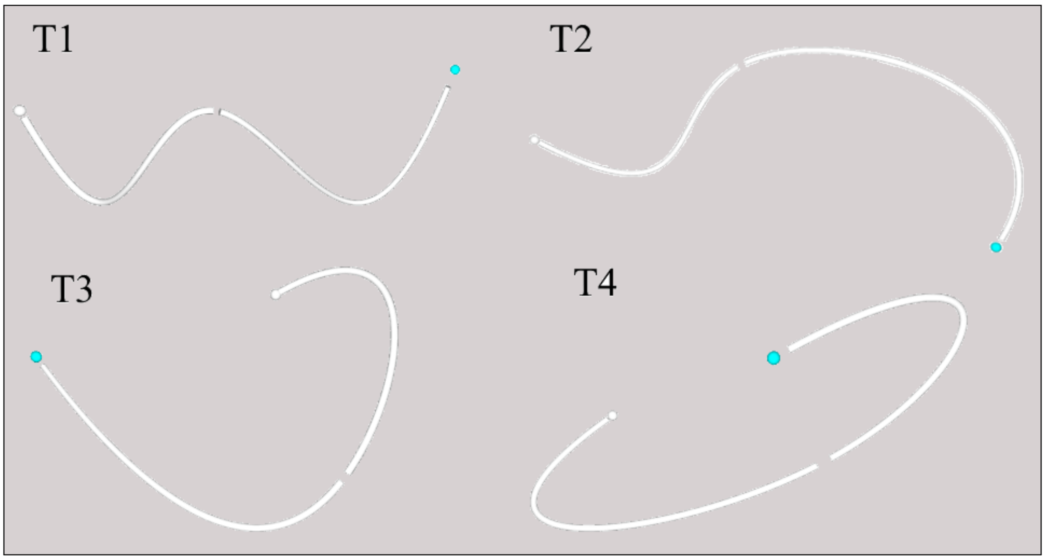


*Figure 4.3: Trajectories used for the training protocol. They are built as two interconnected cubic Bezier curves.*

## 4.3 Experimental design

To avoid the possibility of over fitting the user skills to the same task, we developed a set of different 3D trajectories.

10 trajectories were created, each of them composed by two interconnected cubic Bezier curves. Through some pilot acquisitions, 4 final trajectories were chosen by homogeneity criteria. As result, we were able to identify four trajectories (Fig. 4.3).

The experiment was composed by four phases:

1. *Familiarization*: subjects freely interacted and played in the virtual environment to familiarize with the device. This phase lasted about five minutes.

2. *Baseline* (B): subjects executed the sequence of trajectories once.

3. *Training* (T): subjects executed the sequence of trajectories 10 times with or without force application, depending on the group they belonged to.

4. *Evaluation* (E): subjects executed the sequence of trajectories once without any force applied.

## 4.4 Force Fields Parameters

Haptic feedback is computed as a force field that linearly varies according to the distance from cursor position to the closest point on the curve (see Chapter 3).

Referring to Figure 3.15 and Equation 3.16, distance thresholds (Thr) are $2.2mm$ and $3.5mm$ for LP and LT respectively. Also $G$ parameter is different between the two approaches: $GLP$ is equalt to $0.095N/mm$, while $G_{LT}$ is $-0.19N/mm$.

These choices have been conducted via empirical trials, evaluating the best trade-off between force feedback strength and error inducted by the force itself.

## 4.5 Performance Metrics

During task execution, trial time, cursor position, and closest point on curve have been acquired to analyze trainee performance improvement.

Trial time ($t_{Trial}$) has been defined as the time from touching the start and end points (distance from point $< 0.01mm$).

From these raw data, we computed the distance between cursor position and closest point on curve, root mean squared error, translational path error, speed accuracy index and interquartile range of distances.

- *Root mean squared error* (RMSE) is used as overall index of user performance. It is computed as:

$$RMSE = \frac{\sum_{i=1}^{N} d_{i,Closest}}{N} \tag{4.1}$$

  where $d_{Closest}$ is the closest distance from curve calculated for each sample acquired, and $N$ is the total number of samples for each trial.

- *Translational path error* (TPE) quantifies accuracy as the sum of the punctual surface path errors (Fig. 4.4). It is computed as multiplication of the distance between current closest point on curve and previous one, and distance between current cursor position and closest point on curve [15], representing the surface difference between the end effector path and target path. The lower the TPE is, the higher is the accuracy of the trial executed from the trainee. Analytically, it is calculated as in Equation 4.2:

  $$TPE = \sum_{n=1}^{N} \|\mathbf{X}_n - \mathbf{C}_n\| * \|\mathbf{C}_n - \mathbf{C}_{n-1}\| \tag{4.2}$$

  Unlike RMSE, no averaging on the total number of samples ($N$) is performed, thus TPE is used as accuracy metric, quantifying the total amount of error executed by trainees along a trial.

- *Speed-accuracy index* (SA) is computed as combination of TPE and trial time. It is inferred from Fitts's law as a measure of the trade off between accuracy and time spent to execute a trial [60]. Since a good performance is connected to high accuracy (i.e., low TPE) and high speed (i.e., low trial time), a low SA index, compared to a higher one, indicates a better performance.

  $$SA = TPE * t_{Trial} \tag{4.3}$$

- *Interquartile Range* of closest distances (IQR) is used as variance metric. By definition, it is the range of values between third and first quantile:

  $$IQR = Q_3 - Q_1 \tag{4.4}$$

  where $d_{Q_3}$ and $d_{Q_1}$ are $75^{th}$ and $25^{th}$ percentiles of the euclidean distances between current cursor position and closest point on curve of each trial.

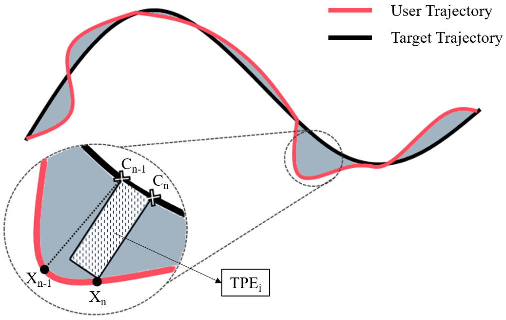


*Figure 4.4: Translational Path Error. Red line represents the end effector movement, while the black is the target path. Considering the actual cursor position and closest point on curve ($X_n$, $C_n$ respectively), and previous cursor position and closest point on curve ($X_{n-1}$, $C_{n-1}$ respectively), $TPE_i$ is the rectangular area pictured by current closest distance from curve, and distance between current and previous closest point on curve. The overall TPE is the sum of every $TPE_i$ computed, thus approximating the surface difference between the two paths.*

## 4.6 Statistical Analysis

### 4.6.1 Baseline

To verify the absence of learning effects during the execution of the four trajectory in the Baseline recording, the measured indexes (RMSE, TPE, $t_{trial}$, SA, IQR) have been grouped between subjects and corresponding trials. Since for some of the indexes these data distributions were not normal (Lilliefort $\alpha = 0.05$) a Kruskal Wallis test ($\alpha = \mathit{0.05}$) was used to evaluate the statistical differences between the consecutive baseline trials.

### 4.6.2 Training effects

To evaluate the different training modalities, we computed the variation between baseline and evaluation for each metric:

$$\Delta_{BE} = I_{i,Baseline} - I_{i,Evaluation} \tag{4.5}$$

with $I = \{\text{RMSE, TPE, Duration, SA, IQR}\}$. Positive training effects are represented by higher $\Delta_{BE}$, while detrimental effects are negative. One $\Delta_{BE}$

sample for each trajectory of each user was computed, yielding to 4-by-6 values (four trajectories, six subjects) for each group. We tested if data were normally distributed for each subject with Lilliefors test ($\alpha = 0.01$). Since no significant difference was found for any performance metric, we considered each trajectory execution as a repeated measure, and we extracted the median value among the trajectories for each subject.

Even if all data followed a normal distribution, since our sample size was small, we decided to use non-parametric statistic for repeated measures; to conduct our statistical analysis, we used the Nonparametric Analysis of Longitudinal Data in Factorial Experiments package in R.

# Chapter 5

# Results

### 5.0.1 Baseline Analysis

Baseline was analyzed to verify that no learning effect would appear within the first execution of the four trajectories. Kruskal Wallis tests with significance level ($\alpha$) of 0.05 have been conducted grouping all subjects of all groups, and considering them across the different trajectories. No significant differences were found for any metrics (p-values ranging from 0.0673 and 0.99).

### 5.0.2 Performance Analysis

Table 5.1 shows median and interquartile ranges for each index for all the subjects in the three force groups. The results of the non parametric ANOVA analysis for repeated measures is presented in Table 5.2, where significant statistical differences (in bold) are highlighted with stars. In Fig. 5.1 the index distributions and statistical differences are depicted.

Apart from the trial duration time ($t_{trial}$), in which no significant difference was found, the $\Delta_{BE}$ variations in all the measured metrics show differences between the three force groups. For RMSE, TPE and IQR a statistical difference subsist between each group, while for the SA index, no difference was found between LP and LT. Throughout the 4 indexes presented in Fig. 5.1 the same trend was identified: the subjects that took part in the Con-

*Table 5.1: MEDIAN AND INTERQUARTILE RANGE OF $\Delta_{BE}$ FOR EACH GROUP.*

| | LP | | LT | | C | |
|---|---|---|---|---|---|---|
| **RMSE** $[cm]$ | 3.02 | 51.7 | -0.41 | 30.3 | 14.3 | 181.3 |
| **TPE** $[cm^2]$ | 0.01 | 1.5 | 0.005 | 2.22 | 1.02 | 8.9 |
| **Duration** $[s]$ | 8.16 | 6.3 | 4.45 | 12.5 | 5.41 | 8.2 |
| **SA index**$[cm^2 * s]$ | 1.10 | 0.9 | 0.31 | 1.1 | 3.28 | 4.0 |
| **IQR**$[cm]$ | 0.002 | 0.02 | -0.009 | 0.01 | 0.03 | 0.09 |

*Table 5.2: P-VALUES FROM THE NON-PARAMETRIC ANALYSIS IN R; ASTERISKS INDICATE STATISTICAL DIFFERENCES BETWEEN GROUPS.*

| | LP:LT | LP:C | LT:C |
|---|---|---|---|
| **RMSE** | $0.003^{**}$ | $0.03^{*}$ | $5 \cdot 10^{-5***}$ |
| **TPE** | $0.004^{**}$ | $0.005^{**}$ | $8 \cdot 10^{-9***}$ |
| **Duration** | 0.5 | 0.4 | 0.9 |
| **SA index** | 0.08 | $0.04^{*}$ | $0.003^{**}$ |
| **IQR** | $0.01^{*}$ | $0.003^{**}$ | $10^{-7}$*** |

trol group showed the highest incremental performances, followed by the LP group which is characterized by less positive incremental effects. Users that were teleoperating under the influence of LT force field showed substantial negative $\Delta_{BE}$ values for RMSE, TPE and IQR denoting a negative effect of the training.

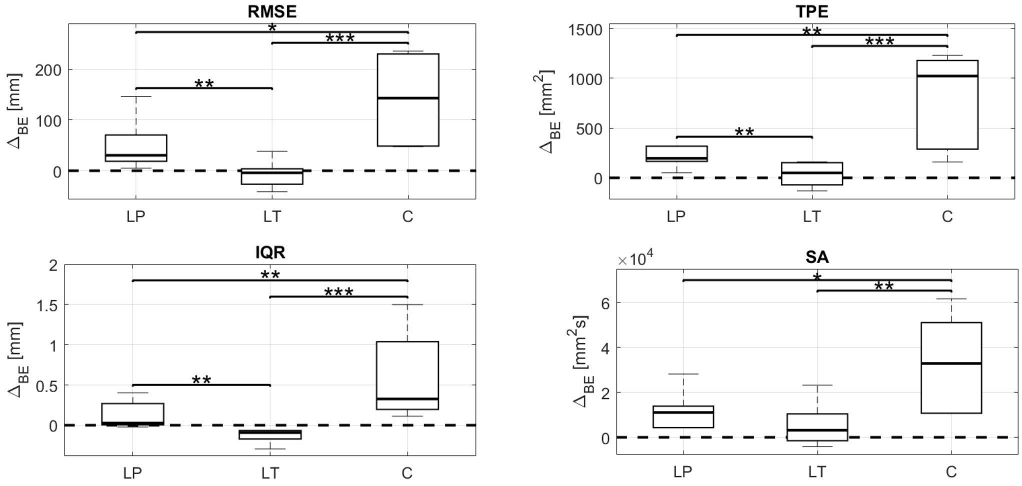


*Figure 5.1: Median, first and third interquartile distance as well as minimal and maximal values are presented as box-plots of the four performance metrics that showed statistical differences. Statistical difference is quantitatively represented with asterisks (* $p \leq 0.05$, ** $p \leq 0.01$, *** $p \leq 0.001$).*

# Chapter 6

# Discussion

In this work we implemented a virtual teleoperation training protocol to evaluate the effects of different force fields over psychomotor skills learning. A thresholded linearly increasing divergent force field (LP) and a hybrid convergent/divergent force field (LT) were developed and compared with a control group.

The effects of the three training protocols have been quantified by evaluating the variation in multiple performance metrics between pre-training baseline and post-training evaluation phase ($\Delta_{BE}$). Non-parametric one-way analysis has been conducted on $\Delta_{BE}$ among the force groups.

A consistent trend has been found for all the analyzed metrics, with the exception of trail time ($t_{trial}$): C group improvements overcome both LP and LT. Regarding the lack of difference in $t_{trial}$, it seems that, independently from the group, all the subjects equally decreased the trial time from Baseline and Evaluation; this evidence seems to suggest that regardless of force fields, users felt the same level of acquired acquaintance .

This result seems to suggest that the force field that have been implemented reduce the subject natural capabilities in learning the tasks by pure undisrupted repetitions.

Moreover, the LT group, whose performance are the worst among the three groups, showed detrimental results, often worsening the subjects capabilities in performing the tasks, especially in terms of RMSE and IQR.

Regarding the SA index only, no significant difference was found between LP and LT, but C is still significantly different.

This dissimilarity could result from the higher interquartile range of $t_{Trial}$ for LT with respect to LP group(standard deviation of 9.93 vs. 6.57 of LP).

The classical error augmentation approach seems to be less effective in teleoperation compared to rehabilitation: the differences between the fine manipulations adopted in teleoperation and the less precise, often ballistic movements characterizing rehabilitation may require additional studies. Analyzing our results along with other studies researching on divergent force fields implementations for motor learning, leads to controversial conclusions. In the study of Lee et al. [40], noise-like random disturbance force fields showed better performance than control group and convergent force field group. On the other hand, a recent work testing convergent and divergent force fields for teleoperation did not find any difference between training with and without force application [15]. In particular, it is interesting to notice how LP and simple divergent forces could lead to different learning effects: it seems that giving to user the possibility to carry out the task within a safe region, where no forces are applied, restricts training effects, thus highlighting possible advantages of simple divergence approach as error augmentation technique for training.

Regarding the novel approach here proposed, it is interesting to notice that a degradation in performance can be seen just for the LT group (negative $\Delta_{BE}$ values in Fig. 5.1), especially for RMSE and IQR. These results could suggest a negative impact of training in an unstable haptic environment: while training with LT approach, trainees struggle to carry out good performances, since forces always reject them from the target path. This condition leads to continuous error, and users experienced faster muscular fatigue with respect to the other groups, causing them to require some minutes of relaxation between consecutive trials. Indeed, users have to generate higher forces to stay as near as possible to the target path, possibly leading to very stiff arm configurations. It seems that constraining the trainee to an unstable environment, without giving him/her the possibility to actually reach a good performance is even detrimental for training.

In the end, we can observe that C group shows the highest variances, indicating a large fluctuation of subjects' learning effect within this group. LP and LT groups show less overall improvement, but the changes induced are more similar between different subjects: it seems that the effect of the single user ability to learn is reduced. These trends could indicate a different way to perceive and metabolize training with or without force feedback. In fact, it seems that, after receiving a haptically-enhanced training phase, users performances are narrowed around limited intervals, thus leading to more repeatable results, and showing that force fields have a consistent effect over subjects. The application of divergent and mixed force fields could therefore carry a high potential in increasing teleoperators' performance repeatability.

# Chapter 7

# Conclusion

From these results and considerations, we could conclude that force fields application during training on a series of different path would not improve performances more than training without force feedback. On the other hand, we were able to create conditions in which force feedback could be even detrimental.

These factors should be highlighted and should keep researchers interested in the error augmentation field: no standard training protocols have been implemented yet, thus different methodologies could be applied to same training approaches, providing different results, and looking forward the optimal training methodologies.

A possible critical aspect and limitation of this work could reside in the overall experiment duration: about 40-50 minutes were necessary to execute a total number of 48 trials. This amount of training in just one session brought to fatigue of the trainees, thus suggesting possible further implementation in multi session training protocols.

In addition, since LP approach is still a novel technique in the motor control field, future works would focus on a deeper analysis of its application in psychomotor skill training: more data would be acquired, recruiting more subjects for training. Furthermore, a multi-session experimental protocol would be proposed to better simulate actual learning of complex skills, which are not possible to acquire in just one session. As consequence, also task difficulty will be reassessed.

Focusing on the novel error-augmentation approach, i.e., LT, it could be interesting to evaluate a non-linear force algorithm in order to maintain the challenging aspects that characterize this implementation, without preventing learning. A critical aspect of this technique is indeed the inability of the trainees to be able to exaclty follow the target path; smoothing divergent forces, the instability would be still applied, but with lower strength.

Furthermore, it could be interesting to understand arm kinematics behind the stiffness condition related to instability, and quantitatively understand its contribution to the performance indexes.

# Bibliography

[1] Ascension Technology Corporation©. `https://www.ascension-tech.com/products/`. [Online; accessed 14-April-2017].

[2] Mimic Simulation. `http://www.mimicsimulation.com/products/dv-trainer/`. [Online; accessed 14-April-2017].

[3] QuickHaptics™ MicroAPI. `http://www.geomagic.com/it/products/open-haptics/overview`. [Online; accessed 14-April-2017].

[4] In American Heritage Publishing Company, editor, *American Heritage Dictionary of the English Language*. Fifth edition, 2011.

[5] Hamid Abboudi, Mohammed S Khan, Omar Aboumarzouk, Khurshid A Guru, Ben Challacombe, Prokar Dasgupta, and Kamran Ahmed. Current status of validation for robotic surgery simulators–a systematic review. *BJU international*, 111(2):194–205, 2013.

[6] Emad T Aboud, Ali F Krisht, Terence O'keeffe, Remi Nader, Moustafa Hassan, C Melinda Stevens, Fahd Ali, and Fred A Luchette. Novel simulation for training trauma surgeons. *Journal of Trauma and Acute Care Surgery*, 71(6):1484–1490, 2011.

[7] Jamie E Anderson, David C Chang, J Kellogg Parsons, and Mark A Talamini. The first national examination of outcomes and trends in robotic surgery in the united states. *Journal of the American College of Surgeons*, 215(1):107–114, 2012.

[8] Greg Anson, Digby Elliott, and Keith Davids. Information processing and constraints-based views of skill acquisition: divergent or complementary? *Motor control*, 9(3):217–241, 2005.

[9] Richard H Bartels, John C Beatty, and Brian A Barsky. *An introduction to splines for use in computer graphics and geometric modeling*. Morgan Kaufmann, 1995.

[10] Brian T Bethea, Allison M Okamura, Masaya Kitagawa, Torin P Fitton, Stephen M Cattaneo, Vincent L Gott, William A Baumgartner, and David D Yuh. Application of haptic feedback to robotic surgery. *Journal of Laparoendoscopic & Advanced Surgical Techniques*, 14(3):191–195, 2004.

[11] Jérémy Bluteau, Sabine Coquillart, Yohan Payan, and Edouard Gentaz. Haptic guidance improves the visuo-manual tracking of trajectories. *PLoS One*, 3(3):e1775, 2008.

[12] M Bosi. Visualization Library. `http://www.visualizationlibrary.org/`, 2011. [Online; accessed 14-April-2017].

[13] Stuart A Bowyer, Brian L Davies, and Ferdinando Rodriguez y Baena. Active constraints/virtual fixtures: A survey. *IEEE Transactions on Robotics*, 30(1):138–157, 2014.

[14] B. Noland Carter. The fruition of halsted's concept of surgical training. *Surgery*, 32(3):518–527, 1952.

[15] Margaret M Coad, Allison M Okamura, Sherry Wren, Yoav Mintz, Thomas S Lendvay, Anthony M Jarc, and Ilana Nisky. Training in divergent and convergent force fields during 6-dof teleoperation with a robot-assisted surgical system. In *World Haptics Conference (WHC), 2017 IEEE*, pages 195–200. IEEE, 2017.

[16] Harrison Philip Crowell and Irene S Davis. Gait retraining to reduce lower extremity loading in runners. *Clinical biomechanics*, 26(1):78–83, 2011.

[17] Jennifer Davies, Manaf Khatib, and Fernando Bello. Open surgical simulationa review. *Journal of surgical education*, 70(5):618–627, 2013.

[18] Neil A. Dodgson. Some mathematical elements of graphics. 2000.

[19] Alexander Duschau-Wicke, Joachim von Zitzewitz, Andrea Caprez, Lars Lunenburger, and Robert Riener. Path control: a method for patient-cooperative robot-aided gait rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(1):38–48, 2010.

[20] A. Faso. Haptic and virtual reality surgical simulator for training in percutaneous renal access. Master's thesis, University of Illinois at Chicago, 2017.

[21] Paul Morris Fitts and Michael I Posner. Human performance. 1967.

[22] C. Gatti. Application of haptic virtual fixtures in psychomotor skill development for robotic surgical training. Master's thesis, University of Illinois at Chicago, 2017.

[23] Cecilia Gatti, Cristian Luciano, James L Patton, and Elena De Momi. Virtual reality test bed for haptic error augmentation in psychomotor skill development. In *In Workshop "Haptics for Accessibility' on World Haptics Conference*, Munich, Germany, June 2017.

[24] Valeria Gazzola, Giacomo Rizzolatti, Bruno Wicker, and Christian Keysers. The anthropomorphic brain: the mirror neuron system responds to human and robotic actions. *Neuroimage*, 35(4):1674–1684, 2007.

[25] Ann M Gentile. A working model of skill acquisition with application to teaching. *Quest*, 17(1):3–23, 1972.

[26] Apostolos P Georgopoulos. Cognitive motor control: spatial and temporal aspects. *Current opinion in neurobiology*, 12(6):678–683, 2002.

[27] Paul J Gorman, Andreas H Meier, and Thomas M Krummel. Simulation and virtual reality in surgical education: real or unreal? *Archives of Surgery*, 134(11):1203–1208, 1999.

[28] ME Hagen, JJ Meehan, Ihsan Inan, and Philippe Morel. Visual clues act as a substitute for haptic feedback in robotic surgery. *Surgical endoscopy*, 22(6):1505–1508, 2008.

[29] Ulrike Halsband and Regine K Lange. Motor learning in man: a review of functional and clinical studies. *Journal of Physiology-Paris*, 99(4):414–424, 2006.

[30] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory.* Psychology Press, 2005.

[31] JD Hernandez, SD Bann, Y Munz, K Moorthy, V Datta, S Martin, A Dosis, F Bello, A Darzi, and T Rockall. Qualitative and quantitative analysis of the learning curve of a simulated surgical task on the da vinci system. *Surgical Endoscopy And Other Interventional Techniques*, 18(3):372–378, 2004.

[32] Herbert Heuer and Jenna Lüttgen. Robot assistance of motor learning: A neuro-cognitive perspective. *Neuroscience & Biobehavioral Reviews*, 56:222–240, 2015.

[33] Maureen K Holden. Virtual environments for motor rehabilitation: review. *Cyberpsychology & behavior*, 8(3):187–211, 2005.

[34] Andrew J Hung, Mukul B Patil, Pascal Zehnder, Jie Cai, Casey K Ng, Monish Aron, Inderbir S Gill, and Mihir M Desai. Concurrent and predictive validation of a novel robotic surgery simulator: a prospective, randomized study. *The Journal of urology*, 187(2):630–637, 2012.

[35] Timothy N Judkins, Dmitry Oleynikov, and Nick Stergiou. Enhanced robotic surgical training using augmented visual feedback. *Surgical innovation*, 15(1):59–68, 2008.

[36] Arie Kaufman. *Rendering, visualization and rasterization hardware.* Springer Science & Business Media, 1993.

[37] Douglas C Kelly, Andrew C Margules, Chandan R Kundavaram, Hadley Narins, Leonard G Gomella, Edouard J Trabulsi, and Costas D Lallas. Face, content, and construct validation of the da vinci skills simulator. *Urology*, 79(5):1068–1072, 2012.

[38] Roger Kneebone. Simulation in surgical training: educational issues and practical implications. *Medical education*, 37(3):267–277, 2003.

[39] Hojin Lee and Seungmoon Choi. Combining haptic guidance and haptic disturbance: an initial study of hybrid haptic assistance for virtual steering task. In *Haptics Symposium (HAPTICS), 2014 IEEE*, pages 159–165. IEEE, 2014.

[40] Jaebong Lee and Seungmoon Choi. Effects of haptic guidance and disturbance on motor learning: Potential advantage of haptic disturbance. In *Haptics Symposium, 2010 IEEE*, pages 335–342. IEEE, 2010.

[41] Michael A Liss and Elspeth M McDougall. Robotic surgical simulation. *The Cancer Journal*, 19(2):124–129, 2013.

[42] Michael A Liss and Elspeth M McDougall. Robotic surgical simulation. *The Cancer Journal*, 19(2):124–129, 2013.

[43] Sergio Maeso, Mercedes Reza, Julio A Mayol, Juan A Blasco, Mercedes Guerra, Elena Andradas, and María N Plana. Efficacy of the da vinci surgical system in abdominal surgery compared with that of laparoscopy: a systematic review and meta-analysis, 2010.

[44] Elspeth M McDougall. Validation of surgical simulators. *Journal of endourology*, 21(3):244–247, 2007.

[45] James Minogue and M Gail Jones. Haptics in education: Exploring an untapped sensory modality. *Review of Educational Research*, 76(3):317–348, 2006.

[46] Dan Morris, Hong Tan, Federico Barbagli, Timothy Chang, and Kenneth Salisbury. Haptic feedback enhances force skill learning. In *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for*

*Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*, pages 21–26. IEEE, 2007.

[47] Erik Valdemar Oberg, Franklin Day Jones, and Holbrook Lynedon Horton. *Machinery's handbook: a reference book for the mechanical engineer, designer, manufacturing engineer, draftsman, toolmaker and machinist*. Industrial Press, 1984.

[48] Allison M Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 31(6):499–508, 2004.

[49] Allison M Okamura, Niels Smaby, and Mark R Cutkosky. An overview of dexterous manipulation. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 255–262. IEEE, 2000.

[50] Marcia K Omalley, Ozkan Celik, Joel C Huegel, Michael D Byrne, Jean Bismuth, Brian J Dunkin, Alvin C Goh, and Brian J Miles. Robotics as a tool for training and assessment of surgical skill. In *Computational Surgery and Dual Training*, pages 365–375. Springer, 2014.

[51] A Partow. Wykoby Computational Geometry Library. `http://www.wykobi.com/index.html`, 2005. [©2005-2017 Arash Partow. Online; accessed 14-April-2017].

[52] James L Patton, Mary Ellen Stoykov, Mark Kovic, and Ferdinando A Mussa-Ivaldi. Evaluation of robotic training forces that either enhance or reduce error in chronic hemiparetic stroke survivors. *Experimental brain research*, 168(3):368–383, 2006.

[53] Cyril Perrenot, Manuela Perez, Nguyen Tran, Jean-Philippe Jehl, Jacques Felblinger, Laurent Bresler, and Jacques Hubert. The virtual reality simulator dv-trainer® is a valid assessment tool for robotic surgical skills. *Surgical endoscopy*, 26(9):2587–2593, 2012.

[54] L. Rapetti. Virtual reality navigation system for prostate biopsy. Master's thesis, University of Illinois at Chicago, 2017.

[55] Richard K Reznick and Helen MacRae. Teaching surgical skillschanges in the wind. *New England Journal of Medicine*, 355(25):2664–2669, 2006.

[56] Gabriel Robles-De-La-Torre. The importance of the sense of touch in virtual and real environments. *Ieee Multimedia*, 13(3):24–30, 2006.

[57] Alan W Salmoni, Richard A Schmidt, and Charles B Walter. Knowledge of results and motor learning: a review and critical reappraisal. *Psychological bulletin*, 95(3):355, 1984.

[58] David Salomon. *Curves and surfaces for computer graphics.* Springer Science & Business Media, 2007.

[59] Coralie Sann and Arlette Streri. Perception of object shape and texture in human newborns: evidence from cross-modal transfer tasks. *Developmental Science*, 10(3):399–410, 2007.

[60] Richard A Schmidt and Tim Lee. *Motor control and learning.* Human kinetics, 1988.

[61] Stéfanie A Seixas-Mikelus, Thenkurussi Kesavadas, Govindarajan Srimathveeravalli, Rameela Chandrasekhar, Gregory E Wilding, and Khurshid A Guru. Face validation of a novel robotic surgical simulator. *Urology*, 76(2):357–360, 2010.

[62] Stéfanie A Seixas-Mikelus, Andrew P Stegemann, Thenkurussi Kesavadas, Govindarajan Srimathveeravalli, Gughan Sathyaseelan, Rameela Chandrasekhar, Gregory E Wilding, James O Peabody, and Khurshid A Guru. Content validation of a novel robotic surgical simulator. *BJU international*, 107(7):1130–1135, 2011.

[63] Neal E Seymour. Vr to or: a review of the evidence that virtual reality simulation improves operating room performance. *World journal of surgery*, 32(2):182–188, 2008.

[64] Ian Sharp and James L Patton. Limit-push training reduces motor variability. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

[65] Roland Sigrist, Georg Rauter, Robert Riener, and Peter Wolf. Augmented visual, auditory, haptic, and multimodal feedback in motor learning: a review. *Psychonomic bulletin & review*, 20(1):21–53, 2013.

[66] Roger Smith, Mireille Truong, and Manuela Perez. Comparative analysis of the functionality of simulators of the da vinci surgical robot. *Surgical endoscopy*, 29(4):972–983, 2015.

[67] George S Snoddy. Learning and stability: a psychophysiological analysis of a case of motor learning with clinical applications. *Journal of Applied Psychology*, 10(1):1, 1926.

[68] Andrew P Stegemann, Kamran Ahmed, Johar R Syed, Shabnam Rehman, Khurshid Ghani, Ricardo Autorino, Mohamed Sharif, Amrith Rao, Yi Shi, Gregory E Wilding, et al. Fundamental skills of robotic surgery: a multi-institutional randomized controlled trial for validation of a simulation-based curriculum. *Urology*, 81(4):767–774, 2013.

[69] E. Tagliabue. Visuo-haptic model of prostate cancer based on magnetic resonance elastography. Master's thesis, University of Illinois at Chicago, 2017.

[70] OAJ Van der Meijden and MP Schijven. The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review. *Surgical endoscopy*, 23(6):1180–1190, 2009.

[71] C Wagner, N Stylopoulos, and R Howe. Force feedback in surgery: Analysis of blunt dissection. In *Proceedings of the 10th symposium on haptic interfaces for virtual environment and teleoperator systems*, 2002.

[72] Yejun Wei, Preeti Bajaj, Robert Scheidt, and James Patton. Visual error augmentation for enhancing motor learning and rehabilitative relearn-

ing. In *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pages 505–510. IEEE, 2005.

[73] George Whittaker, Abdullatif Aydin, Nicholas Raison, Francesca Kum, Ben Challacombe, Muhammed Shamim Khan, Prokar Dasgupta, and Kamran Ahmed. Validation of the robotix mentor robotic surgery simulator. *Journal of Endourology*, 30(3):338–346, 2016.

[74] Camille K Williams and Heather Carnahan. Motor learning perspectives on haptictraining for the upper extremities. *IEEE transactions on haptics*, 7(2):240–250, 2014.

[75] Gabriele Wulf, Charles H Shea, and Sabine Matschiner. Frequent feedback enhances complex motor skill learning. *Journal of motor behavior*, 30(2):180–192, 1998.