Università degli Studi di Udine

Dipartimento di Scienze Matematiche,
Informatiche e Fisiche

Dottorato di Ricerca in Informatica e Scienze
Matematiche e Fisiche

Ph.D. Thesis

Multi-sensory design, physical computing and experimental validation of virtual buttons and knobs for professional appliances

Candidate:
Yuri De Pra

Supervisor:
Prof. Federico Fontana

Co-Supervisor:
Dr. Mauro Sonego

Cycle XXXIII – Year 2021

Author's e-mail:
yuri.depra@uniud.it

Institute Contacts:
Dipartimento di Scienze Matematiche, Informatiche e Fisiche
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine – Italia
`https://www.dmif.uniud.it`

Abstract

Touch screens have become the most popular computer interface in our daily life. Although they respond to finger contact, most of them are able to provide only a visual representation of standard physical controls such as buttons, sliders, and knobs. Indeed, tactile and auditory feedback is neutralized by a touch screen into an interaction with a flat surface. Among other issues that occur during this interaction, the lack of multimodal sensory cues inevitably unbalances the cognitive load toward vision.

This thesis targets the design, realization and validation of user interfaces for the professional appliances domain. In such a specific working environment, populated by multiple and cooperative human activities that frequently and sometimes unexpectedly trigger attendance by the operator, the use of touch screens is generally problematic because often the visual attention must be switched from a machine interface to an emerging task, without suspending any activity in between. In this context, new designs and implementations providing rich auditory and tactile information can contribute to more reliable and robust interactions capable of restoring, partially or absolutely, the multisensory essence of physical controllers hence improving effectiveness and safety of the working environment.

To such an end, this thesis discusses three interface prototypes reproducing the multimodal feedback of two fundamental physical controllers: buttons and knobs. The proposed research starts from the peculiar interaction with professional appliances, and aims at 1) designing and 2) prototyping audio-haptic interfaces that combine innovative haptic and force actuators, as well as 3) evaluating such interfaces through rigorous experiments involving human participants.

The first part of the thesis deals with surface haptics — a research field studying the reproduction of tactile effects on touch surfaces. This part provides a series of experiments on auditory and tactile surface perception, together with the development of a mock up aimed at testing and improving the reliability and robustness of interaction with professional appliances controlled by touch screen interfaces. The second part of the thesis deals with rotary controllers with haptic feedback. As opposed to touch screens, knob controllers do not require visual attention, thus allowing a user to perform multiple actions simultaneously. In this part, the design of two innovative knob controllers is presented, each providing specific haptic features: while the first device essentially consists of a technology improvement in the form of a programmable haptic knob exposing a low-cost force resistance technology, the second device, called "Non-a-knob", affords a new interaction primitive in which rotation gestures are operated over a motionless cylinder. Along with its machine learning-based sensing algorithm, a validation experiment is presented aiming at studying objective and subjective parameters of the user interaction. Finally, the third part of the thesis shows some hardware and software tools that were developed in the context of this research.

Acknowledgments

It is difficult to find the right words to thank all the people who helped me to carry out this research.

First of all, I want to thank my supervisor, Prof. Federico Fontana for guiding and helping me on this long journey with enthusiasm and patience. I want to thank Dr. Stefano Papetti and Dr. Hanna Järveläinen for their mentoring and fundamental contributions to the entire research. I also want to thank them for their warm hospitality during my visit to the Institute for Computer Music and Sound Technology (ICST) at the Zurich University of the Arts.

I want to thank the reviewers for their nice comments and suggestions, which improved the quality and readability of this thesis.

I want to thank all the students of the University of Udine who participated or helped in the experiments; in particular, I want to thank Andrea Boato and Elisa Gaiarin for their contribution in carrying out the experiments.

Many thanks to the Patent Office of the University of Udine for their support in drafting and filing our patent.

I want to thank Electrolux Professional and The Research Hub for funding this research. I want to especially thank Alessandro Morassut and Mauro Sonego for their mentoring, friendship, and constant professional support during my PhD. Their true passion for work has inspired and motivated me a lot during this period. My gratitude also goes to Riccardo Furlanetto and Michele Simonato, who gave me the opportunity to start this journey with Electrolux Professional, supporting my research ideas from the very beginning.

Many thanks to all the other PhD students and colleagues of The Research Hub of Electrolux Professional; in particular, I want to thank Emidio Tiberi, Eleonora Pippia, Débora Pereira and Giulia Diamante for their friendship, their professional support, and their almost infinite patience in putting up with my messy desk and the constant noise of my experiments.

Finally, my greatest thanks goes to my family for the constant support in all my choices and to my beloved Laura, who inspired me in reaching this milestone, constantly supporting me during my journey.

Contents

List of Figures

List of Tables

Introduction

The human interaction with the world is multisensorial [30]. We are constantly provided with a wealth of information to support interaction with the external world thanks to the simultaneous use of multiple sensing modalities. In the research field of Human-Computer Interaction (HCI), the interactions involving multiple senses are called *multimodal* [222, 210]. In spite of this, for decades the technologies in the field of HCI have been mainly related just to visual and auditory interactions [168]. Comparably, the tactile information provided by digital devices was essentially neglected. As a matter of fact, the devices we use everyday provide less and less tactile information: touch screens, for instance, although operated by the fingers, are mostly providing the same tactile response to any action, while vocal controlled interfaces are touch-less at all. Nowadays, touch screens are the prevalent user interfaces used in our daily life. The success of touch screens can be attributed to the directness and intuitiveness of their interaction mechanism. Indeed, the modality of interacting with digital objects by touching them follows exactly the human's interaction primitive of acquiring physical objects. In addition to this, touch screens are chosen by the industry since their layout can be reconfigured without modifying any hardware component indefinitely. These characteristics made touch screens an optimal replacement for many physical layouts (e.g., button, sliders, knobs) [138, 8]: they have been progressively applied to smartphones, tablet PCs, smart devices, automotive and, eventually, to home and professional appliances; for instance, all the physical buttons and knobs that were needed to control washing machines or ovens can now be substituted by a single flat touchscreen. Through these replacements, the display increases its importance by embedding all the functions of buttons, sliders and knobs, thus becoming an input/output device. Consequently, the intrinsic tactile information given by physical controllers disappeared. As a result, even if the controls are visually displayed on the screen, they can not be identified by touching the screen and, without such tactile information, users must rely only on the visual feedback to operate the device. This may cause, for instance, wrong or double selections if the visual refresh rate is slower than expected. Thus, compared with their physical counterparts, touch screen controls increase the visual load and are considerably more prone to selection errors, especially when they do not provide a coherent multimodal (i.e. auditory or tactile) feedback [29, 137]. To overcome such a problem, the current touch screen implementations often provide additional auditory feedback: whereas these feedback are mainly abstract and not very informative (except for alarms and warnings), in specialized working conditions they can be overwhelmed by many sources of noise, resulting unrecognizable.

Although multimodal interaction may be ancillary for the consumer market, where the interactions happen in everyday conditions and the visual attention can be always focused on the user interface, conversely it can be crucial in specialized work environments characterized by safety risk, repeated tasks, unpredictable sources of disturbance and high stress levels. Despite these characteristics belong to many workplaces, in this research the focus will be on the specific work environment of professional appliances.

Professional Appliances

Electrolux Professional, the funder of this research, is a multinational company in food service equipment and laundry solutions. It fulfills the business-to-business [1](B2B) HORECA [2] market by providing a comprehensive range of solutions. Regarding the food service equipment, the company produces many tools to store, prepare, cook, serve food together with dish-washing equipment; regarding the laundry sector, instead, Electrolux Professional produces mainly washing machines, drying machines and ironers. Electrolux food service equipment is tailored for different types of professional kitchens: restaurants, pubs, bars, Quick Service Restaurants, hospitals, schools and military canteens. In the same fashion, laundry solutions cover industrial appliances, coin operated washing and drying machines, and special machines designed for sanitizing and sterilizing garments.

As a leader in market, Electrolux Professional is continuously innovating the products with advanced solutions to improve efficiency and productivity of professional kitchens and laundries. Among others, one aspect of the innovation is the continuous improvement of the user-machine interaction. On the one hand, multiple ergonomic studies carried out during the design of the appliances result in safer machines: in fact, all the appliances are designed to minimize injuries, excessive weight handling, uncomfortable working pose and long term professional illnesses. On the other hand, appliances are everyday more and more connected through Internet of Things (IoT) technologies. Such connection allows for remote monitoring and management, and predictive maintenance. However, on-board user interfaces still play a fundamental role in the appliances' control, being the principal communication channel between the machine and the humans. As mentioned above, touchscreens are literally flooding the user interface market, being, *de facto*, a must-have user interface element for all new devices and appliances; however, the need for innovation can collide with practical and unexpected issues: for instance, anyone regularly experiences the frustration caused by a wrong selection on the smartphone touchscreen while performing an important task (e.g., an emergency call). This simple dialing error can translate in a long recovery time that is perceived as neverending during moments of anxiety. Thus, as working environments can be dangerous and demanding from a physical and cognitive point of view, humans and their safety could benefit from robust interactions, nowadays not afforded by touchscreen interfaces.

---

[1] Business-to-business marketing involves the commercial transaction of a company's product or service to another company.

[2] From international Union of National Associations of Hotel, Restaurant, and Cafe Keepers, HORECA is an abbreviation used in Europe to designate the food service Industry Market (Restaurants, Hotels, Bars And Cafes, Supermarkets, Hospitals And Care Homes, Business, Transport & Industry, Commercial Laundries, Self-Services Laundries).

Research questions

The research reported in this thesis aims at restoring the multimodal interaction that physical controllers inherently provide to the user by rendering physical characteristics (e.g., material properties) over neutral surfaces or by programming mechanical behaviors, thanks to the combination of virtual and tangible haptic displays. In particular, this research will investigate the simulation of two fundamental input controls for professional appliances: the button and the knob.

Considering the limitations imposed by the existing technology, in particular concerning the rendering of tactile properties, this work is guided by the following research questions:

- How different is the interaction robustness and reliability between physical and virtual controllers?
- Which key features of physical controllers are we able to reproduce on virtual controllers holding today's technology? And which are the expectations for future technological advances?
- Can the usability level of specific physical controllers (i.e. buttons and knobs) be approached by virtual controllers, or by hybrid combinations of virtual and physical controllers?

Background and related work

The term *multimodal interfaces* belongs to interactive systems aiming to leverage natural communication; humans, on the one hand, produce speech, gestures, touch, gaze or facial expression, while, on the other hand, simultaneously detect every feedback through their perceptual channels: visual, auditory, tactile, kinesthetic, olfactory and gustatory [30, 62]. Human–computer interaction historically focused on unimodal interactions: since long in the interactive computing history, input and output were respectively modeled on one single channel. The first demonstration of multimodal computer-human interaction is widely considered to be "Put That There", a system controlled through voice and gesture inputs, developed by Richard Bolt in 1980 [24]. However, previous unimodal interactions were, often unintentionally, multimodal to some extent (e.g., mouse movements and clicks, auditory feedback by-products from hard-drives and printers). The interaction between humans and machines has been always multimodal too, but less rigorously studied: whereas HCI has evolved mainly driven by academic research, the development of the user interfaces of products, like industrial machinery or professional appliances, was usually driven by technology outbreaks or costs' reduction. Indeed, until recently, such user interfaces displayed also few configurable parameters, individually controlled by buttons, switches and knobs and displayed through LEDs or 7-digits displays. However, the increasing number of functions, nowadays provided by professional appliances, calls for more complex user interfaces, result of tailored user interaction studies.

Multimodal perception

As known from the literature, the human brain integrates a continuous flow of signals, merging them to form a coherent and robust percept, using different combination and integration strategies [67]. For this reason, studies on multimodal interaction are not limited to methods and technologies encoding the human expressions and providing multimodal feedback, but also investigating the processes behind the human perception and the integration of multimodal stimuli. As a general rule, when multimodal stimuli are coherent, redundant and temporally synchronized, they have an additive effect, resulting in stronger percepts [59, 77]; conversely, when stimuli are not coherent or not synchronized, results can be unpredictable, generating weak or no effects, or even perceptual illusions. Among others, the auditory and tactile perceptual channels interact constructively when congruent stimuli are delivered simultaneously [240, 241, 205, 206]. However, their temporal resolution is greater than audio-visual or visuo-tactile combinations [81], being both very sensitive to temporal delays, detecting especially low latency values relative to each other. Thus, systems providing multimodal audio-tactile feedback must be able to precisely control the temporal synchronism of the feedback. Thanks to several experimental studies, it has been discovered that hearing and touch have mutual influences: whereas Schurmann *et al.* found that congruent tactile stimuli enhanced the hearing perception of low amplitude sounds [196, 195], sounds synchronized with vibrations allow to lower tactile intensity thresholds [190], as well as enhance the tactile intensity perception [220]; more surprisingly, the manipulation of auditory feedback characteristics (e.g., spectral content) can significantly affect the tactile perception [245, 32, 91, 139], often resulting in perceptual illusions: for instance, in the parchment-skin illusion [118], while rubbing their-own palm, participants felt their skin drier or wetter depending on the spectral content of the filtered rubbing sound provided through the headphones. Another interesting perceptual illusion found in the literature is the marble-hand effect: human subjects, whose hands were gently hammered, felt their own hand to become as much harder and heavier as the provided contact sound did [198]. These studies suggest that the integration between auditory and tactile feedback has promising potentials to enhance the interaction robustness, however such integration effect needs to be investigated within the specific applications.

Rendering and perception of material properties

Although human perception has been tested mainly with abstract stimuli (e.g., sine bursts), the use of ecological stimuli can improve their potential applicability since their features are easier to learn and retain compared to abstract stimuli [204]. An interesting research topic making use of ecological stimuli is the material identification from visual, auditory and tactile cues. However, most of the experiments found in the literature consider only a single modality; for instance, the material classification based on the auditory channel has been found to rely mainly on the frequency and decay when sounds are originated from impacts [87]. The tactile channel, instead, can identify multiple properties such as hardness [107], roughness [11], temperature [108], as well as the vibrations originated from impacts [96]. However, differently from the auditory channel, the detection of tactile characteristics is related to specific motor activity, like pressing or sliding. Regarding multisensory integration in material classification task, only few experiments have been performed: for instance, in the evaluation of wood naturalness, a visual-tactile integration was found by Overvliet *et al.* [174]. Conversely, the contribution of the auditory channel to visual feedback was found to be not significant in the perception of material properties [151]. Eventually, in the literature no specific experimental designs evaluating auditory and tactile contribution in material classification were found. In an effort to improve the robustness of user interaction with professional appliances, this research aims at adopting such kind of ecological audio-tactile stimuli to design virtual controllers, e.g., virtual buttons, embedding characteristics of their physical counterparts (e.g., auditory and tactile response to tap).

Virtual buttons

In the literature, several studies already investigated the characteristics of physical buttons, building a database of force/displacement response functions [5, 225]. In addition, many user interaction studies compared the performance of physical buttons to virtual buttons rendering different kinds of multimodal feedback: Lee *et al.* [137] found comparable performances in terms of digitization speed and appreciation only when auditory and/or tactile feedback was provided, whereas Brewster *et al.* [29] found significant performance improvement when comparing virtual buttons providing tactile feedback to virtual buttons displaying only visual feedback. Koskinen *et al.* [129] found that tactile feedback improves the usability of virtual buttons, however the satisfaction of the experience was subjective and included users who preferred sharp and strong vibrations only when the auditory feedback was absent.

Also the temporal synchronicity between audio-tactile feedback has been found to affect the quality of virtual buttons; to this end, Kaaresoja *et al.* were able to change the perceived quality of virtual buttons by varying the stimuli temporal delay between 5 and 70 ms [119]. Considering handled game consoles, instead, multiple studies found that users reached higher scores when playing on physical controllers than playing through virtual controls rendered on touch screen, confirming the contributions given by indentations and mechanics of physical controllers [244]. However, video games specifically designed for touch devices reported comparable performance concerning parameters like satisfaction and ease of control [98]. One of the greatest issues concerning the perceived quality of virtual buttons is the absence of compliance consequent to pressing actions: to resolve such an issue, Liao *et al.* designed a hybrid solution consisting of a physical button capable of displaying programmable force-displacement curves [140]; Park *et al.*, instead, designed an augmented physical button able to reproduce the perceptual illusion of different mechanics thanks to vibration actuators [178]; eventually, more sophisticated vibrotactile cues have been successfully used for conveying illusory intra-modal effects of depth, compliance, roughness or indentation on rigid surfaces [123, 170]. The absence of edges (i.e. physical protrusions) during the exploration of surface displaying virtual buttons represents a great limitation in their use in eye-free contexts: in fact, even when provided with vibrotactile feedback, touchscreens selections resulted not reliable under demanding conditions (e.g., when dialing a telephone number while driving a car) [136]. However, complex technologies operating in dynamic conditions, such as ultrasonic actuator arrays [113, 197, 231] or electrostatic displays [69, 202], can be nowadays used to simulate edges. Taken together, these results confirm that the realistic rendering of all the characteristics belonging to physical buttons still represents an open technological challenge, especially regarding the tactile feedback [9]. However, to some extent, even low-cost technologies can be profitably used to reproduce relevant features: this research will try to maximize the potentials of voice coil and piezoelectric actuators, given their ability to provide fast and powerful transients, furthermore having affordable costs.

Knob controllers

The role of physical characteristics such as form factor and mechanical behaviour becomes fundamental when considering knob controllers: in fact, the knobs flattening, i.e. the projection of the knob 3D form factor on a surface (2D), reduces significantly the usability of such virtual widgets. Experiments comparing physical and virtual knobs have highlighted that tangibility has significant positive effects on several aspects of the interaction [233, 180, 236]. More specifically, it was proved that tangible control allows for better performance in terms of error rate and interaction speed: interaction with physical knobs was found to be 20% faster compared to virtual counterparts and, additionally, subjective performance remained unaltered in eyes-free contexts. Moreover, interactions with tangible knob controllers led to more accurate selections. This suggests that, in the design of virtual knobs, their form factor should be preserved in some way by creating touch surfaces embedding physical protrusions. A hybrid solution between physical and virtual knobs consists of haptic knob controllers with programmable features: such controllers embed the inherent multimodal feedback of a physical knob while, on the other hand, properties like force feedback, displacement or even form factor can be programmed [226]. In the literature, there are many implementations of programmable haptic knobs usually actuated by DC motors [16, 124, 111] or magneto-rehological fluids [232, 10]. Similarly, the same technologies are profitably used to create programmable sliders [19] and buttons [140]. Although the performance of programmable haptic knobs is remarkable, so far their implementation on commercial products has been limited to few exceptions, given their cost, size and complexity. For instance, they can be found in safety-critical control panels [232] or in research laboratories [124].

Compared to their virtual counterparts, physical knobs have also limitations: for instance, they generally need mechanical moving parts that can be damaged or removed; moreover, physical knobs are usually not completely waterproof: this may allow seepage of liquids or dirt intakes. Overall, these two drawbacks can compromise the lifetime of professional appliance user interfaces.

Goals and research hypotheses

The above-mentioned experiments confirm some basic research results about multimodal perception [139, 59, 28]: congruent multimodal stimuli have an additive effect, improve identification and reduce the time of event detection, resulting in more robust and reliable perception-and-action loops. The acceptance of such technologies yet needs to meet several requirements: safety, technology readiness, price, size, power consumption and ergonomics. With these goals in mind, the thesis develops along with the following project points:

1. The development of reliable virtual buttons for professional appliances operating in specialized working environments. To guarantee this, since auditory cues can be affected by environmental noise and visual attention may be not focused on the interface, it is necessary to demonstrate that, even selections based on unimodal tactile feedback, gather enough information to warn the user about incorrect selections. If this statement is proved in ideal experimental conditions, it will be possible to develop a technology demonstrator, equipped with a haptic actuator, to validate the experimental results in real working conditions.

2. The development of low-cost programmable knob controllers able to reproduce variable force feedback with a resistive technology. The hypothesis is that simple resistive feedback patterns, even if originated using low-cost technologies, can still improve accuracy and precision in the rotary selection, allowing to fine-tune without focusing the visual attention on the user interface.

3. The development of a new type of user interface for professional appliances: a haptic knob that, without having moving parts, preserves at least the form factor of its physical counterpart. The hypothesis is that rotary gestures can substitute rotations accomplished with physical knobs, resulting in a sealed interface without mechanical moving parts. This hypothesis must be proved concerning three aspects: user proprioception should not be affected by the gesture primitive (i.e. grasping vs. sliding), having similar accuracy and precision during rotations in non-visual conditions; the development of the sensing technology for precisely detecting rotary gestures should be affordable and, finally, users should welcome this new interaction primitive.

Thesis structure

This thesis is divided into three parts:

Part I covers some aspects related to surface haptics [9], with the specific goal of rendering virtual buttons for professional appliances. In particular, experimental studies on finger press and multimodal perception are used to support the creation of a prototype providing rich tactile feedback. Chapter 1 presents two experiments on multimodal perception: such experiments are designed to investigate the human ability to classify different materials through auditory, tactile or bimodal ecological feedback [46, 45]. Chapter 2 reports the technical description and the characterization of a prototype interface able to display virtual buttons. The chapter shows a user validation experiment carried out using the same haptic feedback used in the previous experiments (Chapter 1), as well as stimuli specifically designed for the device [53]. Finally, Chapter 3 presents an experiment aimed at investigating the effects of force and direction in the vibration perception in order to improve the design of haptic devices. In particular, the experiment evaluates whether the direction of the stimuli (i.e. normal or tangential) or the finger pressing force influences the human sensitivity to 250Hz vibrations [57].

Part II deals with rotary controllers with haptic feedback. In particular, this part reports two innovative technological solutions aimed at replacing traditional mechanical knobs. Chapter 4 reports the design and the validation of a haptic knob providing programmable resistive force feedback [48], whose low-cost hardware/software architecture has been filed with the Italian Patent Office (IPO) [44]. The chapter presents also specific implementations of the device concerning multimedia production tools [51], digital effects control [49] and professional appliances control. Chapter 5 compares the performance of rotary and motionless knobs: the latter is a cylindrical user interface without moving parts, able to encode rotation gestures applied to its sides. In this chapter, several objective measurements and subjective evaluations are used to compare the two interaction primitives [47]. Finally, Chapter 6 reports the design of the *Non-a-knob*: a motionless knob that encodes rotation gestures by processing capacitive signals with a Machine Learning (ML) algorithm. Besides its technical description, the device is furthermore validated in an experiment that compares its performance with the results reported in Chapter 5 [50]. In the final part of the chapter, the Non-a-knob is further discussed as part of a new concept design collecting multimodal input/output features within a single user interface.

Part III describes the hardware and software tools developed in the context of in this research. Chapter 7 reports the design of the Bogus Finger: an open-source robotic tool for the simulation of quasi-static finger pressing on stationary and vibrating surfaces. Besides the technical description and the end-to-end validation [55], this chapter shows the mechanical model of the device, the experimental estimation of its parameters, and the model validation [58]. This open-source project has been presented and discussed within the haptics community at different stages of the development [54, 56]. Finally, Chapter 8 shows the versatility of the programming language Python in several research activities. Indeed, almost all the software presented in this research has been developed in Python: experimental procedures, control algorithms, machine learning algorithms, statistical analysis and chart plots. In particular, this chapter shows an in-depth study to investigate the limits of this programming approach concerning the real-time data processing [52, 43].

Figure 1 summarizes the proof-of-concept prototypes developed and validated along this research. On the left, four physical buttons, providing different compliance and mechanics, are rendered as enhanced virtual buttons through the display presented in Chapter 2: the display, besides showing the buttons, is able to provide a rich haptic feedback dependent on the finger pressing force. On the right, the haptic characteristics of regular mechanical knobs are reproduced and improved through different prototypes: the first device is the low-cost programmable haptic knob described in Chapter 4, whereas the second device is the innovative motionless knob ”Non-a-knob”, conceptualized and validated in Chapter 5.

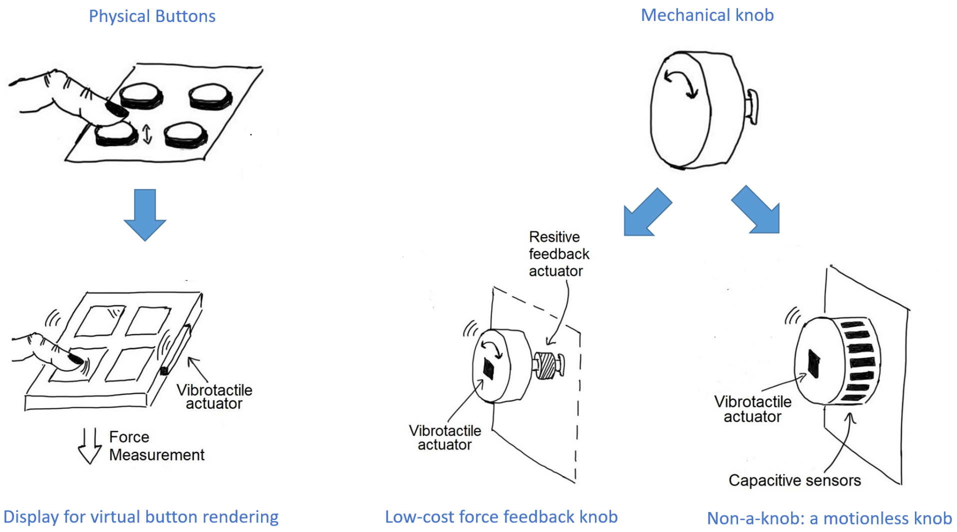


Figure 1: Reconstruction of the physical controllers’ cross-modal feedback (top part) through the design and validation of three low-cost technology demonstrators (bottom part): a haptic display rendering virtual buttons, a resistive programmable knob and a motionless knob.

A note on the ethical issues

All the experiments and data collections were performed in the context of the collaboration with Institute for Computer Music and Sound Technologies (ICST) of ZHDK (Zurich), particularly concerning the treatment and processing of the data. Experiments and data collections which involved human participants were not harmful nor dangerous for the subjects, which participated on a voluntary basis. No health-related personal data were collected during the experiments, and furthermore, data were always anonymized. For all these reasons, the experiments did not fall within the scope of the Swiss Human Research Act https://www.admin.ch/opc/en/classified-compilation/20061313/index.html and, therefore, they did not require the approval of the ethics committee.

I

Surface Haptics

1

Material identification from bouncing events

As Humans, when we approach an object, we initially identify everyday materials from their visual aspect. Visual identification can be later refined through touch [40], by analyzing tactile surface properties such as roughness [11] and temperature [108]. This analysis postulates a material to be fully characterized by its superficial appearance.

Further material properties can be actively explored using touch and hearing, as when an indentation or a tap unveil the hardness of an object. Point-wise tapping, in particular, generates impulsive audio-tactile feedback whose role goes beyond aesthetics [151], giving rise to cross-modal cues which are difficult to disambiguate [82].

In the literature, the classification of material, size and shape from impulsive auditory feedback has been successfully performed by listeners identifying synthetic stimuli reproducing strikes on suspended plates [134, 87, 221] and clamped bars [128, 147]—see also Giordano and Avanzini [86] for a comprehensive review of related work. In fact, the simulation of objects vibrating in mechanical isolation enables fine control of their oscillatory *modes* through the amplitude, frequency and decay parameters of each mode. Depending on their setting, these parameters link the physical properties of an object to its auditory perception by means of fundamental cues such as decay, pitch and timbre. Experiments aiming at applying materials' sound synthesis to auditory displays and interfaces [84] have suggested that everyday materials are first roughly classified into distinct groups (e.g., metals) depending on decay. Once grouped, further categorization may be based on characteristic ("material") frequencies [87]. The latter association was shown to become especially important when the stimuli are short (i.e. less than about 400 ms), as decay cues in this case become difficult to perceive [146].

Classification of materials by impulsive tactile feedback has been researched too, albeit less systematically. Most of the works deal with direct or mediated finger tapping, especially in view of applications to robotic sensing and material augmentation. Kim and Kesavadas [126] parameterized a contact model to reproduce different materials, by acquiring temporal patterns of force from participants tapping on steel, aluminum, wood, and rubber surfaces during an identification task. Hachisu *et al.* designed a stick that, when tapped, cancels its own body's response and then renders haptic sensations of aluminum, wood and rubber by synthesizing a damped sinusoid with characteristic amplitude, decay and frequency parameters [96]. Both works reported successful recognition of the proposed materials, with possible support from sound in the former. The latter was later applied to touchscreen augmentation on a tablet displaying playable percussion instrument boards made of wood and metal [95]. An exception comes from Higashi's systematic research on tactile perception of hardness [107], resulting in intensity curves [104] and mechanical parameter ranges [105] of equal hardness perception, as well as in a psycho-physical map linking materials to perceived stiffness in response to a tap [106].

This chapter presents two experiments that consider both the auditory and tactile sensory channels, in an effort to assess their individual contribution while forming a multi-sensory material category. In fact, only a minority of the literature about the influence of hearing on touch [205, 206, 245, 32] during material classification considers impulsive feedback.

In the following experiments the auditory and tactile responses to an impulsive excitation were recorded from three flat objects made of different materials, first taken in mechanical isolation and then resting on a table. Then, such recorded sounds and vibrations were reproduced, either separately or together, respectively through headphones and on a hard glass plate actuated by a vibrotactile transducer. The tactile display used in the experiments avoided surface texture rendering technologies [64, 246]. Instead, for its simplicity and low cost [60], the proposed setup is ideal for testing the audio-tactile feedback of virtual buttons on touchscreens specifically designed for professional appliances, goal of this research. Coherently with this goal, temperature cues, that could further characterize the materials, were removed from the experiments.

In the literature only the study accomplished by Smith *et al.* indirectly linked material classification to virtual buttons by proving that abstract auditory feedback can be more difficult to learn and retain than environmental sounds [204].

Hence, the experiments presented in the following minimized the abstraction of the audio-tactile feedback using audio recordings rather than synthetic stimuli.

1.1 Experiments

The two experiments used stimuli recorded from single impacts on three flat objects made respectively of wood, plastic, and metal. The experiments differed in the main resonance decay times, as a consequence of recording sounds and vibrations either with suspended objects (Experiment 1), or more realistically with the same objects resting on a table (Experiment 2). Furthermore, a control test was set up using real impact events on the same materials when they were in mechanical isolation, i.e. in the same condition as when sounds and vibrations for Experiment 1 were recorded. The purpose of the control test was to set a reference baseline on the human ability to classify materials based on the original objects' audio-tactile feedback.

Provided, as confirmed by a control test, that humans are able to classify wood, plastic and metal by impulsive auditory, tactile, and finally audio-tactile feedback, then the hypothesis is that the bounce of an unfamiliar light object (e.g. a ping-pong ball) on flat objects made of those materials enables the same process across both the auditory and tactile channels.

1.1.1 Setup

Wood, metal, and plastic materials were selected as they respond rigidly to impacts (i.e. with spectral energy concentrating in the high frequency range), thus enabling realistic tactile reproduction on a glass surface, offering just vibratory feedback, rather than kinesthetic cues that are linked to soft materials [96].

Control test

Three flat objects were built out of fir wood, hard plastic, and steel. They were U-shaped by bending or carving, allowing for a hand or an accelerometer to find sufficient room in the resulting cavity underneath (see Fig. 1.1). All objects were sized 160×160×45 mm. Two circular patches having a diameter of about 4 cm, made of thin adhesive film, were attached at the same location on both sides of the surfaces. Both (i.e. the patch on the reverse side for Experiment 1 and the patch on the top side for Experiment 2) offered a uniform surface spot where participants put their fingers. In this way, subjects could not use surface properties to identify materials. On the opposite side, these spots marked the impact point. Thanks to their low mass, thinness, and firm adhesion to the objects, they introduced almost imperceptible changes in the impact sounds and vibrations.

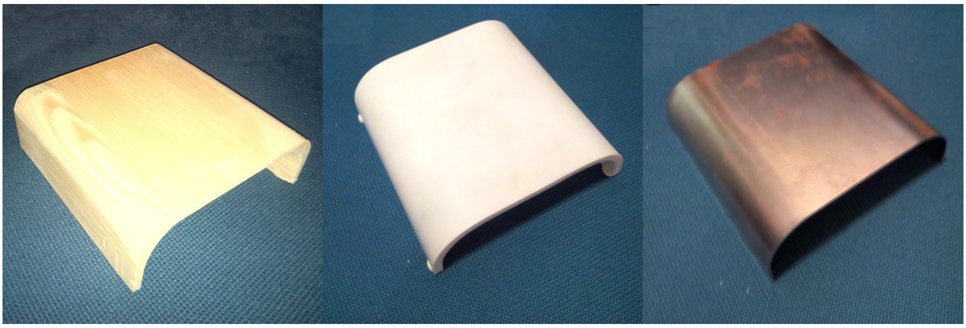


Figure 1.1: Wood, plastic and metal objects used to record stimuli and perform the control test.

Experiments 1 and 2

A flat object was built by mounting a 3 mm-thick borosilicate glass plate on a metal frame suspended by means of rubber strips, and then coupling the frame with a wooden structure, as shown in Fig. 1.2.

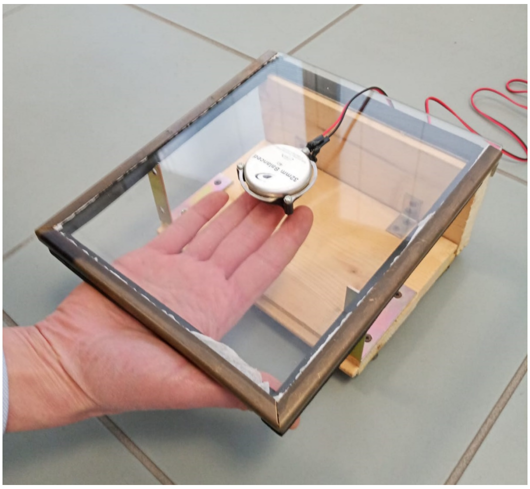


Figure 1.2: Suspended glass plate with attached exciter, used in Experiment 1 and 2.

1.1.2 Stimuli

A ping-pong ball was used to excite the materials, as it has a light yet rigid structure, giving rise to neat impact events characterized by small energy in the low frequency range. Tests were also made with metal, rubber, and wooden balls of different size and weight, however they produced impacts whose energy at low frequency fell outside the range of the small, low-power actuator that is required for vibration reproduction in Experiment 1 and 2.

Control test

The ball was dropped on the three objects. The intensities of the stimuli were equalized across materials by dropping the ball from varying heights: 30 cm for wood, 80 cm for plastic, and 40 cm for metal. A marked rod was placed near the cardboard support, helping the experimenter to release the ball correctly during the experiment.

Mechanical decoupling was realized by putting the objects upside-down on a support made of foam and cardboard sized 200×240×60 mm, shown in Fig. 1.3 (center). The support also forced participants to touch the surface only in correspondence of the adhesive tape, as in Fig. 1.3 (right).

The temperature of the objects was stabilized at approximately 30°C, by keeping them under a halogen lamp starting ten minutes before and throughout the experiment when not in use.

An inspection of the temporal signals immediately after the bounce showed the presence of a low-frequency component identical in all cases, evident consequence of the response of the support. On top of this component, fading transients with a peak occurring within the first 100 ms were clearly visible. After removing the component in low frequency, such peaks showed a relative amplitude of approximately 0.30 mm for wood, 0.14 mm for plastic and 0.08 mm for metal. Decreasing peak values

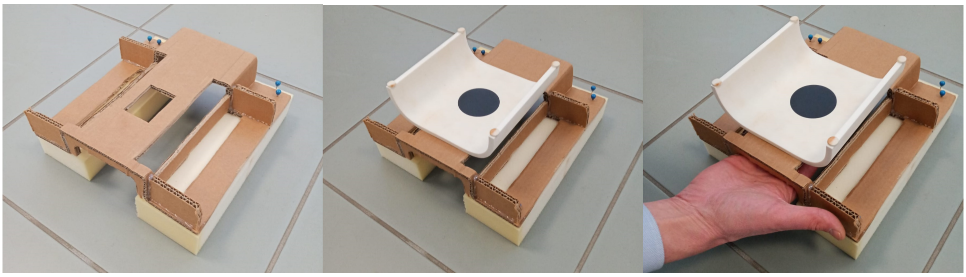


Figure 1.3: Cardboard support: without (left) or with (center) plastic object turned upside-down, and in use during the control test or familiarization with Experiment 1 (right).

are compatible with the implemented intensity equalization, as the corresponding materials produced different decays as explained below.

Experiments 1 and 2

These experiments made use of reproduced audio and tactile stimuli: sound and vibration samples were recorded from a single ball hit on each surface. The objects were either turned upside-down and suspended as in Fig. 1.3 (center), producing samples for use in Experiment 1, or resting on a table (see Fig. 1.6) for Experiment 2.

Sounds were recorded with an Audio-Technica AT4050 condenser microphone placed 40 cm away from the bouncing point. Vibrations were recorded by attaching a Wilcoxon 736 accelerometer in correspondence of the adhesive film. Both devices were connected to a RME Babyface Pro audio interface—the accelerometer through its companion pre-amplifier.

Auditory stimuli were played back through a pair of Beyerdynamic DT 770 PRO closed-back headphones. Tactile stimuli were reproduced by a Dayton Audio 32-mm balanced vibrotactile transducer, attached at the top side of the glass plate. Bimodal stimuli were provided by playing back auditory and tactile stimuli at the same time. In this case, the auditory signal was delayed by 1.14 ms, corresponding to the time needed for airborne sound to travel from the impact to the listening point.

Spectrograms of the audio recordings made for both experiments are shown in Fig. 1.4. In order to highlight the frequency content of the stimuli, the reported spectrograms were generated using a 16384 samples windowing; this inevitably put in the background the temporal information on the same spectrograms. On the other hand, the same attacks were deliberately chosen to be as short as possible; for this reason, their spectral content was inherently difficult to disambiguate in a spectrogram. They show differences below 30 Hz, consequence of the different support employed, which were however inaudible. A closer look to the audible band reveals that the stimuli in Experiment 1 were about 0.1 s longer, with a strong resonance in metal at about 3 kHz, lasting about 0.9 s.

Figure 1.5 shows spectrograms of the recorded vibrations in the top and middle rows, unveiling differences similar to what found for audio. Furthermore, metal in low

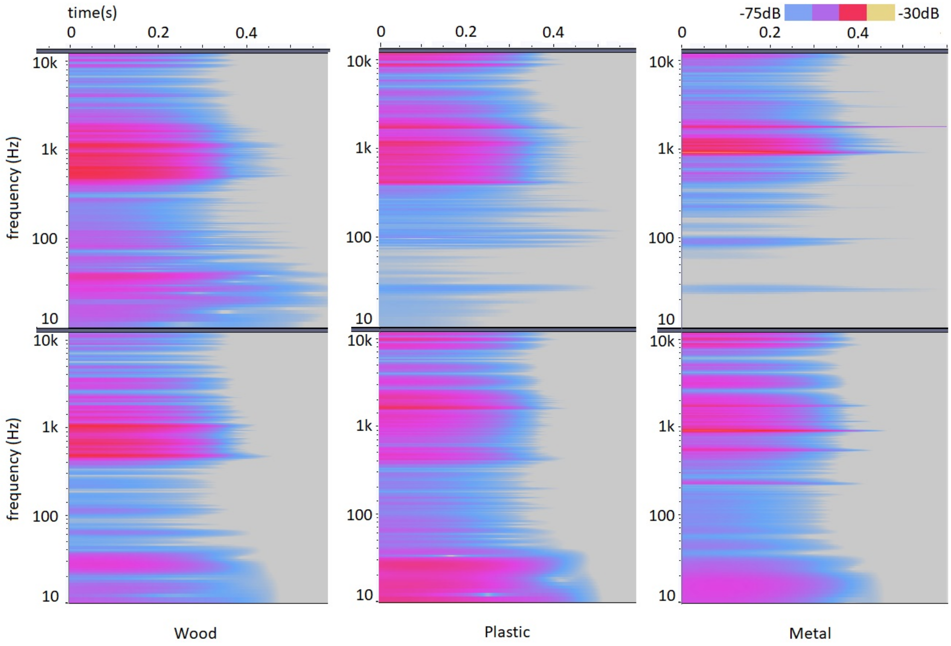


Figure 1.4: Sound spectrograms in Experiment 1 (above) and 2 (below).

coupling conditions generates long-lasting vibrations at about 20 and 250 Hz, that were not efficiently radiated across the air.

Spectrograms of the vibrations after reproduction on the glass plate during Experiment 2 are also shown, in the bottom row of Fig. 1.5. They were acquired by placing the accelerometer in correspondence of the presentation point of the plate—see Fig. 1.2. A comparison between these and the original vibrations in Experiment 2 (middle row) discloses some unavoidable differences affecting the tactile stimuli during reproduction. In fact, the limited admittance of glass at low frequencies and the frequency cutoff of the actuator progressively attenuate frequencies below 200 Hz. Moreover, the denser modal distribution of the glass causes the resonances at higher frequencies to fragment into subgroups gathering two or three original vibration modes together.

Table 1.1 summarizes the characteristics of the stimuli used in Experiments 1 and 2, and in the control test.

1.1.3 Participants

Participants were recruited among students at the University of Udine and employees of Electrolux Professional SpA. They participated on a voluntary basis and were not paid. Their auditory and tactile acuity was informally tested by asking participants to close their eyes, then localize a sound source nearby, and finally identify the materials

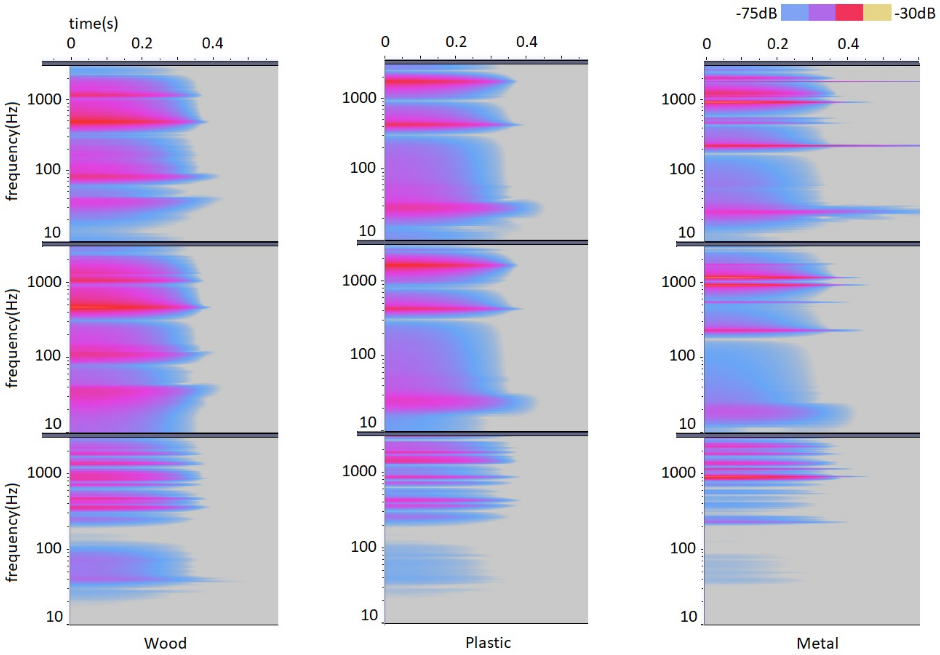


Figure 1.5: Vibration spectrograms in Experiment 1 (top row) and 2 (middle row). Vibrations reproduced on the glass plate in Experiment 2 (bottom row).

used in the experiment by touching the respective object outside the adhesive tape.

Control test

Sixty participants, aged between 19 and 52 (M=24.3; SD=6.7), took the control experiment.

Experiments 1 and 2

Twenty-five subjects between 23 and 61 years old (M=32.1; SD=10.1) participated in Experiment 1, and twenty-seven (21-54 years old; M=29.0; SD=6.8) in Experiment 2. Eight subjects participated in both experiments. Roughly one third of the participants were females.

1.1.4 Design and Procedure

In all experiments, the design consisted of two within-subjects factors: Material and Modality. Material was either Wood, Plastic, or Metal. Modality was either unimodal Auditory, unimodal Tactile, or Bimodal audio-tactile. The factors were crossed and each factor combination was repeated six times, resulting in 6 × 3 Materials × 3

Table 1.1: Characteristics of the stimuli used in the experiments.

Experiment	Stimuli	Setup	Coupling	Resonance decays
control	live	cardboard support	low	slow
1	recorded	cardboard support	low	slow
2	recorded	on the table	normal	normal

Modalities = 54 trials. Trials were organized in blocks according to Modality. Both unimodal conditions were presented before the Bimodal condition, and the order of Auditory and Tactile conditions was balanced among participants. Within each block, six repetitions of each Material were presented in random order. The experiment lasted about 10 minutes.

The task was to classify and report the material by saying its name. Responses were noted by the experimenter and audio-recorded for later reference. Participants were blindfolded during the control test. In all experiments, during unimodal Tactile trials, they received pink noise through headphones to mask unwanted auditory feedback.

Prior to each experiment, participants familiarized with the real audio-tactile events by listening to the impact sounds while keeping one or two fingers of the dominant hand on the adhesive spot (see Fig. 1.6) until they felt they could confidently recognize the respective materials through those cues.

Figure 1.6: Familiarization in Experiment 2.

Control test

A trial consisted in the experimenter dropping a ball on one of the objects from the prescribed height. In Tactile and Audio-Tactile trials, participants placed one or two fingers below the object through the cardboard support, as during familiarization (Fig. 1.3 (right)). The other two objects were in turn kept under the halogen lamp to avoid changes of their temperature during the session.

Experiments 1 and 2

A trial consisted in playing back a recorded impact event, presented through headphones and/or the actuated glass plate, as shown in Fig. 1.2.

1.2 Results

1.2.1 Control test

Table 1.2 reports the confusion matrix for the Auditory, Tactile and Bimodal modalities. Each diagonal contains the total proportion of correct responses in bold symbols, while the other cells report false responses. Columns labeled 'None' report missing responses. Figure 1.7 presents a boxplot of individual proportions correct for Modality (Auditory, Tactile, Bimodal) and Material (Wood, Plastic, Metal).

Table 1.2: Control test: confusion matrix for each condition.

Condition	Auditory				Tactile				Bimodal		
Response → Stimulus ↓	Wood	Plastic	Metal	None	Wood	Plastic	Metal	None	Wood	Plastic	Metal
Wood	**90.0%**	9.4%	.6%	0%	**66.9%**	26.7%	6.4%	0%	**99.7%**	.3%	0%
Plastic	6.4%	**90.3%**	2.8%	.5%	21.7%	**74.7%**	3.6%	0%	0%	**96.7%**	3.3%
Metal	.3%	2.8%	**96.4%**	.5%	2.5%	6.4%	**90.8%**	.3%	0%	2%	**98.0%**

Concerning unimodal conditions, Wood and Plastic were classified much better in the Auditory than Tactile condition, whereas Metal was classified well in both conditions. In the Bimodal condition, performance was nearly perfect across materials. Hence differences in performance were analysed only between the two unimodal conditions as follows. A non-parametric analysis was performed due to considerable ceiling effects in the data. In particular, a Friedman test [79] was conducted, revealing significant differences in Material ($Q = 92.25, p < .001$). Three pairwise comparisons using the Wilcoxon Rank-sum test [103] highlighted that Metal differed significantly from Plastic and Wood in the Tactile condition (Wood-Metal: $Z = 5.5$, Bonferroni-corrected $p < .01$; Plastic-Metal: $Z = 3.6$, $p < .01$). Concerning Modality, the pairwise comparisons highlighted significant differences between Auditory and Tactile for all materials ($Z = 5.2, p < .01$). Finally, a Wilcoxon Rank-sum test confirmed that presentation order (Auditory then Tactile or Tactile then Auditory) did not result in significant differences for either Auditory ($Z = .43, p > .05$) or Tactile ($Z = .7, p > .05$) identification scores.

1.2.2 Experiment 1

Table 1.3 reports the confusion matrices in the same fashion as Table 1.2. Figure 1.8 shows a boxplot of individual proportions correct. Compared to both unimodal conditions, the results suggest that performance was better in the Bimodal condition.

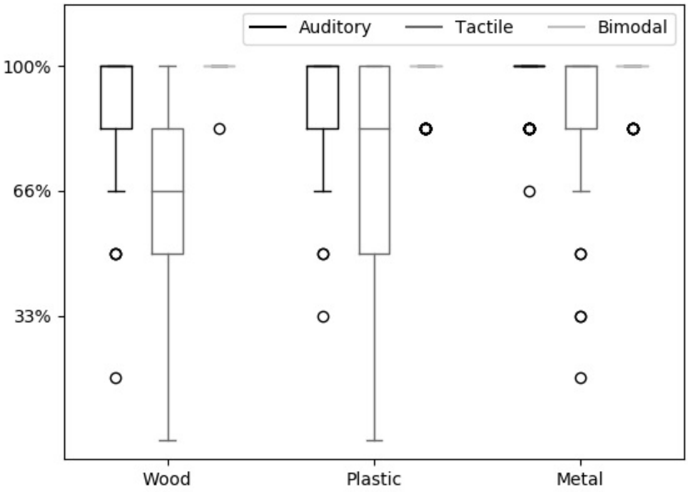


Figure 1.7: Control test: Boxplot of proportions correct for all condition combinations.

Table 1.3: Experiment 1: confusion matrix for each condition.

Condition	Auditory				Tactile				Bimodal		
Response → / Stimulus ↓	Wood	Plastic	Metal	None	Wood	Plastic	Metal	None	Wood	Plastic	Metal
Wood	**79.3%**	20.0%	0%	.7%	**62.7%**	20.0%	16.6%	.7%	**90.0%**	9.3%	.7%
Plastic	24.0%	**72.7%**	3.3%	0%	23.3%	**63.4%**	13.3%	0%	11.3%	**87.4%**	1.3%
Metal	1.3%	2.0%	**96.7%**	0%	22.7%	11.3%	**66.0%**	0%	.7%	2.7%	**96.6%**

Again, the score distributions deviate from normal due to a ceiling effect, hence a Friedman test was used. A significant main effect of Modality was detected ($Q = 37.8, p < .01$). Three pairwise comparisons were performed between modalities using the Wilcoxon Rank-sum test. Significant differences were detected between Auditory-Bimodal ($Z = -2.7$, Bonferroni-corrected $p < .01$) and Tactile-Bimodal ($Z = -5.4, p < .01$).

A more detailed inspection of the two unimodal conditions shows higher median scores for Auditory than Tactile. In the Auditory condition, Metal was classified especially well. A Friedman test, considering each factor combination as one of six conditions of a combination factor, revealed significant differences ($Q = 21.8, p < .01$). Six pairwise comparisons were performed. Three comparisons between materials in the Auditory modality revealed that Metal significantly differed from Plastic and Wood (AuditoryWood-AuditoryMetal: $Z = 4.3$, Bonferroni-corrected $p < .01$; AuditoryPlastic-AuditoryMetal: $Z = 3.4$, $p < .01$). Further three comparisons were performed for each Material between the Auditory and Tactile modalities. A sig-

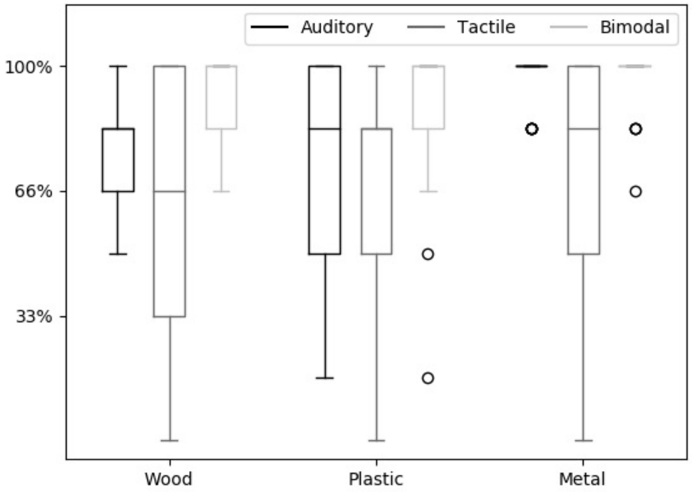


Figure 1.8: Experiment 1: Boxplot of proportions correct for all condition combinations.

Table 1.4: Experiment 2: confusion matrix for each condition.

Condition	Auditory				Tactile				Bimodal		
Response → / Stimulus ↓	Wood	Plastic	Metal	None	Wood	Plastic	Metal	None	Wood	Plastic	Metal
Wood	**75.9%**	16.1%	6.8%	1.2%	**67.9%**	13.0%	17.9%	1.2%	**87.0%**	7.4%	8.3%
Plastic	11.7%	**62.4%**	24.7%	1.2%	17.9%	**53.1%**	27.8%	1.2%	7.4%	**67.6%**	29.6%
Metal	20.4%	29.0%	**50.0%**	.6%	13.0%	36.8%	**49.4%**	1.8%	5.5%	25.0%	**62.1%**

nificant difference was detected only for Metal (MetalAuditory-MetalTactile: $Z = 3.5, p < .01$).

1.2.3 Experiment 2

Table 1.4 reports the confusion matrices in the same fashion as Table 1.3. Performance is now generally lower and in some cases close to chance performance. Most participants performed above chance; however, two participants failed in both unimodal conditions and additional two in one unimodal condition. Metal was frequently misclassified: 36.8% of Metal trials were classified as Plastic in the Tactile condition and 29.0% in the Auditory condition. Wood and Plastic were classified better than Metal, especially from Auditory cues.

Figure 1.9 reports a boxplot and means with Standard Error (SE) of proportions correct for the same conditions as in Fig. 1.8. Again, performance was better in the Bimodal condition than in the unimodal conditions. A non-parametric Friedman test detected a significant main effect of Modality ($Q = 25.0, p < .01$). Pairwise comparisons were performed using the Wilcoxon Rank-sum test, revealing significant differences between Auditory-Bimodal ($Z = -2.5, p = .03$ Bonferroni-corrected) and Tactile-Bimodal ($Z = -3.7, p < .01$).

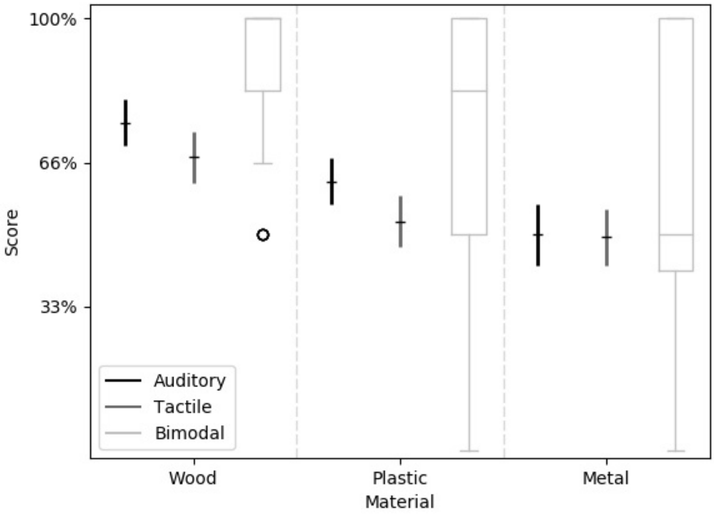


Figure 1.9: Experiment 2: Mean proportions correct with SE bars (Unimodal) and boxplots (Bimodal) for all condition combinations.

Particularly for the unimodal conditions, scores were lower than in Experiment 1. Tests on the unimodal distributions with the D'Agostino method confirmed no significant deviation from normality for all factors [42], concluding that ceiling effects were not present. Even though some skewness was found in the combination (Auditory, Wood), a parametric analysis could be undertaken.

A two-way repeated-measures ANOVA was performed using Greenhouse-Geisser correction for insphericity. A significant effect of Material was detected ($F(1.61, 41.9) = 16.3, p \leq .001$), whereas neither the main effect of Modality ($p = .09$) nor the interaction of Modality and Material ($p = .563$) was significant.

The mean results for Materials were: Wood ($M = .72, SD = .033$), Plastic ($M = .58, SD = .033$) and Metal ($M = .50, SD = .04$). Their respective 95% confidence intervals result in a partial overlap between Plastic ($.51 - -.64$) and Metal ($.42 - -.57$), while Wood is outside their combined range ($.65 - -.78$).

1.3 Discussion

Figs. 1.7, 1.8 and 1.9 show that in all tests auditory cues were more effective than tactile cues for material classification. This is not surprising, since hearing discriminates cues of frequency better than touch [230]. In the control test, however, Metal was classified almost equally well in both modalities. The most plausible explanation for this exception is that participants efficiently discriminated the longer decay of the metallic object vibrations from both sensory channels. This conclusion is consistent with previous findings, concluding that cues of damping/decay times are fundamental during material identification by hearing [87] and also by touch [107].

Further support to the above conclusion comes from Experiment 1 where participants, compared to the control test, were less precise in the Auditory modality when listening to Wood and Plastic, but once again almost infallible when listening to Metal. In fact, the auditory confusion matrix in Table 1.3 disperses the data around the diagonal limited to the sub-matrix reporting for Wood and Plastic. Headphone listening introduces spectral (hence timbral) changes, and internalizes sound sources especially if using closed-back headsets [234]. The use of such devices in our experiments hence altered the auditory recognition process, and disrupted the localization process [150]. The consequent distortion of the ecology that listeners had previously experienced during familiarization with the bouncing event may have caused larger error rates in the Auditory modality. Notably, such artifacts are less relevant for sounds made of few oscillatory components, where pitch instead of timbre cues prevail [215]. Hence, after the onset listeners might have been able to isolate the long-lasting resonance at about 3 kHz (above in Fig. 1.4) equally well for both real and reproduced metal sounds.

A similar motivation may explain the performance drop while recognizing Metal through the Tactile modality in Experiment 1. In fact, an inspection of the bottom row in Fig. 1.5 shows that the reproduction over glass progressively attenuates the resonances from 200 Hz down, and alters those above this frequency. For this reason, participants might have lost both high- [18] and low-frequency tactile pitch cues [17] visible in the top row in Fig. 1.5, which had been acquired during familiarization. Losing the former could have had consequences in identifying the metallic object. In parallel, the generally disappearing spectral energy below 200 Hz might have been responsible for a proportional performance decay of participants in identifying all materials through touch from reproduced vibrations during Experiment 1.

In Experiment 2, participants still performed above chance in both the Auditory and Tactile modalities; however, performance was generally lower than in Experiment 1. Wood essentially confirmed the scores of Experiment 1, while Plastic and especially Metal did not. This performance decay finds an explanation in the spectrograms of Fig. 1.4 and 1.5 relative to this experiment (bottom rows). According to them, both channels ceased to provide the characteristic resonances acquired by subjects during familiarization, and suggest that during the task sounds and vibrations were perceived to have different timbre and no that distinct pitch that was still present in Experiment 1. The Auditory classification of Metal suffered particularly from this situation, scoring down until about 50%. This caused in its turn a general increase of the auditory confusion, as the expected resonant timbre of Metal and

Plastic disappeared in favor of a muffled, unpitched sound inducing participants to occasionally swap the two materials, or classify them indistinctly as Wood.

The above considerations find even more solid ground with the Tactile modality. Indeed, a comparison between the mid and bottom rows of Fig. 1.5 respectively suggests that, during familiarization, these participants received characteristic low-frequency content and resonance modes; yet later, during the experimental tasks with reproduced stimuli, most of the energy below 200 Hz was not present, nor could the original resonances be retrieved from the spectral clusters in the tactile band [230] of the reproduced vibrations. Analogously to Experiment 1, the spectral distortion progressively got worse while moving from Wood to Plastic and finally Metal, with potentially proportional effects in the material identification.

The first general conclusion hence is that participants identified Metal from resonances with longer decays, when available. Then, they relied on less robust timbre and pitch cues which were present in the onset of all stimuli. This conclusion echoes the results obtained by Giordano *et al.* using auditory feedback [87]; additionally, it suggests that participants made proficient use of longer resonances also in the tactile modality, as Higashi found while investigating tactile hardness perception [107]. Wood and plastic in any case had to be classified based on spectral cues, with little or no support from temporal information: in this respect, these results are aligned with existing research on tactile recognition of musical timbre [192].

In both experiments the classification based on Bimodal stimuli was better. Especially in Experiment 1, it seems that the synergistic reproduction of audio and tactile cues was able to restore the information existing in the unimodal cues when they were experienced directly from the objects. More surprisingly, the same synergy was present also in Experiment 2 in which the sensory channels were further distorted. The logical conclusion is that participants were supported in their classification in the Bimodal condition by some form of cross-modal summation of tactile and auditory cues of material.

Sensory integration is known to optimize perceptual acuity [66]. In particular, interactions between such two channels have been reported by several authors [78], with effects depending on the spectral characteristics and temporal relationships between auditory and tactile stimuli. Even if such interactions do not necessarily lead to constructive effects [243], synchronous audio-tactile presentations of matching frequencies have been shown to improve event detection also in presence of broadband auditory noise [240].

Constructive audio-tactile summation of particular interest to our experiment was reported by Shurmann [196]. Participants performed a loudness-matching task with and without touching a bar vibrating coherently with sound. Vibrations were discovered to amplify the perception of auditory stimuli especially when their loudness was low. Further results have highlighted that the frequencies responsible for this effect range between 200 and 400 Hz [2]. In line with that and some previously cited experiments, our participants in the Bimodal condition might have detected audio-tactile cues reporting of resonance modes (be they equal in frequency or consonant [171, 74]) that conversely had disappeared or were perceptually masked in the unimodal stimuli. Their detection, hence, could have improved the classification performance. In this respect, literature from the musical haptics field provides intriguing, but not always

Table 1.5: Material classification from incongruent stimuli.

Stimulus		Response		
Auditory	Tactile	Wood	Plastic	Metal
Wood	Plastic	63.0%	32.4%	4.6%
	Metal	58.3%	22.2%	19.5%
Plastic	Wood	30.6%	48.1%	21.3%
	Metal	6.5%	37.0%	57.5%
Metal	Wood	46.3%	25.9%	27.8%
	Plastic	13.0%	45.3%	41.7%

robust evidences of multisensory perception of frequency cues [191, 112, 76].

1.3.1 Incongruent stimuli

The above considerations on audio-tactile synergy during material classification are even more interesting if considering responses to six *incongruent* bimodal stimuli, obtained by combining sounds and vibrations generated from different materials. Such stimuli were prepared with the recorded short-decay responses, as in Experiment 2. Immediately after the completion of a session in Experiment 2, the participant were asked to classify the same three Materials from four randomized repetitions of incongruent stimuli, for a total of 4 × 6 = 24 additional trials.

Table 1.5 reports how Materials were classified. The histogram in Fig. 1.10 illustrates the distribution of *consistent* classifications across Bimodal stimuli resulting from the 3 congruent and 6 incongruent audio-tactile combinations. For each combination, classifications were considered as consistent if reiterated in more than two (that is, half of the) repetitions irrespective of the identified material. Only the so defined consistent responses are represented in the histogram of Fig. 1.10. Consequently, shorter bars reflect lower consistency and thus greater confusion during classification.

As the incongruent results can not be compared to the congruent cases, the histogram can be interpreted only qualitatively. In spite of this, Fig. 1.10 suggests some interesting considerations. Congruent stimuli supported the Auditory classification of the unique Material they represented, and furthermore such classifications were mostly reliable. As reliability gradually decreases while moving to the right of the figure, consistent classifications started to occur for incongruent stimuli too, again led by the auditory channel. However, the tactile channel prevailed in the last three (on average least reliable) consistent classifications.

To this end it is possible to speculate that tactile feedback, in the limits of its abil-

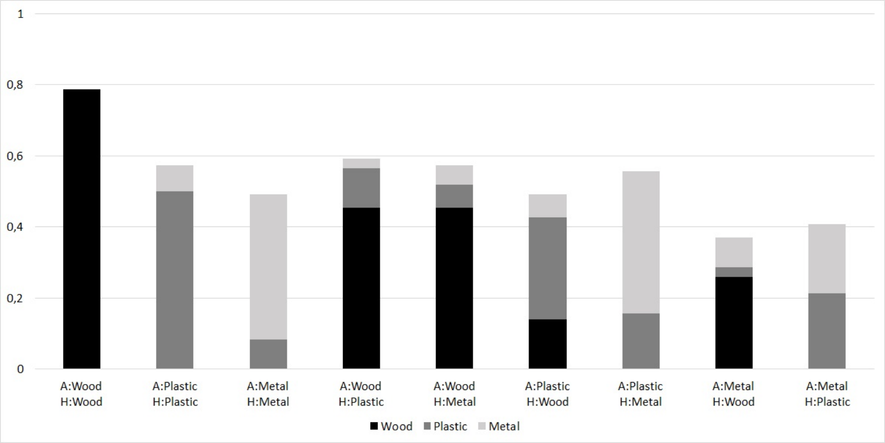


Figure 1.10: Distribution of consistent classifications from congruent and incongruent Bimodal stimuli.

ity to convey timbre, became progressively more important as the auditory channel, in front of incongruent materials, left its leading role while remaining supportive to cross-modal perception. This conclusion finds partial confirmation from experiments demonstrating that simultaneous presentation of sound and vibrations can lower tactile intensity thresholds [190] as well as enhance tactile intensity perception [220]. Concerning material classification, holding the conditions of Experiment 2 in which Metal could not be identified anymore by longer resonances, Wood established the most robust classification also when incongruent stimuli were presented: Wood was generally identified whenever it was present in at least one channel, whereas it was not identified when it was not present in either channel.

1.4 Lesson learnt

The described experiments investigated the relationships and interactions existing between the auditory and tactile channels when humans are engaged in a material classification task, based on impulsive feedback from flat objects built with those materials. These findings suggest that, while both channels are able to perform this task correctly based on real feedback, the reproduction of recorded sounds and vibrations on a touchscreen-like display deteriorates the performance especially if the material's distinctive resonances are damped (e.g., because the display rests on a table). Anyway, even in the worst conditions, unimodal tactile and unimodal auditory modalities scored greater than chance level. On the other hand, the bimodal modality always gave rise to greater classification scores, suggesting an integration effect between modalities. These experiments hence provide a baseline for the design of virtual buttons taking natural interaction with ecological feedback into consideration. The experimental outcomes here reported are in accordance with previously accepted results, showing that few decaying resonance modes are sufficient to characterize the sounds and vibrations of a button: they indeed suggest that simple audio-tactile feedback can be contextualized to reflect material properties, through proper resonance tuning and the design of suitable broad-band onsets. In fact, the design of feedback containing subtle cues of materials would be effective only if relying on technologies able to reproduce them with great accuracy.

2

A haptic display to render virtual buttons

Multimodal feedback is fundamental in realistic conditions where several noise sources can potentially affect more than one perceptual channel. As work environments often present acoustic disturbances as well as possible visual occlusions of the user interface, the professional appliances displaying virtual buttons on their touchscreen should provide at least well differentiated tactile feedback to support a reliable human-machine interaction [83]. This chapter reports the design of a touchscreen device implementing several virtual buttons having different tactile feedback triggered by finger press. Supported by the experimental results of Chapter 1, the prototype is tested on the rendering of virtual buttons having tactile properties related to ecological materials. The chapter presents the subjective evaluations of the virtual buttons accomplished by two user panels in separate case studies. In the first case study, the vibrotactile stimuli recorded from impacts on different materials are manipulated and reproduced by the touchscreen device, whereas, in the second case study, virtual buttons are carefully designed from the ground up, targeting the hardware frequency response.

2.1 The prototype

Thanks to the collaboration with Electrolux Professional, I designed and built a prototype user interface able to generate vibrotactile feedback in response to touch interactions. The device implements virtual buttons triggered by the screen position (x,y axes) and the pressing force (z axis) in the soft-touch range (0-5 N). The main goal of the device is the rendering of virtual buttons that are easy to discriminate based mainly on tactile cues, thus enabling the users of professional appliances to recognize any correct or wrong touch selection in eyes-free conditions, therefore relying on tactile feedback. As active touch enhances the sensitivity to vibrations [176, 169], even better discrimination performance was expected compared to the experiments presented in Chapter 1, which were conducted in passive conditions.

Figure 2.1 shows the layout of the prototype, built using off-the-shelf components. The device displays virtual buttons on a 2.8 inches TFT touchscreen (see Fig. 2.3) whose capacitive layer locates finger contact positions, while the exerted pressing

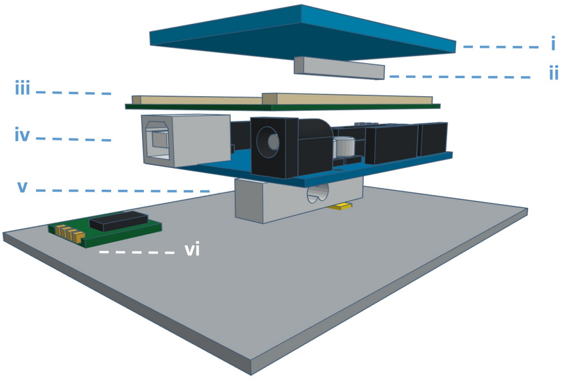


Figure 2.1: Schematic of the prototype device: A piezo-electric actuator (ii) is glued to the back of a 2.8 inches capacitive touchscreen (i) suspended on foam strips (iii); The touchscreen is connected to a microcontroller board (iv) that lays on a load-cell (v); A separate board hosts a driver for the piezo-electric actuator (vi). All the elements are fixed to a metal base simulating the internal panel of industrial appliances.

force is measured using a BND-611N load-cell (0-1 kg) placed at the bottom of the structure. The load-cell is driven by a 24 bit HX711 AD converter with a sampling rate of 80 Hz. Although techniques exist for the estimation of finger force during tapping actions [193], a more direct and accurate measure via a low-cost load-cell was preferred. Indeed, the use of a force sensor allows tracking also the release phase of pressing gestures before a finger loses contact with the touchscreen surface, which would not be reliable based only on capacitive or resistive sensing. Luckily, professional appliances usually have large empty volumes behind the user interface, allowing the interposition of a load-cell between the front and the rear panel of the user interface.

In this prototype vibrotactile feedback is generated by a Samsung Electro Mechanics (SEMCO) PHAH353832 piezoelectric actuator (dimensions L 35×W 3.8×H 3.2 mm, weight 2.7 g) controlled by a Texas Instrument DRV2667 piezo driver, connected to an Arduino Mega 2560 microcontroller board via the I2C communication bus. Compared to other haptic technologies (e.g., LRA, ERM and voice coil), piezo actuators can also render fast transients at different frequencies [9].

In the prototype tested the piezoelectric actuator is attached under the TFT screen in a central position, producing displacements along the Z axis. However, thanks to its compact package, the same actuator can be turned in different orientations in order to produce, e.g. lateral displacements. The piezo driver may operate in analog mode, by amplifying (up to $200\,\mathrm{V_{pp}}$) an audio-level signal at its analog input, or it can use its internal digital-controlled synthesis engine to generate simple sequences of sine waves, whose parameters (frequency, amplitude, attack and decay time, and duration) can be programmed through the I2C interface.

Since the touchscreen is suspended on two foam layers, the device is slightly compliant to normal pressing forces ($\leq 1\,\mathrm{mm}$).

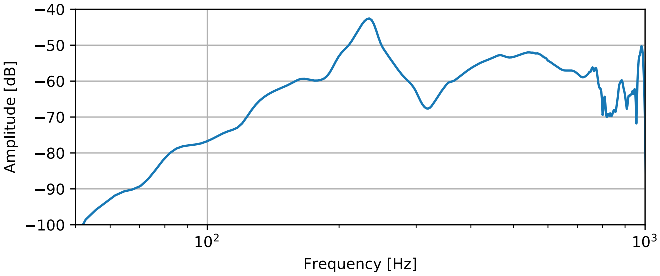


Figure 2.2: Frequency response of the device in the range 50-1000 Hz.

The system frequency response was determined by measuring the response to a 50-1000 Hz sweep signal with a Wilcoxon 736 accelerometer attached on top of the touchscreen (center position), and then deconvolving it [68]. As shown in Fig. 2.2, the device is mostly efficient around the resonance frequency of the piezo actuator (230 Hz), whereas it is substantially unable to reproduce frequencies below 100 Hz. Concerning the upper part of the tested range, distortion is present above 700 Hz, as, in fact, the reproduction of the sweep signal produced auditory artifacts in that range.

The device displays up to four virtual buttons labeled A, B, C, D (see Fig. 2.3), matching the number of main functions commonly found on professional appliances (2 to 6). Their shape and size (squares of 22 mm side) were set based on guidelines from the literature [137, 213].

Three different sets of vibrotactile stimuli were designed and associated with the virtual buttons, aimed at simulating different materials and effects. The first two sets were designed starting from the vibration stimuli used in the classification experiment 2 (Chap. 1), whereas the last set was designed based on the rendering capabilities of the device (Sec. 2.3).

2.2 Case study 1

Based on the reported positive results of tactile material classification (Chap. 1), a first implementation of virtual buttons tested the straightforward reproduction of the ecological vibration stimuli of experiment 2, that are the stimuli recorded with the three objects in contact with the table (short decay). Obviously, such stimuli were considered the most coherent feedback to finger tapping actions. Unfortunately, such straightforward attempt was not effective at all: the original stimuli shown in Fig. 2.4 (orange lines) gave rise to weak and distorted reproductions, as visible in Fig. 2.6 (orange lines). Indeed, the chosen actuator can efficiently reproduce only a few concurrent spectral components, whereas the reproduction of rich spectral and dynamic content is generally unsatisfactory.

In an attempt to overcome such issue, two new sets of stimuli were prepared: the former consisted in a filtered version of the original signals, made using a tenth-order

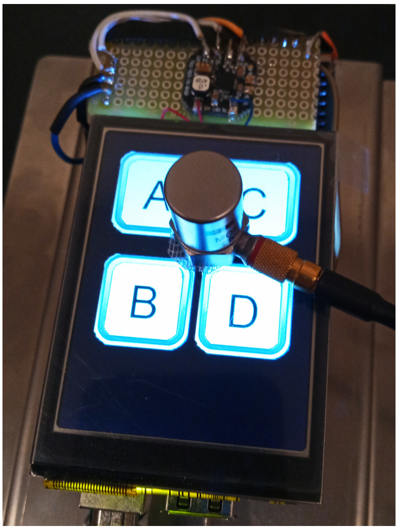


Figure 2.3: Visual appearance of the virtual buttons. For the characterization procedure, an accelerometer was placed in the middle of the touchscreen.

Butterworth filter with pass-band 100-600 Hz; the second set was synthesized by tuning the frequency and decay time of exponentially decaying sine oscillators to the most prominent components of the original signals in the same frequency band, that is two components at 115 and 470 Hz for wood, one component at 430 Hz for plastic, and two components at 230 and 550 Hz for metal. The RMS power of all stimuli was normalised within a 500 ms window, so as to make them uniform and maximize vibration amplitude while avoiding distortion. The signals from both sets are made available via an open-access repository.[1] Figures 2.4 and 2.5 show the spectra and the spectrograms of the obtained stimuli compared to those of the original recordings, while Fig. 2.6 reports the spectra as actually rendered by the device. Although the newly designed stimuli were in general better rendered than the original recordings, their reproduced characteristics are worth noticing: an artifact was introduced by the actuator at around 100 Hz in all signals; likewise it boosted energy in the low frequency ($<$ 100 Hz) concerning the synthesized plastic and especially metal stimuli, while both filtered and synthesized metal stimuli also were boosted around their fundamental frequency, being it close to the resonant band of the actuator; conversely, the first component of the original wood signal (115 Hz) was not reproduced by the device; finally, the spectra of reproduced wood and plastic were quite similar, as they have frequency components that are close to each other (430 Hz vs. 470 Hz).

2.2.1 User Evaluation

The two sets of stimuli underwent a separate subjective evaluation.

---

[1] `https://doi.org/10.5281/zenodo.3630367`

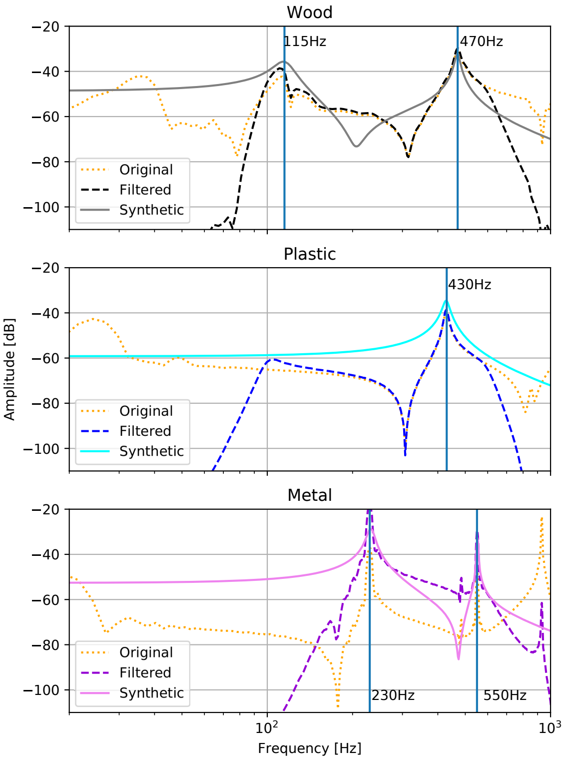


Figure 2.4: Spectra of the signals designed based on the original vibration stimuli from the classification experiment (dashed orange lines). For each material, two design techniques are shown: signals band-pass filtered in the 100-600 Hz range, and signals synthesizing a few relevant components of the original spectra (marked by blue vertical lines).

Three virtual buttons labeled A, B, and C were respectively linked to wood, plastic and metal stimuli, either filtered or synthesized, which were triggered by finger pressure exceeding 1 N. Given that the target use of the device is in generally noisy professional environments, and that the evaluation was performed in a silent room, an auditory distractor reproducing the noise of a crowded room (70 dB(A)) was continuously delivered during the assessment. This way, the collateral auditory signals produced by the piezoelectric actuator were masked as in real working conditions.

Fourteen subjects (6 male, 8 female) aged between 22 and 54 (M = 33.1; SD = 7.4) participated. Each participant performed two sessions, respectively evaluating three buttons using either filtered or synthesized stimuli. The task was to freely operate the buttons while answering an online questionnaire in Italian containing 7-point Likert scale evaluations and multiple choice questions:

1. Degree of difference among the three buttons, based on touch only. The evalu-

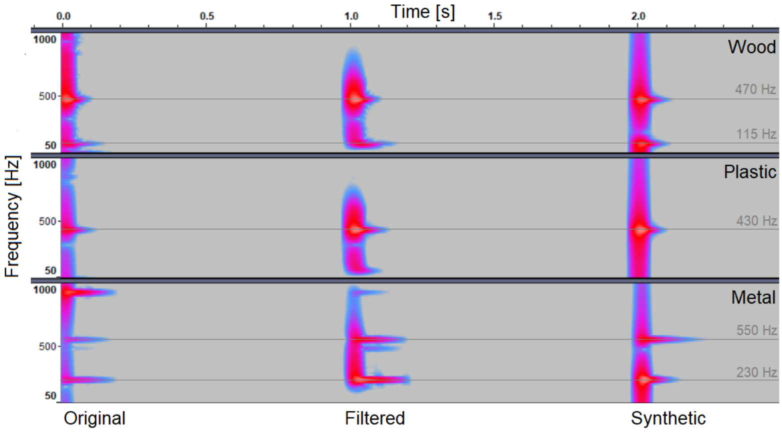


Figure 2.5: Spectrograms of the signals shown in Fig. 2.4. From left to right, each row reports original, band-pass filtered and synthetic stimuli for the three materials (i.e. wood, plastic and metal). The horizontal gray lines mark the relevant components in the original spectra from the filtered and synthetic stimuli.

ation scale ranged from ‘barely different’ to ‘very different’.

2. General tactile quality of all buttons. The evaluation scale ranged from ‘not appreciated’ to ‘much appreciated’.
3. Compliance of each button. Despite the fact that no displacement was rendered, compliance illusion could be elicited thanks to the vibrotactile response to finger pressing [123, 193]. The evaluation scale ranged from ‘weak’ to ‘strong’.
4. Material each button was made of, among five options (metal, plastic, wood, glass, and rubber). There was one question per material, each with possible multiple choice of buttons and an additional ‘none’ option (i.e., A, B, C, none).

For the sake of clarity, in what follows the buttons reproducing filtered stimuli are referred to as Plastic Filtered (PF), Wood Filtered (WF) and Metal Filtered (MF), while those reproducing synthesized stimuli are labeled as Plastic Synthesized (PS), Wood Synthesized (WS) and Metal Synthesized (MS).

2.2.2 Results

Figure 2.7 shows the perceived difference scores among the buttons. In addition, participants reported that buttons reproducing wood (WS, WF) and plastic (PS, PF) rendered similar stimuli, whereas buttons with metal feedback (MS, MF) differed from the others in both sets.

Concerning the appraisal of tactile feedback, the distributions reported in Fig. 2.8 show that the evaluations were more consistent for filtered rather than synthesized

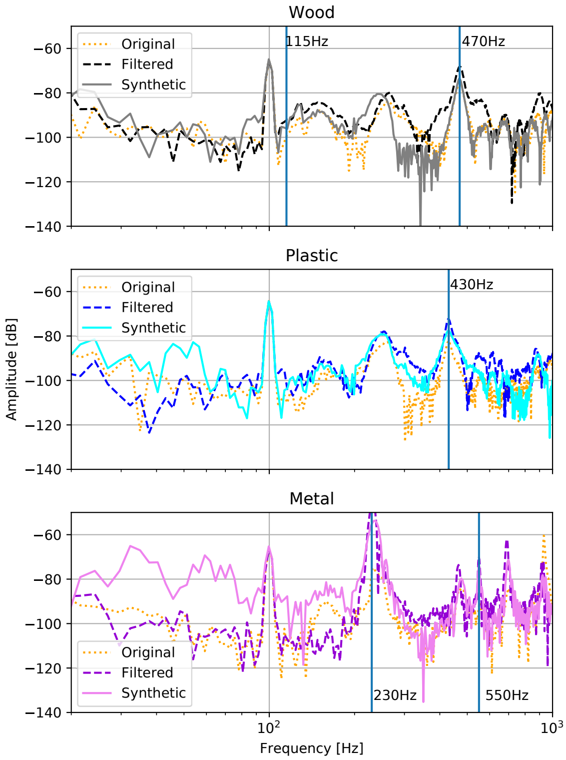


Figure 2.6: Spectra of the stimuli shown in Fig. 2.4 as actually reproduced by the device. The blue vertical lines represent the main frequency components of the original signals.

stimuli. However, nobody assigned the highest score to either filtered or synthesized stimuli.

Regarding the perceived compliance, Fig. 2.9 reports for both sets high scores for stimuli related to metal (MS, MF) and low scores for stimuli related to wood (WS, WF). In general, the perceived compliance seemed to depend more on the simulated material than the type of stimuli (filtered or synthesized).

Material attributions are reported for the two sets separately in Fig. 2.10 and 2.11, revealing high uncertainty in both cases. Notably, wood was the only material not attributed to any button by almost all participants: wood stimuli were mostly identified as plastic or glass, confirming the previous observations regarding the similar spectral content of the original wood and plastic signals. In general also material attribution seemed to be rather independent of the set type. Given the limited differences among the filtered and the synthetic stimuli in terms of spectral content and components decay, this suggests that participants generally confirmed the same material attributions in both sets.

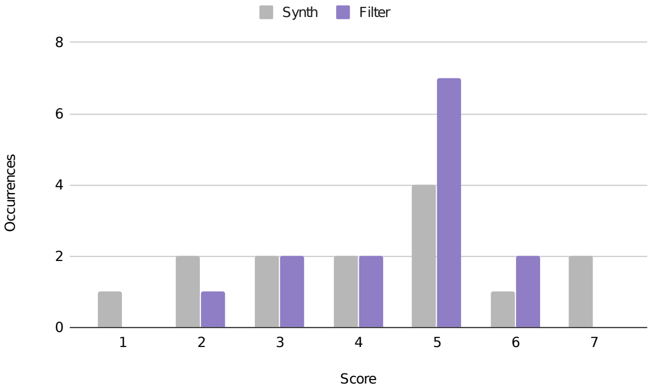


Figure 2.7: Score distributions of perceived difference among the buttons in case study 1 (1 = barely different, 7 = very different).

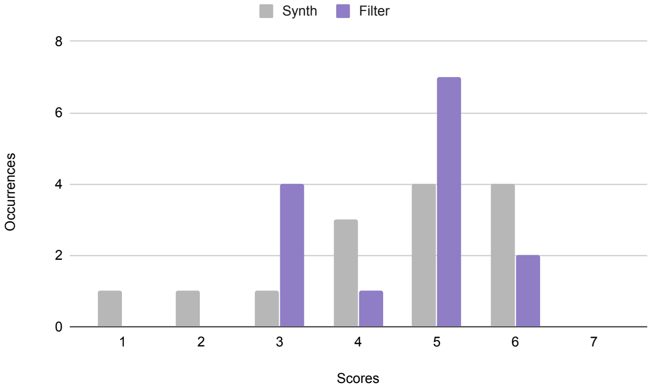


Figure 2.8: Score distributions of tactile feedback appraisal in case study 1 (1 = not appreciated, 7 = much appreciated).

2.3 Case study 2

In the light of the poor overall results obtained in case study 1 with filtered and synthesized stimuli based on the original vibration recordings, a further set of signals was designed from the ground up making direct use of the piezo driver. Its internal synthesis engine can generate temporal sequences of sine waves at frequencies multiple of a fundamental of the piezo (about 7.8 Hz), thus limiting the design space. Four virtual buttons labeled A, B, C, and D were designed, aimed at simulating different tactile materials and effects. Based on known illusory kinesthetic effects, such as compliance and indentation, elicited by vibrotactile feedback [123, 193], some mechanical features of real buttons were also simulated. The spectral and temporal content of

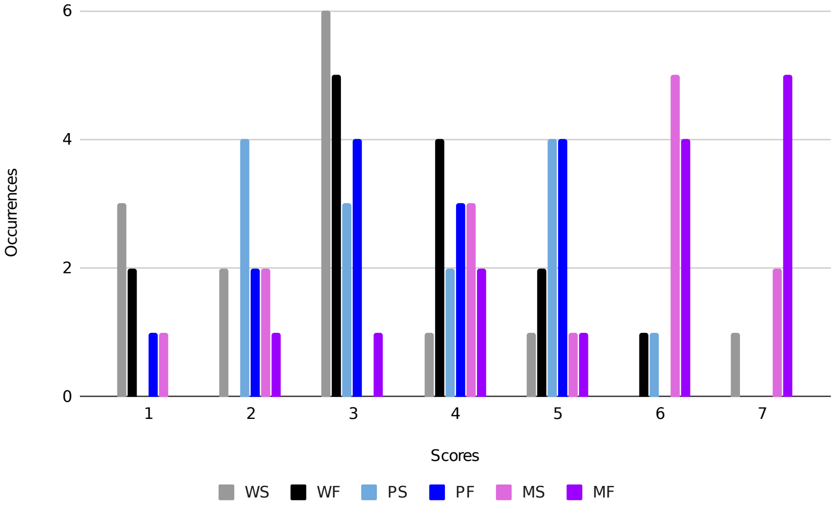


Figure 2.9: Score distributions of perceived compliance for each button in case study 1 (1 = weak, 7 = strong).

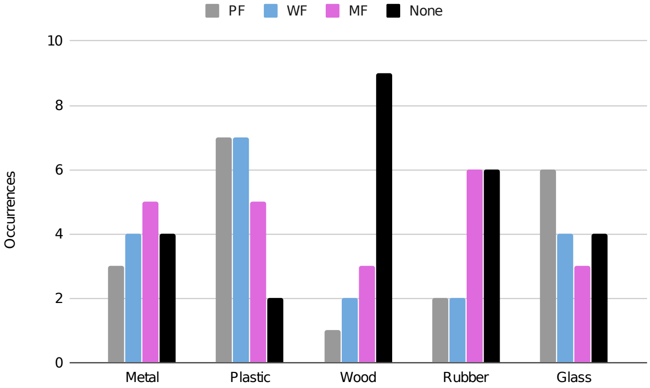


Figure 2.10: Attribution of materials with filtered stimuli in case study 1.

the stimuli, instead, was designed based on the haptic characteristics of the device, primarily its natural resonance frequencies. The main characteristics of the designed buttons are listed below:

- **Button A** simulates a silicon rubber key.[2]

  Onset: when the applied force exceeds 3 N, a sequence of two sine waves (5 cycles

---

[2] `https://en.wikipedia.org/wiki/Silicone_rubber_keypad`

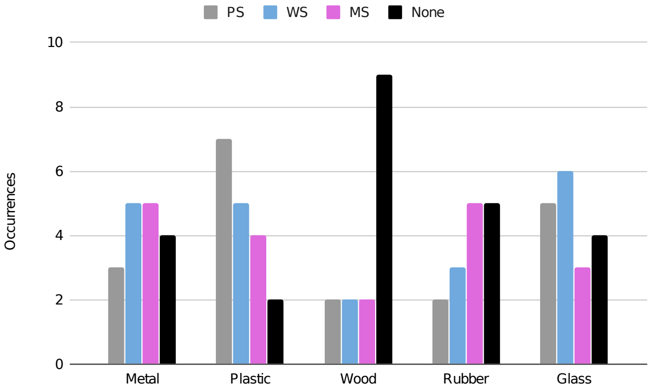


Figure 2.11: Attribution of materials with synthesized stimuli in case study 1.

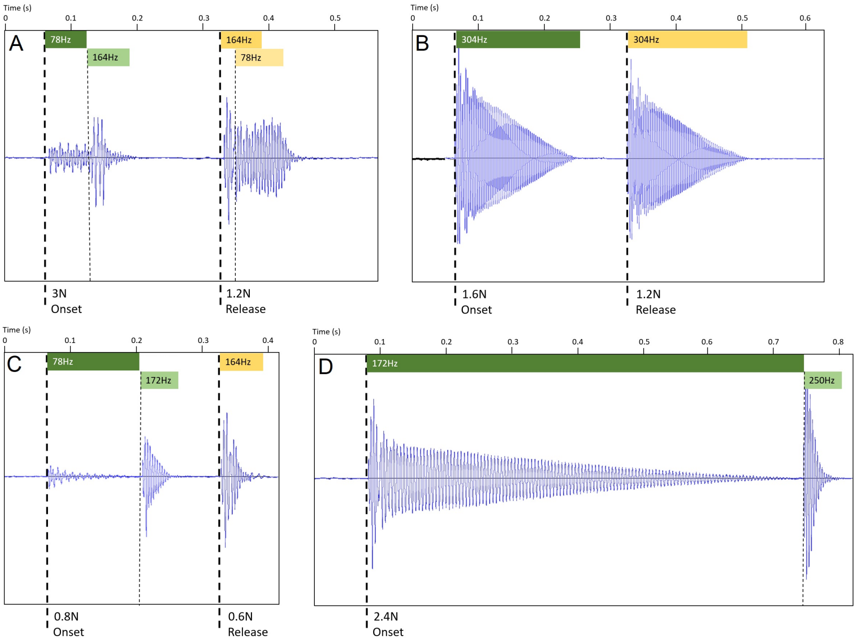


Figure 2.12: Vibration waveforms of the four virtual buttons of case study 2, as measured by an accelerometer (see Fig. 2.3). Different sequences of sine waves are produced at finger-press onset and release, whose frequencies are reported in green and yellow bars respectively. Onset/release triggering forces are shown at the bottom.

at 78 Hz and 3 cycles at 164 Hz) is synthesized producing a peak acceleration of $1.35\,\mathrm{m\,s^{-2}}$. The frequency of the first signal is below the pass-band of the device, resulting in a "rubbery" tactile effect just before a further transient that simulates a soft 'click'.

Release: the same two waves are played in reverse order when the force drops below 1.2 N, producing a $1.9\,\mathrm{m\,s^{-2}}$ peak acceleration.

Together, these sequences simulate the acceleration curves resulting from pressing a finger on soft materials [125].

- **Button B** simulates the behavior of a metal membrane switch.

  Onset: a strong transient consisting of a single cycle of a sine wave at 304 Hz with $3.8\,\mathrm{m\,s^{-2}}$ peak acceleration is triggered when the applied force exceeds 1.6 N, simulating the sudden deflection of a metal membrane.

  Release: the same feedback is generated when the force falls below 1.2 N, resulting in $3.4\,\mathrm{m\,s^{-2}}$ peak acceleration.

- **Button C** simulates a latching push button made of plastic, inspired by the switches found on old table lamps.

  Onset: when a 0.8 N force is exceeded, a 78 Hz sine wave is played for 150 ms, simulating the initial phase of button depression. Right after that, a stronger transient (a short 172 Hz sine wave) is produced with $1.8\,\mathrm{m\,s^{-2}}$ peak acceleration, simulating a 'click'.

  Release: when the applied force falls below 0.6 N, a short 164 Hz sine wave is generated to simulate the release 'click', resulting in $2.2\,\mathrm{m\,s^{-2}}$ peak acceleration.

- **Button D** simulates a more abstract metal resonance with long decay, especially suited to long-press actions.

  Onset: when a 2.4 N force is exceeded, a strong 172 Hz sine wave with long decay is produced to simulate a 'click', and if pressure is held for more than 700 ms a further short feedback (250 Hz sine wave) is generated. The peak acceleration produced is $3.4\,\mathrm{m\,s^{-2}}$.

  No feedback is provided on release.

Despite the fact that wood-related feedback scored best in the material classification experiments of Chapter 1, no button was designed to simulate wood. This mainly because the strong low frequency components typical of this material can not be correctly rendered by the device, and secondly because it was not reputed a common material for the buttons found in professional appliances. Metal and plastic were instead found more appropriate, however since they represent the two materials that were more often confused in the previous classification experiments, one button rendering metal (D) was strongly differentiated by implementing longer decaying resonances inspired to results achieved in the experiment 1 of Chapter 1.

As demonstrated in case study 1, the main spectral components of the vibratory signals used in the material classification experiments (see Fig. 1.5) cannot be accurately rendered on the device, given its limited bandwidth (see Fig. 2.2). The frequencies of the synthesized sine waves, as well as their decays and amplitudes, were therefore empirically chosen based on the pass-band of the device and informal

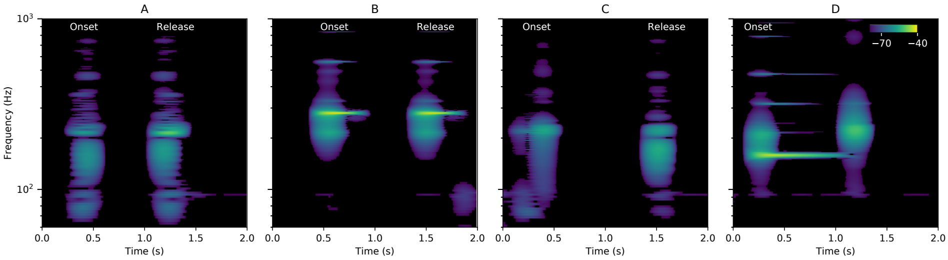


Figure 2.13: Spectrograms of the vibrotactile feedback associated to the four virtual buttons.

testing, while leaving the generation of higher frequency components to the inherent harmonic distortion taking place with strong signals (see buttons B and D in Fig. 2.13). As a result, the main spectral components of the designed stimuli are generally at lower frequency than those in the stimuli used for the material classification experiments reported in the previous chapter. However such pitch-shift is known to have no effect on the perception of a specific material, being it more associated to the varying size of an object [87]. Instead, materials were mainly defined by the designed decays (e.g., shorter for rubber and plastic), amplitudes (e.g., stronger for metal) and harmonic content.

The vibrotactile feedback produced by the buttons was measured by attaching a Wilcoxon 736 accelerometer on top of the touchscreen, between the virtual buttons (see Fig. 2.3). Figures 2.12 and 2.13 respectively show the waveforms and spectrograms of the feedback signals. The measured signals, as well as video footage of the four virtual buttons being operated are made available via an open-access repository.[3]

When vibrations were produced, the system emitted also some parasitic sound, however this was hardly perceivable in the (noisy) environment chosen for the device evaluation, and could therefore be ignored.

2.3.1 User evaluation

Sixteen subjects (9 male, 7 female) aged between 25 and 47 (M = 34.7; SD = 8.1) evaluated the virtual buttons. The assessment took place in a realistic situation (i.e., a crowded open-space office hosting about 40 people), thus no additional auditory distractor was required. The task was to freely operate the buttons and answer an online questionnaire containing the same 7-point Likert scale evaluations and multiple choice questions proposed in the case study 1 (see Sec. 2.2.1).

2.3.2 Results

The following evaluation is qualitative, in view of a deeper quantitative analysis of the results.

---

[3] `https://doi.org/10.5281/zenodo.3630367`

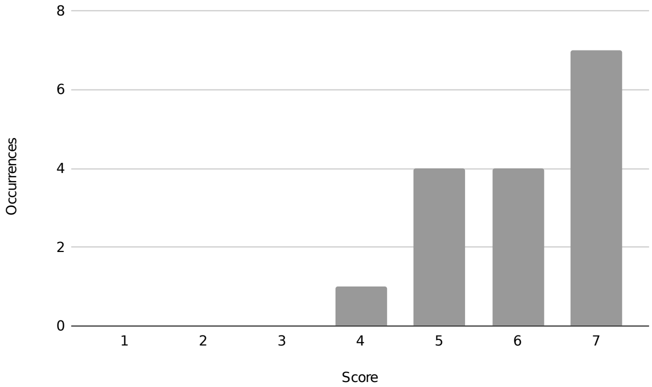


Figure 2.14: Score distributions of the perceived difference among the four buttons in study case 2 (1 = barely different, 7 = very different).

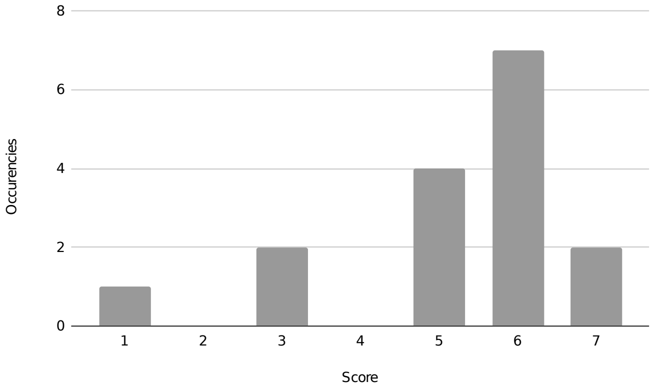


Figure 2.15: Score distributions of tactile feedback appraisal in study case 2 (1 = not appreciated, 7 = much appreciated).

As highlighted in Fig. 2.14, participants generally rated the buttons as clearly distinguishable from each other, furthermore they expressed general appreciation for the quality of tactile feedback, as shown by the score distributions in Fig. 2.15.

Evaluation ratings of the perceived compliance are reported in Fig. 2.16: The effect was most pronounced for button D followed by button B, while ratings related to buttons A and C are distributed in the lower and the mid-upper part of the scale.

With regard to the association of five materials (metal, plastic, wood, glass and rubber) with the virtual buttons, their choice distribution is reported in Fig. 2.17. Attributions mostly agreed with the intended feedback design (see Sec. 2.3): button A was mainly associated with rubber, button C clearly with plastic, and buttons B and

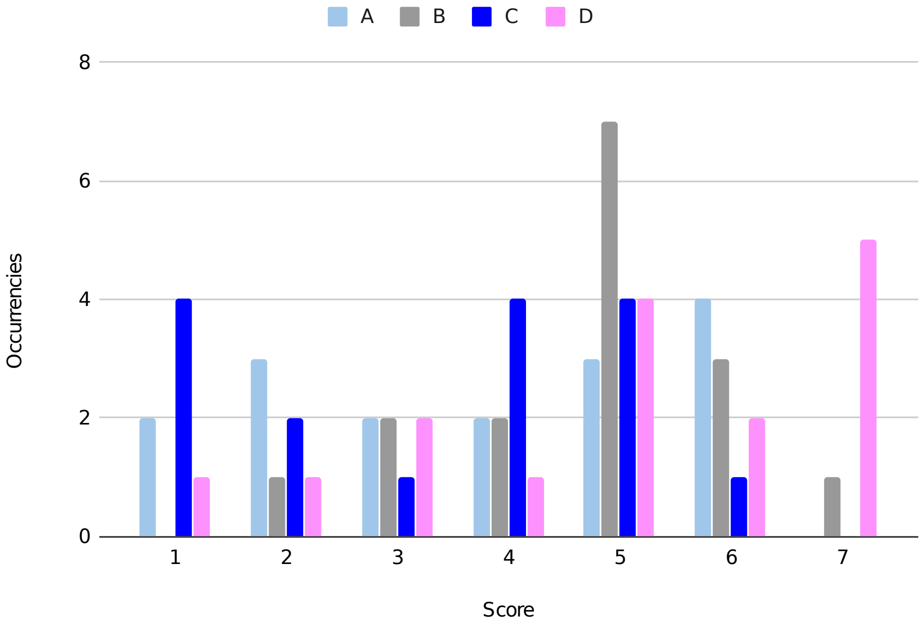


Figure 2.16: Score distributions of the perceived compliance for each button in study case 2 (1 = weak, 7 = strong).

D even more distinctly with metal. Concerning wood and glass – both not simulated – the former was not associated with any button by half of the participants, while a small group associated the latter almost uniformly with all the given possibilities, including the 'none' option.

2.4 Discussion

The two reported case studies revealed the challenges posed by the tactile rendering of well distinguishable virtual buttons on touchscreens.

The most relevant outcome of the two studies concerns the discrimination of buttons: although differences were generally perceived in both assessments, score distributions show that a careful design of tactile signals to exploit device's peculiarities (e.g., resonances and damped frequencies, controlled distortion), in conjunction with the optimization of force thresholds at which feedback is provided, can be even more effective than the use of real vibration recordings, even if adapted to the device's pass-band. Indeed, even if both studies rendered feedback related to ecological materials, only in case study 2 buttons are rated as "very different" by most participants, suggesting that the reproduction of signals with realistic frequency components and decays may be not sufficient to enable a precise discrimination. However, discrimi-

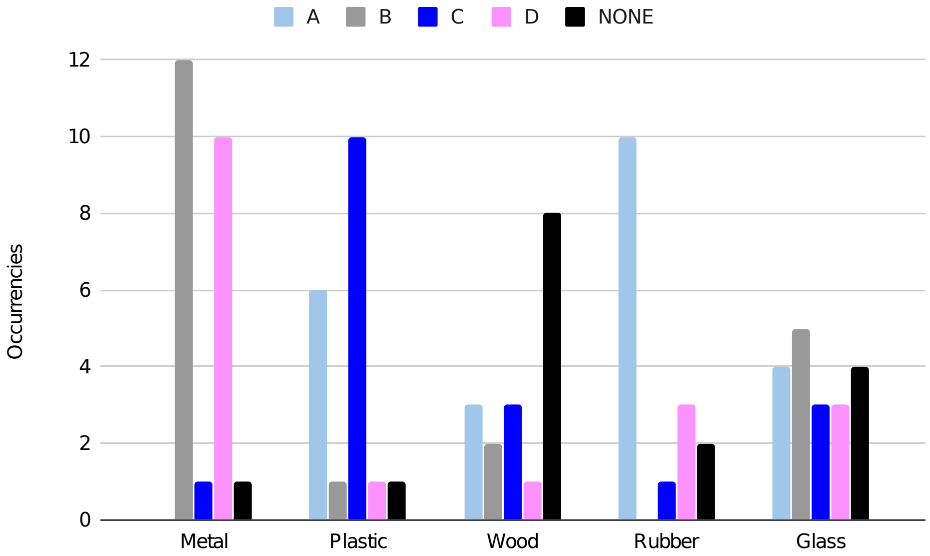


Figure 2.17: Attribution of materials in study case 2.

nation in case study 2 may also have improved by rendering illusory cues related to button mechanics (e.g., switches, material compliance).

The tactile feedback generated by our device was generally appreciated in both studies, however the virtual buttons implementation of case study 2 received higher scores. After comparing the synthesized signals with the same signals provided via the analog input of the piezo driver, it is possible to claim that the advantage of the internal synthesizer is all in its reduced design space which imposes to concatenate sine waves at frequencies that maximize the actuator's efficiency.

A major difference between the two case studies concerns the attribution of materials to the virtual buttons. In case study 1, stimuli originated from metal vibrations (MF, MS) were almost evenly assigned among the available materials, whereas in case study 2 the buttons inspired to metal properties (B, D) were clearly identified independently from the decay. Therefore, the design of effective stimuli simulating metal seems to be linked with the inharmonicity of their spectra: indeed, although buttons B and D in case study 2 render spectral components that differ from those in the original recording of metal vibration, they efficiently generate an inharmonic content typical of metal [86]. The buttons designed to render plastic materials – that is, PF and PS in case study 1 and button C in case study 2 – were correctly assigned by 50% and 62% of the participants, respectively in case study 1 and 2. In both studies, plastic was more confused with glass than other materials. In general, wood was the material more associated with the 'none' option, which is indeed correct for case study 2. Surprisingly, in case study 1 the buttons rendering wood-related stimuli (WF, WS) were mostly associated to every other materials except wood. Actually, the impaired reproduction of frequencies below 100 Hz clearly explains these associations.

As mentioned above, vibrotactile feedback can be used to simulate to some extent button mechanics, thus increasing differences among virtual buttons or easing material identification, as done in case study 2. However, the illusion of kinesthetic feedback can be effectively elicited only if tracking the applied force, and by careful design and control of the delay and duration of the stimuli [149, 178].

This design partially confirms the findings of Sadia *et al.* [193], who investigated forces and accelerations involved in various button press actions (e.g., latch, push and toggle) and emulated such mechanics by reproducing tactile stimuli by means of piezo actuators. In particular, for their latch button a waveform pattern was generated whose temporal evolution is close to button C in case study 2. On the other hand, they triggered stimuli when the applied force was much greater than the force used in this study. To improve the latch button here proposed it would be possible to trigger the two parts of the stimuli onset based on multiple subsequent force triggers (e.g., at 3 and 10 N). Moreover, based on the dataset provided by Alexander *et al.* [5], who characterized the physical properties of more than 1500 push buttons, it would be possible to design further button mechanics.

2.5 Lesson learnt

The aim of this chapter was the validation of the results that emerged in passive conditions in Chapter 1: unimodal tactile feedback can enable material classification, allowing low mismatch rates. Therefore, multiple virtual buttons were designed for an *ad hoc* prototype device, consisting of a touchscreen interface offering rich tactile feedback and force sensing in the range of soft-touch.

Two user panels evaluated several aspects of the tactile feedback associated to various sets of virtual buttons through questionnaires: although the vibrotactile feedback originated from real bouncing events were relatively clearly discriminated (case study 1), participants could more successfully discriminate the buttons designed from the ground up, exploiting the prototype device's response and characteristics (case study 2). The characterization and validation of the prototype highlighted that the stimuli design must carefully match the capabilities of the device, furthermore trying to take advantage of its natural resonances. To design high-quality vibrotactile stimuli, the modifications of the human finger impedance, consequent to the fingertip compression while pressing on the device, should be considered too. To this end, a tool used to simulate the impedance of the human finger while pressing will be described in Chapter 7.

In the light of the experimental results, it can be speculated that the repeated everyday use of a known set of virtual buttons, such as those experimented here, could be proficiently used also in work environments affected by auditory noise and visual distractors, such as professional kitchens or laundries.

The design strategy adopted to trigger the virtual buttons (i.e. the simultaneous tracking of finger position and force) shows potentials also in the improvement of the interaction reliability and robustness concerning professional appliances; indeed, every key feature triggered by touchscreen selections can be programmed to respond only to defined screen positions and to bounded force profiles, therefore minimizing unintentional selections.

As a final note, although the prototype was assessed in the form of a flat rectangular touchscreen interface, the same technology inspired the concept design of a novel user interface that will be described in the final section of Chapter 6.

3

Direction and force influence on vibration perception

The design phase of the prototype described in the previous chapter involved many practical issues regarding the mechanical design of the user interface. In particular, in order to maximize the actuator efficiency, the choice of its position and vibration direction was accomplished by comparing the rendering of the same feedback in multiple configurations. This informal trial-and-error approach revealed that even simple test stimuli were differently perceived by users, thus avoiding a straightforward decision on the device design.

In collaboration with the Institute for Computer Music and Sound Technologies (ICST) of ZHDK (Zurich), this chapter presents an experiment accomplished to investigate whether the direction of vibration (i.e. lateral or normal) influences its perception sensitivity on touch surfaces. Moreover, since the target application of this research involves active touch, the experiment was aimed also at defining a model to relate the normal force exerted by the fingertip on the surface to the perception sensitivity.

The perception of normal and tangential vibration at the finger has been studied by several authors, however in contexts and with modalities slightly different from our case study. For instance, during the 60s, a study considering the whole hand in contact with a flat vibrating surface (in range 3-300 Hz) reported similar thresholds for both directions [163]

More recently, Biggs and Srinivasan [22] investigated the perceptual equivalence of quasi-static forces presented tangentially and normally to the fingertip. In such study, the index fingertip of subjects was glued to a flat-ended, cylindrical 1 mm diameter probe tip mounted on a 3-axis positioning robot able to move in both directions. The tangential motion was found to require higher forces than normal motion for producing equivalent perception. However, lateral forces can generally be generated with much smaller displacements, as the finger presses normally. A study, conducted in similar experimental conditions, revealed that humans are also able to assess the magnitude of the applied forces concerning both directions at low frequencies [177].

Ullrich and Cruz [224] compared transient pulses of normal and tangential accelerations at the fingertip for various pressure levels (0.5, 2, 5 N). Their study showed that the perception of tangential and normal motions depends mainly on the respective acceleration magnitude: the tangential motion was perceived as slightly weaker than the normal motion for low accelerations, whereas it was perceived up to 40% stronger for high accelerations. Although the acceleration perception trend-lines were similar for the considered pressure levels, this effect was found not significant for the lowest value (0.5 N).

Given the adopted methodologies and the tested frequency range tested, all the results reported above are mostly related to accelerations that stimulate FA-I, SA-I and SA-II afferents (i.e. the mechanoreceptors responsible for low-frequency skin motion and force grip control [117]). Indeed, these results can not be directly transposed to high-frequency skin motion (40-800 Hz) detected by FA-II afferents (Pacinian corpuscles), i.e. to the skin's dynamic response to vibration [145].

Although several experiments addressed the perception of frequencies in the 100-700 Hz range, they usually limited the investigation to vibrations in the normal direction. To this end, Verrillo [230] found the lowest amplitude perception thresholds in the region around 250 Hz. On the other hand, though, the perception of these high frequencies vibration can vary among the human population, due to many physiological factors. For example, a cadaver study reported that the Pacinian corpuscles in the human hand are predominantly distributed in the fingers (44-60%) and their number can vary largely (192-424, M=300) [209]. Other studies investigated the effect of aging, finding that such frequencies are generally worse perceived by older people, due to a lower density of mechanoreceptors [3], a reduced functionality of Pacinian corpuscles [6] and to the changes in the skin mechanical properties [100]. Some effects related to gender have been identified, although not statistically significant, with lower perception thresholds reached by women subjects at 250 Hz [229]. The temperature factor has been studied as well, resulting in higher sensitivity thresholds concerning subjects with colder hands [100, 1].

To understand and model the propagation of the high-frequency vibrations in the fingers tissues, Wu *et al.* [242] developed a Finite Element (FE) model of the fingertip. The simulation of such model showed two resonant frequencies at 125 and 250 Hz, not dependent on the motion direction (normal or tangential); in particular, in accordance with their model, normal vibrations should give rise to a horizontal strain of the finger-pulp tissues, but only in the superficial skin layer. Furthermore, the analysis indicates that the resonance magnitude at 250 Hz is independent from the finger's pre-compression that should, therefore, result in perception thresholds, being independent from the finger press force. However, experimental results partially refuted this model: indeed, the measurements performed by Wiertlewski and Hayward [239] on human subjects did not find any evidence of the first resonance predicted by the FE model (125 Hz), whereas other studies found that vibrotactile thresholds proportionally decrease with pressing force in active touch [176, 169]. Finally, Hwang *et al.* [114] designed an experiment aimed at identifying the finger sensitivity thresholds concerning all the 3 motion directions (normal, lateral and fore-and-aft) for two stimuli at 150 Hz and 280 Hz, while applying a constant force of 0.8 N. In general, their study reported lower thresholds for the normal direction; moreover, as opposed to previous literature [230], they found the thresholds at 280 Hz significantly higher than those at 150 Hz. Concerning the tangential directions (i.e lateral and fore-and-aft), Hwang *et al.* [114] found comparable vibration perception thresholds; this result was confirmed also by [239], which measured the fingertip impedance, finding similar stiffness and damping coefficients concerning both tangential directions, thus suggesting an equal perception of vibrations. In the light of those findings, concerning the tangential direction, the following experiment will test only the lateral direction, in order to reduce the number of factor combinations.

As opposed to the findings of Hwang *et al.* [114], the informal tests, accomplished on the prototype interface of Chapter 2 and on further test devices, revealed lower sensitivity thresholds for the lateral direction, specially when light pressing force was applied to the vibrating surface. In order to investigate such effect, a new experiment was designed to compare the perception of normal vs lateral sustained stimuli at 250 Hz in active touch conditions (i.e. finger pressing at various forces).

Under the above conditions, the experiment tested the following two hypotheses:

H1: Vibrotactile sensitivity to lateral vibration is higher than sensitivity to normal vibration.

H2: Sensitivity proportionally improves to the applied pressing force.

3.1 Apparatus

For the purpose of the experiment, a device was designed able to efficiently and independently reproduce normal and lateral vibrations at 250 Hz, while minimizing cross-talk between the two motion directions. This section describes the implementation, the characterization and the validation of the device.

3.1.1 Hardware

The experimental device (or testbed), depicted in Fig. 3.1 and 3.2, is built around a 26×26×36 mm 3D-printed plastic (PLA) cuboid embedding two small actuators which produce vibrations respectively along its normal and transversal axes (see Fig. 3.3).

Figure 3.3 shows a close-up schematic of the actuated element: two Lofelt L5 voice-coil actuators [20] are embedded in the PLA cuboid, arranged perpendicularly to each other, and are driven by a 2-channel Class-T audio amplifier (Dayton Audio DTA-2); a 30×30 mm Plexiglass panel is glued at the top of the cuboid, offering a smooth surface for finger contact.

To maximize the actuators' efficiency, the actuated element is mechanically decoupled from its supporting structure by suspension: this allowed to greatly reduce the effective mass moved by the actuators, and enabled free motion even when external normal forces are applied via finger pressing. Furthermore, the suspension system allows to minimize the cross-talk between the normal and transversal vibration directions. The supporting structure, visible in Fig. 3.1, consists of a bottom wooden panel hosting two rubber shock absorbers, which hold the actuated element by means of nylon wires traversing it via four through-holes (see Fig. 3.3). After testing several materials, such as steel and copper, nylon wires were selected as they offer a good trade-off between rigidity and dampening, resulting in low dissipation particularly for

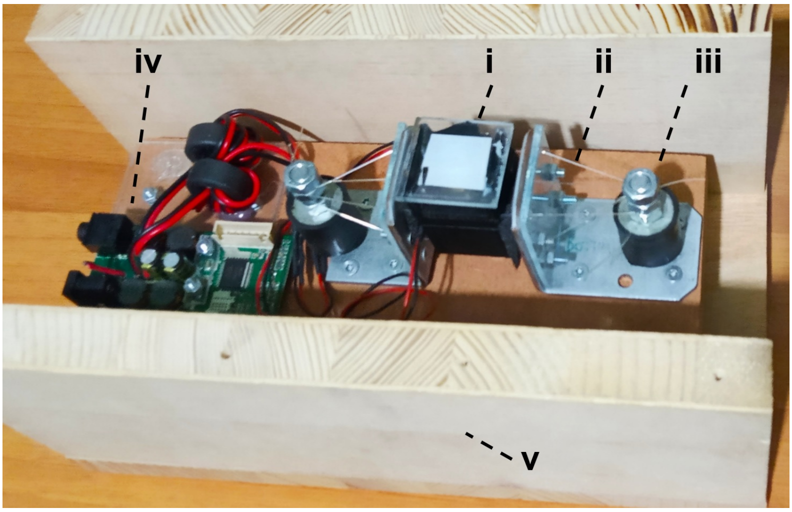


Figure 3.1: Device inside: an actuated element (i), hosting two small actuators, is suspended by means of four nylon wires (ii). These are attached to two rubber shock absorbers (iii) fixed to a bottom wooden panel (iv), which lays on a load-cell sensor.

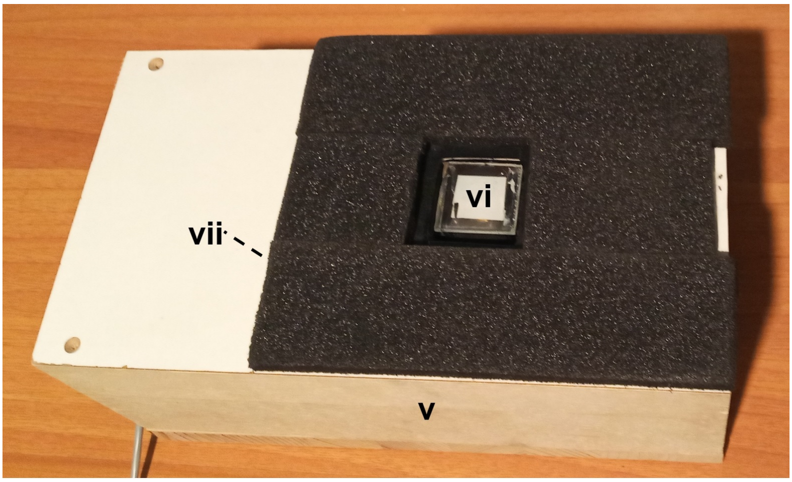


Figure 3.2: Device outside: a wooden box (v) encloses the device exposing to the user only the contact point (vi), surrounded by a black foam layer (vii).

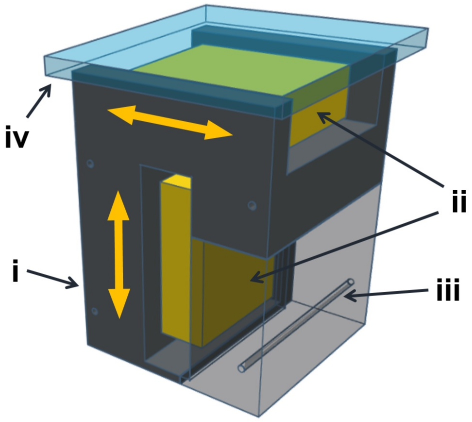


Figure 3.3: Schematic of the actuated element: a PLA cuboid (i) hosts two Lofelt L5 actuators (ii), which are arranged perpendicularly to each other. The cuboid has four through-holes (iii) for suspension. The touch surface (iv) consists of a 30×30 mm thin Plexiglass top panel.

vibrations around 250 Hz, which is the frequency used in the experiment. The total mass of the actuated element is 25 gr, whereas the elastic constant of the suspended element in the normal direction is 2.6 N/mm.

The bottom wooden panel lays on a CZL635 load-cell sensor, used to monitor the applied finger-pressing force (Fig. 3.4). The analog force signal is processed by a INA125P amplifier having the input gain set to read force values in the 0-20 N range. The force signal is acquired with a resolution of 0.02 N through the 10-bit ADC converter of an Arduino Mega 2560 microcontroller. Although the setting that we chose for the Arduino probably resulted in a non uniform sampling of the input data, the finger-pressing force is expected to change far slower than the 9.6 KHz sampling rate.

Finally, a wooden enclosure sized 250×130×120 mm protects the whole structure; its 2 mm plywood top-cover is overlaid by a 6 mm foam layer that surrounds the finger contact area (see Fig. 3.9).

3.1.2 Characterization and validation

The characterization of the device was performed by measuring vibration acceleration while human subjects exerted various forces on top of the actuated element. Such procedure was necessary to determine how vibrations are affected by pressing fingers: indeed, the biomechanics of the human finger allow to increase the applied pressing force while keeping the added mass constant and low.

Vibration measurements were acquired via a triaxial PCB 356A17 accelerometer glued to the top surface of the actuated cuboid. Ten subjects (9 male, 1 female) were asked to reach and hold a given target force (0.5, 1, 2 and 4.9 N) by pressing

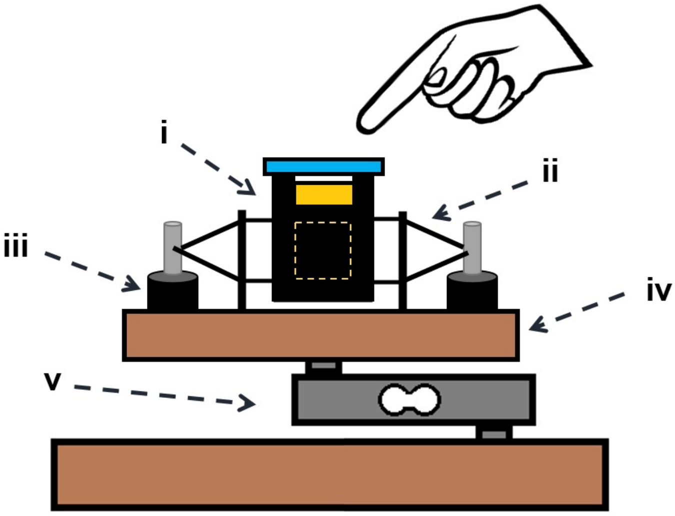


Figure 3.4: Side view of the device: i) actuated element, ii) nylon wires, iii) shock absorbers, iv) wooden panel, v) load-cell.

the index finger of their dominant hand on top of the accelerometer, while vibration was provided. Visual feedback displayed on a computer screen guided the subjects to accomplish the assigned task. For each force level, the actuators played back a sequence of stimuli, one direction at a time: a sequence consisted in repeated 250 Hz sinusoidal stimuli lasting 3 s, whose amplitude decreased by 6 dB at each repetition.

Influence of applied force on vibration amplitude

The resulting vibration recorded by the accelerometer was analysed calculating the median RMS value of eight subsequent time windows lasting 0.2 s, so as to cancel out possible noise due to small finger movements during the acquisition. Then, a set of linear models was fitted on the obtained RMS acceleration data to correlate forces and vibration accelerations for each tested vibration amplitude. Figure 3.5 reports the individual measurements (solid lines) and the fitted linear models for each amplitude and direction (dashed lines).

Although the models show no dependency between the applied forces and resulting accelerations in any direction (i.e. all the dashed lines have slope = 0), the variance of individual acceleration measurements actually changes with the direction: while the variance of horizontal vibration is constant with the applied force, the variance related to vertical vibration increases proportionally with the force; also, both variances increase for the lowest amplitude measured. Figure 3.6 reports the repeated measures for subjects 1 and 2 concerning each vibration direction at the highest amplitude: the plots reveal high variance in the measurements, which can be explained with small variations in finger positioning, as no constraints were imposed for the finger contact angle, rotation, and contact area.

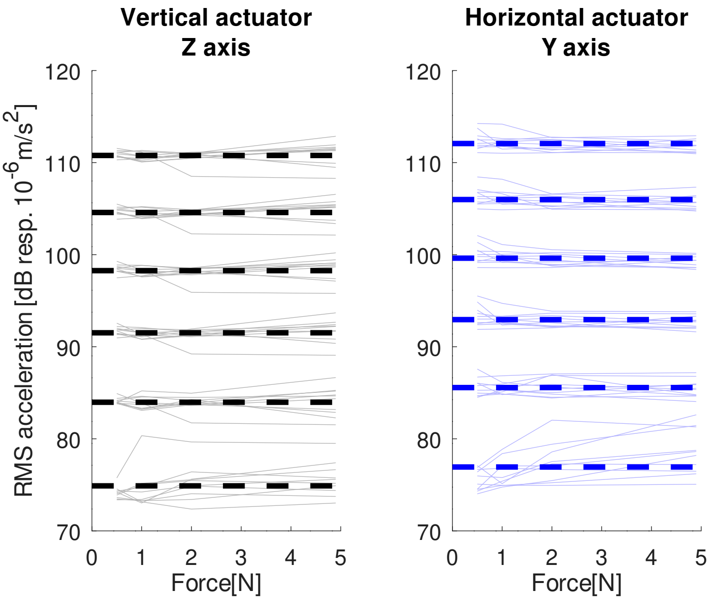


Figure 3.5: Acceleration measurements for all subjects (solid lines) and models (dashed lines) for the main direction of motion: Z axis for the vertical actuator (left plot) and Y axis for the horizontal actuator (right plot).

Vibration amplitude accuracy

Figure 3.7 reports the nominal and collateral vibration accelerations recorded as functions of amplitude variations generated by the vertical and the horizontal actuators, respectively. Nominal and collateral vibration accelerations decrease linearly in amplitude for about 18 dB; for lower amplitudes, the collateral components become slightly more relevant.

Vibration frequency response

The vibration frequency response of the system was obtained through the recording of sine sweeps (10-600 Hz, 15 s), separately reproduced by each actuator [68].

Figures 3.8a and 3.8b respectively report the average transfer function for the vertical and horizontal actuator. In general, the responses along the two nominal directions show a peak around the resonance of the actuator (64 Hz). In the 200-400 Hz range, thanks to the suspension system, the applied force plays a limited role on the nominal vibration directions, resulting in largely overlapping spectra. For lower frequencies, light and medium pressing forces (0.5, 1 and 2 N) do not affect the spectra, while the highest force level (4.9 N) strongly affects the actuators' motion. The ratio between nominal and collateral vibrations increases with frequency, resulting in differences respectively greater than 20 dB and 25 dB for the vertical and horizontal actuator above 200 Hz.

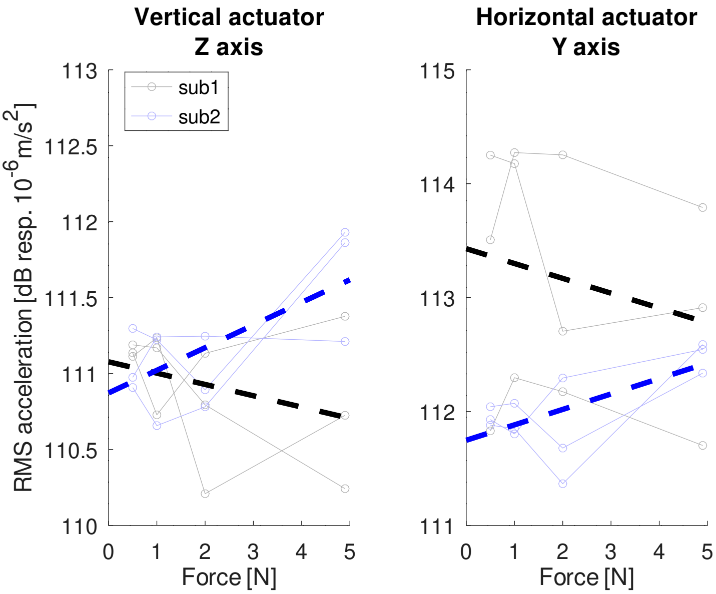


Figure 3.6: Repeated measurements for subjects 1 and 2 concerning the highest amplitude of each motion direction: Z axis for the vertical actuator (left plot) and Y axis for the horizontal actuator (right plot).

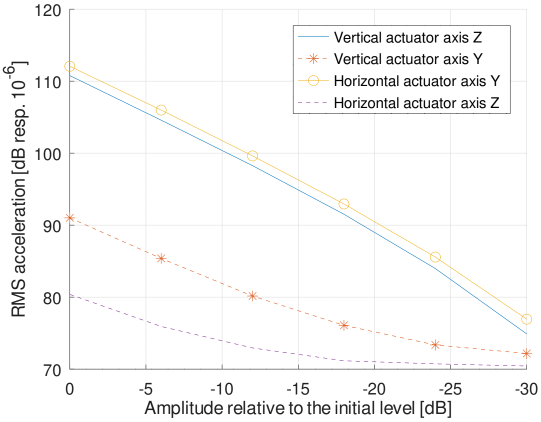
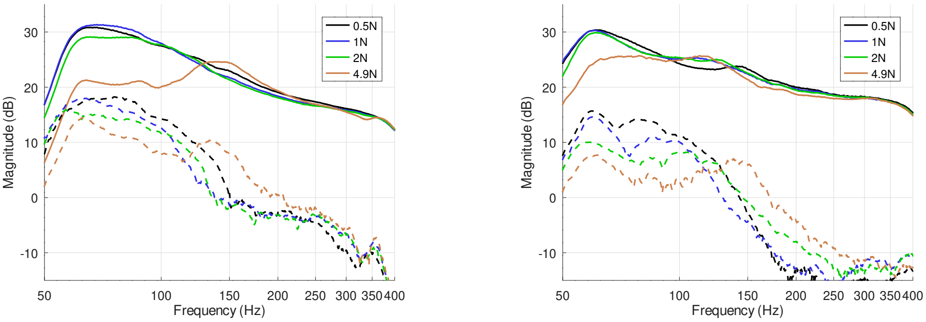


Figure 3.7: Average amplitude variations for vertical and horizontal motion directions of 250 Hz sine: nominal and collateral vibrations are respectively represented by solid and dashed lines.

Characterization issues

As shown in this section, the characterization of haptic devices is not trivial. Indeed, besides measuring tools and laboratories, it often requires a validation procedure



(a) Vertical direction (Z axis).

(b) Horizontal direction (Y axis).

Figure 3.8: Frequency response of vibration reproduction in the vertical and horizontal direction. For each force level, nominal and collateral vibrations are respectively represented by solid and dashed lines

involving human subjects that inevitably introduces several issues regarding the accuracy and the repeatability of measurements. To this end, a tool for the simulation of quasi-static finger pressing called *Bogus Finger* (see Chapter 7) was developed. Therefore, all the data collected from human subjects during this characterization procedure are also used in Chapter 7, in comparison with the Bogus Finger, while pressing on the same shared haptic device.

3.2 Experiment

The main goal of the experiment was the comparison of finger sensitivity to normal and lateral (ulnar-radial axis) vibration, for pressing forces compatible with everyday interaction [5]. Psychometric functions – expressing the proportion of correct responses as a function of vibration acceleration – were estimated for the two motion directions and two force levels.

3.2.1 Design

In the experiment, measurements were carried out using vibrotactile stimuli at predefined amplitudes and a YES/NO procedure similar to what was done in [114]. This method was chosen over a staircase algorithm for better control over several aspects: firstly, device characterization is more accurate for a fixed number of vibration amplitude levels; secondly, modern analysis methods allow parametric estimation of response distributions, and thus deliver more information than a single threshold estimate; finally, this method leads to sessions of known duration, preventing fatigue that prolonged staircase runs may cause in an active finger-pressing task.

Seven amplitude levels were measured for two crossed factors: motion direction (horizontal, vertical) and applied force (f1 = 0.5 N, f2 = 4.9 N). The proportion of correct responses was based on a block of twelve YES/NO trials for each factor combination. In each block, the vibrotactile stimulus was present in 50% of the trials and absent in the rest. This procedure was chosen in order to penalize biased decision criteria, like in the two-alternative forced choice design, while keeping the task simple [167]. Correct responses included both correct detection (hits) and correct rejections, while false rejections (misses) and false detection (false alarms) were recorded as incorrect responses. Thus, accounting for participants' individual decision biases, this design leads to a guessing rate of 0.50.

3.2.2 Setup

Inside a quiet room, the device was positioned over a table with an armrest in front of it and a computer monitor behind it (see Fig. 3.9). A laptop computer running a Python script controlled the experimental procedure. Vibratory stimuli were generated as audio signals via a RME Babyface PRO interface. The Arduino microcontroller, connected to a USB port, provided pressing force data measured by the load-cell. Also connected to the Arduino microcontroller, a control pad equipped with two buttons corresponding to YES and NO answers was used by the participants to report whether they felt vibration. Participants wore Uvex k-series ear muffs (32 dB noise attenuation) during the experiment. Finally, an IR thermometer was used to take the participants' finger temperature before each session.

Figure 3.9: Experiment setup: a subject pressing on top of the actuated element while the coloured bar displays the target force.

3.2.3 Stimuli

The stimuli were sinusoidal waveform with a frequency of 250 Hz and a duration of 1 s, a period sufficiently long for stable perception considering the temporal summation of the PC channel [228].

Seven amplitudes crossed with two motion directions (vertical and horizontal) and two force levels (0.5 and 4.9 N) resulted in 28 factor combinations, each of which was presented 12 times in a block. Within each block, the vibration stimulus was present in six trials. In total, the experiment contained 28 blocks×12 repetitions=336 trials. The presentation order was randomized both within and between blocks.

3.2.4 Participants

Twenty-two participants (15 males, 7 females) aged between 22 and 50 (M=31, SD=7.8) were recruited among students at the University of Udine and employees of Electrolux Professional SpA. They participated on a voluntary basis and did not receive any payment.

3.2.5 Procedure

At the beginning of each session, the experimenter briefed the participants and collected personal data such as age, gender, dominant hand and finger temperature. During the briefing, participants could acquaintance with the experimental protocol. As training, eight stimuli were presented with amplitudes 3 dB above the highest amplitudes used in the experiment.

Although the device was completely silent concerning the amplitude range tested, participants wore ear muffs to avoid any external noise. Participants were asked to press the index finger of their dominant hand on top of the contact area, reaching and holding a given target force displayed on the LCD screen. Once the target force was held for 1 s, another 1 s observation interval would follow, after which participants had to report by pressing the YES/NO buttons whether they had felt a vibration. A new trial started automatically after they lifted their finger from the touch surface.

In case participants could not keep a pressing force within ±20% of the target value across the entire duration of the observation interval, the trial was repeated. Participants were allowed to rest between trial blocks, and each session lasted between 40 and 60 minutes.

For each trial, the following data were collected: trial ID, factor levels, response, response time, mean force applied in evaluation window, and mean force applied in observation window. After each experiment session, all the materials were cleaned and sanitized following the Italian protocols against COVID-19.

3.2.6 Pilot test

Given the high variability of touch sensitivity [100], a pilot test was performed with eight participants (6 male, 2 female) to estimate the ranges of vibration amplitude to be used in the experiment. The technical setup was otherwise similar to the main experiment, but instead of a designed procedure, sequences of vibrotactile stimuli were produced with amplitudes decreasing in 3 dB steps from 120 dB RMS (re 10^{-6} m/s^2). One such sequence was played for each factor combination (i.e. motion direction and applied force). Correct pressing force levels were signaled by visual feedback. Participants reported when they were not able to feel the stimulus anymore. These rough thresholds, expected to overshoot actual detection thresholds somewhat, advised the choice of amplitudes for the main experiment. Figure 3.10 reports the pilot results and the choice of amplitude ranges for each factor combination.

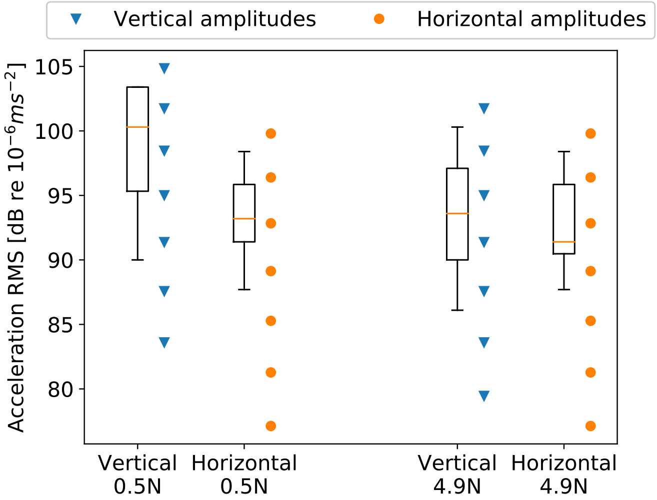


Figure 3.10: Box plot of vibration thresholds identified in the pilot test with corresponding amplitude levels selected for the experiment (i.e. triangles and circles).

3.3 Results

The measured variable, proportion correct (p_c), was computed for all factor combinations and participants as follows [167]:

$$p_c = \frac{\text{hits} + \text{correct rejections}}{\text{total trials (12)}}$$

The data were analysed using the software R 4.0.2 with the quickpsy [141] and brms [31] packages.

3.3.1 Psychometric functions

Psychometric functions were estimated by fitting an s-shaped logistic curve of the form [219] to each factor combination:

$$p_c(x) = \gamma + (1 - \gamma - \lambda) \cdot \frac{1}{1 + e^{-k(x - x_0)}} \tag{3.1}$$

where γ is the guessing rate (0.5), λ is the estimated lapse rate, k is the estimated slope, and x_0 is the sigmoid's estimated midpoint. The explanatory variable x represents vibration amplitude. The fitted curves are presented in Fig. 3.11 and the estimated vibration perception thresholds at 75% correct (the midpoints of the curves) in Fig. 3.12. Between the thresholds differences are apparent, while the slopes seem stable across conditions. For horizontal vibrations, the effect of pressing force is small, with threshold estimates of 90.2 dB RMS and 89.4 dB RMS for force 1 (0.5 N) and force 2 (4.9 N), respectively. In contrast, the effect of force is more notable for vertical vibrations: the respective thresholds are 93.7 dB RMS and 89.9 dB RMS.

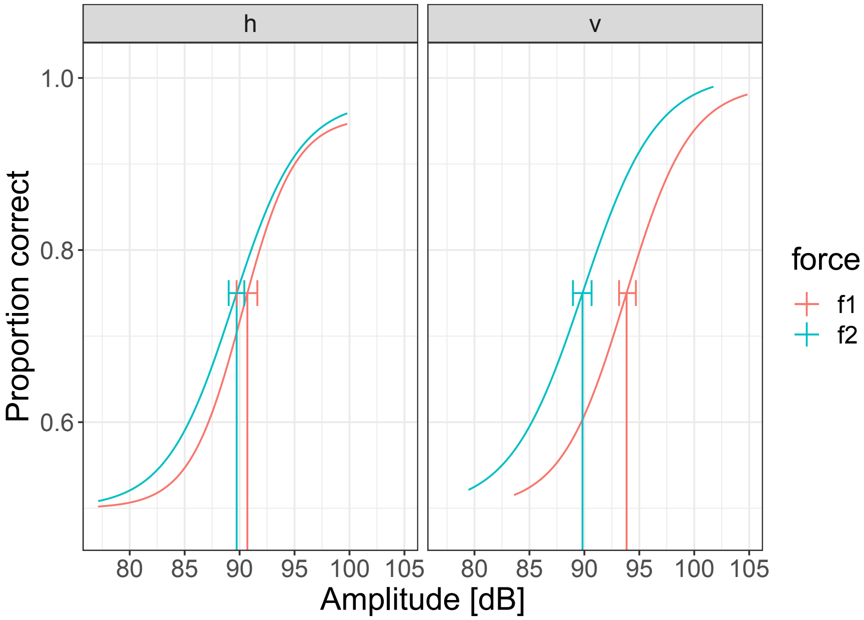


Figure 3.11: Estimated psychometric functions as a function of amplitude for horizontal (left) and vertical (right) vibration.

3.3.2 Statistical analysis

The psychometric functions suggest that at least the motion direction has a significant effect on the midpoint parameters, i.e. the thresholds. In order to estimate the effects of both predictors, the continuous amplitude variable (expressed as zamplitude) was standardized and fitted with the following nonlinear mixed effects model, estimating

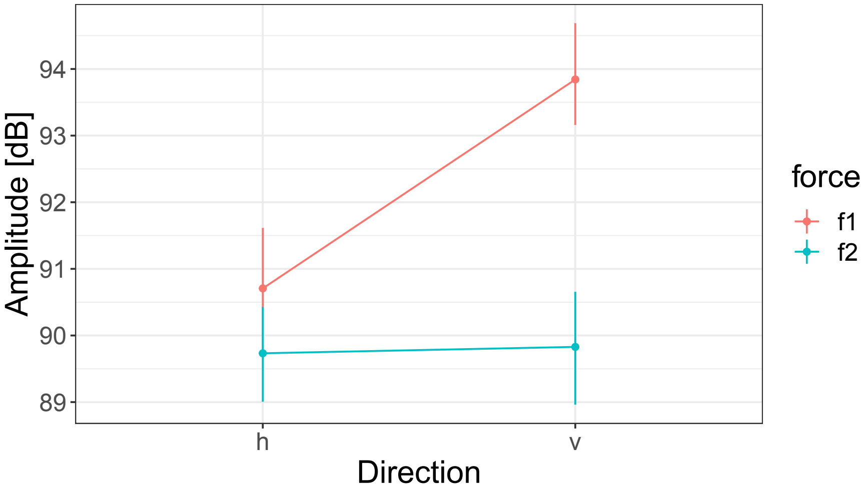


Figure 3.12: Estimated vibration perception thresholds (75% correct).

its parameters by Bayesian inference [133, 31][1]:

$$\begin{aligned} \text{proportion correct} &\sim 0.5 + (1 - 0.5) \cdot \frac{1}{1 + e^{-\eta}} \\ \eta &\sim 1 + \text{zamplitude} + \text{direction} * \text{force} \\ &\quad + (1 + \text{zamplitude}|\text{subject}) \end{aligned} \tag{3.2}$$

where the parameter η of the logistic function depends on vibration amplitude, vibration direction, pressing force, and the direction:force interaction. In addition to these population-level effects, individual intercept and slope were estimated for each subject. A normal prior was used for η. (Because the nonlinear link function is given in the prediction formula, distribution of the response variable was specified as binomial with an identity instead of a logistic link in the function call.)

Population-level effects

The estimates for population-level effects and their 95% Credible Intervals (CIs) are given in Table 3.1. For the nominal factors, the baseline condition is force 1 with horizontal vibrations. Note that the estimate values refer to the nonlinear parameter and are, therefore, hard to interpret in terms of proportions correct; however, positive and negative coefficients respectively implicate an increase and decrease in the proportion correct relative to the baseline condition. As the CIs do not contain zero for any of the main effects nor the force:direction interaction, one can infer that all these effects are significant. In line with the psychometric function estimates , force 2 (4.9 N) has a very slight positive effect in combination with horizontal vibrations (the

---

[1] The statistical analysis was performed by Dr. Hanna Järveläinen

baseline case in Table 3.1), and a notably stronger effect in combination with vertical vibrations.

Table 3.1: Population-level effects of the nonlinear mixed effects model.

Effect	Estimate	l-95% CI	u-95% CI
η Intercept	-0.32	-1.35	0.68
η amplitude	3.82	3.11	4.68
η directionv	-1.22	-1.66	-0.80
η force2	0.47	0.06	0.90
η directionv:force2	1.35	0.78	1.96

Random effects and static predictors

To associate participants with their repeated measures in the model, subjects were treated as a random variable as specified in Eq. 3.2. The estimated random effects produced subject-specific intercepts and coefficients for the effect of amplitude. It was then investigated whether the subject-specific effects could be explained through the two static predictors that were recorded for each participant, namely age and finger temperature. These values were standardized and linear models were fitted between them and both random intercepts and slopes: neither the intercepts nor slopes were significantly associated with participants' age or finger temperature.

3.3.3 Response time

According to Piéron's law [185, 186], response time decreases with increasing stimulus intensity or choice discriminability in detection and decision tasks. In the current detection task, where guessing produces 50% of proportion correct, response time is predicted to decline linearly as a function of stimulus intensity, when intensity is expressed on a logarithmic scale [25]. These chronometric functions [142] are presented in the left panel of Fig. 3.13. The right panel of Fig. 3.13 presents the Speed-Accuracy Functions (SAF), which represent the relationship between response time and logit-transformed proportions correct [25]. Although the response times generally behave as predicted, the psychometric functions fit the data better than the chronometric or SAF functions. This may be understood, as participants were not informed about response time measurement nor instructed to consider how fast they responded.

3.3.4 Pressing force control

The pressing force was recorded during each force evaluation interval (initial load) and its following observation interval (stimulus / no stimulus). Force control error was calculated as the difference between a target force and the respective force exerted by participants. Table 7.3 reports the means and standard deviations of the force control error. A two-way repeated measures ANOVA was performed to test the effect of Target force (0.5 N and 4.9 N) and the type of evaluation interval (initial load,

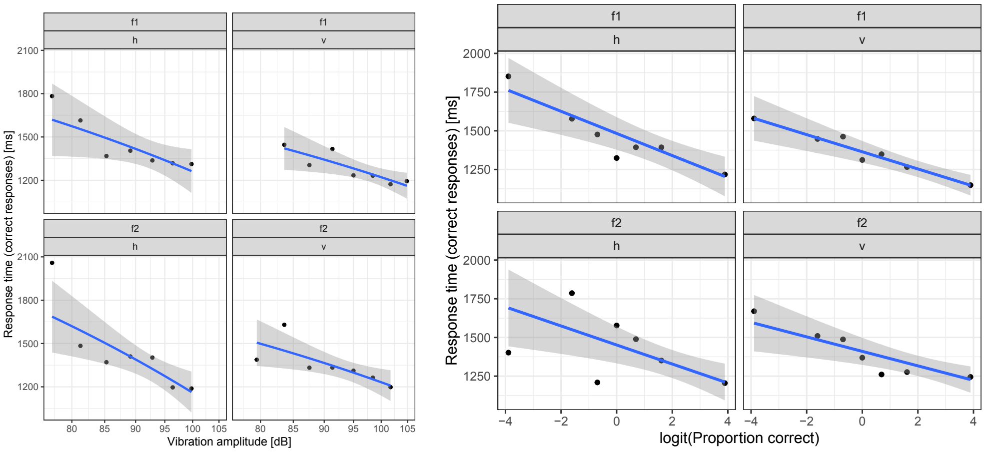


Figure 3.13: Chronometric functions (left) and speed-accuracy functions (right) for combinations of force and vibration direction.

stimulus and no stimulus) on the pressing force control error. The effect of the Target force was found significant (F(1,21)=46.6, p<.001), as well as that of the type of evaluation interval (F(2,42)=4.35, p<.05). Conversely, the factors interaction effect was found not significant. The mean control errors were negative for the higher target force (4.9 N) in all types of evaluation intervals, highlighting an undershoot effect; for the lower force, instead, the mean errors were balanced around zero with undershoots in the evaluation interval and overshoots in the observation intervals. In absolute terms, the control error was higher for the highest force.

The means of the control error were normalized by the target forces as shown in Table 7.3. The normalized accuracy (i.e. mean relative error) stayed within 2.2% for the lower force and within 4.4% for the higher one.

As a measure of precision, the standard deviations of the control error normalized by the means of the measured pressing forces were considered. These values are shown in the last column of Table 7.3: ranging between 1.9% and 3.6%, they suggest that the participants could control accurately and precisely the pressing force for both the chosen force levels. Moreover, during the force evaluation intervals, the precision was higher than in the observation interval, suggesting that the subjects' attention moved from the force precision to the vibration perception.

3.4 Discussion

All participants but one were able to clearly identify at least the highest amplitude for each factor combination, confirming the preliminary results of the pilot test. In particular, that specific participant was unable to identify any of the horizontal stimuli during the experiment.

The experimental results partially confirm the hypothesis H1 and H2 concerning

Table 3.2: Means and standard deviations of force control error, and means of accuracy and precision

Target force (N)	Evaluation window	Error(N) mean ± s.d.	Accuracy mean (%)	Precision mean (%)
0.5	Load	-0.009 ± 0.010	-1.8	2.0
0.5	Vibr.	+0.007 ± 0.017	1.4	3.3
0.5	No Vibr.	+0.011 ± 0.018	2.2	3.5
4.9	Load	-0.215 ± 0.094	-4.4	1.9
4.9	Vibr.	-0.202 ± 0.167	-4.1	3.4
4.9	No Vibr.	-0.168 ± 0.176	-3.4	3.6

the force levels and the stimuli tested.

3.4.1 Effects of motion direction

Although the vibration sensitivity of the whole hand is not much affected by the motion direction [163], at the fingertip Hwang *et al.* [114] found significant differences among directions, with lower perception thresholds associated with vertical motion. By contrast, the psychometric functions estimated from this experiment highlighted lower thresholds for horizontal stimuli, confirming our hypothesis H1, even if the significance of the differences depended on the pressing force level. According to Biggs and Srinivasan [22], the forces produced by horizontal displacement of the fingertip are greater than the forces produced by the same displacement in the vertical direction. This can explain the lower thresholds observed for the lateral vibrations. Also, as modeled by Wu *et al.* [242], the vertical vibration at 250 Hz produces a shear strain on the first skin layer, reducing the effective energy reaching the Pacinian receptor, thus resulting in lower sensitivities.

The discrepancy between the results of Hwang *et al.* [114] and the results of this experiment can be attributed to multiple factors: for instance, the finger press influences the vibrations produced by the actuator [194] but, as far as it is known, the experimental setup of Hwang *et al.* was calibrated without taking into consideration such effects. To overcome such characterization bias, the device here presented was calibrated with human subjects pressing on top of the vibrating surface; the playback of 250 Hz sine signals showed limited acceleration variance among participants, despite their physiological differences and the absence constraints to the finger posture during the measurements. Furthermore, the output of the device was not affected by the finger pressing force, reporting a difference between the nominal and the collateral vibrations (i.e. the cross-talk) of at least 20 dB in both directions. Again, in the study of Hwang *et al.* it was not possible to find any reference to the collateral vibrations measured on complementary axes.

A direct comparison between the absolute thresholds found by Hwang *et al.* [114] and this experiment is not possible since the accelerometer used for the characterization was absent during the experiment; therefore, the level of vibrations reaching the finger increased for all the provided amplitudes. However, it is possible to speculate that the found absolute thresholds should be increased by 3 to 6 dB, resulting in values comparable to [114]. A further difference between the experiment of Hwang *et al.* [114] and this experiment regards the stimuli playback: whereas all the trials provided in the experiment of Hwang *et al.* contained the stimuli, conversely, in this experiment, half of the trials did not contained the stimulus, resulting in more unbiased responses [167].

3.4.2 Effects of pressing force

As reported by [176, 169], the increase of pressing force improves the perception of vibrations in the normal direction. Our experiment confirmed those findings, furthermore proving that the perception improvement is independent from the device mechanical behavior. Indeed, the finger press does not affect the magnitude of the vibrations reaching the fingertip, using the experimental device (see Sec. 3.1.2). Therefore, the lowering of the sensitivity thresholds observed in the normal direction for the highest pressing force (f2) depends only on the fingertip's mechanics: according to [97] the finger contact force against the surface changes the impedance of the finger-pulp, probably resulting in more vibrations reaching the Pacinian receptor. By contrast, the vibration perception in the horizontal direction is not much affected by the pressing force. According to [239], the fingertip compression increases also the horizontal stiffness and damping, following a 1/3 power-law for both distal-ulnar and lateral directions. However, the signal reaching the Pacinian mechano-receptors does not seems to be affected by these changes in the impedance, at least for the tested frequency.

3.5 Lesson learnt

This chapter presented an experiment aimed at assessing whether the vibration direction (normal or lateral) affected the sensitivity to 250 Hz vibration stimuli in a finger-pressing task involving two force levels (0.5 N, 4.9 N). The chapter showed the design and the characterization of a testbed device able to independently reproduce vertical or horizontal vibrations. The latter also allowed to measure the normal force applied by a finger on top of its actuated surface. The experimental results showed that the sensitivity to normal vibrations is affected by the pressing force level, having higher sensitivity associated to the highest pressing force. Conversely, the sensitivity to lateral vibrations were found to be almost independent of the pressing force. The comparison between stimuli directions, instead, showed higher sensitivities for the horizontal direction than for the vertical direction. As a consequence, haptic devices with limited power capabilities should provide lateral vibrations in order to maximize the perception of their feedback. Finally, the dependency existing between the sensitivity in the normal direction and the normal force applied can be exploited to design a wide palette of vibrotactile feedback and, potentially, further tactile illusions.

II

Rotations with haptic feedback

4

A low-cost haptic knob with programmable force feedback

As seen in the previous chapters, touchscreens can be equipped with haptic technologies to enable more robust multimodal user interactions. However, whenever direct manipulation provides safer and simpler interactions, tangible controls could be still be preferred. In today's cars, for instance, whereas most auxiliary functions are accessible from touchscreens or capacitive switches, the functions which closely affect driving are still a domain of physical knobs and selectors [158]. Tangible controls, indeed, expose multiple haptic feedback allowing safer eyes-free selection while performing other main tasks, such as driving [180]. Furthermore, by distributing part of the cognitive load onto the somatosensory channel, they ensure better control accuracy in several everyday contexts and work environments [136].

In between virtual and physical layouts, tangible digital controls can be provided with variable haptic feedback: programmable buttons [140] and sliders [19, 143] can render different force/displacement curves, whereas haptic knobs can generate multiple effects such as detents, barriers, spring repulsion through variable torque. The main downside of devices offering force feedback usually resides in their size, power requirements, and specially costs, which may be several orders of magnitude higher than those of their purely mechanical counterparts. For instance, a mechanical detented knob can cost less than 1 USD, whereas a programmable haptic knob may easily be 100 times as expensive (e.g., [10, 33]); moreover, a haptic knob requires electrical power and is usually much larger.

To this end, this chapter reports the design and the validation of a low-cost, compact knob with programmable resistive force, built around an electromagnetic braking system. Although not required in the context of professional appliances, the power consumption and size of this haptic controller are much lower than those required by similar existing products, due to the absence of motors or other active components. This rotary controller was specifically designed considering the cost limits existing in the market of professional appliances; however, its features can be implemented in many other applications, such as multimedia controllers, cars and piloting systems, wired or wireless PC controllers. For this reason, the hardware and software implementation has been filed with the Italian Patent Office (IPO) [44] thanks to the support of the Patent Office of the University of Udine.

Therefore, this chapter will only partially disclose the hardware/software implementation, whereas it will focus on user interaction aspects, such as the effectiveness of the resistive feedback in a visual-tactile target matching task.

4.1 Related work

Different studies focusing on the manual control of visual tasks, occurring especially during machine operation, have found that tangible controllers can improve human performance, particularly when the cognitive load is high and multiple tasks are performed at the same time [187, 36, 136, 233, 180]. According to such studies, some recent smart gadgets incorporate or offer additional physical knobs: examples include Google Nest [70], an intelligent touchscreen-based thermostat, and the remote multimedia controllers Griffin PowerMate [90] and Microsoft Surface Dial [159]. This evidence suggest that the tangibility of physical knobs is still appealing. At the same time, today it would be desirable to expand their flexibility towards multi-parametric control as currently offered by virtual knobs, which, on the other hand, cannot be operated when visual attention is focused elsewhere (e.g., during teleoperation) [179, 36].

One way to make the manipulation of multiple quantities through a single physical knob more intuitive is to provide it with haptic effects simulating various mechanics. However, this obviously comes at the cost of embedding actuators and their relative electronics in the knob, adding cost, complexity, encumbrance and power consumption. As a result, force-feedback rotary controllers are found only where this can be tolerated: car dashboards, piloting systems, professional audio/video editors, robot controls and medical devices. Most such controllers make use of DC motors to generate force feedback [16, 201, 111, 127]. Hybrid solutions that combine motors and brakes have been proposed as well [33]: while allowing the design of subtle effects, they further increase hardware complexity. Even more expensive and technologically advanced solutions make use of magneto-rheological fluids in which the knob shaft is immersed [232, 10]: in this case, magnetic-field variations are used to change the density of such fluids, allowing precise control of the resistive torque. Finally, programmable haptic feedback is going beyond the mere reproduction of mechanical features like torque and detents, as the idea of branding machine interfaces with a unique haptic "feel" is progressively finding a relevant place in current technology trends [207, 124, 26, 201]. In synergy with rigorous functional design constraints, this idea has nurtured controllers such as the jog wheel aboard the recent Traktor Kontrol S4 DJ console [115]. In the same fashion, a particular and advanced programmable knob controller, such as the one here presented, could be a unique design feature also in the context of professional appliances.

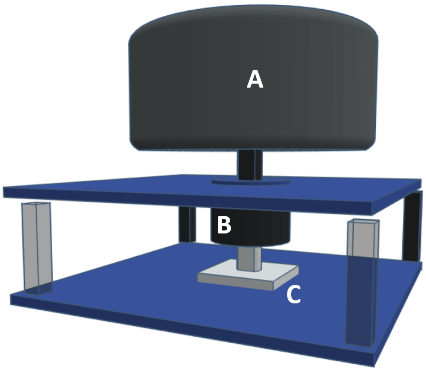


Figure 4.1: Schematic of the haptic knob: the end-effector (A) is connected to an encoder (C) by means of an electromagnetic braking system (B).

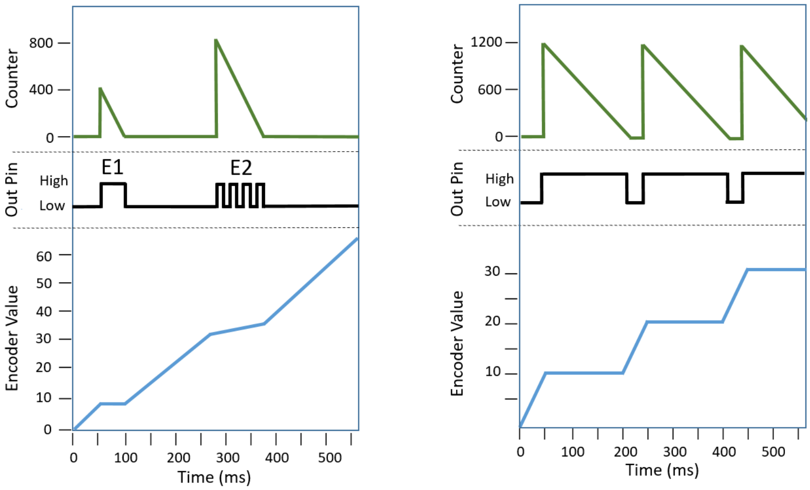
4.2 Hardware / software design

The knob here presented can be programmed to generate frictional patterns which, by resisting against rotation, induce resistive force-feedback effects. Its main hardware components consist of a rotary encoder coupled with an electromagnetic braking system that is controlled by an Arduino microcontroller through its pulse-width modulation (PWM) output.

Compared to existing haptic knobs, the proposed electromagnetic braking system has advantages and drawbacks. The main advantage is that it can easily couple a standard knob to an encoder (see Fig. 4.1), resulting in a thin, lightweight and low-cost device. It also needs less power than DC motors [16], hence enabling battery-supplied portable solutions. The main shortcoming is lack of active force, which limits the feedback to changes in torque, and reproduction of detents as well as end-stops. The implemented control algorithm samples the encoder position and estimates its rotation speed during unidirectional shifts; these data are then used to set the magnitude and duration of the resistive force. The current prototype mounts a magnetic encoder (AMS5600) with a resolution of 4096 points per revolution, corresponding to about 0.1 degrees.

The output voltage of the microcontroller (between 0 and V_{CC}) depends on the relative length of the PWM duty cycle, and directly controls the resistive torque of the knob. For example, if the output maintains a constant voltage, then a proportionally constant torque is applied against rotation.

Detents are reproduced as follows: once the encoder detects the position occupied by a detent, the algorithm sets the length of the PWM duty cycle and the value of a counter, depending on the programmed resistance and estimated rotation speed. The counter is decremented at every cycle, and the electromagnetic system operates at a constant PWM value until the counter reaches zero. The harder the detent, the longer the knob resists against motion.



(a) Two example effects: E1 simulates a soft detent by setting the PWM duty cycle to 100% for 50 ms, resulting in a brief rotation stop; E2 simulates a lower constant resistance by repeating several PWM duty cycles at 50%.

(b) Hard detent effect, repeated every ten encoder steps. Hard detents are simulated by almost stopping hand movement. This is achieved by setting the PWM duty cycle to 100% for 150 ms, resulting in a strong resistive force.

Figure 4.2: Temporal evolution of control variables for various haptic effects: when the encoder position (blue line) reaches a predefined value, resistive force is generated by activating the output signal (black line) until the counter (green line) is decremented to zero. The generated resistive force is proportional to the length of the PWM duty cycle.

Fig. 4.2a shows the temporal evolution of the mentioned variables for two different haptic effects: in the first event (E1), a soft detent is simulated by applying the highest possible resistance for 50 ms; in the second (E2), the action of the braking system is modulated with short activations so as to generate a lower resistance, slowing down hand movement however without stopping it.

In addition to rotation, the knob can also be pushed as a button. Overall, six different input modes are made available by combining clockwise (CW) and counterclockwise (CCW) rotation with no, short or long push.

4.3 Experimental evaluation

A visual target-matching task was set up to test the effect of resistive feedback while performing an action that was functionally equivalent to a drag-and-drop action.

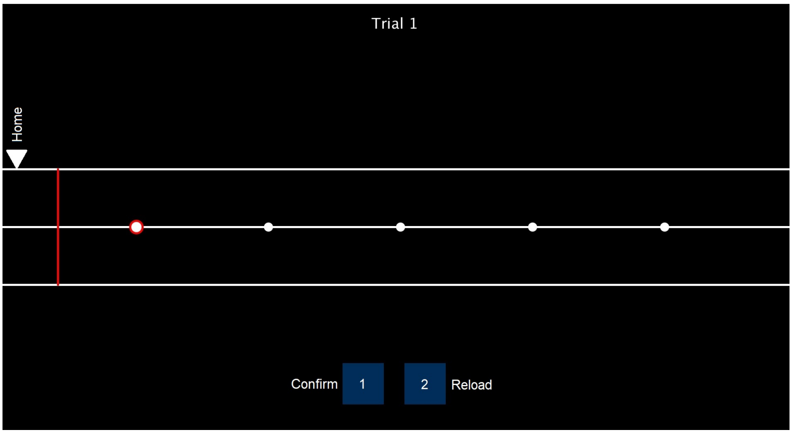


Figure 4.3: The GUI used in the experiment. A vertical red cursor is controlled by the haptic knob; it moves along a virtual horizontal slider which hosts five round markers. At each trial, one marker is highlighted with the red color, thus becoming the current target. The blue squares are activated by pressing the corresponding buttons on the computer keyboard.

4.3.1 Procedure

Participants were asked to sit at a desk where a computer screen, keyboard and the knob were placed. The screen displayed a Graphical User Interface (GUI) developed in Processing 3.5 (see Fig. 4.3): using the knob, a red vertical segment representing a cursor could be dragged horizontally along a virtual slider which contained five equally spaced round markers. Two buttons completed the GUI, labeled 'Confirm' and 'Reload': they were operated by the computer keyboard respectively for confirming that a target was successfully matched, thus ending the trial, or for repeating the current trial.

The shift of the cursor had an accuracy equal to 11.4 pixels per degree of rotation; the markers were set 320 pixels apart from each other, corresponding to a rotation by about 28 degrees.

Figure 4.4 describes the procedure followed in the test. At the beginning of each trial, the cursor was reset to the position marked with 'Home' in Fig. 4.3, and one of the markers along the slider was surrounded by a red circle, symbolizing the current target. Participants had to select this target by dragging the cursor over it: in order to move the cursor they had to push down the knob and turn it. When they reached the target, they released the knob and chose whether confirming or repeating the trial by pressing the respective button on the keyboard using the other hand. At each trial, haptic feedback was either present or absent; when present, it reproduced a hard detent (see Fig. 4.2b) in correspondence of a marker. Each factor combination of 5 targets × 2 feedback conditions was repeated 10 times, hence resulting in 100 trials which were randomly balanced for each participant.

A repetition was typically invoked when participants released the knob off-target,

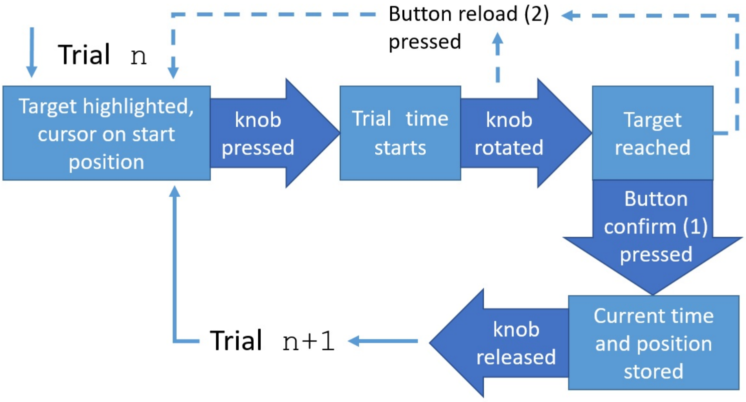


Figure 4.4: Test procedure

similarly to what happens when a wrong drag-and-drop action made with the mouse must be undone. If a trial was confirmed, then the time elapsed from the knob initial press to its release was measured, along with the distance of the cursor from the target (called mismatch from now on).

Ten participants (7 males, 3 females) aged between 24 and 57 (M = 39.9, SD = 10.3), all right-handed, took part in the test on a voluntary basis and were not paid. Before the test, participants performed a training session consisting of 10 trials, in which each factor combination was presented once in random order.

4.3.2 Results

Fig. 4.5 and 4.6 respectively show box plots of the mismatch in pixel and time-to-match in ms, for each factor combination. Tests on the data distributions with the D'Agostino method [42] confirmed no significant deviation from normality.

A two-way ANOVA was conducted to study the influence of the two independent variables (target position, feedback condition) on mismatch. Using a Greenhouse-Geisser correction for insphericity, the main effect for feedback yielded $F(1,9) = 13.4$, $p < .005$, indicating a significant difference between trials in presence of haptic feedback (M = 3.7, SD = 0.31) and without it (M = 4.54, SD = 0.31). The main effect for position instead was not significant with $F(1.5, 13.6) = 0.2$, $p > .05$, and so was the interaction effect: $F(2.5, 22.5) = 2$, $p > .05$. In agreement with previous studies [187], our results show that haptic feedback significantly improved the precision of the action. The smaller mismatch did not come unexpected though, as the increased resistance to rotation occurring while traversing a marker supported participants to stop the cursor at the right location. A further two-way ANOVA was performed to study the influence of the two independent variables on the time-to-match. After a Greenhouse-Geisser correction for insphericity, the main effect for feedback was not significant with $F(1,9) = 0.43$, $p > .05$. The main effect for position yielded

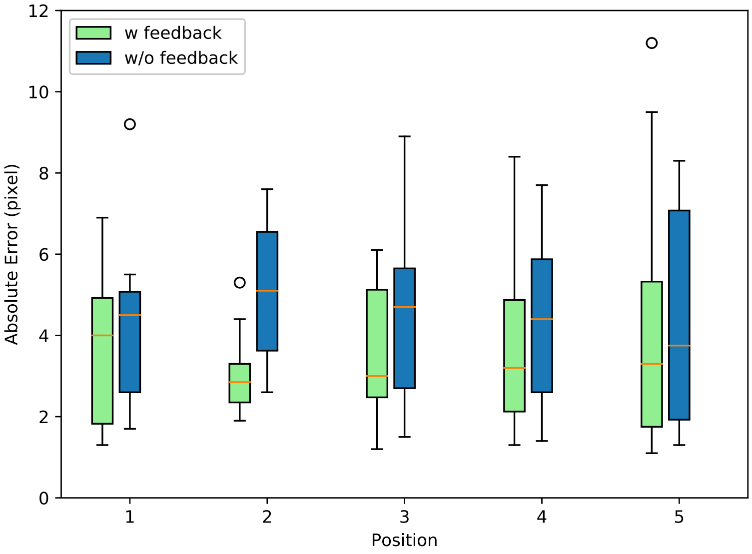


Figure 4.5: Box plots showing the mismatch for each factor combination.

$F(1.4, 12.8) = 22.3$, $p < .001$, indicating a significant difference between position 1 (M = 1396, SD = 150), position 2 (M = 1720, SD = 145), position 3 (M = 1799, SD = 133), position 4 (M = 2049, SD = 184) and position 5 (M = 2223, SD = 170). The interaction effect was not significant: $F(1.5, 13.3) = 1.4$, $p > .05$. The time-to-match depended on the position, with longer time associated to farther targets. While this result may appear obvious at first, one must consider that haptic feedback caused resistance to knob rotation each time a marker had to be traversed, however this had no significant effect on time-to-match. Even further, the time-to-match in presence of haptic feedback was on average generally lower, suggesting that resistive force potentially enables even faster execution of target-matching tasks.

4.4 Resistive feedback design

The resistive force feedback technology calls for a specific design of the haptic effects, too. Whilst haptic knobs based on active force feedback, such as the Torquetuner [127], encode the haptic effects as transfer functions between angle/velocity and torque, a resistive knob needs to be programmed with respect to the resistance duration, in order to design many of the effects. For instance, the illusion of soft and hard detents is generated by locking the knob for different amounts of time. Some haptic feedback effects, that can be designed with the proposed controller, are listed below.

Haptic markers: movements along a graduated scale trigger resistance points simulating soft detents. The top-left plot in Fig. 4.7 shows the encoder position

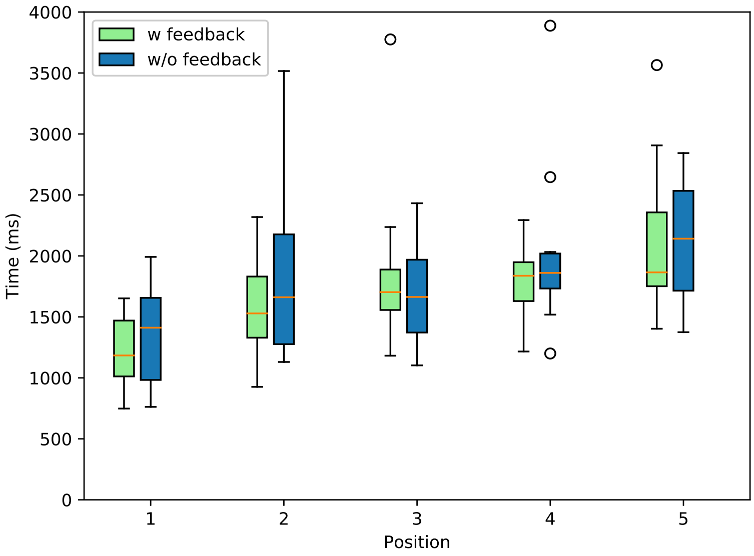


Figure 4.6: Box plots showing the time-to-match for each factor combination.

during 500 ms in presence of a constant torque input. The absence of resistance enables fast motion, resulting in the completion of multiple steps in a short time (steep segments); otherwise, the knob resistance almost stops hand movement for 100 ms (flat segments). This effect has a visual counterpart corresponding to a graduated slider or knob.

Multiple selector: movements across markers generate a resistive feedback simulating hard detents. The bottom-left plot in Fig. 4.7 shows the corresponding feedback: hard detents are rendered by almost stopping hand movement for 350 ms; otherwise, no resistance is generated. This effect has a logical counterpart corresponding to e.g., a set of radio buttons or a rotary selector. A common practice is to use such controls to select source signals, waveforms, or to switch among audio channels.

Variable resistance: resistance is proportional to the encoded value (see the top-right plot of Fig. 4.7). This control can be used to linearly regulate effects' parameters (e.g., wet/dry, intensity, etc.). Additionally, sudden discontinuities of the resistance can be programmed to signal positions of the knob featuring some specific event (i.e. a binary switch following a parameter maximization, or a peculiar value of the same parameter).

Constant resistance: the resistance of the knob is constant, independent of the position (see the bottom-right plot of Fig. 4.7). Different values of resistance may be used to implement softer/harder knobs.

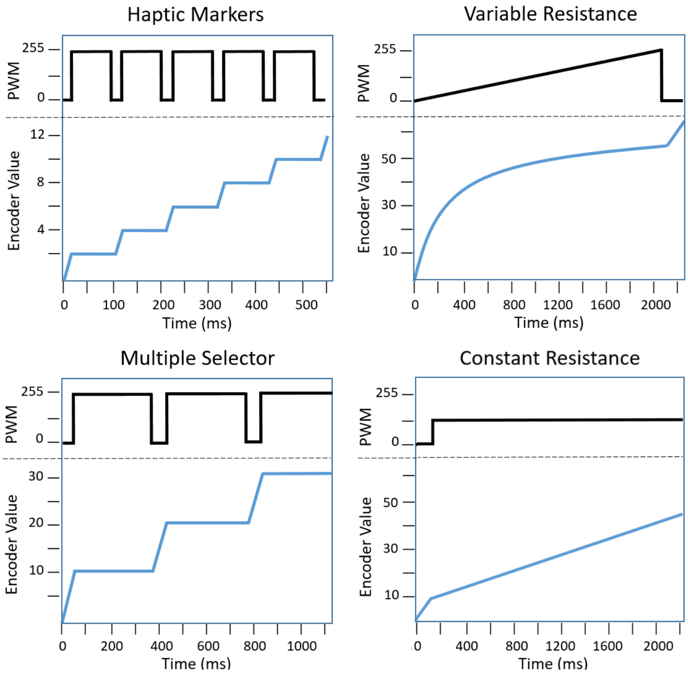


Figure 4.7: Example of different haptic feedback effects for a constant torque input.

The first two effects provide discrete information, whereas the third and fourth effects are continuous.

4.5 Applications

In the following, some implementations of the effects are reported.

4.5.1 Multimedia production

A pilot software application was developed in Processing to demonstrate the use of the device limited to some simple control of effects in music production. A screenshot of the GUI is reported in Fig. 4.8. The Processing application communicates with the device through a serial bus. The operations made available by the software are described below:

Volume control: the controller is associated to the blue virtual knob of Fig. 4.8. Each discrete value is visually denoted with a small tick. Movements across ticks generate a resistance simulating soft detents.

Effect selector: the controller is associated to the green virtual knob of Fig. 4.8. Five possible choices are displayed with ticks. Movements across ticks generate

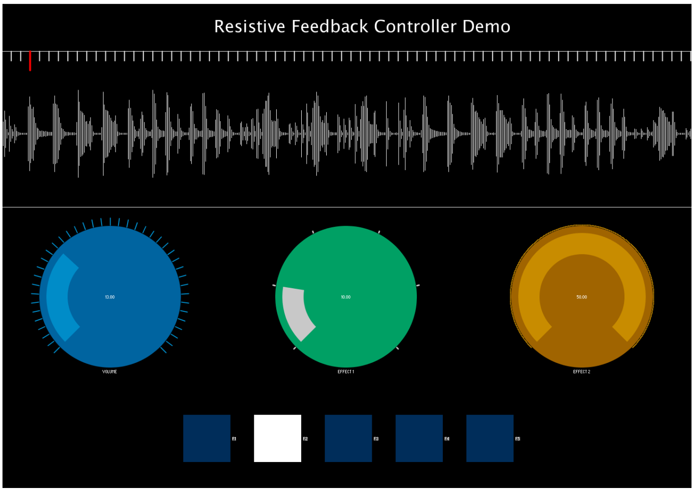


Figure 4.8: Software developed in Processing to test the resistive haptic effects for multimedia production.

a resistance feedback that simulates hard detents.

Parameter control: the controller is associated to the orange virtual knob of Fig. 4.8. Depending on the effect selected with the green virtual knob, the resistance changes (i.e. constant or variable).

Amplitude transient detection: the rotary controller allows to explore the track shown in Fig. 4.8 across time. A red cursor visually prompts the position on the waveform. Moving along the waveform produces a varying resistance, proportional to the energy of the signal crossed by the cursor.

Time window navigation: the rotary controller allows to explore the track across time. Time units are displayed above the waveform. A red dot marks the current position. A hard detent effect is generated each time the cursor moves through the marker.

A video footage of the haptic knob being used in the five operations mentioned above is available in an open-access repository.[1] Possible formal evaluations of these features are left for the future.

4.5.2 Control of digital audio effects

The subsequent step in the development process was the test of the device as a multi-parametric input device for the control of digital audio effects executed at audio rate. In particular, the device was set as controller of the digital simulation of the EMS VCS3 voltage-Controlled Filter (VCF) [75]. Given the computational complexity of

---

[1] `https://doi.org/10.5281/zenodo.3757901`

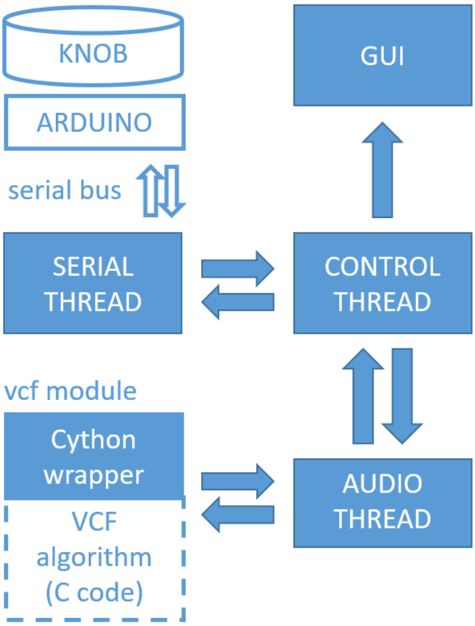
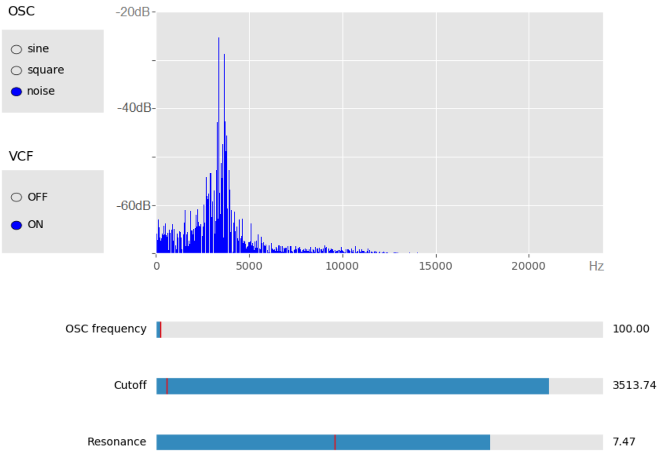


Figure 4.9: Python software architecture for interactive control of an audio effect with the haptic knob.

the digital model, the trivial implementations developed in Processing 3.5 were abandoned in favor of a structured multi-threading software architecture. The software was developed in Python 3.6 adopting the real-time audio programming approach described in Chapter 8. Sound algorithms, simulating the behavior of electronic circuits (e.g., audio filters), can be considered a good benchmark to stress the performance of a software architecture regarding CPU usage and memory management. This way, a fluid and interrupted interaction accomplished while computing complex audio algorithms will somehow ensure the reliability of the system while testing the prototype for the control of professional appliances simulated in Python.

Figure 4.9 shows the implemented software architecture: the resistive knob device communicates with the digital effect through the Arduino microcontroller via serial connection. The related thread asynchronously receives commands and positional data, whereas the main control thread makes them available to the DSP algorithm, and in parallel updates the GUI showed in Fig. 4.10. The audio thread realizes a simple audio source by synthesizing a sine wave, a square wave, or white noise; in the meantime, it processes sample-by-sample chunks of audio through the VCF algorithm. Such algorithm can be easily developed in Python and processed by the Numba library or can be developed in C code and, hence, made available to the Python interpreter through the so-called *cythonization* (see Chapter 8). Two input modalities were made available through the knob:



OSC
○ sine
○ square
● noise

VCF
○ OFF
● ON

OSC frequency 100.00

Cutoff 3513.74

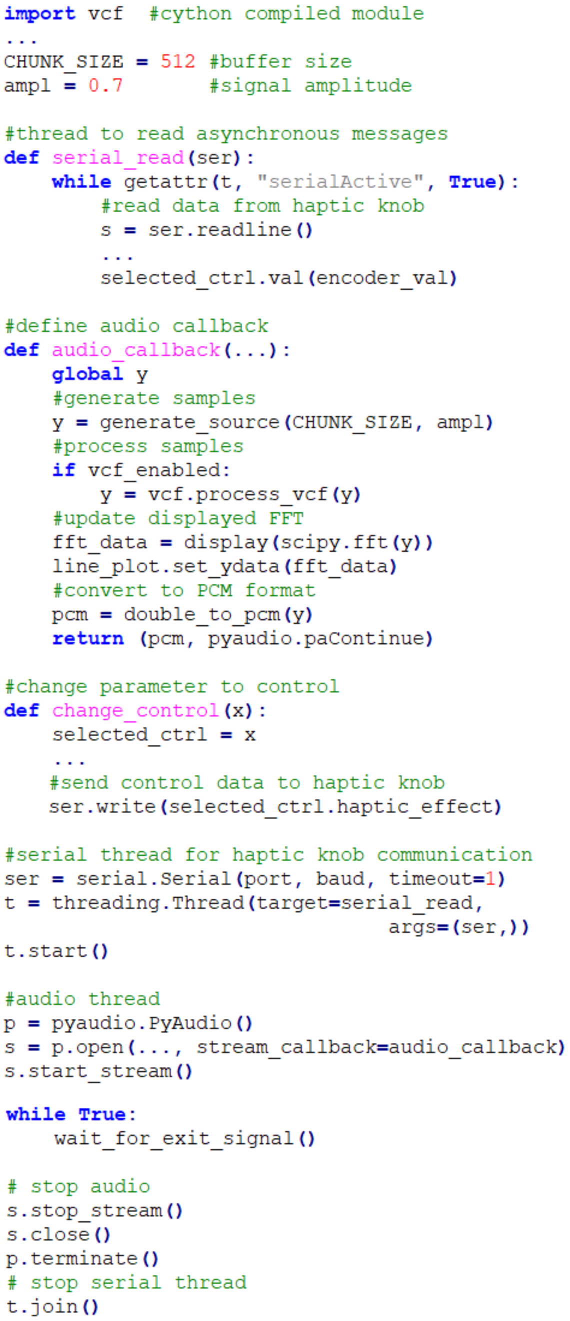
Resonance 7.47

Figure 4.10: Audio effect GUI: Python-based visual interface displaying the available user controls and spectrum of the processed sound output.

- The **navigation mode** allows to select the active control. The modality is enabled by keeping the knob pressed while browsing over the controls through CW or CCW rotations;
- The **control mode** is enabled by simply rotating the knob CW or CCW once a control parameter has been selected, and allows to change its values.

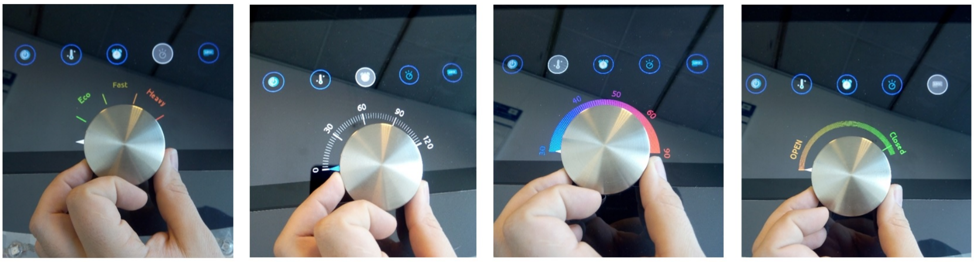
With regard to the implemented resistive feedback, in navigation mode a hard detent (Multiple selector of Fig. 4.7) is generated each time the user jumps to a different control, while in control mode the following effects are produced: the selection of the sound source, as well as the filter on/off switching, are rendered by hard detents (visually displayed as radio buttons); the VCF cutoff frequency (visually represented by a slider) is rendered with a fixed low resistance; finally, the VCF resonance (visualized using a slider) is rendered with a variable resistance. The knob can be also used to select the fundamental frequency of the oscillator: in this case, the knob actuates no feedback, just conveying its own physical resistance. All the controls, visually represented by sliders, furthermore provide a barrier effect corresponding to the sliders' endpoints.

Figure 4.11 shows code excerpts of the Python threads forming the software architecture represented in Fig. 4.9. When a control is selected using the knob in navigation mode, the function `change_control(x)` is called, which populates the `selected_control` object attributes with the content of the argument `x`. The object `selected_control` embeds information such as the controlled parameter and its current value, the functions mapping the knob position into this value and, finally, information needed by the visual display and haptic effects. Thus, when the active instance of `selected_control` is changed, the GUI is updated and a consequent command is sent to the knob controller to configure the corresponding haptic effect. Next,



```
import vcf  #cython compiled module
...
CHUNK_SIZE = 512 #buffer size
ampl = 0.7       #signal amplitude

#thread to read asynchronous messages
def serial_read(ser):
    while getattr(t, "serialActive", True):
        #read data from haptic knob
        s = ser.readline()
        ...
        selected_ctrl.val(encoder_val)

#define audio callback
def audio_callback(...):
    global y
    #generate samples
    y = generate_source(CHUNK_SIZE, ampl)
    #process samples
    if vcf_enabled:
        y = vcf.process_vcf(y)
    #update displayed FFT
    fft_data = display(scipy.fft(y))
    line_plot.set_ydata(fft_data)
    #convert to PCM format
    pcm = double_to_pcm(y)
    return (pcm, pyaudio.paContinue)

#change parameter to control
def change_control(x):
    selected_ctrl = x
    ...
    #send control data to haptic knob
    ser.write(selected_ctrl.haptic_effect)

#serial thread for haptic knob communication
ser = serial.Serial(port, baud, timeout=1)
t = threading.Thread(target=serial_read,
                                    args=(ser,))
t.start()

#audio thread
p = pyaudio.PyAudio()
s = p.open(..., stream_callback=audio_callback)
s.start_stream()

while True:
    wait_for_exit_signal()

# stop audio
s.stop_stream()
s.close()
p.terminate()
# stop serial thread
t.join()
```

Figure 4.11: Threads forming the Python software architecture.



Cycle selection:
hard detents

Cycle time:
soft detents

Cycle temperature:
constant resistance

Door lock:
variable resistance

Figure 4.12: The resistive knob controller applied to a washing machine interface: the knob, partially overlapped to a touch screen, renders different haptic effects tailored on the specific function: hard detents to mark the three programs, soft detents to set the timer, constant resistance to adjust the temperature and a variable resistance to lock/unlock the door

after some processing by the `serial_read()` thread, the selected control is set with positional data coming from the knob.

The `audio_callback` function, called by the stream object `s` of the PyAudio library, generates and processes audio signal buffers according to user parameters setting the sound source and the effect properties (see Chapter 8). The function `process_vcf(y)` realizes the VCF and belongs to the library `vcf`, independently compiled and then imported as shown at the beginning of the code example. The GUI, generated using Matplotlib widgets, is regularly updated by the audio thread concerning the FFT window, and by the control thread concerning the visual elements forming the controls.

4.5.3 Professional appliances

The device was finally tested as a user interface for professional appliances. In particular, a demo user interface was designed for controlling a coin-operated professional washing machine. The interface was assembled in the Electrolux Professional workshop making use of a UDOO Quad [223] development board running a Linux Debian OS. The main control software was developed in Python 3.6 using the QT4 graphical framework, whereas the firmware of the resistive knob was compiled for the Sam3x processor of the same UDOO board. The communication between the mock-up user interface and the washing machine was established using the Electrolux Professional proprietary protocol. Figure 4.12 shows different visual-tactile configurations of the knob while controlling the parameters of the washing machine. In this implementation, the resistive knob partially overlaps a ten-inches touchscreen, which displays a needle-shaped cursor controlled by the knob. This way, the pointer can be perceived as an extension of the knob, improving quality and naturalness of the interaction. The resistive feedback and the visual representation of the controls were selected in accordance with their function:

Cycle selection: the limited number of washing program available on coin operated machines (e.g., from 3 to 6) can be controlled through a selection switch. The borders surrounding the program names (leftmost image of Fig. 4.12) are marked as haptic hard detents remembering an old-fashion mechanical selection switch.

Cycle time: time selection was inspired to the mechanical timer, generally found on the user interface of domestic ovens. Soft detents marked each tick visually displayed on the screen.

Cycle temperature: since the washing temperature is a continuous coarse parameter, no haptic effects were rendered during this selection, keeping the resistance constant.

Door lock: the rightmost image of Fig. 4.12 shows the control proposed for the machine porthole lock. The lock mechanism of real washing machines has a peculiar haptic feedback curve: the clockwise rotation of the handle gives rise to a resistance that increases proportionally up to the final lock position; at that position, the resistance ceases completely confirming the porthole lock. The same haptic feeling was realized with the programmable knob, whose resistance was function of the rotation angle up to the "Closed" position. Then, the knob resistance was set to the lowest level, confirming the close action.

The user interface was informally tested by Electrolux Professional AD&T members in the context of the rapid prototyping of innovative user interfaces; however, the results are omitted due to the non-disclosure agreement with Electrolux Professional.

4.6 Lesson learnt

This chapter reported the design of a low-cost knob controller with programmable resistive feedback. Its effectiveness was assessed in a visual target-matching test that showed a significant reduction of the positional mismatch when resistive feedback was applied to the knob. This study suggested that, even in absence of active forces as those provided by more expensive, bulky and power hungry motorized knobs, resistive feedback has concrete potential to support effective visual browsing of sequential domains such as sliders, program menus, and, in general, all those maps where physical rotary controls can be conveniently employed for the selection of targets. Moreover, thanks to the resistive feedback, the controller can be profitably used in eyes-free context, such as the operation of professional appliances. Several resistive haptic effects were designed and tested in different application contexts, such as multimedia production, digital audio effects control and professional appliances. Future development of the presented technology may involve the addition of vibrotactile actuators to the knob handle in order to obtain richer haptic effects during rotation (e.g., roughness). However, most of the possible implementations must face all the problems related to power and signal transmissions to rotating interfaces (e.g., wire de-rotations, radio transmissions).

Concerning software architecture, whereas the trivial mock-up applications used to test the user interaction were developed in Processing, more complex applications were designed using a multi-threading approach developed in Python. A more detailed discussion regarding the implementation of the audio digital signal processing applications in Python is reported in Chapter 8.

5

A performance comparison between rotary and motionless knobs

In the previous chapter, physical knobs were presented and discussed, suggesting their possible advantages with respect to touchscreen interfaces. However, compared to touchscreens, physical knobs are exposed to mechanical wear, seepage of liquids or dirt, accidental shocks or vandalism. These factors may reduce the lifetime of the knob and, consequently, the operation of the appliances. Moreover, physical knobs bind the design of user interfaces and reconfigurable layouts. Therefore, the substitution or virtualization of physical knobs still represents a milestone to come, since the straight knob virtualization on touchscreens has several disadvantages:

- the diameter of physical knobs should stay within a certain range to allow for a comfortable rotation gesture [109];
- fingers can partially occlude the screen during rotation;
- the implementation of realistic multi-point haptic feedback with virtual knobs requires complex technologies [175].

As a consequence, even if displayed as circular objects, virtual knobs are often controlled using sliding gestures: once selected, the user performs linear motions with one finger on an arbitrary area of the screen while the knob visually changes its value. This interaction technique is often implemented in touch-enabled digital audio software (e.g., sound mixers, audio effects, synthesizers) where, usually, the visual attention can be exclusively focused on the interface. On the other hand, it may results impractical for professional appliances having the user interface placed outside the visual working area (e.g., under the appliance worktop).

To address some of the issues pointed out above, one can envision a trade-off between physical and virtual knobs: a *motionless knob* consisting in a fixed protruding cylinder whose side surface is able to sense rotation gestures.

A motionless design brings together many advantages of both virtual and tangible knobs: a short protrusion can be operated non-visually while being part of a larger user interface; it prevents seepage of dirt and liquids and, finally, no mechanical parts (nor wear) are involved. To this end, a motionless knob can also provide rich haptic feedback by means of vibrotactile actuators, without caring about the signal and power transmissions to a rotating device (i.e. wires de-rotation). At the same time, such design poses issues that need to be investigated, since i) an accurate recognition of rotation gestures can be complex and difficult to model, and ii) users may not find it intuitive to slide their fingers around a fixed cylindrical object. In this regard, an experiment is presented aimed at measuring objective and subjective differences between the haptic rotation of a rotary knob and fingers sliding around a motionless knob. The term *haptic rotation* was recently suggested for a rotation providing only tactile information [130]

5.1 Related work

The haptic rotation of knobs and, in general, of rotary objects has been widely studied. Besides the literature focusing on force feedback [179, 180] mentioned in the previous chapter, haptic rotation was also investigated in its proprioceptive aspects. Rods have been used to assess the oblique effect and perceived parallelism through orientation reproduction tasks [85], executions of verbally ordered orientations [148], or comparisons with reference orientations [120]. Other experiments analyzed haptic rotation using an interface equivalent to a rotary knob. Krieger *et al.* tested whether the shape of the handle (e.g. rounded, edged, flat), the number of fingers, or the initial angular position influenced accuracy and precision. They found that a round shape provides the lowest precision [130], whereas progressively increasing the number of fingers (from 2 to 5) grasping the handle led to greater accuracy; on the other hand, the initial angular position affected rotation precision (i.e. repeatability) but not accuracy, with better results associated with the horizontal grasping position [131].

Also, excess rotation (overshooting) was generally found with respect to a given target.

In parallel, rotation gestures have been studied on touchscreens, that is under visual-tactile conditions, especially considering gestures made with two fingers. A study involving 90° rotations [109] showed multiple outcomes: gesture duration and ergonomic failures increased with the distance (i.e. diameter, between 40 and 85 mm) existing between the fingers, furthermore depending on the initial angular position and rotation direction; moreover, rotations in which the index finger was close to the horizontal axis were the slowest. Concerning motor control, Olafsdottir *et al.* found that awareness of the target angular position helps to optimize the initial one [172]. A further study involving elderly people showed that rotation gestures can likely be performed using the thumb finger as a pivot, while the index finger draws a semicircle, thus easing motor control [144].

Conversely, no user studies related to motionless knobs are currently present in the literature. For this reason, the following experiment draws its methods from the aforementioned studies.

With regard to sensing technology suitable for motionless knobs, capacitive sensors are a common choice for multi-touch position tracking, also for circular surfaces, and it was recently proposed to embed them also in the encoder of rotary knobs [37, 208]. However, capacitive sensors generally have low spatial accuracy, especially when fingers are close to each other. Instead, since spatial accuracy was of utmost importance for our experiment, the position of multiple fingers was tracked using a high-speed video camera.

Two devices were implemented for the purpose of the experiment, object of comparison: one built around an optical finger tracker which could be configured either as a motionless or a rotary knob, and the other in the form of a standard rotary knob attached to an encoder. In this way, it will be possible to directly compare the performance of rotary and motionless knobs implemented with a shared tool, furthermore validating the optical acquisition system with respect to an encoder-based rotary knob.

5.2 Setup

5.2.1 Optical finger tracker

The finger tracker device, shown in Fig. 5.1, was implemented based on a self-developed optical tracker and a milled aluminium cylinder with a smooth side surface, around which fingers could easily slide. In compliance with ergonomics studies [173, 214], an external diameter of 65 mm was chosen for the cylinder. The inner side was further milled to host a 24-LED ring (WS2812 5050 RGB) to enlighten the user's fingertips. The cylinder was fixed to a Plexiglas panel (200×300 mm) by means of an M10 bolt. The function of the bolt was twofold: first, it provided a reference point for a finger-tracking algorithm; second, it acted as a pivot for the cylinder, also allowing to set its resistance against rotation. To this end, a half-threaded bolt having a straight neck of 20 mm was used: in this way, the cylinder rotates with low resistance when the bolt is fully screwed-in, whereas it can be firmly locked by adding a 2 mm thick washer. The Plexiglas panel was enclosed in an aluminium frame mounted perpendicularly to a metal support. This support also hosted a 40×40 mm platform enabling micrometric adjustment of the position, pan and tilt of a Sony PlayStation Eye high-speed camera (60 fps @ 640×480 pixel resolution) connected to a PC and facing the back of the Plexiglas panel.

Video processing

Figure 5.2 shows the computer vision system at work. A custom software application, written in Python 3.6 and using OpenCV 4.1.2, tracks the illuminated fingertips and extracts positional data. As a first processing step, the software compensates for lens distortion by applying a geometric transformation. Lens distortion correction – visible on the vertical borders of Fig. 5.2 – was assessed using chessboard image templates [94].

Video data are then processed as follows:

1. while displaying the first video frame, the software prompts the operator to

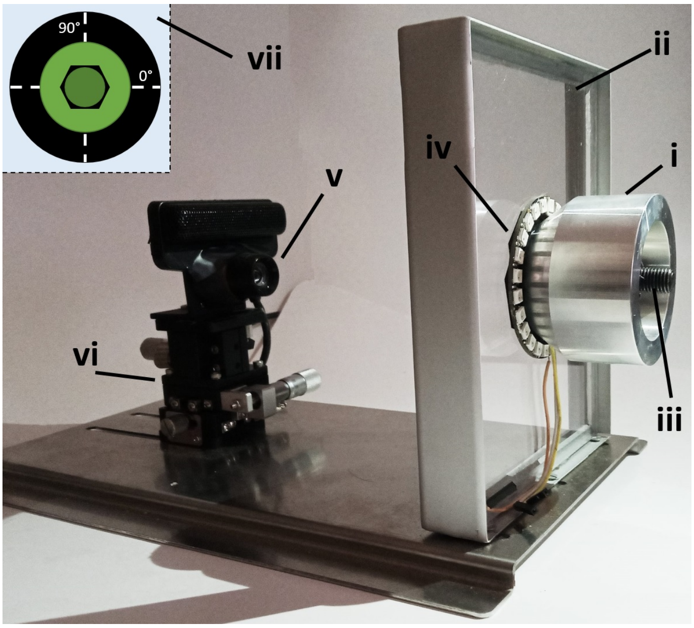


Figure 5.1: The finger tracker system: a metal cylinder (i) is fixed to a Plexiglas vertical panel (ii) by means of a M10 bolt (iii); a ring of 24 RGB LEDs (iv) enlightens the user's fingertips grasping the cylinder; on the back of the panel, a high-speed camera (v) standing on an adjustable metal support (vi) tracks the angular position of the fingertips using as a reference two green markers placed on and around the bolt head (vii).

draw three rectangles with the mouse, each enclosing a fingertip, in the following order: 1) thumb, 2) index and 3) middle finger;

2. for the subsequent frames, the software automatically tracks the fingertips using the CamShift algorithm [27], with the additional constraint that their distance from the center of the cylinder must be constant. Three small red circles are overlaid on the video frame to represent the position of the fingertips, with respective numbers;

3. automatic tracking is supervised offline by the operator, who could inspect that the red circles constantly matched the respective fingertips;

4. as a final check, while displaying the last frame of the video the software asks the operator to optionally adjust the position of the circles, in case errors occurred during tracking (e.g., if two fingers overlapping during the task caused the algorithm to swap the association with the corresponding circles).

For each video recording, the software generates a list of frames containing the three fingers' (x, y) coordinates and the respective angle. The difference between the

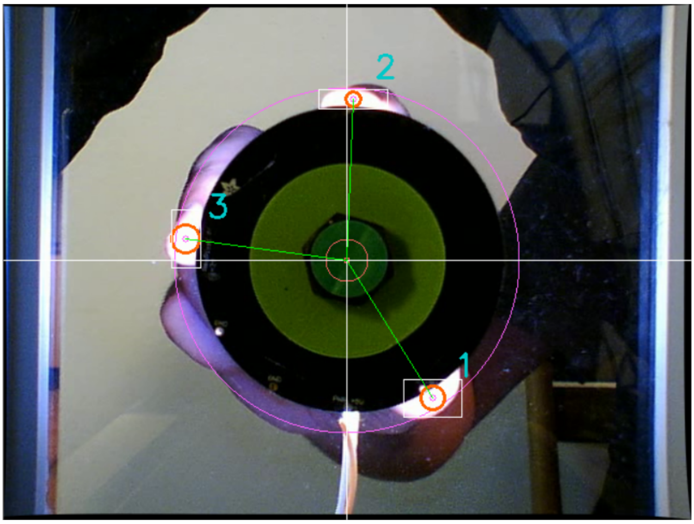


Figure 5.2: Finger-tracking: a Python application detects the position of three illuminated fingertips (small red circles) bounded within a circular region (large red circle enclosing the cylinder). The thin white cross marks the center reference.

last and the first frame of each list was used to compute rotations, resulting in the following data for each video recording: initial angle, final angle and overall rotation of each finger, average rotation of all the fingers and respective standard deviation.

Focal alignment and validation

A preliminary calibration procedure was needed to align the camera with the cylinder: to this end, a plastic ring (internal $\oslash = 65$ mm, external $\oslash = 85$ mm) was placed around the cylinder, then the support was adjusted so as to align and center the camera with three reference markers: a disc coinciding with the bolt head, a ring around it, and the mentioned outer plastic ring around the cylinder (only the first two are visible, in green, in Fig. 5.2).

To assess the accuracy of alignment and tracking, three white circular spots, surrounded by a black background were glued at the side of the plastic ring, thus simulating three illuminated fingertips facing the camera, and 64 random rotations of the cylinder were recorded. The tracking algorithm was then applied to the recordings: the resulting tracking error ranged between 0.15 and 1.46°, with mean standard deviation of 0.64°. Moreover, this error was evenly distributed around the circumference of the cylinder, proving that the camera was correctly aligned with it.

5.2.2 Standard rotary knob

An equivalent device based on a standard rotary knob was also implemented. A metal knob identical to that of the finger tracker was fixed to a vertical metal panel having the same size and attach points as the Plexiglas panel of the finger tracker, so as to make them structurally identical. The cylinder was connected to an AMS 5600 magnetic encoder, having a resolution of 0.1°, which was connected to the PC used for the experiment, and asynchronously read through the serial channel.

5.2.3 Trial control

Finally, a device was built to let participants record or repeat trials via two buttons labeled REC and UNDO. The buttons are connected to an Arduino UNO microcontroller which also controls the collection of data from the encoder and the finger tracker. In particular, when the rotary knob is used, the REC button enables the recording of data from the encoder; conversely, when the finger tracker is used, the same button switches on the LEDs ring along with video recording. The UNDO button instead was optionally used by participants who would like to repeat their latest trial.

5.3 Experiment

The experiment was conducted during the COVID-19 pandemic, in compliance with the local safety rules that posed specific access restrictions to the location where the experiment took place and required to sanitize the setup before each session.

5.3.1 Participants

Sixteen people (9 males, 7 females) aged between 22 and 45 years (M=28.6, SD=5.95) took part in the experiment. None of them reported visual or sensorimotor impairments. Before each session, participants were briefed on the experimental procedure: the instructions for using the setup were displayed on a computer screen. All subjects were right-handed but one. They all participated on a voluntary basis and were not paid.

5.3.2 Procedure and task

Before each session, participants were briefed on the experimental procedure: the instructions for using the setup were displayed on a computer screen and further clarified by the experimenter. In particular, the experimenter checked that each participant had a clear mental representation of the target angles used in the experiment (45° and 90°, see Fig. 5.3). Also, participants were instructed to always use three fingers during the experiment, namely the thumb, index and middle finger: this limited the factors in the experiment, meanwhile not impeding a natural operation of the knobs.

The input devices (either the finger tracker or standard knob) were placed side by side and covered by an overhanging dark cloth which kept them out of sight (see Fig. 5.3). Participants sat in front of the computer screen and could adjust the chair until achieving a comfortable position of the arm while grasping the metal cylinders with their dominant hand. The cloth also prevented participants to see their hand

Figure 5.3: Participant grasping the cylinder of the finger tracker while reading the rotation to be performed (45° counterclockwise in the represented case).

and wrist while operating the input devices. Then, they were instructed to use the trial control device with the other hand, and to focus their vision on the computer screen across the entire experiment.

During the session, various target rotation angles were prompted on the screen, at which point the participants had to i) grasp the knob of the requested device with three fingers of their dominant hand, ii) press and hold the REC button, iii) perform the requested rotation and, finally, iv) release the REC button. During the recording phase, a red dot followed by the word REC was displayed on the screen; after the recording was stopped, participants were allowed to press the UNDO button if they were not satisfied with their performance and preferred to repeat the trial.

At the end of the session, participants were asked to rate the naturalness and pleasantness of the interaction with the input devices, as well as the perceived ease of execution with each input device. Moreover, participants were asked about the number of fingers they would used to turn the standard rotary knob without experimental constraints.

5.3.3 Experimental design

The experiment followed a within-subjects design and compared three different configurations:

- Rotation: participants rotated the cylinder of the finger tracker, set free to rotate.
- Gesture: participants performed a rotation gesture by sliding their fingers around the finger tracker cylinder, kept locked.
- Routine: participants rotated the standard knob.

The factor Configuration (Rotation, Gesture and Routine) was crossed with the factor Angle (-90°, -45°, 45°, 90°) for a total of 3×4=12 factor combinations. Each combination was repeated 8 times, resulting in 96 trials per participant. The experiment was carried out in 3 blocks, one for each Configuration.

Half of the participants performed the Routine block first, and the other half performed it as last block; the Gesture and Rotation blocks were presented in random order. The 32 trials within each block were also randomized. No time limit was imposed for task completion, however a session typically lasted 30 minutes.

5.3.4 Dependent variables

After the collection in lists of frames and optional manual correction (see Sec. 5.2.1), the data in each list were merged in a single comma separated value (CSV) file. On such data, four dependent variables were computed and analyzed: signed error, variable error, finger deviation and initial position.

Human performance during haptic rotation has been measured in the literature using signed and variable errors [130, 148]: the former measures the *accuracy* in performing a task, whereas the latter provides insight on the related *precision*.

The signed error for the Routine configuration was computed as the difference between the recorded and target angle rotation; for the Gesture and Rotation configurations it was instead calculated by averaging the difference between the recorded and target angle rotation across the three fingers. Then, following the literature [131, 148, 110], the mean signed error was computed for each factor combination by averaging the respective signed errors made across eight repetitions. For clockwise rotation, positive values corresponded to mean angle rotation larger than the target (overshoot), whereas negative values corresponded to mean angle rotation smaller than the target angle (undershoot). For counterclockwise rotations the opposite applied. In both cases, smaller absolute mean error values signify higher *accuracy*.

The variable error was computed as the standard deviation of the signed errors belonging to a factor combination. Smaller values signify higher *precision*. The finger deviation was computed (obviously limited to the Rotation and Gesture configuration) by averaging the standard deviations of the angle covered by each finger for each factor combination. The initial angular position was measured (again, limited to the Rotation and Gesture configurations) at the thumb.

5.4 Results

A statistical analysis, described in Sec. 5.4.1, 5.4.2, and 5.4.3, was conducted to check whether the dependent variables (signed error, variable error, finger deviation and initial position) were significantly affected by the independent variables (i.e. factors Configuration and Angle). Finally, Sec. 5.4.4 reports a descriptive statistics of the participants' questionnaires.

Separate analyses were carried out for the dependent variables, as they were found to be uncorrelated (Pearson Index$\approx$0, p-values$\gg$0.05), except for the pair signed error-variable error resulting in a moderate positive correlation (Pearson Index=0.48, $p<0.01$).

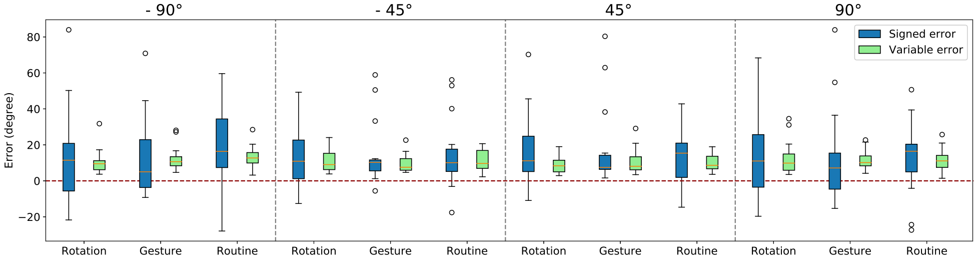


Figure 5.4: Boxplot of signed and variable errors for all factor combinations.

In order to keep type I error under control, standard confidence thresholds $\alpha = 0.05$ were Bonferroni-corrected ($\alpha/4 = 0.0125$)[23, 203]. Analyses were carried out in Python 3.6 using the module *scipy.stats* 1.4.1.

5.4.1 Signed and variable error

Table 5.1: Signed and variable errors grouped by Configuration: each column reports means (M), medians (Mdn),standard deviations (SD), standard Error (SE), lower 95% CI bound (LOW 95CI) and upper 95% CI bound (UP 95CI) for all angles.

	Routine		Rotation		Gesture	
	signed	variab.	signed	variab.	signed	variab.
M	14.7°	11.5°	13.3°	10.9°	14.4°	11.0°
Mdn	15.3°	11.2°	11.1°	9.3°	7.5°	9.5°
SD	±19.5°	±5.8°	±21.4°	±6.9°	±21.8°	±6.0°
SE	2.45°	0.72°	2.69°	0.87°	2.74°	0.76°
LOW 95CI	9.87°	10.02°	8.07°	9.19°	9.04°	9.48°
UP 95CI	19.51°	12.87°	18.63°	12.61°	19.80°	12.46°

Table 5.1 shows mean signed error and mean variable error distributions grouped by Configuration, whereas Fig. 5.4 reports the related boxplot for all factor combinations.

An inspection of Fig. 5.4 reveals that the signed error was only occasionally negative, and on average always positive. This means that, independent of Configuration and Angle, participants on average overshot during the task. The significantly positive correlation existing between signed and variable errors implies that *accuracy* in task execution was paired with *precision*, and vice-versa. However, as visible in Table 5.1, means (M), medians (Mdn), and standard deviations (SD) of the signed and variable errors are similar across configurations.

An analysis of the distributions was carried out on individual factor combinations, revealing severe violations of the ANOVA assumptions for both dependent variables, namely the presence of outliers and deviations from normality. Outliers are marked with circles in Fig. 5.4, while deviations from normality, computed using the D'Agostino method [42], were discovered to be caused by skewness rather than kurtosis.

Therefore, separate two-sided Friedman's tests [79] were performed for both signed and variable errors considering each factor combination as one of twelve conditions of a combination factor (i.e. 3 configurations × 4 angles): no significant differences were detected for either dependent variables ($Q=7.21$, $p=0.78$ for signed error; $Q=12.1$, $p=0.36$ for variable error), meaning that no combination of the independent factors Configuration and Angle affected significantly the average signed error nor the variable error.

For the Rotation and Gesture configurations, the signed and variable errors were further assessed for each finger, however no effects due to a specific finger were found.

5.4.2 Finger deviation

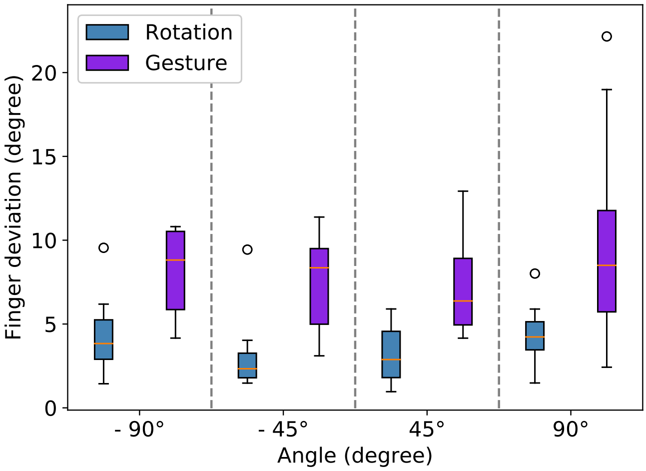


Figure 5.5: Boxplot of finger deviation in Rotation and Gesture configurations for each angle.

Figure 5.5 shows boxplots of finger deviation in Rotation and Gesture configurations for each angle. As expected, the finger deviation for Rotation was close to zero, resulting in a flooring effect affecting the distributions. Therefore, a non-parametric analysis was performed considering in turn each factor combinations as one of eight conditions of a combination factor (i.e. 2 configurations × 4 angles).

Friedman's tests revealed significant differences ($Q=63.85$, $p<0.01$), suggesting to perform multiple pairwise comparisons using the Wilcoxon Rank-sum test [103] with Bonferroni correction. Such comparisons, carried out by grouping all conditions sharing the same configuration, revealed that the finger deviation for Rotation was significantly smaller than that for Gesture (Z=-7.61, p<0.01). On the other hand, pairwise comparison of conditions sharing the same angle amplitude (i.e. 45° or 90°) did not reveal significant differences (Z=-2.15, p=0.09).

Pairwise comparison of conditions grouped by direction (i.e. clockwise and counterclockwise) did not reveal significant differences, too (Z=0.19, p=0.84).

5.4.3 Initial position

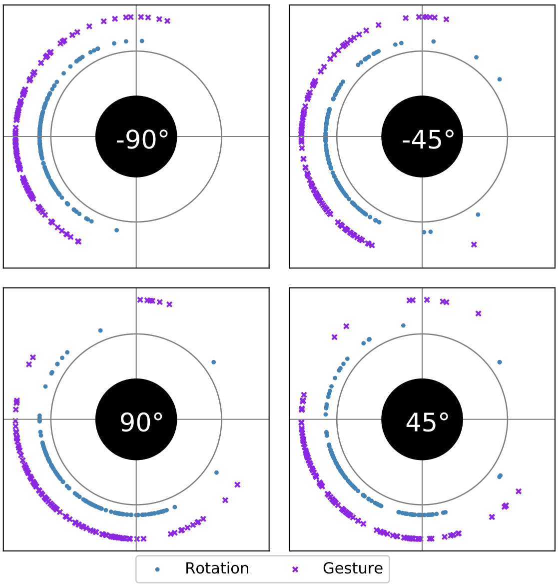


Figure 5.6: Initial position of the thumb for all the considered angles in Rotation and Gesture configurations. Data are presented from the user's perspective.

Figure 5.6 shows the initial positions of the thumb in each trial, grouped by Angle and Configuration. The original data were horizontally flipped to reflect the user's perspective, rather than that of the camera.

A wide distribution of the data is visible. Agreement between subjects was highest in factor combinations having angle equal to -90°.

Interestingly, the two clusters of outliers in Gesture configuration for angle equal to +45° and +90° were produced by only one participant. A general main effect seems to depend on the direction of rotation (i.e. clockwise for the plots below and counterclockwise for those above).

Again, an analysis of the distribution revealed violations of the ANOVA assumptions, therefore a non-parametric analysis was performed, in turn considering each factor combination as one of eight conditions of a combination factor (i.e. 2 configurations × 4 angles). A Friedman's test with Bonferroni-corrected $\alpha = 0.0125$ revealed significant differences (Q=259.8, p<0.01).

Multiple pairwise comparisons were performed using the Wilcoxon Rank-sum test with Bonferroni correction. Pairwise comparison of direction (clockwise, counter-clockwise) resulted in significant differences (Z=-15.31, p<0.01). On the other hand, pairwise comparisons carried out by grouping conditions with the same angle amplitude (i.e. 45° or 90°) revealed no significant differences (Z=-0.75, p=0.45), and so were pairwise comparisons grouping all conditions sharing the same configuration (Z=-0.34, p=0.73).

5.4.4 Questionnaires

Figures 5.7 and 5.8 report the participants' answers to the questionnaire. Figure 5.7a compares the 5-points Likert scale distributions of the perceived naturalness while performing the task in the Gesture, Rotation and Routine configurations. The distributions of Rotation and Routine are similar, whereas Gesture scored worse. The same trend is visible in Fig. 5.7b and 5.7c, respectively showing the related pleasantness and easiness to reach the target.

Figure 5.8 shows the number of fingers participants would used to turn the standard knob without experimental constraints: the majority of participants (62.5%) would used four fingers, three fingers were their second choice, while two and five fingers were preferred only once.

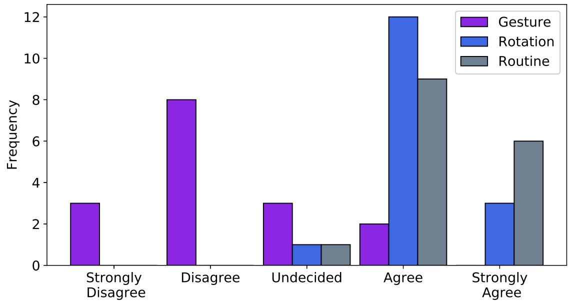
5.5 Discussion

The experiment assessed several objective and subjective issues having implications in the development of motionless knobs.

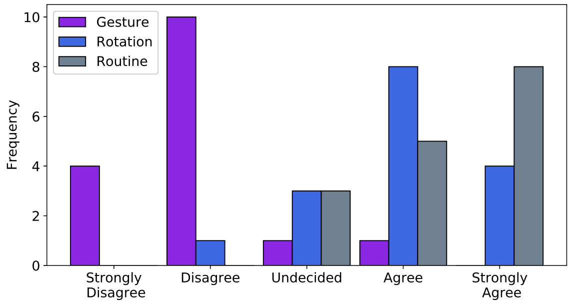
A general result, which is shared by related studies [130, 131], is the tendency to overshoot the rotation by about 10-15° independent of Configuration and Angle (see Fig. 5.4). However, signed and variable errors were not affected by any of the independent variables, suggesting stability of the task in terms of accuracy and precision. Although the lack of significant differences in the dependent variables might be also ascribed to the relatively low number of participants (N=16), the individual distributions of factor combinations did not show any particular trend or deviation (see Table 5.1), suggesting that similar results would be achieved also if involving a larger pool of participants. Indeed, our results are in accordance with previous larger scale studies on haptic rotation performance with rotary knobs [130, 131]. In what follows, the effects of each independent variable are separately discussed.

5.5.1 Effect of Configuration

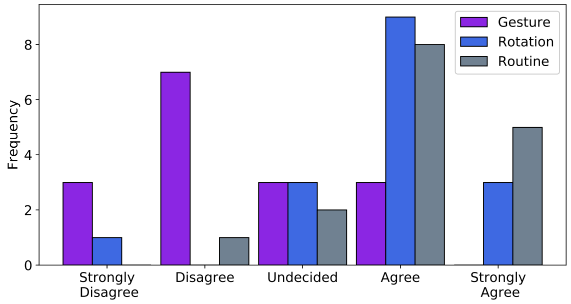
Concerning signed and variable errors, in both Rotation and Gesture configurations the finger tracker showed results comparable to the standard rotary knob (Routine), showing that nor the input device neither the configuration affected the rotation performance.



(a) Q: I feel the interaction is natural.



(b) Q: I feel the interaction is pleasant.



(c) Q: It is easy to reach the target position.

Figure 5.7: Responses to the questionnaire based on 5-point Likert scales: comparison among configurations on the perceived naturalness, pleasantness and easiness to reach the target.

This may suggest that participants assessed angles based on the rotation of the wrist rather than the fingers; in this way, regardless of whether the fingers are rotating a knob or sliding around it, the resulting gesture as well as proprioception are on average similarly effective.

Conversely, Configuration had an effect on the finger deviation measured by the

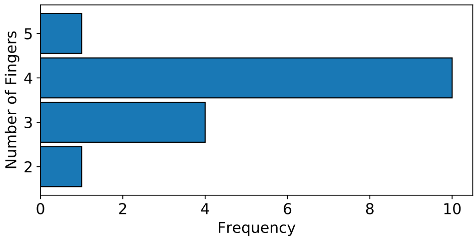


Figure 5.8: Number of fingers participants felt more comfortable to use to turn the knob.

finger tracker. This is hardly surprising, since in the Gesture configuration fingers can slide in different ways, depending on the hand pose and the frictional or adhesive forces arising from the contact interaction with the handle: as a result, whereas the fingers' average rotation does not change significantly, each finger follows its own trajectory, making it difficult to model a gestural recognition system, unless perhaps by constraining the number of fingers to two. As a final note, although ideally the finger deviation for Rotation should be null, deviations of up to 5° were measured (see Fig. 5.5). Such offsets actually depend on occasional tilting of the index and middle finger, which caused the illumination angle to change with consequent tracking of the fingers' apparent movement. However, this optical effect was common to both Rotation and Gesture, and furthermore smaller than the measured physical movement of the fingers. Finally, Configuration also did not affect the initial angular position, that is the initial hand pose.

5.5.2 Effect of Angle

Angle did not affect the signed and variable error magnitudes, meaning that such errors were relatively larger for smaller angles (i.e. 45°).

This suggests that accuracy and precision depend on a local process which is activated in the proximity of the target, rather than by the rotation angle.

Angle had no significant effect on finger deviation either: this in fact slightly increased proportionally to the angle amplitude, but differences were not significant. Thus, while prospectively modeling a gestural recognition system, finger deviation could be treated independently of the rotation angle.

Concerning the initial angular position, although participants were not instructed on how to grasp the handle, their initial hand pose depended on the target's direction. Conversely, the angle amplitude had no effect on it (see Fig. 5.6).

5.5.3 User preferences

The distribution of preferences for the Rotation and Routine configurations were similar. Based on this, it is possible to speculate that the finger tracker in the Rotation configuration was perceived as indistinguishable from the standard rotary knob. Despite the general agreement of the measured objective parameters among different configurations, participants rated Rotation and Routine positively, whereas in most cases they rated the Gesture configuration negatively. Such negative ratings may be partially ascribed to the fact that it was their first interaction with a motionless knob. Another factor that may have negatively influenced the rating is the unexpected friction experienced in the Gesture configuration: since the experiment took part during the summer season, warm temperatures and humidity resulted in non-negligible friction between the fingers and the metal handle, making it more difficult and unpleasant to perform the sliding task.

Finally, the distributions of ratings for the three questions were very similar; this suggests that a more pleasant and natural interaction was perceived to facilitate the task. However, this perceived difference was not confirmed by the objective measurements: the signed error in the Gesture configuration actually scored the lowest median (see Table 5.1).

5.6 Lesson learnt

This chapter presented a preliminary study to the design and development of a motionless knob. In this experiment, the user interaction with rotary and motionless knobs was assessed by measuring objective and subjective parameters.

From a technical perspective, the experiment revealed that the high variability of finger trajectories while sliding around a motionless knob may be the main hindrance to a precise gesture recognition, furthermore considering the amount of different gestures that users can perform without experimental constraints (e.g., variable number of contacting fingers, hand poses). Moreover, several other external factors (e.g., size of the hands, temperature, humidity) may interfere with the gesture tracking system. In light of the above issues, a tracking system based on capacitive sensors embracing a machine learning approach may be more robust and reliable for real-world applications, than a system based on the temporal and spatial modeling of the acquired signals.

Participants rated the interaction with rotary knobs as more pleasant and natural than that with a motionless knob. Interestingly though, while they claimed that it was easier to reach the target position using a regular knob, the experimental data showed that rotation performance (i.e. average accuracy and precision) was not affected by the type of the used knob. The lower score given to the Gesture configuration, perhaps was due to the increased friction, or to the larger finger deviation experienced in that configuration, reflecting the minor affordability of the motionless interaction.

Overall, the study suggests that, besides the technical issues related to gestures detection, considerable effort in terms of design and material selection will be required for a motionless knob to be enjoyed by users.

6

Non-a-knob: a motionless knob powered by artificial intelligence

In the light of the findings reported in the previous study, this chapter reports the design and validation of a motionless knob device. In particular, since knob rotations and rotation gestures resulted in similar accuracy and precision during the haptic rotation experiment, the motionless knob here presented is tested in both configurations: this will allow to investigate whether a large finger deviation of the sliding gestures affects the detection algorithm performance, as suggested in the previous chapter. The device is also evaluated regarding the quality of the interaction by offering the same questionnaire of the previous study to the participants. Finally, after the evaluation of the motionless knob, the last section of this chapter proposes a new concept design that integrates the motionless knob into a compact user interface providing multimodal feedback.

6.1 Hardware design

The design of the motionless knob started from the selection of the detection technology; given the promising results of capacitive sensing reported in [37, 208], several layouts of capacitive sensor pads, also called electrodes, were designed and tested in different sensing configurations. Among others, the PSoc4 demo board produced by Cypress [39] was selected and configured to measure the capacitance variations generated by finger proximity to a plurality of copper sensor plates arranged on the side of the motionless knob. The PSoc4 board was configured to read and process the sensors' signals, furthermore sending asynchronously their values through a virtual COM port using a serial protocol.

6.1.1 First prototype

The first prototype of the motionless knob (see Fig. 6.1) consisted of a 3D printed PLA cylinder having six rectangular copper sensing plates (i.e. s1,..,s6) interleaved by six ground plates (GND) connected together. All the copper plates were separated by small dielectric PLA protrusions toward the inside of the cylinder.

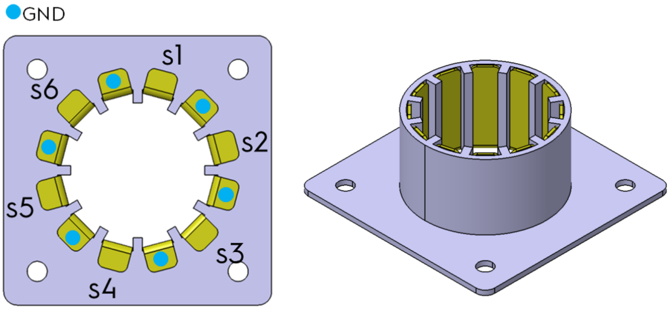


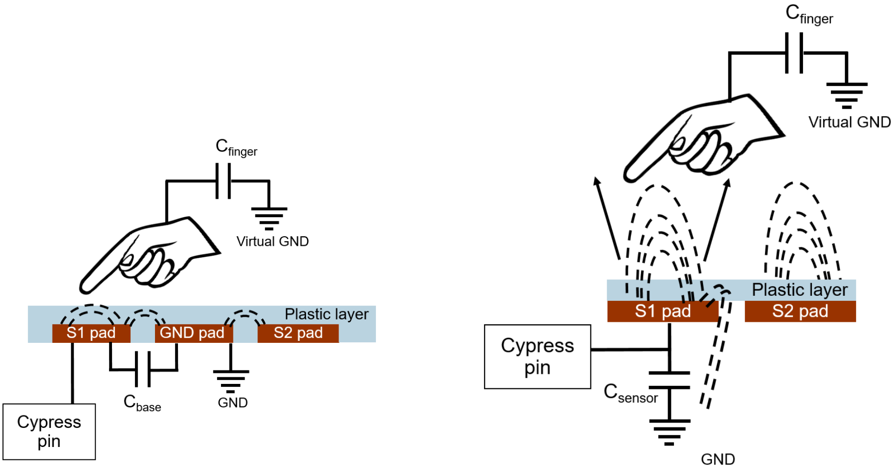
Figure 6.1: The model of the first prototype of motionless knob: a 3D printed PLA cylinder hosts six rectangular sensing plates (i.e. s1,..,s6) that are interleaved by six ground plates. All copper plates are separated by PLA protrusions.

On such prototype two capacitive sensing configurations were applied: the first attempt exploited mutual capacitance, which is commonly used in touchscreens (see Fig. 6.2a); in this configuration, each copper sensing plate (e.g. S_1 pad) is surrounded by ground plates (GND), separated by a dielectric material (e.g. plastic). Such configuration is equivalent to a capacitor (C_{base}) which generates a magnetic field (the dashed lines in Fig. 6.2a) when powered. When the finger approaches the dielectric plastic layer, it forms a second capacitor (C_{finger}) coupled with the sensing plate, changing the capacitance of the sensor. Within certain limits, the PSoc4 control board is able to measure the capacitance variation as defined during its parameters' setup. The capacitance variations are reported as counts, with large values corresponding to high capacitive loads.

Figure 6.3 reports the signals of the six sensing plates (s1,...,s6) shown in Fig. 6.1 being recorded when the index finger was sliding around the motionless knob prototype at constant speed:

- when finger positions correspond to the center of a sensing plate, the signal of such sensing plate reaches the maximum value resulting in a flat plateau (e.g., sensor s2 from 3.0 to 3.8 seconds).
- when finger positions correspond to the center of a ground plate (e.g, between 4.0 and 4.5 seconds), the value of the two adjacent sensing plates does not exceed 100 counts.
- when finger positions correspond to the small dielectric PLA protrusions, the value of the adjacent sensing plate drops to 10-20 counts
- when the finger position is far from a sensor plate, the value of such sensor plate is zero.

The signals of Fig. 6.3 were successfully used to accurately decode rotations across twelve positions, operated by a single finger sliding around the motionless knob. A similar model was successfully used to encode rotations operated by two fingers by



(a) Mutual capacitance principle: sensing plates (S1 pad, S2 pad) are surrounded by ground plates (GND pad), interleaved by a dielectric material (e.g. plastic) creating the equivalent of a capacitor (C_{base}).

(b) Self capacitance principle: each sensing plates (e.g. S1 pad) forms a capacitor (C_{sensor}) with the ground plane, which is the earth (GND).

Figure 6.2: Mutual and self capacitance configurations. In both configurations, the proximity of the finger generates the equivalent of a second capacitor plate (C_{finger})) modifying the magnetic field of the sensing plate (dashed lines) and thus changing the capacitance of the sensor.

imposing constraints on their mutual distance (e.g., finger positions must be specular on the knob circumference). However, it was not possible to model rotations with more than two fingers: in that configuration, many sensing plates were activated at the same time resulting in uninformative saturated signals; to this end, the mutual capacitance sensing design did not match the expected requirements for the capacitance variations measurements. Indeed, this sensing configuration is usually used to trigger on/off events (e.g., touchscreen taps) rather than precisely measure changes of capacitance. Moreover, the size of the copper sensing plates was constrained by the knob size, limiting the capacitance settings available on the PSoc4 board. Consequently, in order to obtain more informative signals the self capacitance sensing configuration was tested.

In the self capacitance configuration (see Fig. 6.2b), a sensing plate forms a capacitor with the ground plane, which in this case is the earth ground. Therefore, when the circuit is powered, the driving circuit performs a calibration and forms a broad magnetic field to the earth ground. When the user finger approaches the sensor pad, the finger capacitance takes over the earth ground and form the second capacitor plate (C_{finger}). Again, this change in the magnetic field changes the capacitance of the sensor that is detected by the PSoc4 board. Also in this configuration, the size and

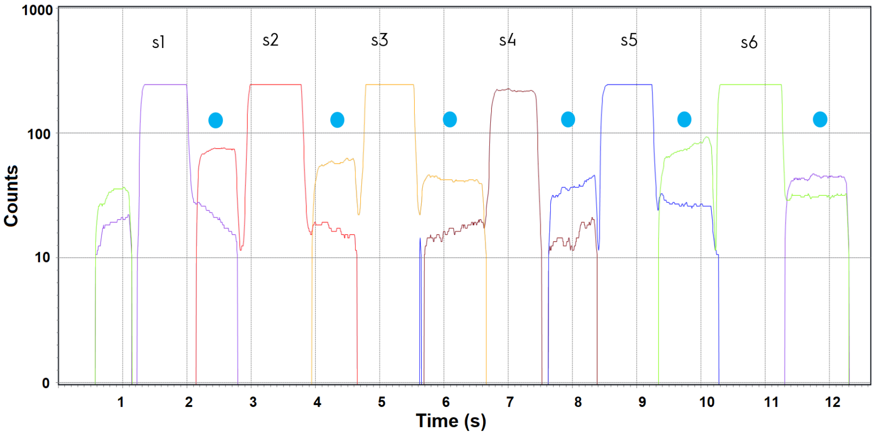


Figure 6.3: Temporal evolution of the signals encoded by the six copper plates in mutual capacitance configuration: the index finger is used to slide around the motionless knob at quasi-constant speed. The blue dots correspond to the positions of the ground copper plates

the form factor of the sensing plates, together with the settings of the Cypress PSoc4 board, determine the strength and the size of the magnetic field generated. However, even using the same sensing plates, in self capacitance configuration the board allows to reach great sensitivities, resulting in meaningful output values in a wider range of 0-65535 sensor counts.

6.1.2 Second prototype

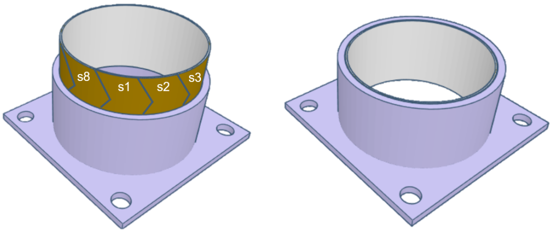


Figure 6.4: The model of the second prototype of motionless knob: a 3D printed PLA cylinder hosts a strip of copper tape where eight sensing plates (i.e. s1,..,s8) were engraved on the copper layer (left picture). The tape was attached to the internal side of the cylinder (right picture).

After the inspection of the signals recorded in the self capacitance configuration, a second prototype of the knob was developed removing the protrusions inside the cylinder (see Fig. 6.4). Moreover, the copper plates were substituted by a strip of copper tape, which was engraved with a *zigzag* pattern proposed in the guideline for capacitive sensor design [7].

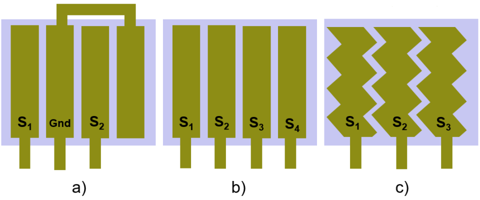


Figure 6.5: Three layouts of the copper plates designed for the motionless knob: in layout a) the plates follow the mutual capacitance design, in layout b) the same plates are configured for the self capacitance and, in layout c) new sensing plates with a more complex design are engraved on copper tape for the self capacitance.

Figure 6.5 summarizes the sensor plate layouts tested: the layout *a* was used for the mutual capacitance configuration, having sensing and ground plates interleaved; the layout *b* was used for the self capacitance having all the copper plates configured as sensing plates; finally, the layout *c* was used to improve the signals in self capacitance configuration by engraving the sensing plates from a thin copper tape. As shown in Fig. 6.6, which reports the profile of the signals recorded from a finger sliding around the motionless knob, the "zigzag" pattern used in layout *c* allows recording smooth changes in the signals, in particular when adjacent sensors are crossed by the finger during sliding.

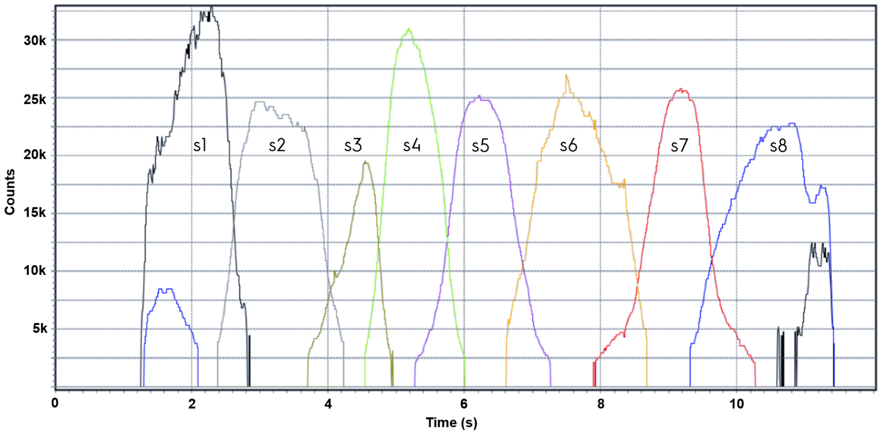


Figure 6.6: Temporal evolution of the signals encoded by the eight sensors in self capacitance configuration (layout c of Fig. 6.5) when the index finger slides around the motionless knob at quasi-constant speed.

6.2 Gesture detection

Once the highest quality of the signals was obtained by setting the PSoc4 board parameters (e.g. sensitivity of each sensor, control of the circuits' driver), a second gesture model was attempted. However, even with improved and more informative signals, a general model of rotation gestures detection based on signal variations turned out to be difficult to design and parameterize: indeed, the variability of the rotation gestures (e.g. number of fingers used, inter-distance between fingers, hand pose) together with the variability of anthropometric factors (e.g. hand size, finger length) resulted in hundreds of possible signal configurations. As an example, Fig. 6.7 reports the temporal evolution of the signals encoded by the eight sensors during a 90° clockwise rotation gesture, performed using 3 fingers.

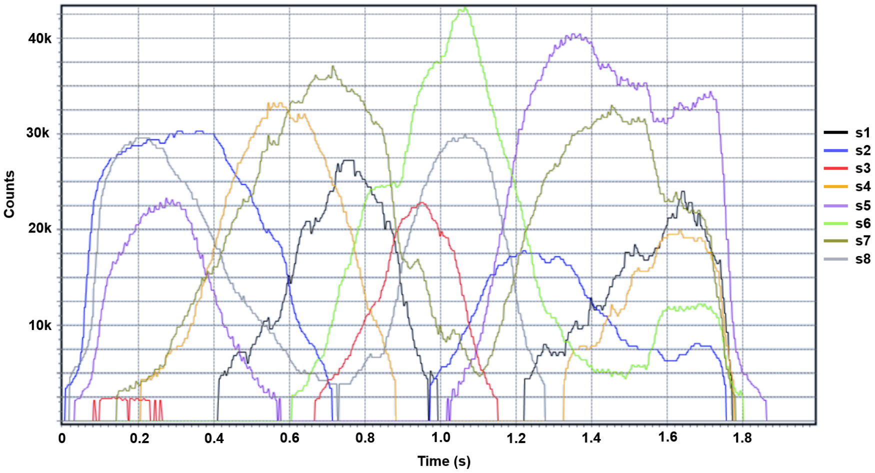


Figure 6.7: Temporal evolution of the signals encoded by the eight sensors during a 90° clockwise rotation gesture performed using 3 fingers.

For this reason, an approach based on machine learning (ML) was preferred to the design of multiple models to encode specific gestures or fingers configurations. In particular, an ordinary fully connected feedforward Neural-Network (NN) [71] was designed.

6.2.1 Dataset

Machine learning algorithms use huge tagged datasets to train and test their parameters. In the specific case of NN with supervised learning, once the developer has defined the structure of the NN, part of the dataset (i.e. the training set) is used to iteratively tune the *weights* associated to each neuron of the NN, whereas the remaining part of the dataset is used for test and validation of the NN. See [71] for a complete review on the design, training and test of feedforward NN.

Usually, the creation of reliable and complete datasets is one of the greatest challenges of a ML project. Indeed, the generation of a great amount of data, being representative of the overall features space used by the ML algorithm often requires long, complex and expensive data acquisitions campaigns, especially when labeled data are required. Luckily, in this specific case, the creation of a big dataset was straightforward. The data collection was carried out with 8 human subjects aged between 18 and 54 years (4 males, 4 females) who performed rotation gestures on the motionless knob prototype. Obviously, such subjects had different physiological hand characteristics. For the data collection, each subject was asked to perform unconstrained rotation gestures clockwise. This way many different hand poses, multiple starting and target angles were acquired. Subjects were also asked to perform rotation gestures using all fingers' possible combinations (i.e. from 1 to 5 fingers). In addition to the rotation gestures, subjects were asked to operate the motionless knob as a standard knob by performing clockwise rotations while grasping a thin plastic ring applied over the PLA cylinder (see Fig. 6.8). The same procedure was applied also for counterclockwise gestures and rotations.

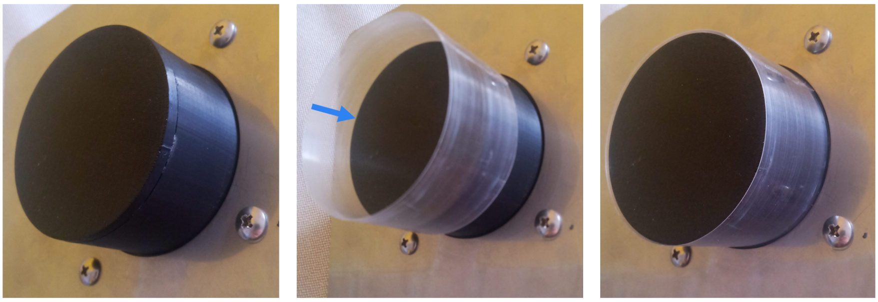


Figure 6.8: The prototype motionless knob used for the dataset creation (and the experiment reported in Sec. 6.3): whereas the gestures were performed directly on the black PLA cylinder (leftmost picture), rotations were performed grasping and rotating the thin plastic ring (center picture) completely inserted around the PLA cylinder (rightmost picture).

After the data processing described in Sec. 6.2.2, the data collection resulted in 32409 records, equally distributed among clockwise and counterclockwise rotations. Each data record consisted of an array containing the 8 processed signal values and a label corresponding to the rotation direction step (i.e. CW motion Step, CCW motion Step). Therefore, each rotation gesture generated several data records, all in the same motion direction.

Figure 6.9 reports the schematic of the control algorithm developed in Python 3.6, using Tensorflow 1.15 as back-end for the NN design. Each stage of the algorithm is reported in the following:

6.2.2 Input and data processing

The data processing stage used the eight capacitive signal values $S = (s_1,..,s_8)$ acquired by the PSoc4 board and sent through the Virtual COM to a Dell Precision 5520 PC.

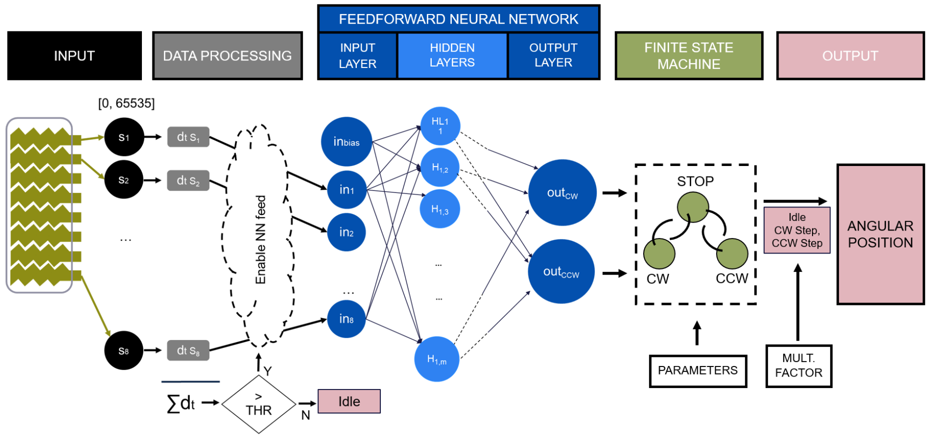


Figure 6.9: Schematic of the control algorithm: starting from the [INPUT] stage, the raw values of the sensors (s_1,..,s_8) are processed calculating the differences ($d_t s_1$, ..., $d_t s_8$) with the last values fed into the NN. At the [DATA PROCESSING] stage, only when the sum of the differences is greater than a defined threshold (THR), data are passed to the [FEEDFORWARD NEURAL NETWORK] stage. The input layer of NN has 9 neurons (i.e. one for each sensor plus the bias neuron); the NN has one or more hidden layers connected to the output layer having 2 neurons (OUT_{CW}, OUT_{CCW}). The output of the NN is used in the [FINITE-STATE MACHINE] stage that, considering user defined parameters, generates one of the following outputs: CW Step, CCW Step, Idle. Finally, the history of the output values multiplied by a user defined factor (MULT. FACTOR) determines the angular position of the knob.

Instead of considering the raw signal values in the range (0, 65535), for each sensor the algorithm calculated the difference $d_t s_n$ between the current value s_n(t) and its last reference value saved s_n(t-1), normalized in range (-1,+1).

The array of differences D = ($d_t s_1$, ..., $d_t s_8$) was used by the next algorithm stage, only when:

$$\sum_{i=1}^{8} abs(d_t s_i) > THR \tag{6.1}$$

where the threshold value (THR) was defined to reduce the noise result of micro-movements of the fingers. Each time the array D satisfied Eq. 6.1, the corresponding array S was saved as a reference value for subsequent inputs. Since the algorithm was designed to consider signal differences instead of the raw data, its results were largely independent of differences in signal magnitude due to hand size.

The data processing stage was common to both the acquisition algorithm used to create the dataset, and to the control algorithm: whereas the acquisition algorithm saved each entry of the array D in a CSV file, the control algorithm used the array D as the input of the NN.

6.2.3 Feedforward Neural Network

The structure of the NN was designed to be as simple and light as possible, in order to be easily coded into embedded systems. The NN had the following structure:

Input layer

The input layer has 9 input neurons: 8 neurons receive the sensors' differences and one neuron contains the bias fixed value +1, used to generalize the training of the NN. Since the NN stage is activated only when the threshold (THR) is exceeded, the sensitivity of the capacitive sensors must be calibrated to generate congruent signal variations during rotation gestures. However, by lowering the sensors' sensitivities, it is possible to introduce a sort of safety mechanisms: indeed, if the signals' variations generated by small hands are too low to trigger the rotation detection, this setting can be used, for instance, to avoid children to unintentionally operate the motionless knob.

Hidden layers

Usually, the most complex part in the design of NN is the definition of the hidden layers, since their functions, structure and connections determine how the input data are combined and transformed through the NN. Given the limited complexity of the task (i.e. to divide into 2 categories tuples of 9 values), this specific NN was expected to have a low number hidden layers, each containing a limited number of neurons.

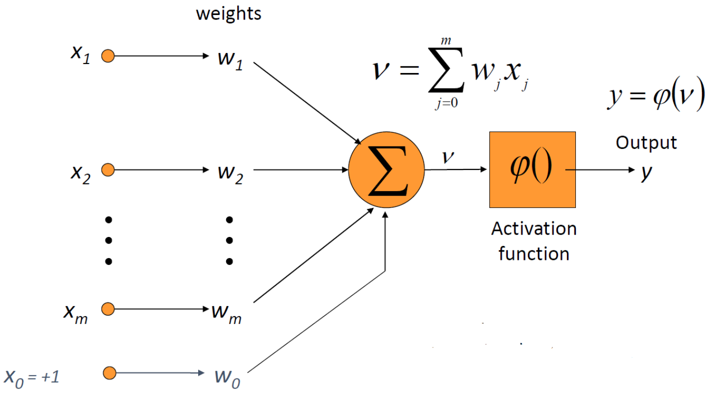


Figure 6.10: Schematic of the neuron model used in the first hidden layer: each element of the input layer (x_0, x_1, ..., x_n) is multiplied by a given weight (w_0, w_1, ..., w_n) and then summed by the neuron as the value v. Such value is used as the argument of a non-linear activation function $\varphi()$.

The k neurons belonging to the first hidden layer are designed following the general schematic reported in Fig. 6.10, where the input values (x_1,...,x_8) correspond to the eight sensors' differences and the input x_0 corresponds to the bias input. Each input value is multiplied by its corresponding weight (w_0,...,w_8) and summed into the variable v, used as the input parameters of the sigmoid activation function $y = \varphi(v)$. The neurons belonging to the (eventual) subsequent hidden layers are designed in the same fashion, using as inputs all the computed output values (y_0,...,y_k).

Output layer

All the neurons of the last hidden layer are fully connected to the output layer. The output layer has 2 neurons, one for each category identified (i.e. CW motion step, CCW motion step). Concerning the neurons of the output layer, all their inputs are multiplied for the corresponding weights and then summed in the variable v. Therefore, the category identified by the NN corresponds to the output neuron having the highest sum v. Moreover, the distance between the v values can be used as a metric of the prediction accuracy. Therefore, in addition to the category identified, also the two values of prediction accuracy are used as inputs of the finite-state machine stage.

6.2.4 Finite-state machine and Output

The finite-state machine stage was designed to encode three motion states: clockwise rotation, counterclockwise rotation and no rotation. The motion state of the machine changes with the NN output (i.e. category and accuracy) and the parameters set by the user.

At the output stage, the control algorithm can provide the following outputs based on the current machine motion state:

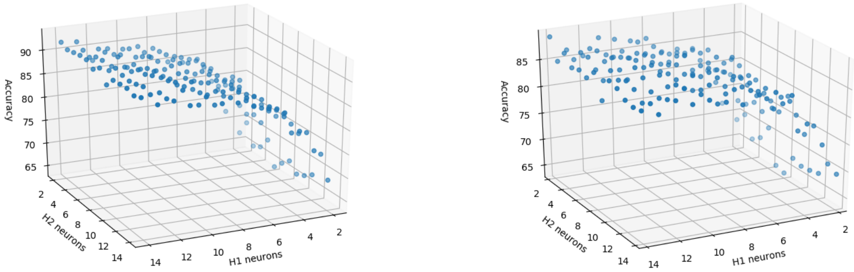
- **CW Step** if the NN infers with sufficient accuracy the CW category AND the current motion state is (*clockwise rotation* OR *no rotation*)
- **CCW Step** if the NN infers with sufficient accuracy the CWW category AND the current motion state is (*counterclockwise rotation* OR *no rotation*)
- **Idle** in all the other cases

Finally, the control algorithm encodes the absolute angular position of the motionless knob based on the history of the CW and CCW steps and a multiplication factor set by the user.

6.2.5 NN Training and performance

The NN was trained and evaluated by comparing multiple configurations. In particular, the final structure of the hidden layers was defined experimentally by testing the classification performance of NNs having a different number of layers and different number of neurons per layer. Each configuration of hidden layers was tested on the entire dataset (Sec. 6.2.1) by randomly assembling its samples into a training set (65%), a validation set (10%) and a test set (25%).

Figure 6.11 reports the classification accuracy achieved combining all the configurations of two fully connected hidden layers, both having a variable number of neurons ranging between 2 to 14. As reported for both the training set (Fig. 6.11a) and the test set (Fig. 6.11b), the increase in the number of neurons in the first layer is associated with improved accuracy; on the other hand, the increase in the number of neurons of the second layer has no effects on the accuracy. In the light of these experimental results the second hidden layer was removed, resulting also in a simpler single hidden layer NN.



(a) Training set (b) Test set

Figure 6.11: Classification performance on training and test dataset as a function of the number of neurons used for the first (H1) and the second (H2) hidden layers.

The performance of the single hidden layer configuration was further tested for an increased number of neurons and training epochs. Figure 6.12 reports the classification accuracy achieved with different numbers of neurons (8 to 39) and two training epochs (40 and 240). Although the dependence of accuracy on the training set is monotonic, the dependence on the test set stabilizes when more than 21 neurons are used. This means that the surplus neurons (i.e. from 22 to 39) were trained on specific features belonging only to the training set. Furthermore, an increased number of training epochs improved only the accuracy of the training set, too. In the light of such experimental results, the final NN configuration was designed as a single hidden layer NN having 21 neurons, trained for 40 epochs. This specific configuration resulted in a prediction accuracy of 91.4% concerning the training set and of 89.2% concerning the test set.

6.3 Experimental Validation

In addition to the measurements of the NN classification accuracy, an end-to-end validation of the device was performed by repeating the experiment on haptic rotations described in Chapter 5 using the motionless knob. The evaluation of the gesture detection algorithm was not obvious: whereas objective measurements without human participants would have been not realistic, in visual-haptic conditions the visual feedback would have probably biased the results with visual cues of finger position. Repeating the experiment of Chapter 5, instead, allowed us to compare the performance of the motionless knob with the objective data acquired by the optical finger tracker. Finally, this choice made it possible to compare the subjective evaluations

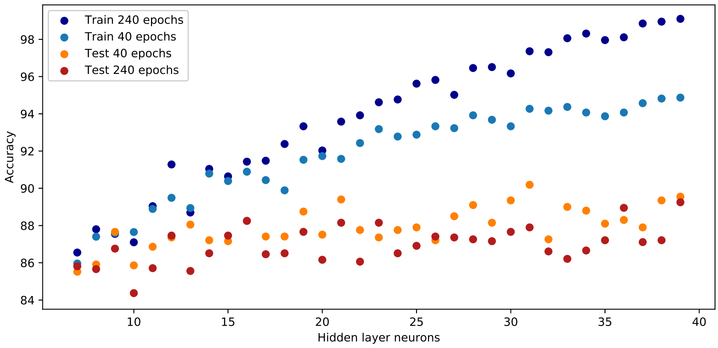


Figure 6.12: Classification performance as a function of the different number of neurons (from 8 to 39) and training epochs (i.e. 40 and 240)

collected through the questionnaires at the end of the experiments.

6.3.1 Participants

Twenty one subjects (17 males, 4 females) aged between 17 and 57 years (M=31.3, SD=10.4) took part in the experiment. All subjects were right-handed, and no one reported visual or sensorimotor impairments. They all participated on a voluntary basis and were not paid.

6.3.2 Experimental design and procedure

The new version of the experiment made use of the same materials and methods described in 5.3.2, except for the experimental apparatus: indeed, the motionless knob prototype was used instead of the finger tracker(see 5.2.1). Figure 6.13 shows a participant while performing the experiment with the new setup, having the standard knob and the motionless knob placed side-by-side.

This new experiment compared three different configurations, too:

- Gesture: participants performed a rotation gesture by sliding their fingers around the motionless knob (see Fig. 6.8 left).
- Rotation: participants rotated a thin plastic ring applied around the motionless knob (see Fig. 6.8 right).
- Routine: participants rotated the standard knob used in the previous experiment (see 5.2.2).

The factor Configuration (Rotation, Gesture and Routine) was crossed with the factor Angle (-90°, -45°, +45°, +90°) for a total of 3×4=12 factor combinations. Each combination was repeated 8 times, resulting in 96 trials per participant. The experiment was carried out in 3 blocks, one for each Configuration. Also in this experiment, half of the participants performed the Routine block first, and the other

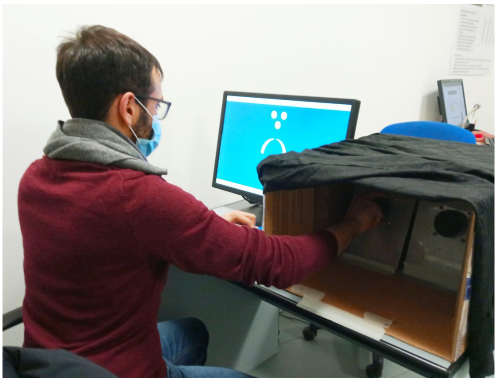


Figure 6.13: Participant grasping the standard knob while reading the rotation to be performed.

half performed it as last block; the Gesture and Rotation blocks were presented in random order. The 32 trials within each block were also randomized.

At the end of the experimental session, participants were asked to rate the naturalness and pleasantness of the interaction with the input devices, as well as the perceived ease of execution with each input device. Moreover, participants were asked about the number of fingers they would use to turn the standard rotary knob without experimental constraints.

6.3.3 Dependent variables

Each trial of the experiment produced a CSV file containing the angular positions measured using the motionless control algorithm or the encoder of the standard knob, depending on the Configuration. On such data, two dependent variables, i.e. signed error and variable error, were computed and analyzed.

In all configurations the signed error was computed as the difference between the recorded and target angle rotation; then, as in the previous experiment, the mean signed error was computed for each factor combination by averaging the respective signed errors made across eight repetitions. The variable error, instead, was computed as the standard deviation of the signed errors belonging to a factor combination.

6.3.4 Results

A statistical analysis was conducted to check whether the dependent variables (signed error and variable error) were significantly affected by the independent variables (i.e. factors Configuration and Angle). Separate analyses were carried out on the two dependent variables since only a moderate positive correlation was found (Pearson Index=0.22, $p<0.01$). A descriptive statistics of the participants' questionnaires is reported at the end of the Section. Analyses were carried out in Python 3.6 using the module *scipy.stats* 1.4.1.

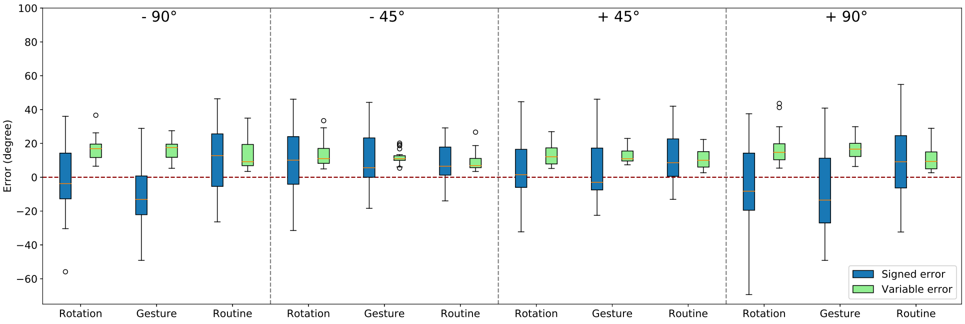


Figure 6.14: Boxplot of signed and variable errors for all factor combinations.

Variable and Signed error: descriptive analysis

Table 6.1: Signed and variable errors grouped by Configuration: each column reports means (M), medians (Mdn) and standard deviations (SD), standard Error (SE), lower 95% CI bound (LOW 95CI) and upper 95% CI bound (UP 95CI) for all angles.

	Routine		Rotation		Gesture	
	signed	variab.	signed	variab.	signed	variab.
M	10.4°	11.1°	2.3°	15.2°	-1.0°	14.5°
Mdn	9.0°	8.9°	1.1°	13.5°	-1.9°	12.9°
SD	±18.4°	±7.1°	±22.5°	±8.0°	±21.1°	±5.4°
SE	2.02°	0.78°	2.47°	0.88°	2.32°	0.59°
LOW 95CI	6.43°	9.60°	-2.55°	13.49°	-5.53°	13.36°
UP 95CI	14.35°	12.66°	7.16°	16.93°	3.57°	15.68°

Table 6.1 shows mean signed error and mean variable error distributions grouped by Configuration, whereas Fig. 6.14 reports the related boxplot for all factor combinations.

An inspection of Fig. 6.14 reveals that the mean signed errors, compared to the previous experimental results (see Fig. 5.4), are more equally distributed around zero. The moderate positive correlation existing between signed and variable errors implies that *accuracy* in task execution was paired with *precision*, and vice-versa. As shown in Table 6.1 and Fig. 6.15, standard deviations (SD) of the signed and variable errors are similar across configurations, reporting values close to the previous experiment (see Table 5.1); conversely, means (M) and medians (Mdn) seems affected by the Configuration factor. The inspection the signed error distributions, grouped by the factor Angle (see Fig. 6.16), revealed similar distribution for angles having the same magnitude (i.e. 45 or 90), independently from the direction (i.e. CW or CCW).

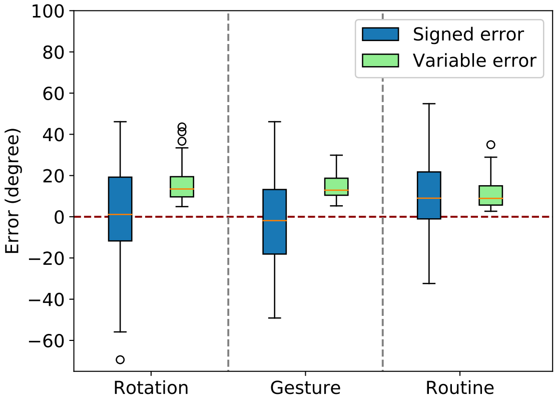


Figure 6.15: Boxplot of signed and variable errors grouped by Configuration.

For this reason, concerning the signed error statistical analysis, the independent variable Angle was furthermore separated into two factors: Angle Magnitude and Angle Direction.

Signed error

The analysis of the signed error univariate distributions reported no violations of the ANOVA assumptions. Therefore, a three-way ANOVA analysis was conducted to study the influence of the three independent variables (Configuration, Angle Magnitude, Angle Direction) on the signed error. Using a Greenhouse-Geisser correction for insphericity, the main effect for Configuration yielded $\mathrm{F}(1.77, 35.3) = 4.03$, $\mathrm{p} < .05$, indicating a significant difference between Routine ($\mathrm{M} = 10.4$, $\mathrm{SE} = 2.0$), Rotation ($\mathrm{M} = 2.3$, $\mathrm{SE} = 2.4$) and Gesture ($\mathrm{M} = -1.0$, $\mathrm{SE} = 2.3$). The main effect for Angle Magnitude yielded $\mathrm{F}(1, 20) = 22.8$, $\mathrm{p} < .01$, indicating a significant difference between 45° ($\mathrm{M} = 8.4$, $\mathrm{SE} = 1.54$) and 90° ($\mathrm{M} = -0.6$, $\mathrm{SE} = 2.1$). The main effect for Angle Direction, instead, was not significant $\mathrm{F}(1, 20) = 0.8$, $\mathrm{p} > .05$. Finally, the only significant interaction effect was found between the factors Configuration and Angle Magnitude, yielding $\mathrm{F}(1.67, 33.5) = 11.49$, $\mathrm{p} < .01$.

Variable error

The analysis of the variable error univariate distributions revealed several deviations from normality [42] and presence of outliers.

Therefore, two-sided Friedman's tests [79] were performed for both the independent variables Configuration and Angle. In order to keep type I error under control on such non-parametric analyses, standard confidence thresholds $\alpha = 0.05$ were Bonferroni-corrected ($\alpha/2 = 0.025$)[23, 203].

Friedman's tests on Configuration revealed significant differences (Q=26.64, p<0.01), suggesting to perform multiple pairwise comparisons using the Wilcoxon Rank-sum

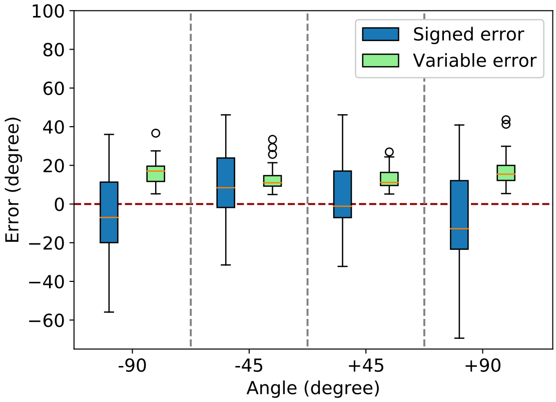


Figure 6.16: Boxplot of signed and variable errors grouped by Angle.

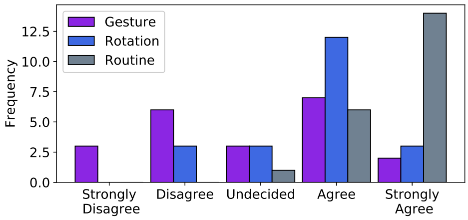
test [103] with Bonferroni correction. Such comparisons revealed that the variable error in Routine configuration was significantly smaller compared to both the Gesture (Z=-4.34, p<0.01) and the Rotation (Z=-3.99, p<0.01) configurations. On the other hand, the pairwise comparison between Rotation and Gesture yielded no significant differences (Z=-0.2, p>0.05).

Friedman's tests on Angle revealed significant differences, too (Q=21.76, p<0.01). Pairwise comparisons carried out grouping the angles with the same magnitude revealed that the variable error of 45° haptic rotations was significantly smaller than if equal to 90° (Z=-3.0, p<0.01). On the other hand, the pairwise comparison carried out grouping the angles with the same direction found no significant differences between clockwise and counterclockwise directions (Z=-0.3, p>0.05).

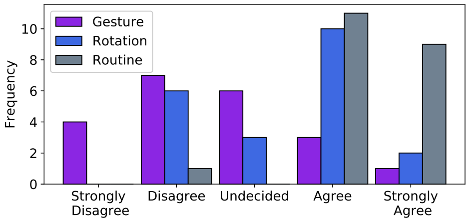
Questionnaires

Figures 6.17 and 6.18 report the participants' answers to the questionnaire. Figure 6.17a compares the 5-points Likert scale distributions of the perceived naturalness while performing the task in the Gesture, Rotation and Routine configurations. The three distributions are quite different: the Routine configuration received lumped high scores, the Rotation received more distributed scores, while the Gesture configuration received evaluations along the entire scale. Similar distributions are visible in Fig. 6.17b and 6.17c, respectively showing the related pleasantness and easiness to reach the target.

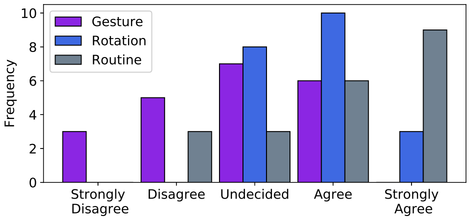
Figure 6.18 shows the number of fingers participants would used to turn the standard knob without experimental constraints: half of the participants (52%) would used four fingers, three fingers were their second choice, while five fingers were preferred only by two participants. None of the participants chose two fingers.



(a) Q: I feel the interaction is natural.



(b) Q: I feel the interaction is pleasant.



(c) Q: It is easy to reach the target position.

Figure 6.17: Responses to the questionnaire based on 5-point Likert scales: comparison among configurations on the perceived naturalness, pleasantness and easiness to reach the target.

6.3.5 Discussion

In the following, the results of the experiment are discussed and then compared to the findings presented in Chapter 5: in fact, although the results were generally improved in terms of accuracy when the motionless knob was used, these results differed from the baseline experiment detailed in previous chapter.

Effects of Configuration

In the experiment the Configuration factor played an effect on both the dependent variables. In particular, in both cases significant differences were found between the control condition (i.e. the Routine configuration) and the configurations that made used of the motionless knob (i.e. Rotation and Gesture).

Considering the Routine configuration, distributions of both dependent variables

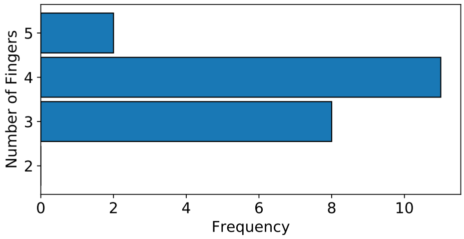


Figure 6.18: Number of fingers participants felt more comfortable to use to turn the knob.

are quite similar to the distributions found in the previous experiment: this is not surprising since such configuration was identical in the two experiments. However, whereas the 95% confidence intervals (CI) of variable error perfectly overlap, the mean value of the signed error is slightly different (see Tables 6.1 and 5.1). To this end, the limited number of participants involved in the experiments may partly explain such differences.

In this new experiment, both the Rotation and the Gesture configurations showed significantly lower signed errors compared to the Routine configuration, reaching mean signed errors close to zero. However, given the findings on the overshoot trend highlighted by the previous experiment and in other literature results [130, 131], the smaller values identified in the Rotation and Gesture configurations must be ascribed to some adjustment introduced by the control algorithm of the motionless knob. Conversely, the variable error equal to both Rotation and Gesture configurations in the new experiment showed significantly higher values, furthermore showing values close to each other (i.e. the means are 15.2° for the Rotation and 14.5° for the Gesture configuration, respectively). Therefore, the haptic rotations recorded by the motionless knob have signed errors that are lower than expected and variable errors that are greater that expected: this means that the control algorithm probably fails to encode some parts of the rotation gestures. However, since standard deviations in the two experiments are very close to each other concerning both signed and variable errors (see Tables 6.1 and 5.1), the mis-detections of the algorithm are not related to specific subjects.

Finally, since no significant differences were found between Rotation and Gesture configurations concerning both signed and variable, this means that the control algorithm of the motionless knob was able to detect rotations and gestures in the same way. In conclusion, the higher finger deviation found during the previous experiment concerning the Gesture configuration (see 5.4.2) was incorporated in the knowledge base of the NN.

Effects of Angle

The magnitude of the Angle had an effect on both dependent variables, too. Concerning the signed error, small rotation angles (i.e. +45°,-45°) gave rise to significant overshoot (mean 8°) compared to large rotations (i.e. +90°,-90°), that were associated with errors close to zero. Conversely, in the previous experiment signed errors were found to be as independent from the angle magnitude, showing mean overshoots of about 14°. Concerning the variable error, smaller angle magnitudes resulted in more precise haptic rotations (M=12°, SD=5.8) than larger angle magnitudes (M=15°, SD=7.8). In the previous experiment, instead, variable errors were slightly smaller (M=11°), furthermore independent of the magnitude of the angle.

Taken together, these results confirm that the control algorithm fails to encode some parts of the rotation gestures: in fact, whereas short rotations resulted in overshoot values close to the previous experiment, larger rotations showed an increased accuracy, previously not observed. Although no results on the effects played by the angle magnitude on haptic rotation were found in the literature, the results of our previous experiment were considered reliable, since data were acquired with an optical system and carefully checked. Finally, since the precision decreased with the angle magnitude, this means that the probability to miss some rotation steps is proportional the rotation magnitude.

The direction of the haptic rotations, instead, had no effect on the measured dependent variables. This result, common to the previous experiment, shows that the control algorithm of the motionless knob was adequately trained in both directions.

Interaction effects

The statistical analysis on the signed error found also a significant interaction effect between the Configuration and the Angle magnitude. The analysis of the estimated marginal means shows that the Angle magnitude had an effect mainly on the configurations making use of the motionless knob (i.e. Gesture and Rotation). Even more interestingly, such effect brought to notable undershoots when 90° rotations were performed on the motionless knob (Gesture=-8.5°, Rotation=-3.2°), resulting instead in overshoot with the standard knob (Routine=9.9°).

User preferences

The questionnaires reported slightly improved results compared to the previous experiment, in particular regarding the Gesture configuration. As expected, the evaluations on the standard knob in both experiments had similar distributions concerning all the questions.

Concerning the Rotation configuration, participants rated the interaction with the motionless knob as natural as the interaction with the finger tracker, thus slightly lower than the interaction with the standard knob. On the other hand, the perceived naturalness in the Gesture configuration sometimes scored better than the previous experiment, resulting in wide distributed evaluations. This difference can be explained by the different material used: the PLA, in fact, gives rise to less friction during the fingers' sliding compared to aluminium used by the finger tracker.

The same effect involved also the pleasantness of the interaction, that resulted slightly improved for the Gesture configuration. On the other hand, with the Rotation configuration results were worse than in the previous experiment. The reason of such evaluations can be ascribed to the compliance and the friction of the thin plastic ring used to simulate the rotation: harder materials with less friction probably will help restore evaluations similar to the standard knob.

Also the perceived easiness in reaching the target proved better in the Gesture configuration: the reduced friction exposed by the PLA cylinder probably improved the perceived proprioception during the rotation gestures. Instead, the evaluations on the Rotation configuration remained stable between the two experiments.

Finally, also in this case participants would have used four fingers to grasp the knob without experimental constraints. However, the number of user that would have used three fingers improved.

6.3.6 Future work

The results of the experiment showed that the control algorithm of the motionless knob introduces systematic errors, proportional to the magnitude of the angle, but independently from subjects. The reason behind such bias can be found in the finite-state machine output procedure: in fact, once determined the rotation direction, the control algorithm quantifies each detected rotation in 3°, independently from the current angular position and the past history. Such conversion factor, determined during informal trials accomplished in visual-haptic conditions, was chosen as a fair starting value for the experiment. Therefore, starting from the data acquired in this experiment it will be possible to model improved strategies having, for instance, the conversion factor depending on the angular distance covered during rotation gestures.

6.4 Full concept design

In its whole concept design, the motionless knob was imagined as the part of a more complex user interface. Indeed, the currently empty circular region and the cylindrical box behind it allow to host further interface elements as well as the hardware components needed to enable them. The complete interface layout consists of a unique object featuring multiple elements:

- a motionless knob, optionally equipped with a plastic ring providing standard rotational input
- a force-sensing visual touchscreen placed on the circular region
- an audio-tactile actuator behind the screen

Following this concept design, the frontal panel of the motionless knob, for instance, can embed the haptic touchscreen previously described and evaluated in Chapter 2. In the intended interaction design, the haptic touchscreen interface is used in combination with the motionless knob to perform few fundamental everyday operation that may require eyes-free interactions (e.g. one to four program sections, power increase/decrease, manifolds control). Conversely, the standard visual-based interaction is left available for all the other functions (e.g. setup of parameters, service).

Figure 6.19 shows how this layout would appear: a refined version of the haptic touchscreen described in Sec. 2.1, having a round touchscreen panel, is coupled with the second prototype of the motionless knob (Sec. 6.1.2) by means of a rubber o-ring. The function of the o-ring is twofold: whereas, on the one hand, prevents from any dirt or liquid intake, on the other hand, it allows the touchscreen panel to indent proportionally to the normal pressing force applied. This way, the load-cell mounted on the back of the touchscreen can precisely track the normal force applied by the fingers on the device. The audio-tactile feedback is generated by means of an actuator attached, for instance, between the load-cell and the touchscreen panel.

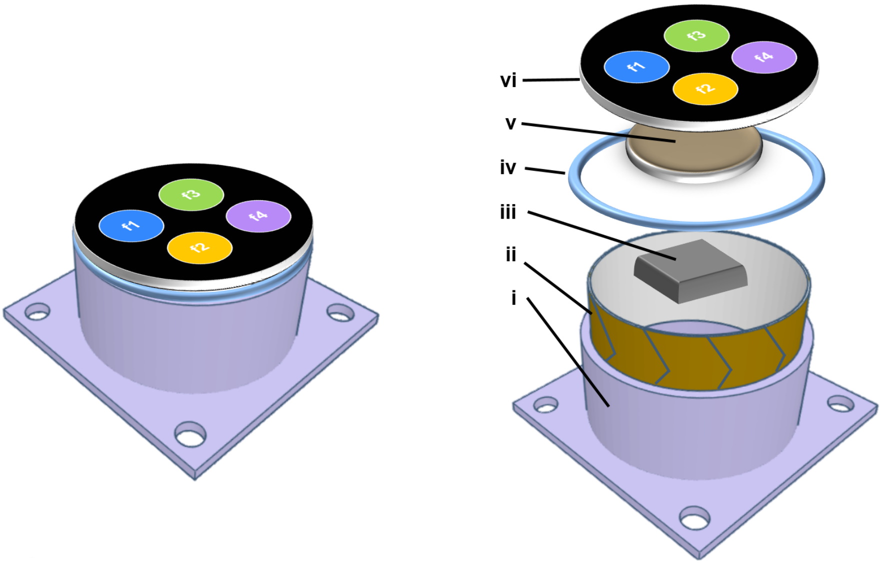
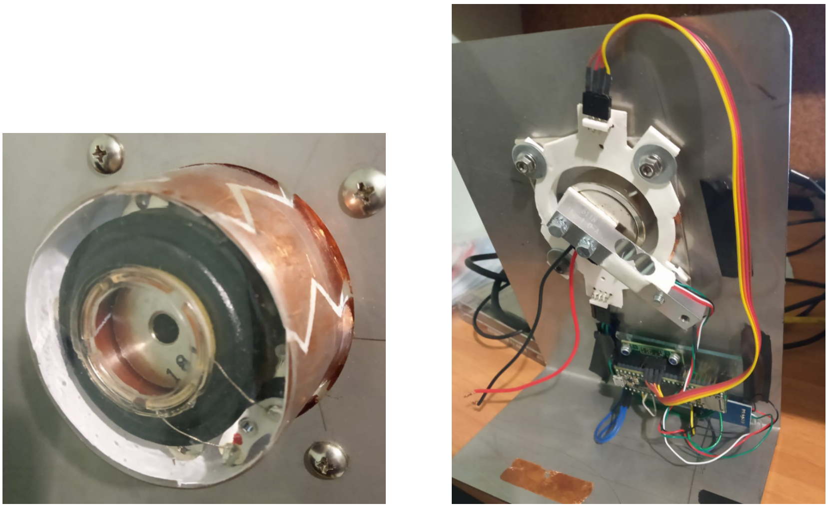


Figure 6.19: The model of the final concept design of the motionless knob. The leftmost picture shows the device as it appears to the user. The rightmost picture shows the schematic of the hardware used: a 3D printed PLA cylinder (i) hosts a strip of copper tape (ii). Inside the cylinder, a load-cell (iii) tracks the normal force applied to the touchscreen (vi) by the finger. The touchscreen is coupled with the cylinder by means of a rubber o-ring (iv), allowing its indentation. An audio-tactile actuator (v) attached between the load-cell and the touchscreen panel provides rich haptic feedback during rotation gestures and finger press.

6.4.1 An intermediate step

The design of such a concept requires to carefully study the user interaction at incremental stages. For this reason, the prototype shown in Fig. 6.20 was designed and assembled. The aim of this prototype was to improve the capacitive sensors sensitivity and to provide a first informal assessment of the vibrations' propagation through the



(a) Front view: a transparent plastic sealed cylinder host a strip of copper tape engraved with eight sensors and a Dayton DAEX25SHF-4 exciter attached to the front side.

(b) Rear view: a load-cell is attached to the back of the exciter, measuring the normal force applied to the motionless knob frontal panel; the electronics (i.e. Teensy 3.6 board, DSP module and load-cell driver) is attached at the right-bottom part of the device

Figure 6.20: An intermediate prototype of the concept design of the motionless knob: the prototype is able to produce rich haptic feedback during rotation gestures and finger pressing thanks to a powerful exciter. A Teensy 3.6 board encodes the eight capacitive signals of the copper strip and the signal of a load-cell; the board processes all the input signals and generates the output audio-haptic signal through a dedicated DSP module.

cylinder. As it can be seen in Fig 6.20a, the PLA 3D printed cylinder has been substituted with a thermoformed thin plastic handle: this way, the sensitivity of capacitive sensors is improved, whereas the friction during rotation gestures is further reduced. In order to generate powerful vibrations in range 30-500Hz, as well as audio signals up to 16000Hz, a 20w Dayton DAEX25SHF-4 exciter has been attached between the cylinder frontal panel and a load-cell (see Fig 6.20b); the load-cell is used to measure the normal force applied by the finger on the front panel of the cylinder, when the latter features a button. Concerning the electronics, in this prototype a compact Teensy 3.6 board is used to process the capacitive sensors' data and the load-cell signal. The same board is used in conjunction with a DSP module to generate and manipulate the audio-haptic output signal.

Although, in the current stage of research, such prototype has been informally tested for both gesture recognition and haptic feedback rendering, further developments are required before starting any experimental phase. In particular, the next development steps will be:

- the design of the mechanical model of the device
- the study of the propagation of vibrations in the cylinder structure
- the improvement of the mechanical coupling between the actuator, the motionless knob and the metal frame based on numerical simulations
- the design of an experiment to evaluate different haptic feedback delivered during rotation gestures

Only by carrying out all these development steps and the experimental evaluations it will be possible to integrate the touchscreen interface described in Chapter 2 and to finally evaluate the user interaction with this new compact user interface, which collects many findings of this research.

III

Tools

7

The Bogus Finger

This part of the thesis describes hardware and software tools developed to support this PhD research. In particular, the tool presented in this chapter required several months of development and validations carried out in collaboration with the Institute for Computer Music and Sound Technologies (ICST) of ZHDK (Zurich). The target of the project was the development of a DIY robotic tool able to simulate the mechanical response of the human finger to vibrations. As shown in Chapter 3, the simulation of the characteristics of the human finger is not trivial, since its mechanical characteristics change dynamically with the normal force exerted: after the initial contact, the finger compression increases the skin-surface contact area, the stiffness and the damping of the fingertip [99, 200], affecting also the friction coefficient with the surface [218]. Therefore, the mechanical impedance of the finger changes accordingly.

As shown in Chapter 2, the dynamic of finger-pulp compression can be exploited to produce tactile illusions through the generation of tailored stimuli bounded to the force level. However, without a precise mechanical model of the haptic device, the stimuli design is a time consuming task, always requiring validations with human subjects. To this end, several issues arise when designing active touch experiments, particularly concerning the control over the subjects' gesture (e.g., repeatability) [161], and the measurement accuracy when manipulable apparatuses are employed.

To overcome such issues, two solutions are viable: since the mechanical impedance of the human finger is well known and modeled [122], one can build a mechanical model of the haptic device and simulate the effects of the finger being in contact with it; on the other hand, by having a robotic tool simulating the human finger, it will be possible to characterize the haptic devices without modeling their structure and components.

Given the increasing number of haptic devices becoming available on the market, the second solution guarantees better results, allowing to compare multiple devices with a common tool independently from their mechanical complexity. However, such artificial tool must provide force control accuracy and precision together with a reliable simulation of the human finger mechanical impedance, concerning multiple force levels and different directions.

Thanks to the ICST, which provided facilities and materials, the Bogus Finger was developed: a robotic tool for the simulation of quasi-static finger-pressing forces of various magnitudes, suitable for interaction with both stationary and vibrating surfaces.

The finger press in static conditions has been studied by measuring on human subjects stiffness, contact area, mean pressure with different levels of force and velocities [181, 200, 99]. Such data has been used to create and validate physical finger-analog implementations [102, 237]. Among others, Friesen *et al.* compared different types of complex artificial fingers consisting of bone, tissue, skin, and outer skin layers analogs [80], while Controzzi *et al.* [35] built a bio-inspiered artificial finger described by a Finite Element (FE) model. Beside the excellence of these artificial fingers, their dynamic response is usually not considered in the analysis. By contrast, the device described in this paper targets mainly the issue of reproducing systematic finger pressing force simulations in dynamic conditions.

The dynamic behavior of the finger can be fully characterized by knowing the mechanical impedance it exposes at the contact point with a surface. The impedance of the finger depends on its mass, stiffness and damping and it is furthermore dependent on the finger-pulp compression [34]; for instance, the compression increases the stiffness [200] while the mass exposes quasi-constant values for increasing press force levels [97, 239].

In general, the impedance depends on the considered motion direction, being different in the tangential or in the normal direction [61]. Concerning the frequency, instead, fingertips behave elastically up to about 100 Hz, whereas the damping effect dominates the response between 100 Hz and 1 kHz. Given the small mass exposed by the finger, the inertial contributions can be neglected up to 500 Hz [239].

In the light of such considerations, the fingertip can be modeled as a viscoelastic non-linear spring-mass-damper lumped parameter system. The model can be arbitrarily complex: for instance, it can consider the finger as a single mass connected to the external world through a spring-damper connection, or it can represent the contributions of the different elements (e.g., skin, bones, tissues and joints). Several finger models with variable complexity are described and discussed in [101]. To this end, the network model proposed by Kern *et al.* [122] considers two main masses, representing the finger and the hand/arm interconnected by three spring-damper connections.

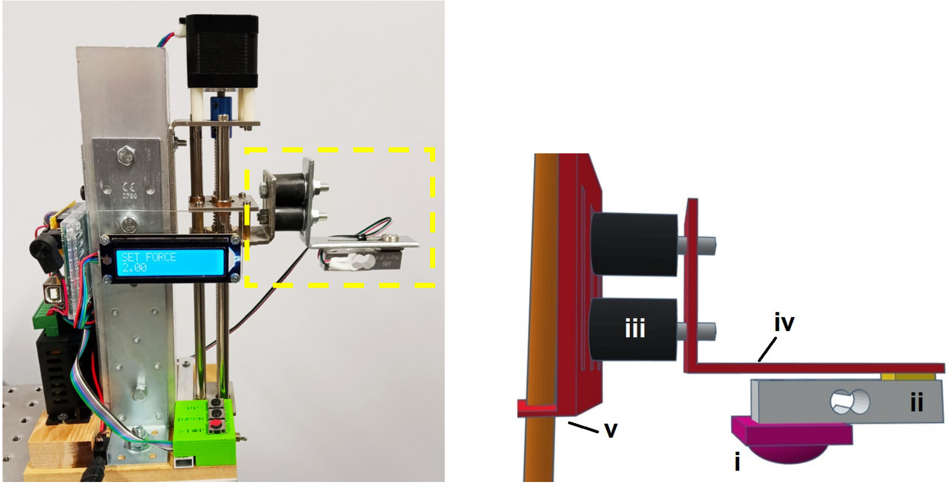
Once defined a reasonable mechanical model, the major issue is the accurate estimation of the parameters, especially considering that in systems like the human finger the elements can not be evaluated separately; for this reason, Kern *et al.* [122] used a least-mean-square technique to fit their model parameters starting form impedance measurements.

The characterization of a mechanical device, like the tool here presented (i.e. a series of connected elements), is usually more precise and straightforward; the masses can be detached and weighed separately, while the elastic coefficients can be obtained by measuring the relation between force and displacement at each joint. The evaluation of the damping, instead, requires dynamic conditions: in fact, the procedures to estimate the damping require the analysis of time or frequency domain responses to force input signals [21, 93].

Given the DIY nature of the project, an easy-to-implement and inexpensive tool was built and made available as an open-source project, furthermore characterized with procedures making use of commercial available tools, thus easily reproducible. In this chapter, a general model of the contact interaction is provided, accompanied by a procedure to estimate the coefficients of the model parameters, providing insights to converge to the mechanical impedance of the human finger. In addition to the technical implementation, the model and the characterization, it is reported an end-to-end validation where vibration measurements accomplished with human subjects are compared to the measurements realized using the Bogus Finger.

In this project, my personal contributions were related to the device hardware/-software design, to its characterization, and to its experimental validation. Dr. Mattia Dal Borgo and Dr. Eleonora Pippia developed the mathematical model of the device. Dr. Stefano Papetti and Prof. Federico Fontana coordinated and supervised the project.

7.1 Device design



(a) Full view with end-effector surrounded by a yellow dashed line.

(b) End-effector schematic: hemispheric silicone layer (i), load-cell force sensor (ii), rubber shock-absorbers (iii), angle metal bracket (iv), vertically sliding metal plate (v).

Figure 7.1: The Bogus Finger

Figure 7.1a shows a working prototype of the Bogus Finger realized by assembling off-the-shelf components and custom-designed parts. The device is released as an open-source project (CC BY-NC 4.0) documented by a public repository linked to GitHub.[1] The repository stores DIY instructions, mechanical and electronic speci-

---

[1] `https://github.com/yuridepra88/Bogus-Finger`

fications and schematics, 3D models, Arduino code and Python script examples for remote control.

7.1.1 Hardware design

The vertical displacement of the Bogus Finger's end-effector is operated by a slide stroke linear motion actuator: a 250 × 50 × 50 mm vertical metal profile holds the actuator, whose motor (NEMA17 42 mm stepper motor, torque 4.5 kg·cm) is controlled by a TB6600 driver connected to an Arduino Mega 2560 microcontroller board. The driver also limits the current provided to the motor, in this way protecting it, and can increase its spatial accuracy by subdividing the motor's step into up to 32 sub-steps (maximum resolution 0.6 µm). The force signal measured by a load-cell mounted on the end-effector is fed back to the microcontroller, allowing to reach and hold stable target forces over time.

The end-effector was designed to model a human finger pressing down vertically. Its components were selected among off-the-shelf material, aiming at matching the mechanical properties of the finger, as described in the literature [41]. Figure 7.1b shows a schematic of the end-effector, whose main components are:

i) A hemispheric silicone layer (radius 10 mm, thickness 6 mm) with squared base (side 24 mm, thickness 4 mm) simulates the viscoelastic properties of the finger. The choice of silicone type, mass and shape has great impact on the exposed characteristics of the device such as its stiffness, damping and inertia. The current prototype mounts soft silicone (Silastic 3481) having mass 4 g, young modulus 0.93 MPa and shore-A hardness 25. In Sec. 7.3 and 7.5, a comparison is reported with a harder silicone (Sylgard 184) having young modulus 1.45 MPa and shore-A hardness 40.

ii) A CZL635 load-cell monitors the exerted pressing force. The analog force signal is processed by a INA125P amplifier and sampled with 10-bit resolution by the Arduino ADC converter. The amplifier gain was set to read force values in the 0-20 N range with 0.1 N resolution. Although the load-cell can read values up to 50 N, greater values were considered outside the scope of our application.

iii) A pair of rubber shock-absorbers connect the end-effector to the linear motion actuator, preventing external vibration noise from reaching the accelerometer during measurements.

The structure chosen for the end-effector is fundamental to characterize the components of the mechanical system, as reported in Sec. 7.2. Three buttons, labeled *Up*, *Down*, *Stop/Function*, offer basic on-board controls, while a switch enables/disables the motor (e.g., once a target force is reached) making the device completely silent and vibration-free. A 16 × 2 LCD display connected to the I2c bus of the Arduino provides various information to the user.

Mechanical supports and electronics lay on a thick wooden board. In our test setup, the device was fixed to a vibration-isolation table (CleanBench TMC).

7.1.2 Force calibration

The load-cell was calibrated by measuring the forces exerted by the end-effector on a Kern 440-47N digital scale: the output voltage of the load-cell was associated to the weight measured by the digital scale. The interpolation of multiple measurements resulted in two different model fits: due to the nonlinear behavior of the load-cell for values between 0 and 3 N, a 3rd-order polynomial was fitted in this range, whereas a linear model was adopted for higher forces.

Once a target force is reached by pressing against a surface, negative drifting may take place over time due to mechanical backlashes, proportionally to the magnitude of the applied force. To counteract such effect, the force-control algorithm overshoots the desired force by about 5% and then adjusts the end-effector's displacement until a stable force is reached.

7.1.3 Controls

The Arduino board processes both on-board and remote commands. A reduced set of commonly used functions is available on-board, operated by the device's physical buttons: i) read the current force at the end-effector; ii) set a target force level to be reached and held; iii) freely move the end-effector; iv) set the home/zero position.

Remote control is provided through the Arduino USB serial connection. The on-board functions are also made directly available through remote commands. Communication with the device is asynchronous, while the data type depends on the selected function mode. Three types of messages can be exchanged: force values, command acknowledgments and events. A remote control API is offered, implemented using Python 3.6 and OSC.[2] Example applications are supplied which showcase the device's functionalities.

7.2 Model

In this section, the mechanical model of the Bogus Finger is presented [3]. The prototype of Sec. 4.2 was used to derive and validate a mechanical lumped parameter model, which allows to concentrate the physical parameters on specific elements of the device. Given the DIY nature of the project, a procedure to estimate the parameters of the model using commercial tools is detailed, providing the final users with a guideline to characterize the system, when materials or components change.

The dynamic response of a mechanical system is determined by its impedance. Once the mathematical model has been derived, the point where the external force is applied is selected, the responses (acceleration, velocity or displacement) of each mass of the system to the external force are then calculated. Thus, once the mass being in contact with the external force has been selected (i.e. the haptic device in this case), it will be possible to calculate the impedance at that point.

---

[2] A widely used communication protocol optimized for multimedia and networking technology: `http://opensoundcontrol.org/`.

[3] The mathematical model was developed thanks to Dr. Mattia Dal Borgo and Dr. Eleonora Pippia.

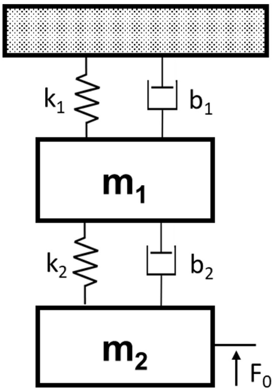


Figure 7.2: Mechanical model of the *Bogus Finger*. The same model applies for both normal and lateral direction.

The prototype of the Bogus Finger can be modeled as two degree-of-freedom (DoF) system. Figure 7.2 shows the mechanical model of the Bogus Finger being in contact with a haptic device. The external force (F_0), generated by the haptic device (i.e. the vibration source), is applied to the mass of the silicone interface m_2. In the model, the spring k_1 and the damper b_1 represent the shock absorbers, whereas k_2 and b_2 belong to the silicone interface. Regardless of the compression load, the shock absorbers are defined by fixed elastic and damping coefficients. Conversely, the hemispherical shape of the silicone interface makes the contact area to increase proportionally with the compression, resulting in increasing elastic and damping coefficients. Finally, the mass m_1 represents the remaining mass of the end-effector up to the shock absorber. All the remaining parts of the *Bogus Finger* (i.e. linear actuator, supports and electronics) are considered as fixed to the ground.

The mechanical impedance $Z(s)$ of the *Bogus Finger* model, calculated as ratio between the external force $F_0(s)$ acting on the mass m_2 and the velocity $\dot{x}_2(s)$, is described by equation 7.1, where s is the complex variable of the Laplace transform. The impedance of the system in normal and tangential direction is different, even if the structure of the model is shared. Indeed, whereas the value of the masses m_1 and m_2 remain obviously constant, the elastic and damping properties of the spring-damper connections change due to their form factor that is different along the directions (e.g., the silicone interface is wider than tall).

$$Z(s) = \frac{F}{\dot{x}} = \frac{(m_1 m_2)s^4 + (m_1 b_2 + m_2 b_1)s^3 + (m_1 k_2 + b_1 b_2 + m_2 k_1)s^2 + (b_1 k_2 + b_2 k_1)s + k_1 k_2}{(m_1 + m_2)s^3 + (b_1 + b_2)s^2 + (k_1 + k_2)s} \tag{7.1}$$

Beside this, two further minor differences associated with lateral forces can affect the mechanical model: firstly, the end-effector itself can be considered as a cantilever beam, thus its lateral oscillations depend on the point where the force is applied along the beam; secondly, the worm gear shown in Figure 7.1a presents a finite stiffness, which changes depending on the vertical position of the end-effector (i.e. the distance from the motor anchor point).

7.3 Modeling parameters

The parameters of the model can be identified from experimental measurements. Parameters can be estimated considering the overall mass of the end-effector (m_{sys} = load cell, accelerometer, metal supports, half of silicone and shock absorber masses) as being suspended between two spring-damper connections (i.e. shock absorber and silicone interface). Thus, by applying harmonic oscillations to the mass m_{sys}, its acceleration can be acquired and analyzed in the frequency domain to identify the stiffness and damping coefficients [93], furthermore repeating the procedure for different pressing forces while the silicone interface is in contact with a rigid surface.

Figure 7.3 shows the experimental setup used to estimate the parameters of the model: a voice coil actuator Lofelt L5 was fixed to the central part of the end-effector by means of an aluminium square and driven by a Dayton DTA-1 amplifier. Depending on the direction measured, the actuator was rotated in order to oscillate along the vertical or the lateral direction. For both motion directions, six levels of pressing force were tested (0.5, 1, 2, 3, 5, 8 N), resulting in 12 test conditions. For each condition, the response of the system was recorded with a PCB 356A17 accelerometer, fixed on top of the end-effector. For each measurement, the transfer function from the input signal (a logarithmic sine sweep between 10 and 600 Hz lasting 15 s) to the output acceleration was calculated using the function `tfestimate` of the GNU Octave 5.1 software.

The electro-mechanical lumped parameter model of the entire system (Bogus Finger + L5 actuator) is reported in Fig. 7.4. A current flowing through the winding of the voice coil actuator generates the input signal F_0, which results applied, in opposite directions, both to the mass of the permanent magnet m_{act} and to the case of the actuator that is fixed to the end-effector mass m_{sys}. The overall system can be modeled as a 4th-order system, with equations of motions given by equation 7.2a for m_{sys} and equation 7.2b for m_{act}.

$$\begin{cases} m_{sys}\ddot{x}_{sys} = -k_{sys}x_{sys} - b_{sys}\dot{x}_{sys} - k_{act}(x_{sys} - x_{act}) - b_{act}(\dot{x}_{sys} - \dot{x}_{act}) - F_0 & (7.2a) \\ m_{act}\ddot{x}_{act} = k_{act}(x_{sys} - x_{act}) + b_{act}(\dot{x}_{sys} - \dot{x}_{act}) + F_0 & (7.2b) \end{cases}$$

where $k_{sys} = k_1 + k_2$ and $b_{sys} = b_1 + b_2$

As shown for example in Fig. 7.5a, the transfer function of the acceleration data (blue line) is used to fit the 4th-order model (orange line) based on the equations 7.2a and 7.2b. Indeed, once the values of the masses m_{act} and m_{sys} are defined, only one set of (k_{sys},b_{sys},k_{act},b_{act}) minimizes the error with the experimental data at each frequency. In particular, k_{sys} and b_{sys} determine respectively the position and the amplitude of the lower frequency resonance peak, whereas k_{act} and b_{act} count for the higher frequency resonance peak that remains constant for all the measurements. The parameters of the voice-coil actuator have been identified as m_{act} =3.9 gr, k_{act} =0.564 N/mm and b_{act} =0.55 Ns/m. These parameters are related to the reso-

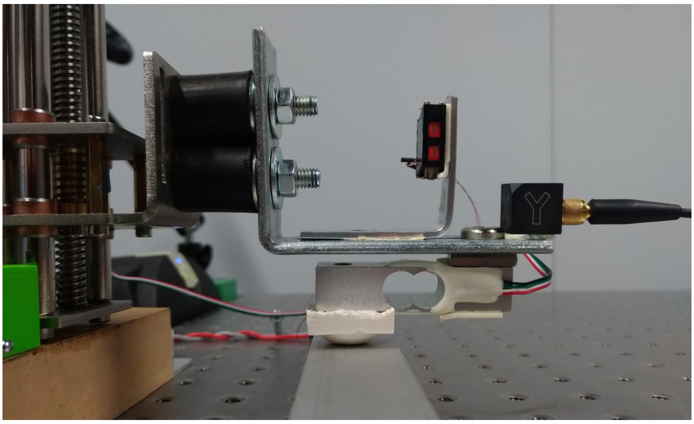


Figure 7.3: Experimental setup used for the model parameter estimation: a Lofelt L5 voice coil was attached to the mass m_{sys} (load cell, accelerometer, metal supports, half of silicone and shock absorber masses) while the Bogus Finger was pressing against a metal bar.

nance peak at 63 Hz that can be identified in all the reported transfer functions (see Fig. 7.5a, 7.5b and 7.6).

Figures 7.5a and 7.5b report the frequency response in vertical and horizontal directions, respectively, when the device is suspended, thus not in contact with any surface through the silicone interface. This configuration was adopted to characterize the stiffness and the damping of the shock absorbers: according also to the static measurement reported in Fig. 7.9, the vertical stiffness is constant, and is equal to 7.67 N/mm; the vertical damping coefficient, instead, is equal to 2.34 Ns/m. These two parameters characterize the shape and the frequency of the resonance peak (f0=36 Hz), correspondent to the mass m_{sys} reported in Fig. 7.5a. Concerning the horizontal direction, the model fitting of the data reported in Fig. 7.5b results in a stiffness coefficient of 2.45 N/mm and a damping coefficient of 1.8 Ns/m.

Whereas the parameters of the connection k_{act} and b_{act} are constant (i.e. they depend on the stiffness and damping coefficients of the structure holding the magnet of the actuator), the parameters k_{sys} and b_{sys} change with the force applied. To tune these two parameters, a greedy optimization algorithm was set up for each tested pressing force. In particular, the Particle Swarm Optimization (PSO), implemented in Matlab® 2019b software, was applied with the default parameters. The optimization procedure minimizes a weighted root mean squared error between the modeled magnitude and the acquisition. Since more accuracy was required around the two peaks, a window of [-50,50] Hz was defined around the maximum peak: in that range, the weight was set to 1, whereas, in the remaining domain, the weight was reduced to 0.1; this way, the model was less affected where the acquisition was more unstable. The window was manually set to capture the slope and the magnitude of the two peaks, characterizing the shape of the frequency response.

Fig. 7.6 reports the obtained frequency response of m_{sys} in the vertical direction

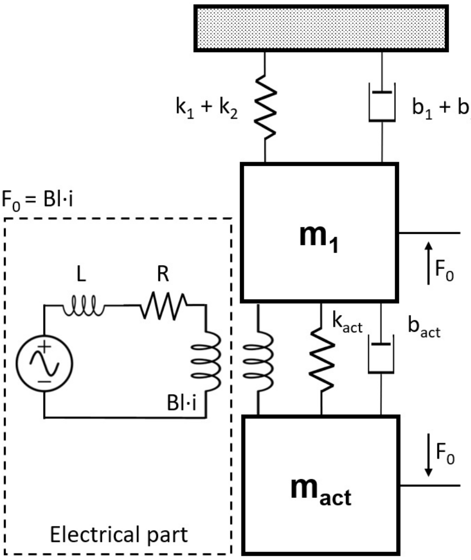


Figure 7.4: Electro-mechanical model of the setup used in the characterization process: the current (i) flowing through the voice coil generates the input signal F_0 proportional to the transduction coefficient (Bl). Such input signal is applied, in opposite directions, to both the masses m_1 and m_{act}.

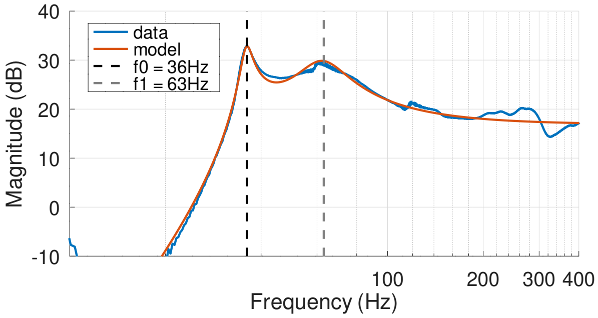
while different pressing force levels are applied: the progressive force increment produces a stiffer (connection/joint) that brings to higher natural frequency of m_{sys} (i.e. from 40.2 to 47.9 Hz).

Table 7.1 reports the parameters estimation of stiffness and damping for all the pressing force concerning the vertical direction. Since the shock absorbers and silicone interface are modeled as parallel connections, the silicone properties (k_2 and b_2) can be calculated by subtracting the constant values of the shock absorbers (k_1, b_1) from the model estimated values (k_{sys}, b_{sys}).

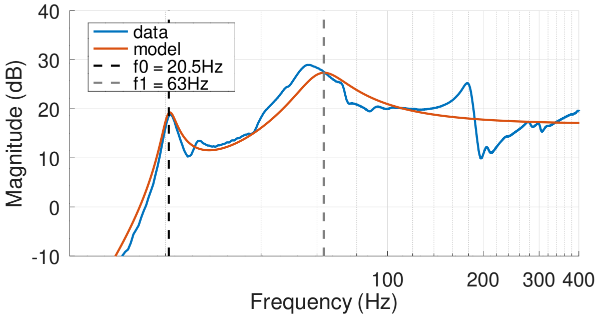
Table 7.2 reports the parameters estimation of the stiffness for all the pressing force concerning the horizontal direction. Unfortunately, it was not possible to calculate the exact damping coefficients from the experimental data, due to a strong anti-resonance found in the frequency region of the m_{sys} natural resonances. On the other hand, it was still possible to roughtly estimate the k_{sys} values matching the slopes of the functions.

The model of the Bogus Finger, reported in Fig. 7.2, was used to calculate the impedance at the contact point, applying the parameters of Tables 7.1 and 7.2 to Eq. 7.1. Figures 7.7a and 7.7b show a comparison between the impedance of the Bogus Finger with the silicone interfaces (s1 or s2) and the impedance of the human finger reported by [122, 101] with vibrations in the normal direction.

Figures 7.8a and 7.8b, instead, show the same comparison but with vibrations in



(a) Vertical



(b) Horizontal

Figure 7.5: Frequency response of m_{sys} without the silicone interface in contact with the bottom surface. The model (orange line) is fitted to experimental data (blue line). The frequencies, corresponding to the natural resonances of the suspended mass (f0) and the actuator (f1), are highlighted with vertical dashed lines.

the lateral direction. Since it was not possible to estimate the damping variations consequent to the applied increasing force, the damping coefficient was set to 2 Ns/m for all the force level; however, as reported for the vertical direction, an increment of damping for increasing forces is expected.

7.4 Model comparison

This section reports a comparison between the models of human finger and the lumped model developed for the Bogus Finger concerning the vertical and the horizontal directions. In particular, the impedance measurements accomplished by Hatzfeld and Kern [101, 122] were used to compare the theoretical impedance of the Bogus Finger calculated from the model.

When the human finger touches a flat large surface vibrating in the vertical or in the lateral direction, the model of [122] shows an impedance function with a unique minimum in the range 20-1000 Hz (see dashed lines of Fig. 7.7a and 7.8a). Such minimum corresponds to the natural resonances of the finger: the frequency of such

Table 7.1: Parameters estimation for all the forces with different silicone interfaces in the vertical direction. Stiffness values are expressed in N/mm whereas damping values are expressed in Ns/m

Sil.	Param	Press(N)					
		0.5	1	2	3	5	8
s1	k_{sys}	10.41	11.08	12.16	13.25	15.30	–
	k_1	7.67	7.67	7.67	7.67	7.67	–
	k_2	**2.74**	**3.41**	**4.49**	**5.58**	**7.63**	–
s2	k_{sys}	9.49	9.74	10.45	11.18	12.26	13.33
	k_1	7.67	7.67	7.67	7.67	7.67	7.67
	k_2	**1.82**	**2.07**	**2.78**	**3.51**	**4.59**	**5.66**
s1	b_{sys}	3.98	4.38	5.80	6.80	8.20	–
	b_1	2.34	2.34	2.34	2.34	2.34	–
	b_2	**1.64**	**2.04**	**3.46**	**4.46**	**5.86**	–
s2	b_{sys}	3.35	3.48	3.87	4.12	4.88	5.61
	b_1	2.34	2.34	2.34	2.34	2.34	2.34
	b_2	**1.01**	**1.14**	**1.53**	**1.78**	**2.54**	**3.27**

resonance mainly depends on the ratio between the stiffness and the mass of the finger, whereas the resonance magnitude depends on the finger damping, with great resonances (i.e. low impedance) associated with low damping coefficients.

In what follows, a detailed comparison between the components of the model for both vertical and horizontal direction is reported.

7.4.1 Vertical direction

Mass

In a spring-mass-damper system, the mass contributes in defining the natural frequency of the system and its behavior in the high frequency range. Human finger mass was found constant or tending to a constant plateau of 6 g within the force interval 2-20 N [97]. The model of Bogus Finger has two masses: the first mass (m_1) is the end-effector (130 g) that affects the impedance of the Bogus Finger, mainly in the low frequency range having a resonance at 36 Hz. The second mass (m_2) is the silicon specimen (between 3 and 4 g). Given the structure of the Bogus Finger, only m_2 mass is affected by high frequencies.

Stiffness

The stiffness in the normal direction can be determined in static conditions measuring the force/deformation ratio. The measurements on the Bogus Finger, reported in Fig. 7.9, show a linear dependency regarding the direct contact (DC) of the load cell pressing over a rigid surface, whereas the addition of a silicone interface between

Table 7.2: Stiffness estimation for all the forces with different silicone interfaces in the horizontal direction. Values are expressed in N/mm.

Sil.	Param	Press(N)					
		0.5	1	2	3	5	8
s1	k_{sys}	2.8	3.0	3.2	3.4	3.7	3.9
	k_1	2.45	2.45	2.45	2.45	2.45	2.45
	k_2	**0.35**	**0.55**	**0.75**	**0.95**	**1.25**	**1.45**
s2	k_{sys}	3.0	3.2	3.4	3.8	4.1	4.4
	k_1	2.45	2.45	2.45	2.45	2.45	2.45
	k_2	**0.55**	**0.75**	**0.95**	**1.35**	**1.65**	**1.95**

the load cell and the surface produces a non-linear trend, meaning that the stiffness increases with the force. This nonlinear trend resembles that of the human fingertip [99, 200], which however is more compliant for low forces, as shown in Fig. 7.9 (dashed line). Finally, the human finger stiffness assumes different values during press and release actions, furthermore depending on the compression speed [35].

In dynamic conditions, the stiffness of shock absorber and silicone layers acts on the frequency of the system natural resonances. In particular, the stiffness of the shock absorbers is responsible for the local minimum at 36 Hz, whereas the stiffness of the silicone interface determines the frequency of the absolute minimum. Such minimum varies between 160 and 250 Hz for the silicone s1 (Fig. 7.7a) and between 100 and 160 Hz for the silicone s2 (Fig. 7.7b) depending on the normal force applied. The local minimum of the Bogus Finger at 36 Hz can be lowered in frequency by increasing the mass of the end-effector and/or using softer shock absorbers, resulting thus closer to the human finger.

Damping

The damping coefficient of the shock absorbers counts for the magnitude of impedance inversion in the low frequency range with small magnitudes associated to high damping coefficients and vice versa. The damping coefficient of the silicone interface, instead, determines the magnitude of the main resonance in the high frequency range. As show in Fig. 7.7a and 7.7b, the damping increases with the pressing force with values between 1.64 and 5.86 Ns/m concerning the silicone s1 and values between 1.01 and 2.54 Ns/m for the silicone s2. Human finger damping was found dependent on the applied normal force, too, with values rising from a mean of 2.2 to 4.0 Ns/m in the interval 2-20 N and high variability among the subjects of the study [97].

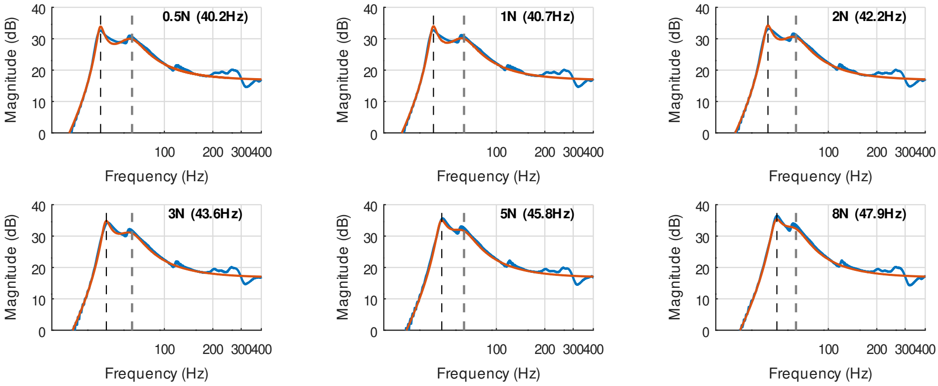


Figure 7.6: Frequency responses to vertical vibrations of m_{sys} with the silicone s2 in the characterization setup of Fig. 7.4. Each subplot reports the frequency response measured (blue line), the model fitting (orange line), natural resonances of m_{sys} (black dashed line) and m_{act} (gray dashed line) for different pressing forces in range 0.5-8 N.

7.4.2 Tangential direction

Mass

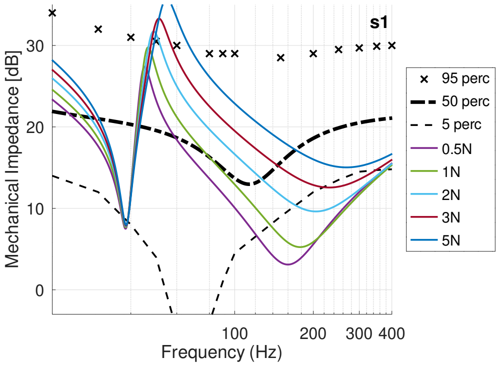
In the lateral direction the mass exposed by the Bogus Finger is the same described above. Concerning the tangential direction, the mass of the human finger results practically constant in the force interval (i.e. 0 - 2N), showing lower values compared to normal direction (from 0.1 g to 0.5 g). According to the literature, the inertial contributions of human finger in tangential direction could be neglected from DC to 500 Hz [239].

Stiffness

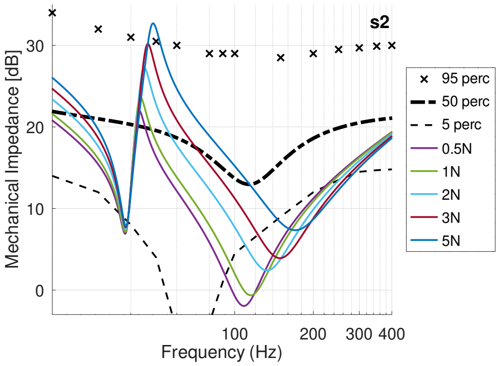
For the time being, the shearing stiffness of the Bogus Finger in static conditions has not yet been addressed, as it would require an additional degree of freedom (motion along X or Y) and an end-effector able to measure lateral forces. Moreover, the shearing stiffness does not depend only on the finger deformation but also on the friction of the surface in contact with the fingertip: when lateral forces overtake friction, a slip effect occurs [166]. For this reason, the stiffness of the human finger was compared with that of the Bogus Finger only in dynamic conditions on an common surface, so as to determine their contribution to the respective frequency response.

According to the literature [239], in dynamic conditions the human finger stiffness is proportional to the normal force applied following a $\frac{1}{3}$ power-law in both proximal-distal and medial-lateral directions, with values between 0.6 and 3 N/mm in the interval (0 - 2N).

The lateral stiffness of the Bogus Finger reports values between 0.35 and 1.45 N/mm for the silicone s1, and between 0.55 and 1.95 N/mm for the silicone s2. Compar-



(a) Silicone s1



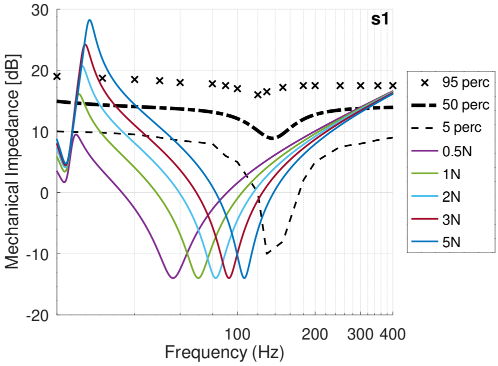
(b) Silicone s2

Figure 7.7: Comparison between the impedance of Bogus Finger with different silicone interfaces and the impedance of human finger (dashed lines) concerning the vertical direction.

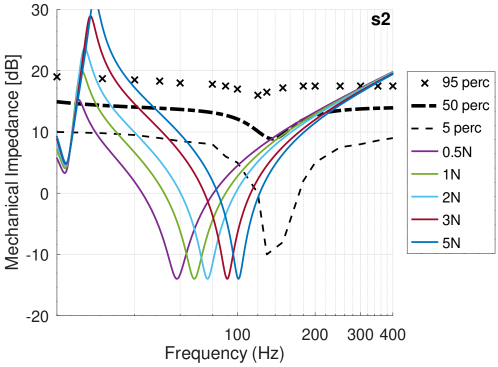
ing the impedance calculated from the Bogus Finger model to the measurements on human fingers, the stiffness of the Bogus Finger seems to be lower than the latter.

Damping

The human finger damping is proportional to the normal force applied following a $\frac{1}{3}$ power-law, too. Values increase from 0.4 to 3 Ns/m in the interval 0 - 2N [239]. As previously said, concerning the lateral direction, it was not possible to assess the value of the damping from experimental data. Thus, the damping coefficient was set to 2 Ns/m for all the force levels; however, also the damping of the Bogus Finger is expected to slightly increase with the applied normal force.



(a) Silicone s1



(b) Silicone s2

Figure 7.8: Comparison between the impedance of Bogus Finger with different silicone interfaces and the impedance of human finger (dashed lines) concerning the horizontal direction.

7.4.3 Friction

Regarding the tangential direction, also the friction has to be taken into account. The friction is defined as the force resisting the relative motion of solid surfaces, fluid layers, and material elements sliding against each other. Thus, the total friction force between the human finger and a flat surface is a sum of following forces: adhesive force, skin deformation force, capillary or viscous shearing force caused by water or natural grease on the skin, and friction due to the deformation of finger ridges [217]. Conversely, in its current configuration, the Bogus Finger exposes only a low adhesive force and no viscous shearing force.

7.5 Validation

The behavior of the Bogus Finger prototype pressing on a vibrating surface was compared to that of the human finger, in order to validate the model proposed and

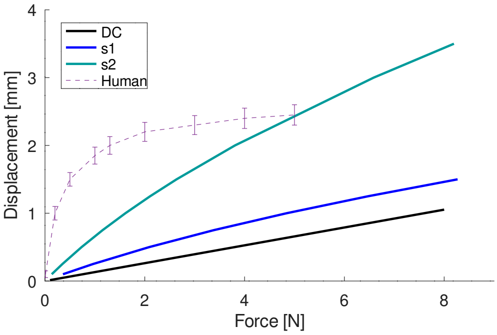


Figure 7.9: Displacement of the end-effector as function of the normal force applied to a stationary rigid surface. The plots show respectively the effects of direct contact of the load cell (DC), and the use of two different silicone layers at the interface (s1: Sylgard 184; s2: Silastic 3481). For comparison, the force/deformation curve of the human fingertip is shown, as reported in the literature [99, 200].

its parameters.

For this purpose, the testbed device described in Chapter 3 and the previous measurements accomplished by ten human subjects were used for validating the Bogus Finger. Therefore, by repeating the same procedure described in Sec. 3.1.2, the Bogus Finger coupled with silicone layer s1 and s2 was used instead of the human subject, providing different force levels.

As test vibrations, the logarithmic sine sweeps between 10 and 600 Hz lasting 15 s (two repetitions) were used. Also in this case, vibrations were recorded with a PCB 356A17 triaxial accelerometer fixed to the top of the testbed cuboid. Pressing forces were recorded across the whole validation to compare accuracy and precision between the Bogus Finger and the human subjects.

The collected acceleration recordings and the data previously acquired from the human subjects were analyzed to determine the effect of the following factors: agent (human subject, Bogus Finger), pressing force (0.5-4.9 N) and stimuli direction (vertical, horizontal).

In particular, for each measurement, the transfer function from the input signal to the output acceleration was calculated using the function `tfestimate` of the GNU Octave 5.1 software. Figure 7.10 compares the frequency responses of the human finger (dashed lines) to that of the Bogus Finger with silicone s1 and s2 for the four force levels measured in the vertical directions. Figure 7.11, instead, reports the same comparison concerning the horizontal direction. The data measured from human subjects were aggregated in magnitude averages and standard deviations, providing more simple visualization and comparison.

The responses to human fingers (gray and blue areas) show a generally narrow confidence interval, especially in the range 200-400 Hz; in the same range, the response to the Bogus Finger with silicone s2 is rather close to that of the finger, while, in the lower range, the response to our device diverges noticeably. Such differences are greater with silicone s1 than s2 for all factor combinations. The most prominent difference involves the amplitude and the frequency of the main resonance peaks, that are both higher when the Bogus Finger is applied to the testbed. The amplitude of the main resonance peak depends mainly on the damping at the contact point, with low damping coefficients associated to large amplitudes. On the other hand, the frequency of the peak is related to stiffness: the larger the interface hardness, the higher the frequency of the peak. As a matter of fact, the silicone s1 (which is harder) always shows higher frequency peaks compared to the soft silicone s2. Finally, the frequency responses change with motion direction: concerning the vertical direction the frequency of the peaks measured in presence of the Bogus Finger is always higher than in presence of the human finger; concerning the horizontal direction, instead, the frequency of the peaks is close to that of the human finger (at least considering silicone s2), whereas the amplitudes are about 6 dB higher.

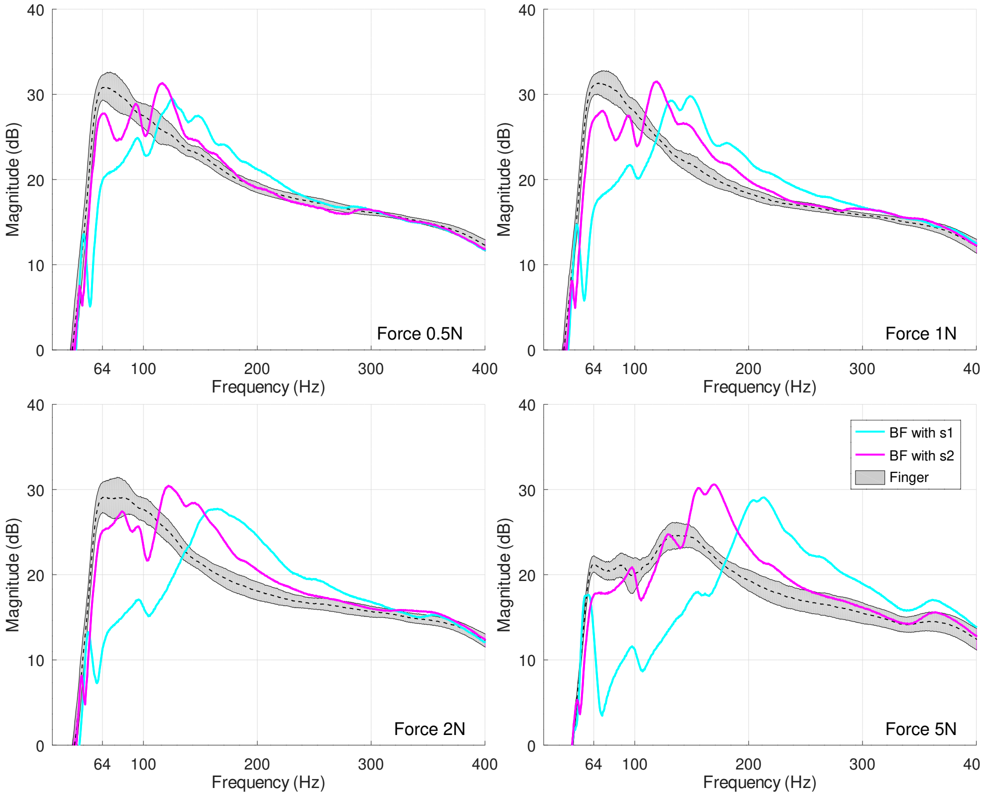


Figure 7.10: Comparison of frequency responses of the testbed in the vertical direction, for different pressing forces. The responses of the human participants are represented by grey shaded areas, while the respective average responses are depicted in dashed lines. Solid cyan and magenta lines report respectively the response of the Bogus Finger with silicone s1 and s2.

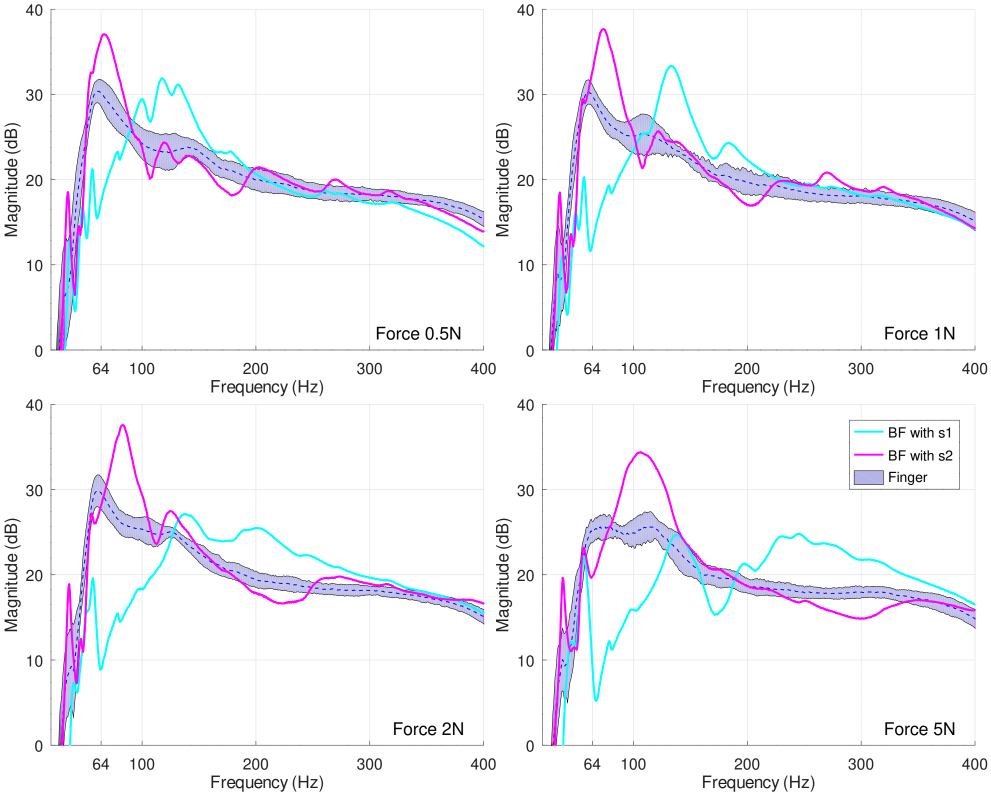


Figure 7.11: Comparison of frequency responses of the testbed in the horizontal direction, for different pressing forces. The responses of human participants are represented blue shaded areas, while the respective average responses are depicted in dashed lines. Solid cyan and magenta lines respectively report the response of the Bogus Finger with silicone s1 and s2.

Pressing force control

Force-control error was analyzed in the data recorded during the experiment.

Table 7.3 reports means and standard deviations of the normalized control error for the human participants and the Bogus Finger. Means account for the accuracy of the pressing force, whereas standard deviations are related to the force-control precision.

Overall, the best accuracy and precision are associated to the 2 N force level. For lower levels, the accuracy of the human participants and the Bogus Finger is similar, while the precision of the device is much higher. When applying the highest pressing force (4.9 N), humans show the lowest accuracy, while the device shows uniformly high accuracy and precision for forces $\geq$ 2 N.

Table 7.3: Force-control error.

Force (N)	Error (%) mean ± s.d.	
	Human finger	Bogus Finger
0.5	1.6 ± 7.0	1.0 ± 1.6
1.0	-1.6 ± 4.7	1.0 ± 1.0
2.0	-1.1 ± 4.4	0.2 ± 0.25
4.9	-2.7 ± 3	-0.6 ± 0.6

7.6 Discussion

The validation generally confirms the differences predicted from the model reported in Fig. 7.2. Indeed, since the frequency response of the device testbed was not linear, this validation can be used only to compare the relative differences between human finger and the Bogus Finger with different silicone layers. Furthermore, it allows to compare the theoretical impedance calculated from the model and the literature to the real values measured with a common haptic device.

As predicted by the model, the Bogus Finger shows two natural resonances for each test condition, whereas the human finger shows always a single resonance. Moreover, the subplots of Fig. 7.10 and 7.11 show that the stiffness of the human finger slightly grows with the pressing force for both motion directions, shifting its natural resonance towards higher frequency. As predicted by our model, the Bogus Finger shows a similar trend.

However, whereas in the vertical direction the frequency of the resonances predicted by the Bogus Finger model is in agreement with the experimental data, in the horizontal direction the peak frequencies are different concerning the silicone s1. Such difference can be ascribed to a different friction coefficient that can introduce slip effects, affecting the frequency response.

The most important model prediction assessment regards the value of the damping exposed by the silicone layers, being always too small compared to the human finger. To this end, the damping at the contact point could be increased by finding alternative composite materials having similar mass/stiffness ratio, but greater damping coefficient.

7.7 Conclusions and future development

This chapter presented several design aspects and key features of the Bogus Finger. Based on the characteristics of the device, a mechanical model was designed and parametrized with experimental data. The end-to-end validation of the prototype revealed good approximation of the finger impedance in the frequency range of highest human sensitivity to vibrations. Despite this, the impedance of the Bogus Finger diverges in the lower range, with resonance peaks either located at higher frequency or having larger amplitude for vertical or horizontal vibration, respectively. However, the experimental results are in good agreement with the responses calculated offline on the model. Thanks to this, the resonances of the Bogus Finger may be made to match the response of the human finger by fine-tuning the silicone layer in its mass, stiffness, damping and form factor parameters through numerical simulations, thus reducing the number of tests with physical prototypes [35].

Future extensions encompass also the simulation of force envelopes by means of lookup tables directly controlling the motion of the stepper motor: these would allow to achieve increased acceleration/velocity (e.g., reproducing impacts) at the expense of spatial accuracy, which could however be recovered via force-feedback control.

Finally, the availability of this low-cost DIY tool in open-access form, rather than proprietary and expensive finger-analogues, has the potential to grant access to realistic simulation of finger-based interactions to a larger community of researchers in the fields of touch psychophysics and haptic interfaces.

8

Python for real-time signal processing

This final chapter is a sort of appendix that shows one challenging application about the characteristics of the Python programming language: the digital signal processing at audio rate. In this thesis, Python has been already mentioned in multiple activities: for instance, it was used to create all the automatic experimental procedures, to manage the input/output, the communications with many devices, the randomization of the experimental trials and the data collection. Concerning the data representation and the analysis, Python was used to generate all the plots shown in this thesis, and to perform almost all the statistical analyses. Finally, Python was used also to develop the control algorithm and the machine learning algorithm of the motionless knob.

As mentioned in Chapter 4, Python has been profitably used to develop and execute digital simulations of analogical electronic circuits described by non-linear equations. However, the efficient and uninterrupted execution of such code was not straightforward, requiring the refactoring procedures presented in the following.

8.1 Computer music applications

Among the many definitions, the development of computer music applications has been qualified as “the expression of compositional or signal processing ideas” [154]. In fact, a computer music language should be oriented in particular to the development of real-time software. To this end, a common characteristic of computer music languages is their orientation toward real-time software. Due to the efficiency of the resulting machine code, early computer music developers used to program on their personal computer in C or C++. This necessity has sometimes caused excessive dependency of the application on the characteristics of the programming language itself [157]. In particular, the high technicality of C and C++ has often discouraged computer musicians to approach problems, requiring solving skills which are domain of computer scientists and engineers instead[116, 212]. To reconcile this divide, abstractions have been proposed leading to specialized languages such as Csound, Max, Pure Data, SuperCollider, Chuck, Faust, and not only [4]. These abstractions embrace the imperative, functional and visual programming paradigm. On the one hand they allow computer musicians to create software with lower effort; on the other hand, the long-term life of their sound applications obviously depends on the continued support of such peculiar languages on standard operating systems and graphic user interfaces. Although, many positive examples of communities that succeed in developing and maintaining open-source hardware and software exist [165], due to the limited business moved by this niche market, and because many such languages are maintained (sometimes even for free) by programmers who are also computer musicians, unfortunately this community has been suffering probably more than others from the lack of a systematic, durable approach to the development and maintenance of sound software [164]. Conversely, Python is continuously increasing in popularity, thanks also of a non-commercial license of use [132]. Its community maintains a lot of official packages, including libraries (e.g. Numpy, Scipy, Matplotlib) providing scientific computing tools comparable to those equipping dedicated software such as Matlab and R. Thanks to a fast learning curve, its rapid prototyping features and intuitive readability of the code, the Python community now includes also users paying particular attention to the interaction aspects of their software, such as academics [92, 152] employing Python as a teaching-by-examples tool [63]. Despite this quest for interactivity, use of Python in real-time applications is testified by exceptions: RTGraph, for instance, instantaneously processes physiological signals and then displays the results trough the Qt framework [199].

This limitation is common for interpreted software. Unlike C and other compiled languages, Python in fact generates bytecode which is interpreted by a virtual machine operating at application level. This, however, is not the only way to run an application. In particular, Python puts available more than one tool to compile the code, hence affording performances that are accessible only at processor level. For instance, the Numba library through its just-in-time compiler speeds up numerical iterations such as those needed by vectorial operations called by Numpy. Furthermore, chunks of C code can be embedded within a Python program [12] using the Cython library, which includes a static compiler accepting instructions belonging to several compiled languages as part of a program written in Python. Using Cython it is also possible to declare static variables as C programmers do for substantially reducing the time to bind them at runtime. This way, algorithms translated from e.g. Matlab or already in C can be compiled through Cython furthermore preserving the static memory space. Later they can be called from the Python environment as standard modules, however computed at machine level. In both Numba and Cython the refactoring of few lines of code is often sufficient to optimize the most computationally intensive parts of an algorithm, with dramatic speedup of the application. In the sound processing domain, this optimization is often limited to the loops that are indefinitely iterated at sample rate.

8.1.1 Related work

The number of computer music programming languages is notable. Each language differs in terms of abstraction, coding approach, learning curve, portability across architectures and operating systems. In parallel, all make real-time sound programming easier. In particular, those in the Music-N tradition [153] support dynamic instantiation of graphs of *unit generators* (UG's), the basic building blocks of a sound synthesis and processing algorithm. Csound [227] structures the code in two parts: an instrument file contains the UG's, while a score file controls them along time through note and other event parameters. Pure Data [188] and Max [38] allow the visual programming of UG networks and related control messaging. SuperCollider [154] implements a client-server architecture enabling interactive sound synthesis and algorithmic composition through live coding, also interpreting different languages thanks to the flexibility of the client. ChucK supports deterministic concurrency and multiple control rates, providing live coding and enabling live performances [235]. Faust follows the functional programming paradigm, and builds applications and plugins for various real-time sound environments thanks to automatic translation into C++ code. It also provides a powerful online editor, enabling agile code development [72].

Existing sound applications in Python mainly focus on the analysis and presentation of audio, as well as on music retrieval [88, 155, 14]. Most such applications have no strict temporal requirements, and hence are conventionally written for the interpreter: examples of this approach to sound programming can be found in e.g. [238], where latency figures are also detailed. Concerning sound manipulation, computer musicians often rely on the Pyo library [13], a client-server architecture allowing to combine its UG's together into processing networks. Hence, the abstraction from the signal level is realized through Pyo also in Python, making the creation of sound generation and effect chains possible as most sound programming languages do.

On the other hand, the low-level development of sound algorithms is not trivial when working with UG's, as they encapsulate the signal processing by definition. Also because of the existing excellence in this sound programming paradigm, the approach here described puts the accent to coding at signal level. Addressing such a level in Python requires to profile and refactor usually few signal processing instructions. The advantages of code refactoring go beyond sound applications; in fact, refactoring can be applied to contexts including, among others, real-time data collection, systems control and automation.

8.1.2 On real-time processing

By definition, real-time processes produce an output within a given time. Concerning sound processing, nominally this time is inversely proportional to the audio sampling rate. Our test environment is an Intel-i5 laptop computer running Windows 10, connected to a RME Babyface Pro external USB audio interface. Contrarily to hardware/software systems specifically oriented to real-time audio [65], in such a standard architecture a sound process can be stopped by the operating system (OS) scheduler for too long or too many times within an allowed time window, hence becoming ultimately unable to regularly refill the audio output buffer at sample rate, with consequent sound glitches and distortions in the output. A common workaround to this problem, known as buffer underflow, consists of increasing the size of the audio buffer. As a sound process normally produces samples much faster than the audio sample rate, this workaround decreases the probability for the audio interface to find the buffer empty.

However, longer buffer size comes along with a proportionally higher latency of the output. Although latency can be a negligible issue in feed-forward sound interactions, conversely it can annihilate a closed-loop musical perception-and-action, where typically no more than 10 milliseconds are allowed for a computer music setup to respond to musicians [156]. More in general, all the applications involving multimodal feedback should limit the processing latency, particularly when tactile feedback is provided. Therefore, the buffer size must compromise between probability of signal artifacts and tolerable latency.

Irrespective of the buffer size, higher latency is generally beneficial for sound quality since the OS scheduler in that case can stop the sound process occasionally for a longer while, e.g. to handle an external interrupt. However, the developer should always avoid to include unbounded time operations in a sound processing thread. Rather, input/output (I/O) instructions, graphic functions, and in general all procedures in charge of the interaction with the system, should be implemented by parallel threads sharing lock-free data structures with the sound processing thread. In this regard, practical general suggestions on real-time programming can be found in thematic discussions on the Internet [15]. At any rate, Python is not designed to support the servicing of audio threads within deterministic temporal constraints. For this reason, regardless of the performances reported for this test environment in the following sections, Python applications should not be programmed with the purpose to guarantee real-time sounds at low latency.

8.1.3 Structure of the chapter

The chapter is structured as follows: Sec. 8.2 explains real-time software interpretation in Python through a simple example. Sec. 8.3 introduces programming and code profiling with Numba and Cython. Sec. 8.4 applies the above concepts on the VCF sound algorithm, being profiled during its running in real time. All the code examples have been put available on GitHub,[1] along with the scripts that have been used to benchmark the algorithms. Finally the Section 8.5 discusses the results in front of the constraints imposed by the OS to real-time process running.

8.2 Standard interpreted approach

Real-time program development first of all needs to manage sound I/O through a low-level application programming interface. Concerning Python, the library PyAudio [184] among others provides bindings for portaudio, an open-source cross-platform audio device. As most low-level libraries do, PyAudio allows to operate sample-by-sample on audio chunks whose size is set by the user. Typical chunks range between 64 and 2048 samples. PyAudio provides a blocking mode, enabling synchronous read and write operations, as well as a non-blocking mode managing the same operations through a callback by a separate thread. On top of I/O, Python provides libraries supporting the agile development of virtual sound processors: the module *scipy.signal* within the Scipy library, for instance, contains some standard signal processing tools. An exhaustive list of Python libraries supporting audio analysis and processing can be

---

[1] `https://github.com/yuridepra88/RealtimeAudioPython`

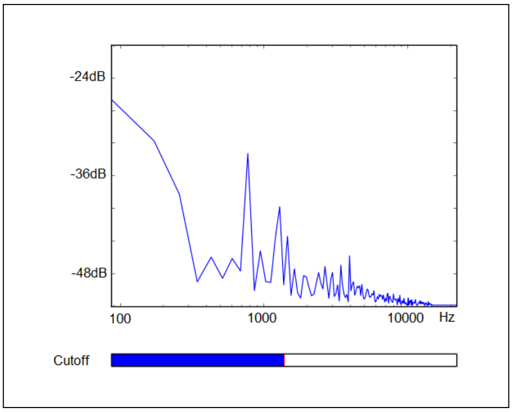


Figure 8.1: Real-time spectrum of the output signal and low-pass filter cutoff parameter control.

found in [211]. Further examples of the interpreted approach to sound programming are presented in [238].

Listing 8.1: Audio I/O – Callback function

```
def callback(i_d, frame_count, t_info, f):
  data = wavefile.readframes(frame_count)
  samples = pcm2float(data)
  y = process(data)
  out = float2pcm(y)
  return (out, pyaudio.paContinue)
```

In Listing 8.1, the basic structure of a callback procedure enabling the asynchronous processing of an audio chunk at sample rate is reported. Each time the procedure is called, one chunk is read from the audio buffer and then assigned to the array *data*, containing accessible sound samples. The functions *pcm2float(byte[])* and *float2pcm(float[])* convert each sample from raw bytes to [-1., 1.]-normalized floats and vice versa. Finally, the function *process(data)* encapsulates the processing algorithm.

Building upon this structure, a simple procedure is exemplified implementing the following low-pass digital filter [162]:

$$y[n] = \alpha x[n] + (1-\alpha)y[n-1] \quad , \quad 0 < \alpha < 1\,. \tag{8.1}$$

Fig. 8.1 shows a simple graphical interface for this low-pass filter control. The interface displays also the Fast Fourier transform computed at runtime on each chunk by the *scipy.fft(float[])* method, provided by Scipy. An example that implements different Butterworth filters meanwhile providing parameter controls to the user is available on our GitHub repository.

Listing 8.2: Low-pass filter implementation – iPhyton

```
def process(data):
  global last_sample

  b = 1-alpha
  y= np.arange(CHUNK_SIZE, dtype=float)

  y[0] = alpha*data[0] + b*last_sample
  for i in range(1,CHUNK_SIZE-1):
      y[i] = alpha*data[i] + b*y[i-1]
  last_sample = y[CHUNK_SIZE-1]
  return y
```

Listing 8.2 shows the respective implementation: each time the *process(data)* function is called, the samples in the array are sequentially processed by the algorithm to form one output chunk; moreover, the last output sample is stored in the variable *last_sample* for processing the first sample of a new chunk when *process(data)* is called next. The variable *last_sample* must belong to the global scope since carrying a value between subsequent function calls. As opposed to other languages, global variables in Python must be declared at the beginning of a function using the *global* identifier if they are later written within an instruction appearing inside the same function. Otherwise, a local variable with the same name is automatically generated. Conversely, the variable *alpha* does not need such declaration; in fact this variable is read inside the function and, concurrently, written by the control thread that assigns a value depending on the slider position visible in Fig. 8.1. These considerations are crucial not only to prevent from incorrect use of the variable scope. In fact, as explained in the next Section, global variables must be correctly refactored depending on the optimization tool.

Alternatively, Python offers the possibility to program the UG's as objects, hence inherently encapsulating every UG state as part of the corresponding object variables. By guaranteeing code modularity through the unit generator abstraction, the UG-based/object-oriented approach essentially removes the need to manage global variables, with major advantages when sets of identical UG's, such as oscillators or filters, must be first instantiated and then put in communication with each other. As an example, the object class *OscSine()* was uploaded in GitHub, allowing multiple instances of a simple sinusoidal oscillator. Similarly to the previous simple low-pass filter, this example will be refactored and hence proposed again in the next sections. However, coherently with the general aim of exploring low-level sound programming, it will give emphasis to procedural instead of object-based examples. Some arguments are in favor of this choice. For instance, especially in the case of nonlinear systems such as those being presented in Sec. 8.4, the UG-based approach shows limits as soon as a sound algorithm improvement requires to concatenate existing UG's in the form of a delay-free loop [73]. In such a case, programming a new object lumping together such UG's can be much more time-consuming and error-prone than adapting an existing procedure to compute the delay-free loop. Unfortunately, the Python interpreter fails to compute sounds signal in time even at audio sample rate as soon as the processing algorithm falls outside simple cases. Although on the one hand dynamic interpretation allows faster development and reduces programming errors, on the other hand it slows down the computation. Among the causes of this slow down, dynamic typing has been recognized to prevent the interpreter from achieving the real time. In this regard, the Python library *line_profiler* [121] can be used to profile and analyse code performances: such library measures the code execution time line-by-line. An example of interpreted code profiling is reported in Listing 8.3, where the cost of incrementing a variable and computing a *sin()* function are measured. This example will be proposed again in the next section, to benchmark the refactored code.

Listing 8.3: Global variable profiling – interpreted Python

```
Function: try_global
Hit time unit: 3.41e-7 s
Total time: 0.1037 s

  Hits          Time   Per Hit    % Time   Line Contents
==================================================
     1           3.0       3.0       0.0    a=0
                                            def try_global():
                                              global a
     1           3.0       3.0       0.0      i=0
100001       85289.0       0.9      28.0      while i < 100000:
100000       92719.0       0.9      30.5          i += 1
100000      126083.0       1.3      41.5          a = math.sin(i)
```

8.3 Code speedup

Due to the dynamic interpretation of the bytecode, Python does not allow to declare static variables. On the other hand static variables speed up access to the data, since they are not allocated in the local memory of a function every time it is called. In practice, the conversion into static of any possible variable that is used intensively in a program can bring substantial performance benefits. As part of their optimization features, Numba and Cython allow to declare static variables.

8.3.1 Numba

Numba is a just-in-time (JIT) compiler for vectorial computing in Python [135]. Using Numba, it is possible to speed up functions containing instructions that can be computed more efficiently at machine instead of application level, such as those applying to NumPy objects. In this case, the instructions are sent to the JIT compiler by adding the decorator *@jit* before declaring a function. At this point, the bytecode is translated in Low Level Virtual Machine Intermediate Representation (LLVM

IR), an architecture-independent language similar to assembly. Numba provides two operating modes, that are assigned during the decoration: the *nopython* mode (i.e., *@jit(nopython=True)*) compiles the code completely, whereas the *object* mode tries to compile the loops while leaving to the interpreter the instructions outside them. The performance improvement resulting from the *object* operating mode is limited and, in general, not sufficient for achieving the real time.

Numba is easy to use, furthermore it requires almost no changes of the Python code. On the other hand it translates only a fraction of the Python instruction set, even if its expressive power is increasing release after release. Most importantly, the functions compiled in *nopython* mode cannot access Python objects at runtime; thus, data can be exchanged between such functions and a Python program only in form of input and return arguments. This limitation asks for refactoring the communication among functions and restructuring the code into simple functions, containing only critical instructions (i.e., arithmetic loops).

Listing 8.4: Global variable profiling – Numba

```
from numba import jit, float32
import math
a=0.
@jit(float32(float32),nopython=True)
def numba_global(a):
        i=0
        while i < 100000:
                i+=1
                a = math.sin(i)
        return a

# Execution and benchmark of the refactored function
Function: wrap_numba_global
Hit time unit: 3.41e-7 s
Total time: 0.0474 s
 Hits          Time   Per Hit    % Time   Line Contents
===========================================================
                                          def wrap_w_numba_global():
                                               global a
    1      129328.0  129328.0      93.0        wrap_numba_global()
    1           3.0       3.0       0.0        a=1.
    1        2183.0    2183.0       1.6        wrap_numba_global()
```

Numba is instructed about static variables through the decorator, that must explicitly refer to the input and return arguments as Listing 8.4 shows. Without this information, the compiler in some cases could be unable to determine the variable scope and, consequently, force the *object* mode. In the specific case a wrapper function named *wrap_numba_global()* is needed to declare the global variable *a*, and then pass it to the function *numba_global(a)* as an argument. Otherwise, *numba_global* cannot access its content since belonging to the runtime space. Listing 8.4 shows also the time profiling concerning the computation of the *sin()* function applied to iterated variable after Numba refactoring. *wrap_numba_global()* is called twice: in the first call the JIT compiler optimizes the code, hence slowing down the execution; from the second call on, the optimized cycle is computed in as fast as 7.45 ns.

8.3.2 Cython

Cython extends Python by allowing explicit type declarations and direct compilation of C code [12]. Almost all Python libraries for scientific computing use Cython to minimize the computational burden, also because most algorithms provided by these libraries were already optimized on pre-existing pieces of C, C++ or even Fortran code. In fact, Cython is able to wrap functions written in different compiled languages. Unlike Numba, Cython is an ahead-of-time (AOT) compiler: during the so-called *cythonization*, C code is generated from additional (.pyx) files written by the developer during refactoring. Such files contain Python code (that is optimized automatically) and/or function wrappers of external routines in other languages. The efficiency of the cythonization can be measured as usual through code profiling: by just adding an option to the cythonization command, Cython even offers pictorial highlighting of the instructions proportionally to their computational cost.

Listing 8.5: Global variable profiling – Cython

```
Function: cython_try_global
Hit time unit: 3.41e-7 s
Total time: 0.0526 s
  Hits         Time  Per Hit   % Time  Line Contents
=====================================================
     1          3.0      3.0      0.0  cdef float a=0
                                       def cython_try_global():
                                          global a
     1          4.0      4.0      0.0     cdef int i=0
     1          1.0      1.0      0.0     while i < 100000:
100000      55087.0      0.6     37.4         i+=1
100000      92154.0      0.9     62.6         a =math.sin(i)
```

Listing 8.5 repeats the refactoring example seen in Listing 8.4, this time in Cython. Before the functions declaration, the global variable is defined static using the annotation *cdef*. Although Cython reduces the processing time by about 50% the interpretation of the same procedure, Numba generates faster code at the cost of the initial optimization.

On the other hand, compared to Numba, Cython abstracts less layers hence allowing easier debugging and optimization of the code. However, the creation of additional files (at least one) is needed. As an example, Listing 8.6 revisits the simple low-pass filter implementation seen in Sec. 8.2. After cythonization, which makes available the function *lp* by compiling *low_pass.pyx*, the resulting *low_pass* module can be launched from a Python program:

- **c_low_pass.c** contains the procedure within the C function *lp*.
- **low_pass.pyx** binds the C file to the corresponding Python functions. From it, Cython compiles the corresponding module *low_pass*.
- **test_lp.py** calls the *low_pass* module functions.

For the sake of comparison, the Cython and Numba refactored code of the *OscSine()* object class have been uploaded in GitHub, together with time profiling.

Listing 8.6: Low-pass filter implementation – Cython

```
==============================
c_low_pass.c
==============================
double last_sample = 0;
double b = 0;

void lp(double* data, int samples, double* a){
   b = 1 - a;
   data[0] = a*data[0] + b*last_sample;
   for (int i = 1; i < samples; i++) {
          data[i] = a*data[i] + b*data[i-1];
   }
   last_sample = data[samples-1];
   return;
}

==============================
low_pass.pyx
==============================
import cython
...
#declare interface to C code
cdef extern void lp(double* data, int samples, double* a)

# c-array that contains samples
cdef double *cdata

#function compiled as module
def process(np.ndarray[double] data, int samples, double alfa):
  ...
  lp(cdata, samples, a)
  ...
  return data

==============================
test_lp.py
```

```
import low_pass
...
# ''a'' is the user parameter
def process(data):
    return low_pass.process(data, CHUNK_SIZE, a)
```

8.3.3 Comparisons & Benchmarking

For what we have seen, Numba and Cython target two different approaches to code refactoring. Below, the most important features and differences are briefed.

- In general, code refactoring is easier and faster in Numba than Cython.
- Numba performs well if arithmetic operations are repeatedly computed without communicating with Python objects at runtime.
- Cython code is compiled once (AOT), while Numba optimizes a script at each new execution (JIT).
- Cython requires to write C instructions.
- Cython allows to debug and profile the code after translation.
- Cython allows to embed existing code written in C or other languages.

The computational performances of cythonized functions encoding standard algorithms such as Fibonacci, FASTA, binary-tree visits in different programming languages can be benchmarked, thus measuring processor and memory occupation. For instance, the Computer Language Benchmark Game [216] compares the processing time differences between interpreted Python and C code, highlighting a dramatic performance drop of some Python implementations. Using Pybenchmarks [160] instead, several pairwise comparisons among Python algorithm implementations (e.g. Python3 vs. Numba or Numba vs. Cython) can be performed. In general, Numba requires less computational resources than Cython. On a final note, Numba and Cython are not the only libraries available for code optimization and speedup. A thorough discussion about compiled Python can be found in [89], whereas examples of C functions written for processing sounds using the portaudio device can be found in [189]. Together, such documentary resources provide a broad picture of the pros and cons coming from using interpreted or compiled Python instead of C/C++.

8.4 Applications to Virtual Analog

Virtual analog is a branch of sound processing dealing with the modeling and real-time simulation of analog effects [247]. To this end, a digital simulation of the voltage-controlled filter algorithm, previously shown in Chapter 4, is reported. The code was benchmarked using the *line_profiler* and the *memory_profiler* [182] libraries concerning processing time and memory occupation, respectively.

8.4.1 Voltage-controlled filter

The code development and refactoring of the voltage-controlled filter aboard the EMS VCS3 sound synthesizer is presented. The core part of the algorithm consists of iterating the computation of five nonlinear and five linear equations, until the loop converges to a fixed point solution; at this point the state of the filter is updated and a new input sample v_{IN} can be processed:

```
repeat
    read $v_{IN}$
    repeat
        compute $v^*_{OUT} = v_{OUT}$
        compute $u_1 = \tanh\big((v^*_{OUT} - v_{IN})/2V_T\big)$
        compute $v_{C1} = I_0(u_2 + u_1)/(4CF_S) + s_1$
        compute $u_2 = \tanh\big((v_{C2} - v_{C1})/2V_T\big)$
        compute $v_{C2} = I_0(u_3 - u_2)/(4CF_S) + s_2$
        compute $u_3 = \tanh\big((v_{C3} - v_{C2})/2V_T\big)$
        compute $v_{C3} = I_0(u_4 - u_3)/(4CF_S) + s_3$
        compute $u_4 = \tanh\big((v_{C4} - v_{C3})/2V_T\big)$
        compute $v_{C4} = I_0(u_5 - u_4)/(4CF_S) + s_4$
        compute $u_5 = \tanh(v_{C4}/6\gamma)$
        compute $v_{OUT} = (K + 1/2)v_{C4}$
    until $|v_{OUT} - v^*_{OUT}| < 10^{-4}|v^*_{OUT}|$
    compute $s_1 = v_{C1}/(2F_S) + s_1$
    compute $s_2 = v_{C2}/(2F_S) + s_2$
    compute $s_3 = v_{C3}/(2F_S) + s_3$
    compute $s_4 = v_{C4}/(2F_S) + s_4$
until ever.
```

The translation of this core in C code is straightforward since it contains only the instructions describing the equations. The final software architecture of the Cython refactoring using an external C file consists of *c_vcs.c* and *vcs.pyx*, forming the algorithm library, and the main file *test_vcs.py* that calls the external functions importing the *vcs* module. The code is made available in the Github repository. . A Numba and

Table 8.1: Voltage-controlled filter: mean processing time and memory occupation of different implementations; chunk_size=512; processing time averaged over 1000 chunks. (Real-time limit: 22.6 μs/sample)

	Processing time	**Memory occupation**
Interpreted Python	199.3 μs/sample	1KB
Cython annotations	6.5 μs/sample	16KB
Cython + native C	0.14 μs/sample	12KB
Numba	1.36 μs/sample	12KB

an annotated Cython version of the same algorithm were prepared as well. The related Numba refactoring consists of the creation of a wrap function *compute_numba()*, passing the global variables to the compiled function *compute_numba_low()* that computes the equations. Moreover, the compiler is instructed about the static character of the input and return arguments.

As it can be seen in Table 8.1, Cython is about five times as fast as Numba. This difference highlights the computational advantage for the C application of keeping the filter state within global variables instead of continuously passing their values through arguments, as Numba does instead. Moreover, the benefits of importing few C instructions accelerate Cython by about fifty times as much. In fact, the manual reprogramming of the hyperbolic tangent function using the *math.c* library translates in a notably more efficient implementation of the same code.

In any case, Table 8.1 reports computation times for the program when it is in running state. As anticipated in Sec. 8.1.2, the same program can take much longer to deliver sound samples to the audio interface in the multitasking environment. This fact is discussed in more detail in the next section.

8.5 Discussion

Table 8.2 reports statistics for the voltage-controlled filter implemented using the Cython+native C approach under different OS load conditions. In all cases a chunk size of 64 samples was set. The real-time limit for this chunk is equal to 64 × 22.6

Table 8.2: Statistics for the voltage-controlled filter running at different OS loads, with audio chunks of 64 samples: audio application only, audio application with I/O operations in the Python GUI and audio application with copy of a directory tree in background.

Load	Audio only			GUI operations			File copy		
	Idle	Proc.	Total	Idle	Proc.	Total	Idle	Proc.	Total
Min (ms)	0.005	0.039	0.044	0.005	0.037	0.042	0.005	0.038	0.044
Max (ms)	36.5	8.7	36.8	33.9	31.6	40.9	27.9	29.9	29.9
Avg (ms)	1.31	0.14	1.44	1.32	0.13	1.44	1.35	0.09	1.45
Viol. (%)	11%	0.17%	11.2%	11%	0.8%	11.8%	11%	0.15%	11.2%

μs $\approx$ 1.45 ms. According to the values reported in Table 8.1, the process takes 64 $\times$ 0.14 = 8.96 μs processor time for each chunk. However, in the best case a chunk was computed in about 37 μs in our test environment. The extension of this extra time depends on the priority assigned to the real-time process by the OS. Using the highest load that it was possible to reproduce, the consequent extra time (31.6 ms in Table 8.2) exceeded the real-time limit for this chunk size by no less than 20 times.

Furthermore, Table 8.2 in the last row reports the percentage of violations of the real-time limit, measured as the time between two subsequent callbacks (idle time). Since no buffer underflow and consequent sound artifacts occur in consequence of such violations, certainly the system alternated audio chunks preemption with prioritization of non real-time processes. In fact, portaudio creates a further buffering level hence smoothing out delays due to algorithmic and/or OS bottlenecks, at the cost of introducing additional latency—about 50 ms in our tests. Fig. 8.2 shows that, in practice, few longer delays interleaved with several short idle times. Such delays happen when the Python process waits for the next chunk of samples to become available from the audio driver.

The total latency of the system depends on multiple factors: the audio interface hardware, the audio drivers, the buffering levels introduced by the system and the chunk size. In our test environment, latency grew up to about 500 ms when the voltage-controlled filter was computed using the internal sound card instead of the Babyface. More generally, Python does not acquire special privileges from the OS or address possible priority inversion problems, and even an audio thread must contend with non-real-time Python threads for the interpreter lock. Furthermore, real-time Python programmers should always check the virtual memory occupation of their software as page faults can have a huge impact on real-time performance. Unless these issues are rigorously managed, uninterrupted audio in principle cannot be guaranteed by a Python application.

8.6 Lesson learnt

This chapter made a short tour showing the potential and limits of Python for real-time sound programming, starting with a simple digital filter to a complex virtual analog model. If the interpreter alone has limited application, conversely refactoring critical parts of the code for libraries like Numba or Cython allows for processing also complex algorithms in real time. In particular:

- the code parts dealing with environment setup, type conversions, user interaction, signal analysis and interface development can be managed by the Python interpreter;

- procedures which do not need to intensively interact with the program workspace can be compiled with Numba (e.g., CPU bound loops);

- procedures, which, conversely, must often interact with the program workspace in terms of e.g., memory allocation or signal resampling and filtering, can be compiled with Cython;

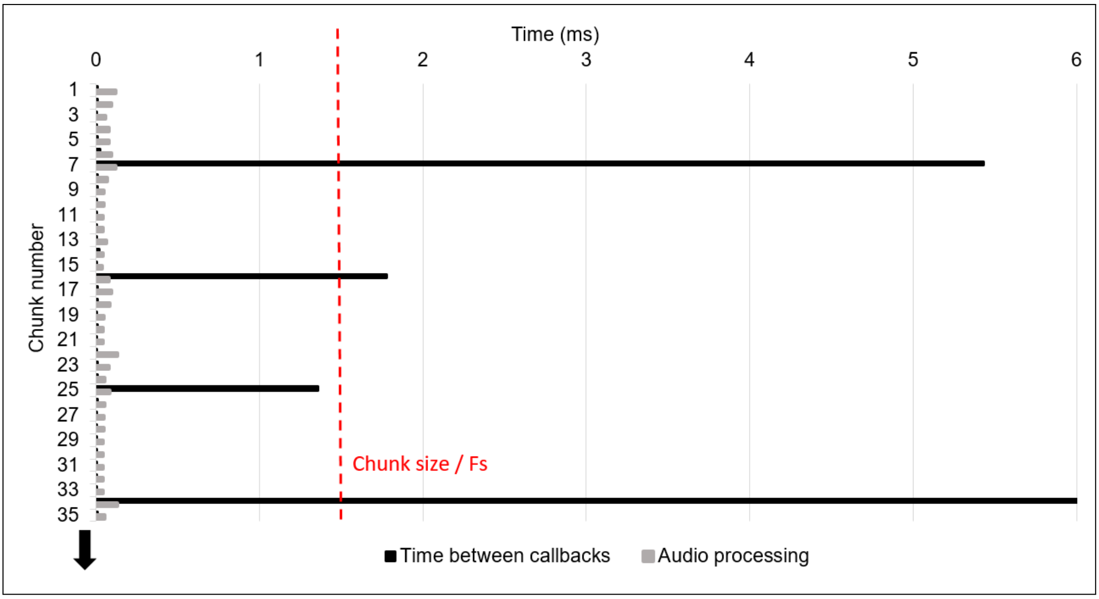


Figure 8.2: Timing of audio callbacks and audio processing in the voltage-controlled filter algorithm.

- finally, already existing procedures written in C or other languages can be efficiently cythonized as well.

Tests of a computationally-intensive sound algorithm were made under different OS loads, leading to accurate audio outputs with a latency of about 50 ms. Unfortunately, lower latency values were out of reach, due to the Python threading mechanism and its interaction with the OS scheduler. This limit prevents from using the resulting applications in contexts where latency is critical (e.g., interactive music). However, Python has certainly potential for integrating sets of computationally-intensive algorithms in frameworks where real-time processing is not bound to low-latency interactive controls.

Conclusions

This thesis showed some effects of multimodal feedback to the user interaction with professional appliances, mostly highlighting the importance of haptic feedback for the reliability and the robustness of the interaction in specialized environments. As a first step, several literature studies on physical/virtual input controllers, multimodal interaction, and material classification were analyzed and discussed. Secondly, research questions which were relevant to the context of professional appliances were investigated throughout rigorous experiments. Finally, based on experimental and literature results, prototypes were designed and validated through user interaction studies.

In the first part of the thesis, three studies (each made of multiple experiments) addressed various aspects related to the rendering of reliable virtual buttons for professional appliances. In particular, the first study assessed the capabilities of humans to classify materials based on unimodal (auditory, tactile) and bimodal feedback in passive conditions. Compared to the literature, the experiments here used ecological stimuli, such as the reproduction of a light ball bouncing on top of different materials (i.e. wood, plastic, and metal). Experimental results suggested that even in the worst conditions, both unimodal auditory and tactile modalities scored greater than chance level, allowing material classification based on the decay of the feedback and its spectral content. On the other hand, the dual modality always gave rise to greater classification scores, suggesting a profitable integration effect between auditory and tactile modalities. The second study further investigated the material classification based on the unimodal tactile modality in active conditions: multiple virtual buttons inspired to ecological materials were designed for an *ad hoc* prototype device, consisting of a touchscreen interface offering rich tactile feedback dependent on the specific normal force applied. Two user panels evaluated several aspects of the tactile feedback associated with various sets of virtual buttons: while the vibrotactile feedback originating from real bouncing events was relatively clearly discriminated, users could more successfully discriminate the buttons designed from the ground up, exploiting the prototype device's response and characteristics. In general, the study suggested that the unimodal vibrotactile feedback can be profitably used to discriminate few virtual buttons rendering ecological stimuli. Consequently, touchscreens providing such haptic feedback can be adopted to improve the user interaction in work environments affected by auditory or visual distractors, such as professional kitchens and laundries. However, in order to strengthen the haptic effects to the most, an effective haptic feedback design must match the inherent resonance frequencies of the haptic device in the frequency region where the human vibration perception is best. To this end, the third study assessed whether the vibration direction (normal, lateral) or the normal force applied to the surface (0.5 N, 4.9 N) affected the sensitivity to haptic feedback. The first experimental results, now limited to sinusoidal vibrations at 250 Hz, showed that the sensitivity to normal vibrations is lower than the sensitivity to lateral vibrations. Moreover, the sensitivity to normal vibrations was found to depend on the pressing force, with increased sensitivities associated to higher pressing force, whereas the sensitivity to lateral vibrations was not. The results of this study suggested that haptic devices with limited power capabilities should, therefore, provide lateral vibrations to maximize the perceived strength of the haptic feedback. These three studies achieved the first goal of the research, drawing a roadmap for developing a touchscreen interface displaying virtual buttons with multimodal feedback. Furthermore, they gave insights on the haptic feedback design and the mechanical optimization of haptic devices. The characterization of the devices used in the above-mentioned experiments, which often required the involvement of human subjects, inspired the development of a robotic tool to simulate the impedance of the human finger while applying different force levels. The robotic tool, aimed at reducing the effort needed to characterize haptic devices, was modeled, parameterized, and validated, reporting a good simulation of the finger impedance in the frequency range 200-400 Hz.

The second part of the thesis addressed a number of research questions about knobs. The first study achieved the second goal of the research: thanks to the investigation of new technologies and hardware configurations, a low-cost knob providing programmable force-feedback was designed, developed and patented. The related user studies proved that, even simple resistive feedback patterns, improved accuracy and precision in the rotary selections. Moreover, the study demonstrated practical applications of the resistive feedback design in the context of professional appliances and multimedia controllers. In accordance with the third research goal, the other studies reported in the second part investigated a novel interaction primitive. The Non-a-knob is the concept design of a new user interface for professional appliances that, without having moving parts, embeds the form factor of a physical knob in the layout of the appliance. A preliminary experiment successfully tested the hypothesis that rotary gestures were eligible to substitute the standard rotations accomplished with physical knobs. The experiment assessed the accuracy, precision, starting position and finger deviations between such two configurations, finding significant differences only regarding finger deviation. Indeed, as opposed to standard rotations, during the rotary gestures each finger slides almost independently from the others. Given this significant variability, the prototypes of motionless knob were controlled by neural networks rather than predefined models. Several configurations and detection technologies were discussed and compared, achieving a working solution. The most promising prototype was evaluated in an experiment that compared the detection performance of the device with the objective data acquired in the preliminary experiment. Although the results showed significant differences between the two experiments in terms of accuracy and precision, such differences depended on the parameters of the control algorithm, being therefore be easily adjustable. Conversely, the control algorithm of the motionless knob was able to encode gestures and rotations in the same fashion, independently of the physical characteristics of the subjects. Eventually, the last part of the study proposed a full concept design that merged together multiple research findings: the motionless knob was ideally merged with the haptic touchscreen described in the first part to create a compact user interface enabling an advanced user interaction with professional appliances.

Future work will encompass the progressive refinement of the prototypes accompanied by rigorous validations carried out with the material and methods developed during this PhD. Indeed, the design and test of rich haptic feedback for a motionless knob will be the natural subsequent step in the development of an interface. The final steps of the research will be about the engineering of the discussed prototypes in terms of mechanical and electronic design. Further research is also needed regarding the materials of the interfaces. The measurement tools and software solutions described in the thesis will be used to investigate yet not considered experimental conditions, extending and strengthening the results of this research.

Complementary activities

During my Ph.D., I accomplished several complementary activities as member of the Advanced Development and Technology (AD&T) team at Electrolux Professional. All those multidisciplinary activities strongly contributed to my personal and professional growth.

Communication Protocols

During the first year of Ph.D., I developed an ASCII based client/server communication protocol over TCP/IP to interact with a COMAU Racer 4 robotic arm. The server on the robot side was configured to accept different type of commands, like: absolute positions, relative positions, point-based trajectories or paramterized trajectories. A demo client was developed in Python and the communication was successfully tested in multiple task. For instance, the system was profitably used to perform flipping actions during food cooking tasks. The control strategies of the trajectories between successive positions were managed by the robot's proprietary planning system.

I also developed several Python clients to communicate with Electrolux Professional machines through proprietary protocols.

Computer Vision

I developed many computer vision algorithms for different applications:

- recognition and estimation of steam emissions in controlled environments to monitor and compare the performances of Electrolux appliances (continuous integration of background/foreground elements)
- object detection of dishes and glasses inside washing baskets applying classic computer vision algorithms (e.g., edge detection, histograms analysis)
- food recognition and tracking during grilling task using a re-trained version of the Mask-R CNN. The algorithm was used to build a dataset of food pose during grilling task, which complementary article is currently under review [183].

Rapid Prototyping

Beside prototyping the user interfaces described in this thesis, I used the platform Arduino to deliver many prototypes to Electrolux Professional regarding data acquisition and data visual representation.

Bibliography

[1] G A. Gescheider, JM Thorpe, J Goodarz, and SJ Bolanowski. The effects of skin temperature on the detection and discrimination of tactile stimulation. *Somatosensory & motor research*, 14(3):181–188, 1997.

[2] Banu Abdikadirova, Evagoras Xydas, and Nurgeldy Praliyev. Effect of frequency level on vibro-tactile sound detection. In *Proc. of the 3rd Int. Conf. on Human Computer Interaction Theory and Applications*, 02 2019.

[3] A. Abdouni, G. Moreau, R. Vargiolu, and H. Zahouani. Static and active tactile perception and touch anisotropy: aging and gender effect. *Scientific Reports*, 8(1):14240, dec 2018.

[4] Alexander T. Adams and Celine Latulipe. Survey of audio programming tools. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 781–786, New York, NY, USA, 2013. ACM.

[5] Jason Alexander, John Hardy, and Stephen Wattam. Characterising the physicality of everyday buttons. *ITS 2014 - Proc. of the 2014 ACM Int. Conf. on Interactive Tabletops and Surfaces*, pages 205–208, 2014.

[6] Emna Amaied, Roberto Vargiolu, Jean Michel Bergheau, and Hassan Zahouani. Aging effect on tactile perception: Experimental and modelling studies. *Wear*, 332:715–724, 2015.

[7] Atmel. Buttons, Sliders and Weels: Sensor Design Guide, accessed June 21, 2021. Available at `https://www.mouser.de/pdfDocs/QTouch_QTAN0079.pdf`.

[8] Bruce Banter. Touch screens and touch surfaces are enriched by haptic force-feedback. *Information Display*, 26(3):26–30, 2010.

[9] Cagatay Basdogan, Frédéric Giraud, Vincent Levesque, and Seungmoon Choi. A review of surface haptics: Enabling tactile effects on touch surfaces. *IEEE Trans. on Haptics*, 2020.

[10] Stefan Battlogg. Magnetorheological transmission device. July 28 2015. U.S. Patent 9,091,309.

[11] Elisabeth Baumgartner, Christiane B. Wiebel, and Karl R. Gegenfurtner. Visual and haptic representations of material properties. *Multisensory research*, 26(5):429–455, 2013.

[12] S Behnel, R Bradshaw, C Citro, L Dalcin, D S Seljebotn, and K Smith. Cython: The best of both worlds. *Computing in Science and Engineering*, 13(2):31–39, March 2011.

[13] Olivier Belanger. Pyo, the python dsp toolbox. In *Proc. of the 24th ACM Int. Conf. on Multimedia*, MM '16, pages 1214–1217, New York, NY, USA, 2016. ACM.

[14] D. Bellini. Expressive digital signal processing (dsp) package for python. `https://github.com/danilobellini/audiolazy`, 2016.

[15] Ross Bencina. Real-time audio programming. `http://www.rossbencina.com/code/real-time-audio-programming-101-time-waits-for-nothing`, 2011.

[16] Nicolai Beni, Marco Grottoli, Francesco Ferrise, and Monica Bordegoni. Rapid prototyping of low cost 1 dof haptic interfaces. In *IEEE Haptics Symposium (HAPTICS)*, pages 479–483. IEEE, 2014.

[17] Sliman J. Bensmaïa and Mark Hollins. Complex tactile waveform discrimination. *The J. of the Acoustical Society of America*, 108(3):1236–1245, 2000.

[18] Sliman J. Bensmaïa, Mark Hollins, and Jeffrey Yau. Vibrotactile intensity and frequency information in the pacinian system: A psychophysical model. *Perception & Psychophysics*, 67(5):828–841, Jul 2005.

[19] Edgar Berdahl and Alexandros Kontogeorgakopoulos. The firefader: Simple, open-source, and reconfigurable haptic force feedback for musicians. *Computer Music Journal*, 37(1):23–34, 2013.

[20] Amir Berrezag and Daniel Büttner. Vibrating actuator, February 26 2019. US Patent 10,218,250.

[21] C.W. Bert. Material damping: An introductory review of mathematic measures and experimental technique. *J. of Sound and Vibration*, 29(2):129 – 153, 1973.

[22] James Biggs and Mandayam A Srinivasan. Tangential versus normal displacements of skin: Relative effectiveness for producing tactile sensations. In *Proc. 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*, pages 121–128. IEEE, 2002.

[23] J Martin Bland and Douglas Altman. Multiple significance tests: the Bonferroni method. *BMJ*, 310, 1995.

[24] Richard A. Bolt. "put-that-there": Voice and gesture at the graphics interface. *SIGGRAPH Comput. Graph.*, 14(3):262–270, July 1980.

[25] C. Bonnet, J. Fauquet Ars, and S. Estaun Ferrer. Reaction times as a measure of uncertainty. *Psicothema*, 20(1):43–48, 2008.

[26] Monica Bordegoni, Francesco Ferrise, and Joseba Lizaranzu. Multimodal interaction with a household appliance based on haptic, audio and visualization. *Proc. IDMME–Virtual Concept*, 2010.

[27] Gary R Bradski. Real time face and object tracking as a component of a perceptual user interface. In *Proc. Fourth IEEE Workshop on Applications of Computer Vision. WACV'98 (Cat. No. 98EX201)*, pages 214–219. IEEE, 1998.

[28] Jean-Pierre Bresciani, Marc O Ernst, Knut Drewing, Guillaume Bouyer, Vincent Maury, and Abderrahmane Kheddar. Feeling what you hear: auditory signals can modulate tactile tap perception. *Experimental brain research*, 162(2):172–180, 2005.

[29] Stephen Brewster, Faraz Chohan, and Lorna Brown. Tactile feedback for mobile interactions. In *Proc. of the SIGCHI Conf. on Human factors in computing systems*, pages 159–162, 2007.

[30] Harry Bunt, Robbert-Jan Beun, and Tijn Borghuis. *Multimodal human-computer communication: systems, techniques, and experiments*, volume 1374. Springer Science & Business Media, 1998.

[31] Paul-Christian Bürkner. Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, 10(1):395–411, 2018.

[32] Umberto Castiello, Bruno L. Giordano, Chiara Begliomini, Caterina Ansuini, and Massimo Grassi. When ears drive hands: The influence of contact sound on reaching to grasp. *Plos ONE*, 5(8):1–9, 08 2010.

[33] Dominique Chapuis, Xavier Michel, Roger Gassert, Chee-Meng Chew, Etienne Burdet, and Hannes Bleuler. A haptic knob with a hybrid ultrasonic motor and powder clutch actuator. In *Joint EuroHaptics Conf. and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC)*, pages 200–205. IEEE, 2007.

[34] Enrico Concettoni and Michael J. Griffin. The apparent mass and mechanical impedance of the hand and the transmission of vibration to the fingers, hand, and arm. *J. of Sound and Vibration*, 325(3):664–678, 2009.

[35] Marco Controzzi, Marco D'Alonzo, Carlo Peccia, Calogero Maria Oddo, Maria Chiara Carrozza, and Christian Cipriani. Bioinspired fingertip for anthropomorphic robotic hands. *Applied Bionics and Biomechanics*, 11(1, 2):25–38, 2014.

[36] José Corujeira, José Luís Silva, and Rodrigo Ventura. Effects of haptic feedback in dual-task teleoperation of a mobile robot. In Regina Bernhaupt, Girish Dalvi, Anirudha Joshi, Devanuj K. Balkrishan, Jacki O'Neill, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2017*, pages 267–286, Cham, 2017. Springer International Publishing.

[37] Keith Edwin Curtis. Knob based gesture system, January 10 2017. US Patent 9,542,009.

[38] Cycling '74. Max/msp website. `https://cycling74.com/products/max/`, 2020.

[39] Cypress. PSoC4 board, accessed July 23, 2021. Available at `https://www.cypress.com/products/32-bit-arm-cortex-m0-psoc-4`.

[40] Jessica D. Ndengue, Ilaria Cesini, Jenny Faucheu, Eric Chatelet, Hassan Zahouani, David Delafosse, and Francesco Massi. Tactile perception and friction-induced vibrations: Discrimination of similarly patterned wood-like surfaces. *IEEE Trans. on Haptics*, 10(3):409–417, July 2017.

[41] A. K. Dabrowska, G. M. Rotaru, S. Derler, F. Spano, M. Camenzind, S. Annaheim, R. Stämpfli, M. Schmid, and R. M. Rossi. Materials used to simulate physical properties of human skin. *Skin Research and Technology*, 22(1):3–14, 2016.

[42] Ralph B. D'Agostino. An omnibus test of normality for moderate and large size samples. *Biometrika*, 58(2):341–348, 1971.

[43] Yuri De Pra and Federico Fontana. Programming real-time sound in python. *Applied Sciences*, 10(12):4214, 2020.

[44] Yuri De Pra and Federico Fontana. Haptic controller with programmable resistive force, 2021. Italian Patent Office (IPO) n. 102021000009068 Status: Pending.

[45] Yuri De Pra, Federico Fontana, Hanna Järveläinen, Stefano Papetti, and Michele Simonato. Does it ping or pong? auditory and tactile classification of materials by bouncing events. *ACM Trans. Appl. Percept.*, 17(2), May 2020.

[46] Yuri De Pra, Federico Fontana, Hanna Järveläinen, Stefano Papetti, Michele Simonato, and Riccardo Furlanetto. Auditory and tactile recognition of resonant material vibrations in a passive task of bouncing perception. In *Int. Workshop on Haptic and Audio Interaction Design-HAID2019*, 2019.

[47] Yuri De Pra, Federico Fontana, Hanna Järveläinen, Stefano Papetti, and Mauro Sonego. A comparison of haptic rotation performance between rotary and motionless knobs. In *Proc. of IEEE HAPTICS SYMPOSIUM '22, Santta Barbara, CA*, 2022. (MANUSCRIPT SUBMITTED FOR PUBLICATION).

[48] Yuri De Pra, Federico Fontana, and Stefano Papetti. Endless knob with programmable resistive force feedback. In Springer International Publishing, editor, *Human-Computer Interaction – INTERACT2021*, 2021.

[49] Yuri De Pra, Federico Fontana, and Stefano Papetti. Interacting with digital audio effects through a haptic knob with programmable resistance. In *Proc. of the 23rd Int. Conf. on Digital Audio Effects (DAFx-20)*, 2021.

[50] Yuri De Pra, Federico Fontana, Stefano Papetti, and Hanna Järveläinen. The non-a-knob user interface. 2022. (MANUSCRIPT DRAFT FOR PUBLICATION).

[51] Yuri De Pra, Federico Fontana, Stefano Papetti, and Michele Simonato. A low-cost endless knob controller with programmable resistive force feedback for multimedia production. In *Proc. of the 17th Sound and Music Computing Conf. (SMC 2020). Turin, Italy*, 2020.

[52] Yuri De Pra, Federico Fontana, and Michele Simonato. Development of real-time audio applications using python. In *Proc. of XX Colloquio di informatica musicale (CIM). Udine, Italy*, 2018.

[53] Yuri De Pra, Stefano Papetti, Federico Fontana, Hanna Järveläinen, and Michele Simonato. Tactile discrimination of material properties: application to virtual buttons for professional appliances. *J. on Multimodal User Interfaces*, 14(3):255–269, 2020.

[54] Yuri De Pra, Stefano Papetti, Federico Fontana, and Michele Simonato. An open-source remote-controllable hardware device for realistic finger-pressing simulation. In *EuroHaptics 2020 Conf., Leiden (NL)*, 2020. Work in Progress Poster(WIP).

[55] Yuri De Pra, Stefano Papetti, Federico Fontana, and Emidio Tiberi. An open-source robotic tool for the simulation of quasi-static finger pressing on stationary and vibrating surfaces. *IEEE Trans. on Haptics*, 2021.

[56] Yuri De Pra, Stefano Papetti, Federico Fontana, and Emidio Tiberi. An open-source robotic tool for the simulation of quasi-static finger pressing on stationary and vibrating surfaces. In *World Haptics Conf. (WHC2021)*, 2021.

[57] Yuri De Pra, Stefano Papetti, Hanna Järveläinen, and Federico Fontana. Sensitivity to normal and tangential vibrations in active touch. *IEEE Trans. on Haptics*, 2021. (MANUSCRIPT DRAFT FOR PUBLICATION).

[58] Yuri De Pra, Stefano Papetti, Eleonora Pippia, Mattia Dal Borgo, Emidio Tiberi, and Federico Fontana. Design, modeling and validation of a haptic device tool simulating finger pressing. 2021. (MANUSCRIPT DRAFT FOR PUBLICATION).

[59] Adele Diederich and Hans Colonius. Bimodal and trimodal multisensory enhancement: effects of stimulus onset and intensity on reaction time. *Perception & psychophysics*, 66(8):1388–1404, 2004.

[60] Derek DiFilippo and Dinesh K. Pai. The ahi: An audio and haptic interface for contact interactions. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[61] Timothy Thomas Diller. Frequency response of human skin in vivo to mechanical stimulation. 2001.

[62] Alan Dix, Alan John Dix, Janet Finlay, Gregory D Abowd, and Russell Beale. *Human-computer interaction.* Pearson Education, 2004.

[63] Allen B. Downey. *Think DSP: Digital Signal Processing in Python.* O'Reilly Media, Inc., 1st edition, 2016.

[64] Pierre Dupont, Vincent Hayward, Brian Armstrong, and Friedhelm Altpeter. Single state elastoplastic friction models. *IEEE Trans. on Automat. Control*, 47(5):787–792, May 2002.

[65] ELK Audio. Audio latency demystified. `https://elk.audio/audio-latency-demystified-part-ii/`, 2020.

[66] Marc O. Ernst and Martin S. Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870):429–433, 2002.

[67] Marc O Ernst and Heinrich H Bülthoff. Merging the senses into a robust percept. *Trends in cognitive sciences*, 8(4):162–169, 2004.

[68] Angelo Farina. Advancements in Impulse Response Measurements by Sine Sweeps. In *Audio Eng. Soc. Conv. 122*. Audio Engineering Society, 2007.

[69] Tamara Fielder and Yasemin Vardar. A Novel Texture Rendering Approach for Electrostatic Displays. In *Int. Workshop on Haptic and Audio Interaction Design - HAID2019*, Lille, France, March 2019.

[70] John B Filson, Eric B Daniels, Adam Mittleman, Sierra L Nelmes, and Yoky Matsuoka. Dynamic distributed-sensor thermostat network for forecasting external events. December 31 2013. U.S. Patent 8,620,841.

[71] Terrence L Fine. *Feedforward neural network methodology.* Springer Science & Business Media, 2006.

[72] Dominique Fober, Yann Orlarey, and Stéphane Letz. Faust architectures design and osc support. 2011.

[73] F. Fontana and E. Bozzo. Explicit fixed-point computation of nonlinear delay-free loop filter networks. *IEEE/ACM Trans. on Audio, Speech and Language Processing*, 26(10):1884–1896, Oct 2018.

[74] Federico Fontana, Ivan Camponogara, Paola Cesari, Matteo Vallicella, and Marco Ruzzenente. An exploration on whole-body and foot-based vibrotactile sensitivity to melodic consonance. In *Proc. of the 13th Sound and Music Computing Conf. (SMC2016)*, pages 143–150, Hamburg, Germany, Aug. 31 - Sep. 3 2016.

[75] Federico Fontana and Marco Civolani. Modeling of the ems vcs3 voltage-controlled filter as a nonlinear filter network. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(4):760–772, May 2010.

[76] Federico Fontana, Stefano Papetti, Hanna Järveläinen, and Federico Avanzini. Detection of keyboard vibrations and effects on perceived piano quality. *The J. of the Acoustical Society of America*, 142(5):2953–2967, 2017.

[77] Bettina Forster, Cristiana Cavina-Pratesi, Salvatore M Aglioti, and Giovanni Berlucchi. Redundant target effect and intersensory facilitation from visual-tactile interactions in simple reaction time. *Experimental brain research*, 143(4):480–487, 2002.

[78] John J. Foxe. Multisensory integration: Frequency tuning of audio-tactile integration. *Current Biology*, 19(9):R373 – R375, 2009.

[79] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. of the American Statistical Association*, 32(200):675–701, 1937.

[80] Rebecca Fenton Friesen, Michael A Peshkin, and J Edward Colgate. Bioinspired artificial fingertips that exhibit friction reduction when subjected to transverse ultrasonic vibrations.

[81] Waka Fujisaki and Shin'ya Nishida. Audio–tactile superiority over visuo–tactile and audio–visual combinations in the temporal resolution of synchrony perception. *Experimental brain research*, 198(2-3):245–259, 2009.

[82] Waka Fujisaki, Midori Tokita, and Kenji Kariya. Perception of the material properties of wood based on vision, audition, and touch. *Vision Research*, 109:185 – 200, 2015. Perception of Material Properties (Part I).

[83] Markus Funk, Juana Heusler, Elif Akcay, Klaus Weiland, and Albrecht Schmidt. Haptic, auditory, or visual? towards optimal error feedback at manual assembly workplaces. In *Proc. of the 9th ACM Int. Conf. on Pervasive Technologies Related to Assistive Environments*, pages 1–6, 2016.

[84] William W. Gaver. Synthesizing auditory icons. In *Proc. of the INTERACT '93 and CHI '93 Conf. on Human Factors in Computing Systems*, CHI '93, pages 228–235, New York, NY, USA, 1993. ACM.

[85] Edouard Gentaz, Gabriel Baud-Bovy, and Marion Luyat. The haptic perception of spatial orientations. *Experimental brain research*, 187(3):331, 2008.

[86] Bruno L. Giordano and Federico Avanzini. *Perception and Synthesis of Sound-Generating Materials*, pages 49–84. Springer London, London, 2014.

[87] Bruno L. Giordano and Stephen McAdams. Material identification of real impact sounds: Effects of size variation in steel, glass, wood, and plexiglass plates. *The J. of the Acoustical Society of America*, 119(2):1171–1181, 2006.

[88] John C Glover, Victor Lazzarini, and Joseph Timoney. Python for audio signal processing. 2011.

[89] Micha Gorelick and Ian Ozsvald. *High Performance Python: Practical Performant Programming for Humans.* " O'Reilly Media, Inc.", 2014.

[90] Griffin Technology. PowerMate multimedia control knob, accessed March 29, 2021. Available at `https://www.iclarified.com/i1112/`.

[91] Steve Guest, Caroline Catmur, Donna Lloyd, and Charles Spence. Audiotactile interactions in roughness perception. *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale*, 146:161–71, 10 2002.

[92] P. Guo. Python is now the most popular introductory teaching language at top u.s. universities. `https://cacm.acm.org/blogs/blog-cacm/176450`, 2014.

[93] Amit Kumar Gupta and SK Mangal. Experimental damping estimation of material by sinusoidal base excitation. *Int. J. of Science and Research*, 4(6):102–108, 2015.

[94] Jong-Eun Ha. Automatic detection of chessboard and its applications. *Optical Engineering*, 48(6):067205, 2009.

[95] Taku Hachisu, Michi Sato, Shogo Fukushima, and Hiroyuki Kajimoto. Hachistick: simulating haptic sensation on tablet pc for musical instruments application. In *Proc.. of the 24th annual ACM symposium adjunct on User interface software and technology*, pages 73–74. ACM, 2011.

[96] Taku Hachisu, Michi Sato, Shogo Fukushima, and Hiroyuki Kajimoto. Augmentation of material property by modulating vibration resulting from tapping. In *Int. Conf. on Human Haptic Sensing and Touch Enabled Computer Applications*, pages 173–180. Springer, 2012.

[97] Aram Z Hajian and Robert D Howe. Identification of the mechanical impedance at the human finger tip. 1997.

[98] John Halloran and Anna Minaeva. Touch and play? investigating the value of touchscreens for gamer experience. *Entertainment Computing*, 32:100312, 2019.

[99] Hyun-Yong Han and Sadao Kawamura. Analysis of stiffness of human fingertip and comparison with artificial fingers. In *IEEE SMC'99 Conf. Proc. 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics (Cat. No. 99CH37028)*, volume 2, pages 800–805. IEEE, 1999.

[100] Christian Hatzfeld, Siran Cao, Mario Kupnik, and Roland Werthschützky. Vibrotactile force perception–absolute and differential thresholds and external influences. *IEEE Trans. on Haptics*, 9(4):586–597, 2016.

[101] Christian Hatzfeld and Thorsten A Kern. *Engineering haptic devices.* Springer, 2016.

[102] Michael Haverkamp. Effects of Material Touch Sounds on Perceived Quality of Surfaces. In *Forum Acusticum*, 2014.

[103] Thomas P. Hettmansperger and Joseph W. McKean. *Robust nonparametric statistical methods.* CRC Press, 2010.

[104] Kosuke Higashi, Shogo Okamoto, Hikaru Nagano, Masashi Konyo, and Yoji Yamada. Vibration-based rendering of virtual hardness: Frequency characteristics of perception. In *Proc. of the 6th Global Conf. on Consumer Electronics (GCCE)*, pages 1–2, October 2017.

[105] Kosuke Higashi, Shogo Okamoto, Hikaru Nagano, and Yoji Yamada. Effects of mechanical parameters on hardness experienced by damped natural vibration stimulation. In *IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 1539–1544, October 2015.

[106] Kosuke Higashi, Shogo Okamoto, Hikaru Nagano, Yoji Yamada, and Masashi Konyo. Hardness perception by tapping: Effect of dynamic stiffness of objects. In *World Haptics Conf. (WHC2017)*, pages 37–41, June 2017.

[107] Kosuke Higashi, Shogo Okamoto, and Yoji Yamada. What is the hardness perceived by tapping? In Fernando Bello, Hiroyuki Kajimoto, and Yon Visell, editors, *Haptics: Perception, Devices, Control, and Applications*, pages 3–12, Cham, 2016. Springer International Publishing.

[108] Hsin-Ni Ho and Lynette A. Jones. Development and evaluation of a thermal display for material identification and discrimination. *ACM Trans. Appl. Percept.*, 4(2), July 2007.

[109] Eve Hoggan, John Williamson, Antti Oulasvirta, Miguel Nacenta, Per Ola Kristensson, and Anu Lehtiö. Multi-touch rotation gestures: Performance and ergonomics. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI '13, page 3047–3050, New York, NY, USA, 2013. Association for Computing Machinery.

[110] Ian P Howard. *Human visual orientation.* John Wiley & Sons, 1982.

[111] Tan Ping Hua, Yeong Che Fai, Ricky Yap, and Eileen Su Lee Ming. Development of a low cost haptic knob. *Int. J. of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 5(10), 2011.

[112] Juan Huang, Darik Gamble, Kristine Sarnlertsophon, Xiaoqin Wang, and Steven Hsiao. Feeling music: Integration of auditory and tactile inputs in musical meter perception. *Plos ONE*, 7(10):1–11, 10 2012.

[113] Charles Hudin, Jose Lozada, and Vincent Hayward. Localized Tactile Feedback on a Transparent Surface Through Time-Reversal Wave Focusing. *IEEE Trans. on Haptics*, PP(99):1, mar 2015.

[114] Jihong Hwang and Wonil Hwang. Vibration perception and excitatory direction for haptic devices. *J. Intell. Manuf.*, 22(1):17–27, 2009.

[115] Native Instruments. Traktor Kontrol S4, accessed August 21, 2021. Available at `https://www.native-instruments.com/en/products/traktor/dj-controllers/traktor-kontrol-s4/`.

[116] Janusz Jabłonowski. Some remarks on teaching of programming. In *Proceedings of the 5th international conference on Computer systems and technologies.* Citeseer, 2004.

[117] RS Johansson and G Westling. Signals in tactile afferents from the fingers eliciting adaptive motor responses during precision grip. *Experimental brain research*, 66(1):141–154, 1987.

[118] Veikko Jousmäki and Riitta Hari. Parchment-skin illusion: sound-biased touch. *Current biology*, 8(6):R190–R191, 1998.

[119] Topi Kaaresoja, Stephen Brewster, and Vuokko Lantz. Towards the temporally perfect virtual button: Touch-feedback simultaneity and perceived quality in mobile touchscreen press interactions. *ACM Trans. Appl. Percept.*, 11(2):9:1–9:25, June 2014.

[120] Astrid ML Kappers. Large systematic deviations in the haptic perception of parallelity. *Perception*, 28(8):1001–1012, 1999.

[121] R. Kern. Line profiler code repository. `https://github.com/rkern/line_profiler`, 2020.

[122] Thorsten A Kern and Roland Werthschützky. Studies of the mechanical impedance of the index finger in multiple dimensions. In *Int. Conf. on Human Haptic Sensing and Touch Enabled Computer Applications*, pages 175–180. Springer, 2008.

[123] Johan Kildal. 3d-press: haptic illusion of compliance when pressing on a rigid surface. In *Int. Conf. on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, pages 1–8, 2010.

[124] Laehyun Kim, Manchul Han, Sang Kyun Shin, and Se Hyung Park. A haptic dial system for multimodal prototyping. In *Int. Conf. on Artificial Reality and Telexistence (ICAT)*, 2008.

[125] Young-Seok Kim and Thenkurussi Kesavadas. Material property recognition by active tapping for fingertip digitizing. In *2006 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 133–139. IEEE, 2005.

[126] Young-Seok Kim and Thenkurussi Kesavadas. Material property recognition by active tapping for fingertip digitizing. In *Proc. of the 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 133–139, March 2006.

[127] Mathias Kirkegaard, C Frisson, and MM Wanderley. Torquetuner: A self contained module for designing rotary haptic force feedback for digital musical instruments. In *Proc. of the 2020 Int. Conf. on New Interfaces for Musical Expression, NIME*, 2020.

[128] Roberta L. Klatzky, Dinesh K. Pai, and Eric P. Krotkov. Perception of material from contact sounds. *Presence: Teleoperators & Virtual Environments*, 9(4):399–410, 2000.

[129] Emilia Koskinen, Topi Kaaresoja, and Pauli Laitinen. Feel-good touch: Finding the most pleasant tactile feedback for a mobile touch screen button. In *Proc. of the 10th Int. Conf. on Multimodal Interfaces*, ICMI '08, pages 297–304, New York, NY, USA, 2008. ACM.

[130] Kathrin Krieger, Alexandra Moringen, Astrid ML Kappers, and Helge Ritter. Influence of shape elements on performance during haptic rotation. In *Int. Conf. on Human Haptic Sensing and Touch Enabled Computer Applications*, pages 125–137. Springer, 2018.

[131] Kathrin Krieger, Alexandra Moringen, and Helge Ritter. Number of fingers and grasping orientation influence human performance during haptic rotation. In *2019 IEEE World Haptics Conf. (WHC)*, pages 79–84. IEEE, 2019.

[132] Sonal Dahiya Krishan Kumar. Programming languages: A survey. *Int. J. on Recent and Innovation Trends in Computing and Communication (IJRITCC)*, pages 307 – 313, 05 2017.

[133] John K. Kruschke. *Doing Bayesian data analysis - A tutorial with R, JAGS, and Stan.* Academic Press, 2nd editio edition, 2014.

[134] Andrew J. Kunkler-Peck and Michael T. Turvey. Hearing shape. *J. of Experimental psychology: human perception and performance*, 26(1):279, 2000.

[135] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In *Proc. of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, 2015.

[136] Ju-Hwan Lee and Charles Spence. Assessing the benefits of multimodal feedback on dual-task performance under demanding conditions. *People and Computers XXII Culture, Creativity, Interaction 22*, pages 185–192, 2008.

[137] Seungyon Lee and Shumin Zhai. The performance of touch screen soft buttons. In *Proc. of the SIGCHI Conf. on human factors in computing systems*, pages 309–318, 2009.

[138] Michael Levin and Alfred Woo. Tactile-feedback solutions for an enhanced user experience. *Information Display*, 25(10):18–21, 2009.

[139] Daniel J. Levitin, Karon MacLean, Max Mathews, Lonny Chu, and Eric Jensen. The perception of cross-modal simultaneity (or "the greenwich observatory problem" revisited). In *AIP Conf. Proc.*, volume 517, pages 323–329. AIP, 2000.

[140] Yi-Chi Liao, Sunjun Kim, and Antti Oulasvirta. One button to rule them all: Rendering arbitrary force-displacement curves. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proc.*, pages 111–113, 2018.

[141] Daniel Linares and Joan López-Moliner. quickpsy: An r package to fit psychometric functions for multiple groups. *The R Journal*, 8(1):122–131, 2016.

[142] S.W. Link. *The wave theory of difference and similarity.* Lawrence Erlbaum Associates, Inc., 1992.

[143] Lars Lischke, Paweł W. Woundefinedniak, Sven Mayer, Andreas Preikschat, and Morten Fjeld. Using variable movement resistance sliders for remote discrete input. In *Proc. of the 2017 ACM Int. Conf. on Interactive Surfaces and Spaces*, ISS '17, page 116–125, New York, NY, USA, 2017. Association for Computing Machinery.

[144] Shuo-Fang Liu, Yu-Shan Chueh, Ching-Fen Chang, Po-Yen Lin, and Hsiang-Sheng Cheng. A study of performance on multi-touch gesture for multi-haptic feedback. In *Int. Conf. on Human-Computer Interaction*, pages 441–449. Springer, 2019.

[145] WR Loewenstein and R Skalak. Mechanical transmission in a pacinian corpuscle. an analysis and a theory. *The J. of physiology*, 182(2):346–378, 1966.

[146] Robert A. Lutfi and Eunmi L. Oh. Auditory discrimination of material changes in a struck-clamped bar. *The J. of the Acoustical Society of America*, 102(6):3647–3656, 1997.

[147] Robert A. Lutfi and Christophe N.J. Stoelinga. Sensory constraints on auditory identification of the material and geometric properties of struck bars. *The J. of the Acoustical Society of America*, 127(1):350–360, 2010.

[148] Marion Luyat, Christine Moroni, and Edouard Gentaz. The role of contextual cues in the haptic perception of orientations and the oblique effect. *Psychonomic bulletin & review*, 12(4):760–766, 2005.

[149] Jani Lylykangas, Veikko Surakka, Katri Salminen, Jukka Raisamo, Pauli Laitinen, Kasper Rönning, and Roope Raisamo. Designing tactile feedback for piezo buttons. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 3281–3284, 2011.

[150] Philippe P. Maeder, Reto A. Meuli, Michela Adriani, Anne Bellmann, Eleonora Fornari, Jean-Philippe Thiran, Antoine Pittet, and Stéphanie Clarke. Distinct pathways involved in sound recognition and localization: A human fmri study. *NeuroImage*, 14(4):802 – 816, 2001.

[151] Rodrigo Martín, Michael Weinmann, and Matthias B. Hullin. A study of material sonification in touchscreen devices. In *Proc. of the 2018 ACM Int. Conf. on Interactive Surfaces and Spaces*, pages 305–310. ACM, 2018.

[152] M. Mason, G. Cooper, and M.D. Raadt. Trends in introductory programming courses in australian universities languages, environments and pedagogy. In M.D. Raadt and A. Carbone, editors, *Australasian Computing Education Conf. (ACE2012)*, volume 123 of *CRPIT*, pages 33–42, Melbourne, Australia, 2012. ACS.

[153] Max V Mathews, Joan E Miller, F Richard Moore, John R Pierce, and Jean-Claude Risset. *The technology of computer music*, volume 5. MIT press Cambridge, MA, 1969.

[154] James McCartney. Rethinking the computer music language: Supercollider. *Comput. Music J.*, 26(4):61–68, dec 2002.

[155] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto. librosa: Audio and Music Signal Analysis in Python. In Kathryn Huff and James Bergstra, editors, *Proceedings of the 14th Python in Science Conference*, pages 18 – 25, 2015.

[156] Andrew P McPherson, Robert H Jack, Giulio Moro, et al. Action-sound latency: Are our tools fast enough? 2016.

[157] Andrew P McPherson and Koray Tahıroğlu. Idiomatic patterns and aesthetic influence in computer music languages. *Organised Sound*, 25(1):53–63, 2020.

[158] Gerrit Meixner, Carina Häcker, Björn Decker, Simon Gerlach, Anne Hess, Konstantin Holl, Alexander Klaus, Daniel Lüddecke, Daniel Mauser, Marius Orfgen, et al. Retrospective and future automotive infotainment systems—100 years of user interface evolution. In *Automotive user interfaces*, pages 3–53. Springer, 2017.

[159] Microsoft. Surface Dial, accessed March 29, 2021. Available at `https://support.microsoft.com/it-it/help/4036279/surface-meet-surface-dial`.

[160] Daniel Milde. Benchmark numba vs. cython. `https://pybenchmarks.org/u64q/numba.php`, 2020.

[161] Theodore E Milner and David W Franklin. Characterization of multijoint finger stiffness: dependence on finger posture and force direction. *IEEE Trans. on Biomedical Engineering*, 45(11):1363–1375, 1998.

[162] S. K. Mitra. *Digital Signal Processing. A computer-Based Approach.* McGraw-Hill, New York, 1998.

[163] Toshisuke Miwa. Evaluation methods for vibration effect. Part 3. Measurements of threshold and equal sensation contours on hand for vertical and horizontal sinusoidal vibrations. *Ind. Health*, 5(3-4):213–220, 1967.

[164] Fabio Morreale and Andrew P. McPherson. Design for longevity: Ongoing use of instruments from NIME 2010–14. In *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME'17)*, pages 192–197, Copenhagen, Denmark, May 15–19 2017.

[165] Fabio Morreale, Giulio Moro, Alan Chamberlain, Steve Benford, and Andrew P McPherson. Building a maker community around an open hardware platform. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 6948–6959, 2017.

[166] Nobuaki Nakazawa, Ryojun Ikeura, and Hikaru Inooka. Characteristics of human fingertips in the shearing direction. *Biological cybernetics*, 82(3):207–214, 2000.

[167] Neil A. Macmillan and C. Douglas Creelman. *Detection Theory: A User's Guide.* Lawrence Erlbaum Associates, 2nd edition, 2005.

[168] M. Obrist, E. Gatti, E. Maggioni, C. T. Vi, and C. Velasco. Multisensory experiences in hci. *IEEE MultiMedia*, 24(2):9–13, 2017.

[169] Seungjae Oh and Seungmoon Choi. Effects of Contact Force and Vibration Frequency on Vibrotactile Sensitivity During Active Touch. *IEEE Trans. on Haptics*, pages 1–1, 2019.

[170] Allison Okamura, Mark Cutkosky, and Jack Dennerlein. Reality-based models for vibration feedback in virtual environments. *IEEE/ASME Trans. on Mechatronics*, 6:245–252, 09 2001.

[171] Ryuta Okazaki, Taku Hachisu, Michi Sato, Shogo Fukushima, Vincent Hayward, and Hiroyuki Kajimoto. Judged consonance of tactile and auditory frequencies. In *World Haptics Conf. (WHC2013)*, pages 663–666, April 2013.

[172] Halla B. Olafsdottir, Theophanis Tsandilas, and Caroline Appert. Prospective motor control on tabletops: Planning grasp for multitouch interaction. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI '14, page 2893–2902, New York, NY, USA, 2014. Association for Computing Machinery.

[173] Kjell Ivar Øvergård, Knut Inge Fostervold, Hans Vanhauwaert Bjelland, and Thomas Hoff. Knobology in use: an experimental evaluation of ergonomics recommendations. *Ergonomics*, 50(5):694–705, 2007.

[174] Krista E. Overvliet and Salvador Soto-Faraco. I can't believe this isn't wood! an investigation in the perception of naturalness. *Acta Psychologica*, 136(1):95 – 111, 2011.

[175] Lucie Pantera, Charles Hudin, and Sabrina Panëels. Two-point haptic pattern recognition with the inverse filter method. In *Int. Conf. on Human Haptic Sensing and Touch Enabled Computer Applications*, pages 545–553. Springer, 2020.

[176] S. Papetti, H. Järveläinen, B. L. Giordano, S. Schiesser, and M. Fröhlich. Vibrotactile sensitivity in active touch: Effect of pressing force. *IEEE Trans. on Haptics*, 10(1):113–122, 1 2017.

[177] Michel Paré, Heather Carnahan, and Allan M. Smith. Magnitude estimation of tangential force applied to the fingerpad. *Exp. Brain Res.*, 142(3):342–348, 2002.

[178] Chaeyong Park, Jinhyuk Yoon, Seungjae Oh, and Seungmoon Choi. Augmenting physical buttons with vibrotactile feedback for programmable feels. In *Proc. of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 924–937, 2020.

[179] Wanjoo Park, Laehyun Kim, Hyunchul Cho, and Sehyung Park. Dial-based game interface with multi-modal feedback. In *Int. Conf. on Entertainment Computing*, pages 389–396. Springer, 2010.

[180] Sylvain Pauchet, Catherine Letondal, Jean-Luc Vinot, Mickaël Causse, Mathieu Cousy, Valentin Becquet, and Guillaume Crouzet. Gazeform: dynamic gaze-adaptive touch surface for eyes-free interaction in airliner cockpits. In *Proc. of the Designing Interactive Systems Conf. (DIS)*, pages 1193–1205. ACM, 2018.

[181] D. T.V. Pawluk and R. D. Howe. Dynamic contact of the human fingerpad against a flat surface. *J. of Biomechanical Engineering*, 121(6):605–611, 1999.

[182] F. Pedregosa. Memory profiler code repository. `https://github.com/pythonprofilers/memory_profiler`, 2020.

[183] Debora Pereira, Yuri De Pra, Emidio Tiberi, Vito Monaco, Paolo Dario, and Gastone Ciuti. Flipping food during grilling tasks: a dataset of utensils kinematics and dynamics, food pose and subject gaze. *Scientific Data*, 2021. Under review, minor reviews.

[184] H. Pham. Pyaudio website. `https://people.csail.mit.edu/hubert/pyaudio`, 2006.

[185] H. Piéron. Recherches sur les lois de variation des temps de laten- ce sensorielle en fonction des intensités excitatrices. *L'Année Psychol.*, 20:17–96, 1914.

[186] H. Piéron. *The Sensations.* Yale University Press, New Haven, CT, 1950.

[187] Matthew S Prewett, Liuquin Yang, Frederick RB Stilson, Ashley A Gray, Michael D Coovert, Jennifer Burke, Elizabeth Redden, and Linda R Elliot. The benefits of multimodal information: a meta-analysis comparing visual and visual-tactile feedback. In *Proc. of the 8th Int. Conf. on Multimodal interfaces*, pages 333–338, 2006.

[188] Miller Puckette. Pure data: another integrated computer music environment. In *Int. Computer Music Conf.*, pages 37–41, 1996.

[189] Eddie Ringle. portaudio. `https://github.com/EddieRingle/portaudio/blob/master/examples/`, 2020.

[190] Tony Ro, Johanan Hsu, Nafi E. Yasar, L. Caitlin Elmore, and Michael S. Beauchamp. Sound enhances touch perception. *Experimental Brain Research*, 195(1):135–143, May 2009.

[191] Marco Romagnoli, Federico Fontana, and Ratna Sarkar. Vibrotactile recognition by western and indian population groups of traditional musical scales played with the harmonium. In Eric W. Cooper, Victor V. Kryssanov, Hitoshi Ogawa, and Stephen Brewster, editors, *Haptic and Audio Interaction Design*, pages 91–100, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[192] Frank A. Russo, Paolo Ammirante, and Deborah I. Fels. Vibrotactile discrimination of musical timbre. *J. of Experimental Psychology: Human Perception and Performance*, 38(4):822, 2012.

[193] Bushra Sadia, Senem Ezgi Emgin, T. Metin Sezgin, and Cagatay Basdogan. Data-driven vibrotactile rendering of digital buttons on touchscreens. *Int. J. of Human Computer Studies*, 135(November):102363, 2020.

[194] Andreas S Schmelt, Viktor Hofmann, Eike C Fischer, Marc C Wurz, and Jens Twiefel. Design and characterization of the lateral actuator of a bimodal tactile display with two excitation directions. *Displays*, 54:34–46, 2018.

[195] Martin Schürmann, Gina Caetano, Yevhen Hlushchuk, Veikko Jousmäki, and Riitta Hari. Touch activates human auditory cortex. *Neuroimage*, 30(4):1325–1331, 2006.

[196] Martin Schürmann, Gina Caetano, Veikko Jousmäki, and Riitta Hari. Hands help hearing: facilitatory audiotactile interaction at low sound-intensity levels. *The J. of the Acoustical Society of America*, 115(2):830–832, 2004.

[197] Thomas Sednaoui, Eric Vezzoli, Brigida Dzidek, Betty Lemaire-Semail, Cedrick Chappaz, and Michael Adams. Experimental evaluation of friction reduction in ultrasonic devices. In *2015 IEEE World Haptics Conf. (WHC)*, pages 37–42, 2015.

[198] Irene Senna, Angelo Maravita, Nadia Bolognini, and Cesare V. Parise. The marble-hand illusion. *Plos ONE*, 9(3), 2014.

[199] S. Sepúlveda, P. Reyes, and A. Weinstein. Visualizing physiological signals in real-time. In Kathryn Huff and James Bergstra, editors, *Proceedings of the 14th Python in Science Conference*, pages 190 – 194, 2015.

[200] Elaine R. Serina, Eric Mockensturm, C. D. Mote, and David Rempel. A structural model of the forced compression of the fingertip pulp. *J. of Biomechanics*, 31(7):639–646, 1998.

[201] Sunghwan Shin, In Lee, Hojin Lee, Gabjong Han, Kyungpyo Hong, Sunghoon Yim, Jongwon Lee, YoungJin Park, Byeong Ki Kang, Dae Ho Ryoo, et al. Haptic simulation of refrigerator door. In *IEEE Haptics Symposium (HAPTICS)*, pages 147–154. IEEE, 2012.

[202] C. D. Shultz, M. A. Peshkin, and J. E. Colgate. Surface haptics via electroadhesion: Expanding electrovibration with johnsen and rahbek. In *2015 IEEE World Haptics Conf. (WHC)*, pages 57–62, 2015.

[203] Sinclair, Taylor, and S J Hobbs. Alpha Level Adjustments for Multiple Dependent Variable Analyses and Their Applicability: A Review. *Int. J. Sport. Sci. Eng.*, 07(01):17–20, 2013.

[204] Sean E. Smith, Karen L. Stephan, and Simon P. Parker. Auditory warnings in the military cockpit: A preliminary evaluation of potential sound types. Technical report, Defense Technical Information Center, Fort Belvoir, Virginia, 2004.

[205] Salvador Soto-Faraco and Gustavo Deco. Multisensory contributions to the perception of vibrotactile events. *Behavioural Brain Research*, 196(2):145 – 154, 2009.

[206] Salvador Soto-Faraco, Charles Spence, and Alan Kingstone. Congruency effects between auditory and tactile motion: extending the phenomenon of cross-modal dynamic capture. *Cognitive, Affective, & Behavioral Neuroscience*, 4(2):208–217, 2004.

[207] Charles Spence and Alberto Gallace. Multisensory design: Reaching out to touch the consumer. *Psychology & Marketing*, 28(3):267–308, 2011.

[208] Peter Spevak. Enhancing sensitivity and robustness of mechanical rotation and position detection with capacitive sensors, May 14 2019. US Patent 10,288,658.

[209] B Stark, T Carlstedt, RG Hallin, and M Risling. Distribution of human pacinian corpuscles in the hand: a cadaver study. *J. of Hand Surgery*, 23(3):370–372, 1998.

[210] Tanya Stivers and Jack Sidnell. Introduction: multimodal interaction. 2005.

[211] F.R. Stöter. Repository of python scientific audio packages. `https://github.com/faroit/awesome-python-scientific-audio`, 2020.

[212] Bjarne Stroustrup. Thriving in a crowded and changing world: C++ 2006–2020. *Proceedings of the ACM on Programming Languages*, 4(HOPL):1–168, 2020.

[213] Xianghong Sun, Tom Plocher, and Weina Qu. An empirical study on the smallest comfortable button/icon size on touch screen. In *Int. Conf. on Usability and Internationalization*, pages 615–621. Springer, 2007.

[214] Yue Hang Tan, Poh Kiat Ng, Adi Saptari, and Kian Siong Jee. Ergonomics aspects of knob designs: A literature review. *Theoretical Issues in Ergonomics Science*, 16(1):86–98, 2015.

[215] Ernst Terhardt, Gerhard Stoll, and Manfred Seewann. Algorithm for extraction of pitch and pitch salience from complex tonal signals. *The J. of the Acoustical Society of America*, 71(3):679–688, 1982.

[216] The Computer Language Benchmarks Game. Benchmark python vs. c++. `https://benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/gpp-python3.html`, 2020.

[217] Sarah Elizabeth Tomlinson. *Understanding the friction between human fingers and contacting surfaces.* PhD thesis, University of Sheffield, 2009.

[218] SE Tomlinson, R Lewis, and MJ Carré. The effect of normal force and roughness on friction in human finger contact. *Wear*, 267(5-8):1311–1318, 2009.

[219] Bernhard Treutwein and Hans Strasburger. Fitting the psychometric function. *Percept. Psychophys.*, 61(1):87–106, 1999.

[220] Ya-Yeh Tsai and Su-Ling Yeh. Freezing effect in tactile perception: Sound facilitates tactile identification by enhancing intensity but not duration. *J. of Experimental Psychology: Human Perception and Performance*, 39(4):925, 2013.

[221] Simon Tucker and Guy J. Brown. Investigating the perception of the size, shape and material of damped and free vibrating plates. Technical report, 2002. Web accessed on August 21, 2021.

[222] Matthew Turk. Multimodal interaction: A review. *Pattern Recognition Letters*, 36:189–195, 2014.

[223] UDOO. UDOO quad board specifications, accessed August 21, 2021. Available at `https://https://www.udoo.org/udoo-quad-dual/`.

[224] Chris Ullrich and Manuel Cruz. Perceived strength of lateral acceleration display for touchscreen emulation of mechanical switch transients. In *Int. Conf. on Human Haptic Sensing and Touch Enabled Computer Applications*, pages 331–336. Springer, 2008.

[225] Nuno Valverde, AMR Ribeiro, Elsa Henriques, and Mihail Fontul. An engineering perspective on the quality of the automotive push-buttons' haptic feedback in optimal and suboptimal interactions. *J. of Engineering Design*, 30(8-9):336–367, 2019.

[226] Anke van Oosterhout and Eve Hoggan. Reshaping interaction with rotary knobs: Combining form, feel and function. In *Proc. of the 2020 ACM Designing Interactive Systems Conf.*, pages 1973–1982, 2020.

[227] D. Vercoe, B.; Ellis. Real-time csound: Software synthesis with sensing and control. In *ICMC'90 Conf. Proc.*, pages 209–211. Int. Computer Music Association, 1990.

[228] Ronald T Verrillo. Temporal summation in vibrotactile sensitivity. *The J. of the Acoustical Society of America*, 37(5):843–846, 1965.

[229] Ronald T Verrillo. Comparison of vibrotactile threshold and suprathreshold responses in men and women. *Perception & psychophysics*, 26(1):20–24, 1979.

[230] Ronald T Verrillo. Vibration sensation in humans. *Music Perception: An Interdisciplinary Journal*, 9(3):281–302, 1992.

[231] Eric Vezzoli, Zlatko Vidrih, Vincenzo Giamundo, Betty Lemaire-Semail, Frédéric Giraud, Tomaz Rodic, Djordje Peric, and Michael Adams. Friction reduction through ultrasonic vibration part 1: Modelling intermittent contact. *IEEE Trans. on Haptics*, 10(2):196–207, 2017.

[232] MA Vitrani, J Nikitczuk, G Morel, C Mavroidis, and B Weinberg. Torque control of electrorheological fluidic resistive actuators for haptic vehicular instrument controls. In *Proc. of the IEEE Int. Conf. on Robotics & Automation*, 2006.

[233] Simon Voelker, Kjell Ivar Øvergård, Chat Wacharamanotham, and Jan Borchers. Knobology revisited: A comparison of user performance between tangible and virtual rotary knobs. In *Proc. of the Int. Conf. on Interactive Tabletops & Surfaces (ISS)*, pages 35–38. ACM, 2015.

[234] Michael Vorländer. Acoustic load on the ear caused by headphones. *The J. of the Acoustical Society of America*, 107(4):2082–2088, 2000.

[235] Ge Wang and Perry R. Cook. Chuck: A concurrent, on-the-fly, audio programming language. In *ICMC*, 2003.

[236] Malte Weiss, Julie Wagner, Yvonne Jansen, Roger Jennings, Ramsin Khoshabeh, James D. Hollan, and Jan Borchers. *SLAP Widgets: Bridging the Gap between Virtual and Physical Controls on Tabletops*, page 481–490. Association for Computing Machinery, New York, NY, USA, 2009.

[237] Nicholas Wettels, Veronica J Santos, Roland S Johansson, and Gerald E Loeb. Biomimetic tactile sensor array. *Advanced Robotics*, 22(8):829–849, 2008.

[238] M. Wickert. Real-Time Digital Signal Processing Using pyaudio_helper and the ipywidgets. In *Proc. of the 17th Python in Science Conference*, pages 91 – 98, 2018.

[239] Michael Wiertlewski and Vincent Hayward. Mechanical behavior of the fingertip in the range of frequencies and displacements relevant to touch. *J. of biomechanics*, 45(11):1869–1874, 2012.

[240] E Courtenay Wilson, Charlotte M Reed, and Louis D Braida. Integration of auditory and vibrotactile stimuli: effects of phase and stimulus-onset asynchrony. *The J. of the Acoustical Society of America*, 126(4):1960–1974, 2009.

[241] E Courtenay Wilson, Charlotte M Reed, and Louis D Braida. Integration of auditory and vibrotactile stimuli: Effects of frequency. *The J. of the Acoustical Society of America*, 127(5):3044–3059, 2010.

[242] John Z Wu, Daniel E Welcome, Kristine Krajnak, and Ren G Dong. Finite element analysis of the penetrations of shear and normal vibrations into the soft tissues in a fingertip. *Medical engineering & physics*, 29(6):718–727, 2007.

[243] Jeffrey M. Yau, Jonathon B. Olenczak, John F. Dammann, and Sliman J. Bensmaïa. Temporal frequency channels are linked across audition and touch. *Current Biology*, 19(7):561 – 566, 2009.

[244] Loutfouz Zaman, Daniel Natapov, and Robert J Teather. Touchscreens vs. traditional controllers in handheld gaming. In *Proc. of the Int. Academic Conf. on the future of game design and technology*, pages 183–190, 2010.

[245] Massimiliano Zampini and Charles Spence. The role of auditory cues in modulating the perceived crispness and staleness of potato chips. *J. of Sensory Studies*, 19(5):347–363, 2004.

[246] Lu Zhao, Yue Liu, Zhuoluo Ma, and Yongtian Wang. Design and evaluation of a texture rendering method for electrostatic tactile display. In *Extended Abstracts of the 2019 CHI Conf. on Human Factors in Computing Systems*, CHI EA '19, pages LBW2314:1–LBW2314:6, New York, NY, USA, 2019. ACM.

[247] Udo Zölzer, Xavier Amatriain, Daniel Arfib, Jordi Bonada, Giovanni De Poli, Pierre Dutilleux, Gianpaolo Evangelista, Florian Keiler, Alex Loscos, Davide Rocchesso, et al. *DAFX: Digital Audio Effects.* John Wiley & Sons, 2002.