**UNIVERSITÀ DEGLI STUDI**
**DI MILANO**

**DIPARTIMENTO DI**
**INFORMATICA "Giovanni**
**Degli Antoni"**

## Corso di Dottorato in Informatica

**Settore Scientifico-Disciplinare: INF/01**
**XXXIV Ciclo**

# Accessible Digital Musical Instruments for Quadriplegic Musicians

### *Analysis, design theory, development and testing of hands-free digital musical instruments*

*Nicola Davanzo*

**Tutor**: *Prof. Federico Avanzini*

**Coordinatore del corso**: *Prof. Paolo Boldi*

A.A. 2020/2021

Laboratorio di
Informatica
Musicale

# Abstract

This thesis explores a particular research topic in the field of Sound and Music Computing, dedicated to the creation of Accessible Digital Musical Instruments (ADMIs) designed for users affected by quadriplegia or similar motor impairments. With such conditions an user is completely paralyzed from the neck down. The impossibility to control the upper and lower limbs, particularly fingers, makes it impossible for such users to play conventional musical instruments, both acoustic and digital. This makes it necessary to introduce specific and non-trivial design and development solutions. A first part of the work is dedicated to the analysis of the related context. After defining the pertinent jargon, an analysis of different physical interaction channels available to people with quadriplegic disabilities is provided, with a review of the sensors suitable for their detection. Some of these channels are then evaluated through an experimental methodology. Further chapters provide an analysis of the state of the art in ADMIs for quadriplegic users, as well as design tools dedicated to ADMIs in general. A chapter is dedicated to the design of musical interfaces controlled through gaze, one of the most employed channels in this context. The second part describes the design, development and testing of new ADMIs suitable for quadriplegic users. The implementation of a software library for developing or fast-prototyping software instruments is described, as well as two Open-Source Hardware sensor peripherals developed ad-hoc for the detection of breath and head rotation. The remaining chapters describe the design, implementation and evaluation of Netytar, Netychords and Resin, three ADMIs that are played through the detection of gaze, breath, head movement, and stimulated resonances in the upper vocal tract, therefore playable hands-free.

*Whenever I have dealt with people with motor disabilities, I felt this: our body is merely an interface to interact with the outside world. We sometimes erroneusly call "able-bodied" those who have that interface working, according to normality criteria. Yet we try every day to augment our interfaces, our ways to communicate or to create new ones through technological means. A person with motor disabilities can simply make use of another type of interface, through which their inner world can express itself in the outer. They simply need different devices. In the end, since the dawn of humanity, we all strive to augment ourselves through tools.*

(One of the personal thoughts which
carried me along my Ph. D. studies)

# Acknowledgements

No other place so far as the university made me understand how interdependent we human creatures are. As introverted we can be during some times of our lives, to think that even just an event, an experience, an objective and its accomplishment may be independent from the connection with the people who surround us, is misleading. Sometimes we are not even conscious that our cultural substrate allows us to be and express ourselves. I think we are never really alone, even if sometimes we feel like. There were times when I was more like a "wild bear" (a pretty Italian figure of speech, probably) than I am now, as the idea to give up everything about computer science and live off the grid by growing beans on some mountain pasture really caressed my thoughts, I have to admit. If at some point in my life this direction totally changed to the desire to open more and more towards loving our world this is particularly due to university experience and what came with it.

Many times I thought that one of my few merits is my messy creativity, a virtue that could never have been expressed had it not been for people with just as much virtue to understand and channel it. The story of the path which lead to the realization of this thesis begins with my father and mother, the former for giving me the opportunity to understand that computers were a wonderful creation tool, the latter for proudly looking at the deformed arm prosthetics I built when I was a child using paper, cardboard and rubber bands. It has continued with my professors at the University of Pavia (in particular Prof. Marco Porta and Mauro Mosconi) who have seen my ideas for designing a musical instrument dedicated to people with quadriplegic disabilities, and instead of simply telling me *"please, just commit to one of the proposed projects and work on something less ambitious"*, they saw potential in them. When I presented myself at the University of Milan with the same project, Prof. Federico Avanzini allowed me to continue to develop it by supervising my Ph.D. studies, who in some periods followed me as I would have never expected from a tutor, sometimes supporting me beyond what is required by a work context. This despite the fact that my project was new in *Laboratorio di Informatica Musicale*, which required him to study topics from a new field together with me.

In this place I also had the opportunity to know better my colleague Federico Simonetta. Although we were already fellow students at the University of Pavia, only with the Ph.D. studies experience we had the opportunity to discover each other. There I found the motivating collaboration of Matteo De Filippis, who showed interest in my research project by dedicating his thesis to the development of one of the instruments described in this thesis, and the inspiring presence, passion, and hyperactivity of Mael Vena. Maybe the greatest interest came from Davys Moreno, an incredible person I still can't believe has traveled and given so much time for the common ideals we have set for ourselves. Despite (or maybe because of) our different energy levels, he gave me a way to rediscover the meaning of dedication and commitment.

My love for knowledge and my willingness to cultivate different aspects of my life are largely due to the affection of Andrea Galvan, a person with whom I have shared so many wonderful thoughts that to call him a friend is reductive. Friends and former Ph.D. students like Marco Clementi spent entire days with me discussing philosophical, scientific and moral issues, increasing the interest for what I was doing. To remind me how much imagination can produce wonderful things, sometimes far from rationality and pragmatism, there is the group of mystical characters I play Dungeons & Dragons with: a handful of introverted freaks who perfectly knows that in some periods they acted almost like a second family, even if half of them would call this wording "cringe". My days were often spent in the company of one of the few people with whom I can spend entire afternoons carefree, who has influenced me in my enthusiasm for science and in the ability to see things objectively and rationally. A strong mind with a passion for computer science, statistics and probability: Daniele Zago. The same levity of spirit characterized my life with a person who has seen all my best and worst moments, and has managed to love me no matter what. With no other person in the world are my thoughts as connected as with Alice Cimino.

And why not to close the acknowledgements with something that not everyone will be able to grasp? Those who do will probably smile, knowing that this comes from the depth of my thoughts and feelings. Thanks to Electraline, for producing tons of insulating tape, without which the concrete products of my Ph.D. studies could never have taken shape.

Last but not least, I would be hypocritical if I didn't also thank a woman from the far Kazakhstan who has been in my thoughts the most during these years, second only to my girlfriend. A figure who gave her life to stir the waters of a knowledge sharing system which I consider inefficient compared to what it could be, that inevitably ends up damaging small people. In a just world, where knowledge has the place it deserves, there would be no need for her incredible work.

# Preface

## Context

Research in the field of digital musical instruments has undergone substantial developments during the last decades. Due to the exponential increase in computational power, miniaturization and availability of electronic sensing technologies, research on such instruments has expanded over the past two decades into the use of unconventional interfaces, interaction paradigms and channels. Such instruments are less constrained by physical limitations than their acoustic and traditional counterparts: this allowed for the exploration of new expressive possibilities and led to the need of partially revising what we culturally consider a musical instrument.

One of the possibilities offered by a digital instrument is to increase the accessibility of the world of musical performance, extending it to people with important motor disabilities such as quadriplegic users. The desire to undertake a research project on this topic arises from the limited availability of musical technologies dedicated to them, both in the scientific literature and on the market. These must exploit interaction methods which avoid the use of hands or fingers for interfacing, which are arguably the main means to interact with traditional acoustic instruments.

Although this development goal may potentially run into several design limitations, it certainly represents a challenge in several research areas. The project falls within the area known internationally as Sound and Music Computing (SMC). The definition of this discipline in the context of the fundamental areas of computer science, and in particular Human-Computer Interaction (HCI), dates back to the 1990s (e.g. *ACM Computing Classification System - CCS1998, entry H.5.5 [Information Interfaces and Presentation]: Sound and Music Computing*). Sound represents a particularly interesting case study regarding many fundamental aspects of information processing, at various levels of abstraction: physical, symbolic, semantic. The *SMC Roadmap*[1] indicates how issues related to interacting with sound and music are part of the current list of challenges to be addressed.

The project, carried out for the entire duration of my doctoral studies, began at the Computer Vision Lab at University of Pavia (Italy) to obtain the title of Master's Degree, and was then placed in the research context of the Laboratorio di Informatica Musicale *(Laboratory of Musical Informatics)* at the University of Milan (Italy), where I carried out my Ph.D. studies.

---

[1] SMC Roadmap, on the SMC network official website: *https://smcnetwork.org/roadmap*

## Hypotheses and objectives

The aim of this thesis is to provide elements of theory and practice in the design, development and evaluation of accessible musical instruments dedicated to people with quadriplegic paralysis. The main assumptions with which the research work was conducted were as follows:

*a*) several under-exploited physical interaction channels are suitable for developing interfaces dedicated to people with quadriplegic disabilities, which therefore do not require the use of hands;
*b*) it is possible to study and develop new interaction modalities which exploit those channels;
*c*) through digital technologies, it is possible to create accessible musical instruments designed for musical performance which can be played by people with quadriplegic disabilities, exploiting the aforementioned interaction methods;
*d*) such instruments may offer a degree of control and expressive depth comparable to those offered by a traditional instrument;
*e*) it is possible to evaluate these instruments and related interaction channels through methodologies typical of HCI research.

As this is a relatively under-explored field, the achievement of the proposed goals required the definition of the research context, including insights on generic accessible digital musical instruments. The scope is narrowed by the definition of an instrument dedicated to people with quadriplegic disabilities which requires fine temporal control and is dedicated to real-time musical performance, modeled on the musician's residual motor skills. An analysis of the interaction channels available to this users group is therefore proposed, as well as an experiment which lays the groundwork for their experimental evaluation. A review of the state of the art related to this small niche of musical instruments is proposed. These theoretical notions lay the basis on which the development, maintenance and extension of three musical instruments dedicated to people with quadriplegic disabilities was conducted, in turn supported by the development of a software library and two hardware sensor peripherals.

In the development phase, focus has been placed on making economically affordable as well as accessible instruments and technologies, to meet the often limited economic resources available to the target group. This goal has been pursued through the realization of hardware peripherals reproducible through do-it-yourself practices, as well as by exploiting sensors already available on the mass market (e.g. eye trackers). Software source code has also been released under free licenses, in order to stimulate potential community development and customization.

The number and scope of the evaluation experiments was negatively affected by the onset of the COVID-19 pandemic in Italy, started in February 2020. This undermined the evaluation of two musical instruments developed during that period, as it would have been impossible to comply with the necessary hygienic standards, thus deferring such activity to future works.

## Publications and original contributions

A list of publications contributing to the outcomes of this thesis is provided below. The following were **published during my Ph.D. studies** period:

- **(P1)** Nicola Davanzo and Federico Avanzini. "Hands-Free Accessible Digital Musical Instruments: Conceptual Framework, Challenges, and Perspectives". In: *IEEE Access* 8 (2020), pp. 163975–163995
- **(P2)** Nicola Davanzo and Federico Avanzini. "A Dimension Space for the Evaluation of Accessible Digital Musical Instruments". In: *Proc. 20th Int. Conf. on New Interfaces for Musical Expression (NIME '20)*. NIME '20. July 2020
- **(P3)** Nicola Davanzo and Federico Avanzini. "Experimental Evaluation of Three Interaction Channels for Accessible Digital Musical Instruments". In: *Proc. '20 Int. Conf. on Computers Helping People With Special Needs*. Online Conf.: Springer, Cham, Sept. 2020, pp. 437–445
- **(P4)** Nicola Davanzo and Federico Avanzini. "A Method for Learning Netytar: An Accessible Digital Musical Instrument:" in: *Proceedings of the 12th International Conference on Computer Supported Education*. Prague, Czech Republic: SCITEPRESS - Science and Technology Publications, 2020, pp. 620–628
- **(P5)** Nicola Davanzo, Matteo De Filippis, and Federico Avanzini. "Netychords: An Accessible Digital Musical Instrument for Playing Chords Using Gaze and Head Movements". In: *In Proc. '21 Int. Conf. on Computer- Human Interaction Research and Applications (CHIRA '21)*. Online conf., 2021
- **(P6)** Nicola Davanzo and Federico Avanzini. "Resin: A Vocal Tract Resonances and Head Based Accessible Digital Musical Instrument". In: *Proceedings of the 2021 AudioMostly Conf.* Trento, Italy (online conf.), Sept. 2021

The following was **published before my Ph.D. studies** period:

- **(B1)** Nicola Davanzo, Piercarlo Dondi, Mauro Mosconi, and Marco Porta. "Playing Music with the Eyes through an Isomorphic Interface". In: *Proc. of the Workshop on Communication by Gaze Interaction*. Warsaw, Poland: ACM Press, 2018, pp. 1–5

The contributions from published papers can be summarized as follows. Paper **(P1)** defines HeaDMIs, which are accessible digital musical instruments for quadriplegic users dedicated to musical performance activities, resembling the performance capabilities offered by traditional instruments. Thus provides concepts for their design, an analysis of the various possibilities offered by the interaction channels available from the neck up and related sensors suitable to detect them, a state-of-the-art analysis as well as introductory concepts for the evaluation of interaction channels;
Paper **(P2)** provides the implementation of a graphical dimension space tool for the design and the analysis of generic ADMIs dedicated to different types of disabilities;
Paper **(P3)** provides an objective evaluation of three interaction channels suitable for HeaDMIs

interaction through experimental methodologies.
Papers **(B1)** and **(P4)** describe Netytar, a monophonic software HeaDMI controlled by gaze detection (via eye tracking) and breath detection. The first provides a general description of the first implementation of the instrument and a comparative evaluation preliminary test. The second provides an exercise-based learning method, introduces a particular notation, and describes developments of the instrument since the release of its first version;
Paper **(P5)** describes Netychords, a software HeaDMI derived from Netytar which allows to play chords through gaze detection and head rotation;
Paper **(P6)** describes Resin, a monophonic software HeaDMIs controlled through the stimulation and detection of resonances inside the upper vocal tract and head rotation detection.

The following are **unpublished contributions** introduced in this thesis, describing work carried out during the Ph.D. studies period:

**(U1)** An introduction to the context of digital musical instruments and their accessible counterpart, including an overview of health benefits given by music-related activities;

**(U2)** An analysis of quadriplegic users as a target group for accessible digital musical instruments;

**(U3)** Design evaluation of accessible digital musical instruments for quadriplegic users in the state of the art and two developed instruments (namely *Netytar* and *Resin*) through the dimension space analysis tool proposed in Ch. 2;

**(U4)** Elements of gaze-based digital musical instruments design;

**(U5)** A description of *NeeqDMIs*, a software library for rapid prototyping of software digital musical instruments for people with quadriplegic disabilities;

**(U6)** A description of *NeeqBS* and *NeeqHT*, two open-source hardware sensor peripherals, respectively for the detection of breath and head movements;

**(U7)** Further developments of two developed instruments after publication, namely *Netytar* and *Netychords*, introducing new control methods and player aids.

An indication of how these published and unpublished contributions are included in the contents of this thesis is provided in the following section.

## Structure

This thesis is divided into two parts for a clearer and more readable grouping of chapters. Where necessary, contributions from the above sources have been indicated with **P*** and **B***, while **U*** is used to indicate new, unpublished contributions.

**Part I** provides theoretical foundations to define the research context, as well as a design toolkit for accessible digital musical instruments dedicated to people with quadriplegic disabilities.

**Ch. 1** provides an overview on introductory concepts, definitions and context useful to understand the remainder of the thesis. A general overview of digital musical instruments is provided, followed by an introduction to the concept of accessible digital musical instruments, including evaluation strategies and related jargon (virtual instruments, performance instruments) **(U1)**. HeaDMIs, which are performance accessible digital musical instruments dedicated to people with quadriplegic disabilities are defined, including an analysis of the potential user group **(P1)** **(U2)**. Benefits of music playing as a motivation for such instruments are reviewed **(U1)**;

**Ch. 2** proposes a formal graphic tool for analyzing design choices and characteristics of generic accessible digital musical instruments, including their use context, degree of simplification, adaptability to various user needs, design novelty, number of physical interaction channels involved and addressed disabilities **(P2)**;

**Ch. 3** provides a detailed overview, classification and analysis of physical interaction channels available for HeaDMIs design, including a review of performance characteristics, physiological aspects as well as their use in musical or general Human-Computer Interaction applications **(P1)**;

**Ch. 4** describes an experimental procedure carried out to evaluate performance characteristics of three interaction channels suitable for HeaDMIs development: gaze pointing, breath and head rotation. Movement speed, precision and stability are evaluated through a Fitts' Law like experiment **(P3)**;

**Ch. 5** reviews the state of the art in HeaDMIs, exploring their mapping strategies, analyzing and evaluating their design through two dimension space analysis tools **(P1)** **(U3)**. Future challenges and perspectives are discussed **(P1)**;

**Ch. 6** provides an overview of gaze-based digital musical instruments design concepts, issues and solutions. Those include an analysis of physiological aspects, visual cues and techniques, an overview of the known Midas Touch problem and possible solutions for musical interface design **(B1)** **(U4)**;

**Part II** provides an account of software and hardware design, development and evaluation work carried out as part of my Ph.D. research project.

**Ch. 7** describes *NeeqDMIs*, a C♯ software library and framework for HeaDMIs prototyping and development. An overview of similar works, framework concepts and library contents is provided **(U5)**;

**Ch. 8** describes *NeeqBS* and *NeeqHT*, two open-source hardware sensor peripherals for the detection of breath and head rotation, respectively. Those could be replicated through do-it-yourself practice, and can serve as an economically affordable alternative for musical interfacing **(U6)**;

**Ch. 9** describes *Netytar*, a software HeaDMI suitable for playing melodies through gaze pointing and breath pressure detection. An overview of the instrument characteristics and implementation **(B1)** **(P4)**, a learning methodology **(P4)**, a notation system **(P4)**, design **(U3)** and experimental **(B1)** evaluations of the instrument are provided;

**Ch. 10** describes *Netychords*, a software HeaDMI which allows to play chords through gaze pointing and head rotation. An overview of its implementation, layout analysis **(P5)**, experimental player aids **(U7)** and design evaluation **(U7)** are provided;

**Ch. 11** describes *Resin*, a software HeaDMI which allows to play melodies through head movement and the detection of resonances on the upper vocal tract. An overview of mapping strategies and implementation **(P6)**, as well as design evaluation **(U3)** are provided.

# Contents

# I. Theory 1

# II. Development 85

# List of Figures

# List of Tables

# Part I.

# Theory

# 1

# Introductory concepts

In this chapter, some key concepts for understanding the contents of this thesis are outlined. Sec. 1.1 illustrates some beneficial effects derived from listening and participation in musical activities such as playing an instrument, demonstrating that extending musical inclusion and accessibility to different user groups could have solid motivations and importance. An introduction to the concept of Digital Musical Instruments (DMIs) is provided in Sec. 1.2, which defines their structure, classification and differences with respect to their traditional acoustic counterparts. Topics such as their evaluation and introducing terms and jargon used later in the following chapters are also addressed. Accessible Digital Musical Instruments, a sub-category of DMIs dedicated to people with different types of disabilities, are introduced in Sec. 1.3. Also in this case the related terminology is outlined in order to avoid ambiguities. The set of considered instruments furtherly tightens with the definition of *HeaDMIs* in Sec. 1.4, which are Accessible Digital Musical Instruments dedicated to musicians with quadriplegic disabilities suitable for performance contexts, which require fine temporal motion control.

## 1.1. Benefits of music

Music playing is one of the most universally accessible and inclusive human activities and takes part in the social life of all known cultures [8].

A general review of music effects on well-being has been proposed by MacDonald [9]. He categorized five types of contexts in which music could provide proven benefits:

- **Music therapy** focuses on positive psychological and physiological benefits to the listener, with experiences provided by qualified musical therapists.
- **Community music** focused more on increasing access to artistic activities rather than on therapeutic effects. An objective could be to provide an opportunity for creative expression in informal settings.
- **Music education** focuses on the development of individual music skills. Recent research is focused on the positive effects of music education (e.g. technique) upon non-musical development areas.

- **Music medicine** takes place in medical contexts. An example of intervention involves the reduction of pain, distress and anxiety perceptions in medically treated patients. Some of these interventions are closely related to music therapy.
- **Everyday uses of music**, finally, can provide relevant and profound psychological effects. Even the act of choosing which piece of music to listen to involves fine self (although often inconscious) psychological assessments.

Some of these context have overlapped goals and objectives, as evidenced by the same author.

Alongside with simple listening pleasure, music can bring evident effects even in different areas of life. It has been for example shown how it can bring improvements into human cognitive capacity. Črnčec et al. [10] explored the effects of music instruction on children, showing evidence on how it can bring benefits in spatiotemporal reasoning skills. Honing et al. [11] demonstrated that its engaging power applies to all ages, and is known to provide benefits also in terms of non-musical skills. According to an article published in 2004 by Costa-Giomi [12] music, in particular learning to play a musical instrument, can have a positive effect in the development and growth of a person starting from childhood, impacting on school performance and the degree of self-esteem.

Some authors shown that music can bring various psychological effects on the listener. Siedlieck and Good [13] brought experimental evidence on the relieving effects of music on feelings such as pain, depression and powerlessness. Williams [14] demonstrated the positive effects on mental health, communication skills, positive parenting and parent-child interactions given by music therapy sessions delivered to mothers and childs with disabilities. Stensæth [15] in 2013 proposed an analysis of a project for collaborative music creation (musical co-creation) through electronic tools, and showed how this activity can promote the establishment of a feeling of inclusion and participation. Sheppard and Broughton [16] analyzed, through a review, how music and dance activities participation are effective means through which individuals maintain well-being, healthy behaviors such as phyisical exercise and reduce social isolation.

Effect of music listening on stress relief have been highlighted by De Witte [17], having an influence on heart rate, blood pressure and hormone levels. A 2014 paper by Fancourt et al. provides a model for the development of a taxonomy of musical and stress-related variables, investigating how music can have positive impacts on human immune system activity.

## 1.2. Digital Musical Instruments

Digital Musical Instruments (DMIs hereafter) are instruments in which sound generation is based on digital means and is achieved by the performer through physical actions detected by sensing devices.

Thanks to the exponential increase of computational power, miniaturization, and available sensors, research on DMIs has expanded during the last two decades into the use of innovative interaction paradigms and interfaces. Yearly conferences such as the *International Conference on New Interfaces for Musical Expression* (NIME) [18] and *Sound and Music Computing Conferences* (SMC) [19] provide the ground for a thriving research community. Although the majority of commercial DMIs still uses the piano keyboard as the main interface, over the years the NIME and SMC communities explored different physical channels and sensors [20]. Nonetheless, the main employed physical channels remain the hands, the feet, and breath.

One of the main features that characterize a DMI is the separation between a *performer interface* (or gestural/performance controller, hardware interface, input device) and a *sound generation unit*. A classic example is given by the keyboard-synthesizer combination: while a keyboard provides an interface, it is the synthesizer which acts as a sound generator. This separation, as well as the use of digital means for interfacing with the musician and generating sound, allows to overcome some important limits dictated by the physical nature of acoustic musical instruments [21]. Different kinds of *sensors* can be used to detect performer's actions. A primary kind of feedback comes from the actual physical interaction with the interface surface, or from the body of the performer themself, if such surface is absent. This feedback can be amplified by using *actuators*. The sound generation unit provides a second type of feedback or *auditory feedback*, which consist in the sound itself as produced by the *sound synthesizer*, actually closing the feedback loop. This can be paired with a *feedback generator*, which provides signals for the actuators. During the design phases of a DMI, the instrument developer has a very wide range of possibilities to define how the actions of the performer on the musical interface is reflected in sound, defining a layer of interactive behavior which is not bound to the nature of physical controls. Such layer is defined as *mapping* [21]. This layer can be shaped in a potentially infinite number of ways, and it is one of the most determining factors to define the nature and identity of the instrument. The scheme in Fig. 1.1 is inspired by a diagram proposed by Marshall [22, Fig. 2.6] to review the main components of a DMI discussed above.

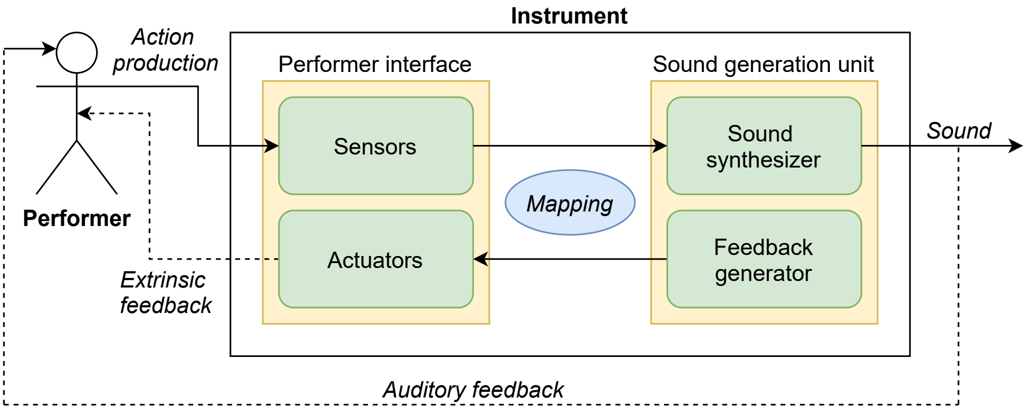


**Figure 1.1.:** *Structural diagram for a generic DMI. Figure based on [22, Fig. 2.6].*

Given this structure, classification systems used for traditional instruments can be unsuitable for DMI cathegorization. Acoustic instruments are often classified by the means of producing sound. For example, the Hornbostel-Sachs system [23] groups them into four cathegories (idiophones, membranophones, chordophones and aerophones), on the basis of which element of the instrument vibrates to generate sound.

Sound synthesis allows for more flexibility in this sense. Miranda and Wanderley [24] proposed a system for the classification of digital gestural music controllers based on similarity with acoustic musical instruments, presenting four categories. *Augmented musical instruments* (or hybrid instruments, hyperinstruments and extended instruments) are acoustic/electric instruments extended by the addition of electronic sensors. *Instrument-like gestural controllers* present a control surface whose design is partly inspired by an acoustic instrument. Similarly, *Instrument-inspired gestural controllers* possess control surfaces directly derived from an acoustic instrument: in such a way, a transition between the latter and the digital instrument should be easier for the musician. Finally, *Alternate gestural controllers* include a large number of instruments which do not resemble traditional ones. A further subdivision of those controllers is based on the degree of physical contact including touch, expanded-range (which require a minimal form of contact, where the musician is free to make movements which do not bear musical consequences) and immersive (where performer movements are not restricted, while being always in the sensing field of the instrument).

Magnusson and Hurtado [25] proposed a questionnaire interviewing musicians about differences between acoustic and digital instruments. From the detected opinions, frequent comments about acoustic instruments included the degree of tactile feedback, depth, uniqueness, absence of latencies, but also lack of memory and intelligence, influence of tradition and history leading to cliché playing, lack of experimentation and lack of microtonality. Digital counterparts, on the other side, have been commented as more free from musical traditions, experimental and explorative, more free from the mapping perspective, not as limited to tonal music. Negative aspects of DMIs included latency, lack of substance, lack of legacy, repertoire and social conventions. Although acoustical instruments impose physical and musical limits, those can be a source of creativity and emotional connection and feeling. A DMI, on the other hand, can overcome those limits by design, and be modeled according to specific needs. This trade off is seen positively or negatively depending on the musician. Lack of legacy repertoire for DMIs can also be seen as a limit, although some notices how this can spur musicians to experiment more, lifting them from social and technical constraints.

In addition to the DMI classification system by Miranda and Wanderley described above, it is useful to provide the definition of some other subcategories to which this thesis work refers. Those are *virtual instruments* and *performance instruments*.

**Virtual instruments.** A definition and a study on this particular type of DMIs is provided by Mulder [26]. Virtual Musical Instruments (VMIs) are DMIs which do not require direct contact between the musician and a control surface. This is made possible by different types of tracking technologies: the body itself of the musician becomes the performer interface, which movement is detected by different types of sensors that can be positioned on the body or trace its position through cameras (e.g. in the spectrum of visible or infrared light). As Mulder suggests, feedback is provided mainly through kinaesthetic and auditory means, leaving aside force and tactile feedback. Some indirect tactile, proprioceptive and visual feedback remains however available to the musician who feels their own body.

Mulder proposes some examples of VMIs, divided in different categories:

- Glove based instruments can translate hands and finger movements into musical performance, e.g. through hall-effect sensors to produce MIDI messages [27];
- Instruments based on whole body movements include for example the Very Nervous Sys-

tem [28], which translates movements into MIDI events through video image processing;
- Bioelectric signals can be used for musical interfacing as well. BioMuse (described in Sec. 5.1.1) provides MIDI control over electromyographic, electroencephalographic and oculographic sensors;
- Gestural controllers for conduction include for example a "Computer Music System that Follows a Human Conductor" [29], which allows to control MIDI performance through a glove, resembling the conduction of an orchestra.

Virtual instruments can be considered gestural interfaces, possibly falling into the categories *expanded range* or *immersive alternate gestural controllers* described by Miranda and Wanderley and reviewed above, depending on whether it is the movement of only some body parts of the musician that contributes to musical interaction, or their entire body. As stated in Sec. 1.4, virtual instruments can be a solution for designing musical interaction systems for people with quadriplegic disabilities (i.e. the main focus of this work), exploiting residual movement abilities. All the instruments presented in Part II of this thesis are indeed VMIs.

**Performance instruments.** The term DMI has somewhat fuzzy boundaries in the literature, and intersects with other forms of musical interfaces. We introduce the wording "performance instruments" based on a rather restrictive definition provided by Malloch *et al.*'s conceptual framework [30], which is in turn inspired by Rasmussen's model of human information processing [31]. In this view, interaction behaviors can be skill-, rule-, or model-based. Briefly, skill-based behaviors are related to activities which take place without conscious attention as smooth, automated, and highly integrated movements controlled on the basis of continuous signals coming from the environment: playing a conventional acoustic instrument falls in this category, along with handwriting, sports, bicycle riding, etc. In rule-based behaviors, activities consist of subroutines controlled by stored rules or procedures which have been learned or derived empirically, and information from the environment is typically perceived as signs: musical examples in this category include sequencing, live diffusion, creating a rhythm on a drum machine, etc. Model-based behaviors refer to more abstract activities in which performance is directed towards a conceptual goal (algorithmic music composition, presentation of recorded material, etc.) and information is perceived as symbols.

The above discussion provides the ground for stating that the present thesis work is focused on skill-based musical instruments, which bear close similarities to traditional ones in terms of both performance behavior and context. It should however be noted that the distinction between skill- and rule-based behaviors is generally blurred [31], and depends on previous training and experience. This is true for musical performance as well [30], where rule- and skill-based behaviors are mixed.

## 1.3. Accessible Digital Musical Instruments

As mentioned in Sec. 1.1, music can provide a number of beneficial effects to players and listeners. Yet, music playing is still not easily accessible for persons with disabilities. A book by Lubet [32] describes the relation of music with disability studies, addressing the ways in which the opportunity to participate in musical activities is often hindered to such people. Similar evidences are brought by the 2020 "Reshape Music" report from Youth Music [33].

Digital Musical Instruments have the potential for augmented accessibility with respect to traditional ones, as we have seen they allow for new, non-conventional modes of interaction [24]. The term “Accessible DMIs” (ADMIs) is often used to refer to instruments designed for persons with disabilities and special needs. The acronym ADMI, although almost always referring to a digital musical instrument dedicated to people with disabilities, is sometimes used differently, placing the words "assistive" or "adaptive" in the place of "accessible". As an example, Frid [34] cites the term “assistive music technologies”, while Graham and Knight [35] refers to ADMIs as “adaptive music technologies”. All these definitions bring about slightly different meanings, and it is useful to draw a distinction between these definitions.

- An **assistive** technology has been specifically designed to help a person with a disability to perform a task. The word “assistive” implies that an external source (technology) provides aid to a person with disabilities to complete a task. For example, a screen reader on a computer can help a person with a disability to read a job posting; a text-to-speech technology can help a visually-impaired user to read a text;
- An **adaptive** technology has the ability to adjust to the context and to the situation of the musician. For example, Clarion (described in Sec. 5.1.12) is an ADMI with high customization possibilities, being possible to adapt it to various use cases depending on the kind of disability from which the user suffers.
- An **accessible** technology has been designed with the needs of different users in mind. It possesses built-in customization features so that the user can really individualize their experience to meet their needs.

In this thesis the acronym ADMI is used with the meaning of "Accessible Digital Musical Instrument" in order to emphasize aspects related to inclusion and universal design. Nonetheless, the concepts of adaptability and assistance remain central properties of accessible instruments.

ADMIs and related works in the context of accessible interfaces have carved out an important niche within the literature. A great number of works have been published as *New Interfaces for Musical Expression* [36, 37, 38, 39]. As indicated by Frid [34] in her review work, several initiatives and charity organizations focusing on these topics were born in recent years, as well as several companies producing ADMIs and having inclusive music practices at the core of their mission.

As reported with the dimension space analysis framework described further in Ch. 2, ADMIs can target multiple types of disabilities. In this work, those disabilities are framed in three categories: physical (affecting motor skills and proprioception), sensory (affecting sight, touch, hearing and the five senses in general) and cognitive (affecting mind, learning skills and brain related skills). Other works propose statistics for similar categorizations. Frid, for example, classifies ADMIs found in literature through overlapping categories indicating that 39.8% of them are focused on users with motor disabilities, in particular 4.8% are dedicated to users affected by quadriplegia. As for ADMIs dedicated to sensory disabilities, 3.6% is dedicated to people with visual impairments and 6.0% to users with hearing impairments. 4.8% of the instruments are instead dedicated to users with cognitive impairments. Frid also indicates additional ADMIs user groups, for which it reports the percentage of dedicated ADMIs: learning difficulties (9.6%), autism spectrum disorder (8.4%), special needs (7.2%), complex needs (6.0%), cerebral palsy (6.0%), and people who cannot communicate verbally (6.0%).

McPherson draws a distinction between "performance-focused" and "therapeutic" instruments [40], where the former include ADMIs designed to enable masterful performances by musicians with disabilities, while the latter include instruments designed to elicit the therapeutic or wellbeing aspects of music making, even for non-musicians. Cappelen and Anderson [41] refer to the ensemble of activities enabled by music access through the word "musicking", originally coined by Small [41, 42]) to subsume all the activities related to music such as listening, playing alone or together, composing and dancing. The benefits of providing access to music to persons with disabilities have been discussed in many of the works mentioned above, and include rehabilitation, social inclusion, personal expression, physical and psychological wellbeing.

The aformentioned analysis framework in Ch. 2 provides a deeper insight on ADMIs design cues. We refer the reader to that section for greater detail on the topic.

## 1.4. HeaDMIs

As multiply stated before, most of the instruments and design concepts described in this thesis work are focused on a specific target group: musicians with quadriplegia. A particular kind of ADMIs, for which we have introduce the name *HeaDMIs*[1], covers such target group. For people with quadriplegia, the only traditional/acoustic means for musical expression are singing or whistling, and a limited number of instruments, such as the mouth harmonica, the kazoo, and possibly a few more.

The use of digital means and electronic sensors for interaction design offer less constraints for instrument development. In order to be able to create instruments which are truly accessible by people with quadriplegia, we should redefine the concept of ADMI as one modeled entirely on the residual motor abilities of the musician, whose interaction parameters adapt to the best available physical channels. This was one of the core motivating concepts for the research work conducted during my PhD studies, which includes the development of a modular and adaptable conceptual framework for their design, by which various types and levels of physical impairments can be addressed.

### 1.4.1. Structure

Fig. 1.2 presents a structural diagram for a HeaDMI, whose components are discussed next. Such diagram represents an alternative formulation for the generic DMIs scheme depicted in Fig. 1.1, more specific for this particular context. Here we further specify the performer's action in terms

---

[1] The name "HeaDMI" summarizes the wording "DMIs controlled through interaction channels placed on the user's Head".

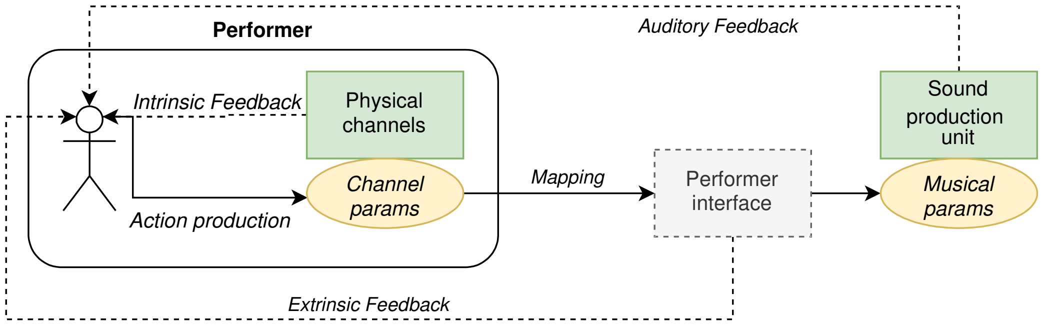


**Figure 1.2.:** *Structural diagram of a HeaDMI.*

of physical interaction channels, parameters, and mapping strategies, as well as the sources of feedback to the performer.

### 1.4.1.1. Physical channels

The physical channels used to interact with traditional musical instruments are hands, fingers, breath, mouth/lips, and feet, with rare exceptions. A HeaDMI can only exploit the remaining able channels of the performer: these may therefore include head movements, gaze pointing, mouth aperture, etc. A comprehensive list and analysis of these channels is presented in Ch. 3 (Tab. 3.1).

Any single physical channel can have multiple associated parameters: as an example, those associated to *Head movements* include pitch, yaw and roll angles, while those associated to *Gaze pointing* are the 2D pointing coordinates on the screen, the duration of fixations, etc. Each parameter can be estimated by appropriate sensors and can be assigned a role in the musical interaction.

### 1.4.1.2. Sound production unit

This block is responsible for the sound synthesis. As mentioned in Sec. 1.2, the possibility of separating the control interface from the sound production unit in a DMI provides an additional degree of freedom with respect to acoustic instruments. The sound production unit generally exposes an interface which is able to receive a set of messages and events influencing musical parameters.

We opt for an operative classification of musical parameters, often used in the context of DMIs [43, 44], which identifies three levels of control over musical processes: the *Note* level requires parameters related to a single note event; the *Timbral* level demands parameters with high temporal resolution, acting on timbral sound properties, even within a single note event; the *Process* level is a “macroscopic” one, which is associated to global or structural musical characteristics. These three levels are reported in Table 1.1, along with a non-exhaustive list of possible associated musical parameters (more precisely, the parameters listed in Table 1.1 are those used by the HeaDMIs reviewed in Sec. 5.1).

| Control Level | Musical parameters | Abbr. |
|---|---|---|
| Note | Note on/off | note |
| | Pitch | pitc |
| | Intensity | inte |
| | Glide | glid |
| | ... | |
| Timbral | Vibrato | vibr |
| | Brightness | brig |
| | Sustain | sust |
| | ... | |
| Process | Instrument selection | isel |
| | Mode/scale selection | msel |
| | Transposition | tran |
| | Harmonic change | hcha |
| | Tempo | temp |
| | Panning | pann |
| | ... | |

**Table 1.1.:** *A non-exhaustive list of musical parameters, associated to different control levels.*

At the Note level, note on/off events control the triggering/releasing of a note; pitch refers to the perceived note height, and may be quantized on a musical scale or may vary continuously; intensity relates to the energy injected into the note emission, and thus to its loudness but also to its spectral coloration; glide refers to a smooth transition between pitches of two successive notes.

At the Timbral level, vibrato is a rapid, slight oscillation in pitch which produces a richer tone; brightness refers to the possibility of manipulating the spectral energy of the sound towards the high or the low frequencies; sustain is a control available on some instruments (e.g., on the piano through the damper pedal) by which the note keeps resonating after its actual release (possibly along with sympathetic resonance from other notes).

At the Process level, instrument selection refers to the used instrumental sound; mode/scale selection and transposition refer to the possibility of redefining the musical scale on which the instrument is tuned or to transpose all the pitches by a given offset (e.g., an octave); harmonic change controls the switching between different chords; tempo refers to the control over the beats per minute (BPMs) of the music being played; panning controls how the sound is distributed on output channels (e.g., in a stereo or multichannel set-up).

#### 1.4.1.3. Mapping

As stated in Sec. 1.2, by mapping we refer to the way in which channels parameters are linked to musical ones. McGlynn [21] discusses various mapping strategies, some of which are especially relevant for the channels analyzed in Ch. 3.

- *Trigger*: an action of the physical channel causes an instantaneous event (an example is a hit on a drum pad);
- *Toggle:* an action causes an instantaneous switch to a different state, and a subsequent one

returns it to the previous state (an example is the use of a selector to transpose by one octave up and down, or to switch from one scale to another);

- *Counter:* different actions allow to scroll between different states in a circular fashion (as an example, pressing a key on an electronic keyboard allows to scroll through different available sounds);
- *Hold:* an action changes the internal state of the system, as long as it is maintained (an example is the pressure on the expression pedal of a piano);
- *Continuous range:* the value of a physical channel parameter in a continuous range is mapped to a musical parameter over an analogously continuous interval (as an example, breath pressure can be mapped to intensity, or head position can be mapped to pitch);
- *Discrete range:* the value of a physical channel parameter is quantized and mapped to a discrete set of values of a musical parameter (as an example, in a harmonica the horizontal head position is mapped to a discrete set of pitches);
- *Excitation:* the rate of change (time derivative) of a physical channel parameter is mapped to a musical parameter (as an example, in a violin the bow speed affects the intensity).

Mappings also have associated qualities, which depend on strategies but also on the physical and musical parameters involved, and have a major influence in instrument playability, expressiveness, and enjoyment. Some of these qualities are particularly relevant for HeaDMIs. Such qualities are reviewed below.

**Transparency.** This quality refers to the "psychophysiological distance" [45] between physical and musical parameters of the mapping, from both the performer and the audience perspective. For the former, transparency depends on cognitive understanding of the mapping and on the level of dexterity with the instrument, while the latter only need to have an understanding of causal relationships between performer's actions and sonic results. For both, understanding is derived from previous knowledge and expectations: as an example, mimicking physical actions on an acoustic instrument, or using metaphors (e.g., pitch increasing from left to right as in a piano keyboard), aids transparency. This aspect is particularly relevant for HeaDMIs, due to the unconventional physical channels considered.

**Energy.** One particularly important ecological principle (i.e., one reflecting expectations derived from everyday experience) is that the acoustic energy of the instrumental sound should be the product of muscular energy injected by the performer's gestures into the instrument. Usability experiments in DMI design [46] have shown that incorporating energy into the mapping provides a more engaging natural instrument and a tighter connection of the performer to it. The physical channels considered in this work allow for limited possibilities of movement and muscular activation. It is therefore necessary to maximize the use of energy in the mapping, and also to devise alternative strategies to compensate for these limitations.

**Cardinality.** Simple mappings employ one-to-one relationships between physical and musical parameters. However it has been long been suggested [46] that relationships involving higher cardinalities (many-to-one, one-to-many, many-to-many) should be preferred especially when several musical parameters are exposed by the sound production unit. These relationships, which are typical of most acoustic instruments, have been shown to be more rewarding and intuitive for musical interaction and to provide more expressive control, possibly at the expense of longer learning times [46]. They may also require additional layers of processing to extract intermediate parameters [47]: as an example, parameters from several physical channels may be combined to estimate the performer's facial expressions and control mode selection in a many-to-one mapping.

#### 1.4.1.4. Performer interface

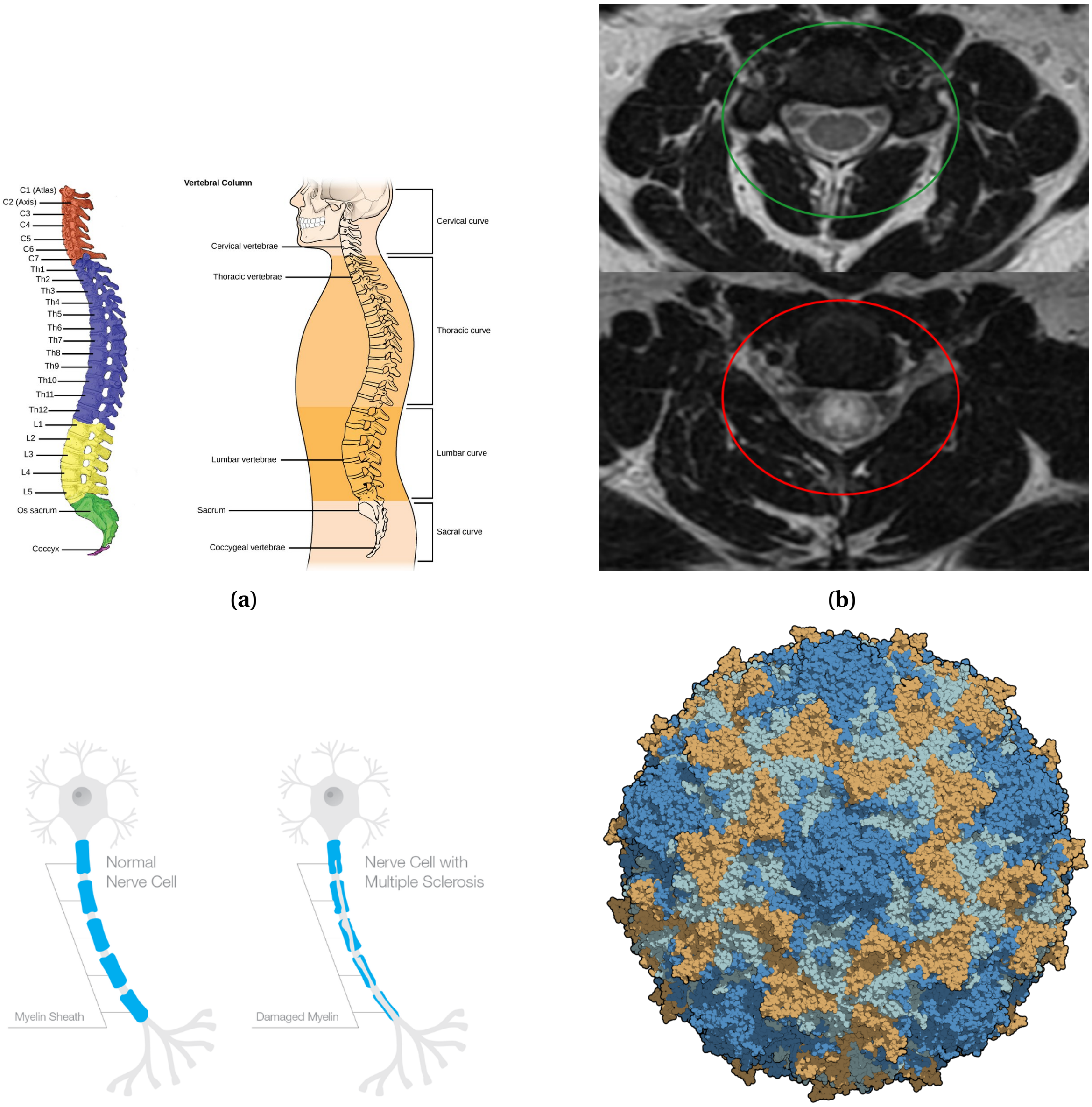
As mentioned in Sec. 1.2, in DMIs the instrument interface is often physically separable from the sound production unit. In HeaDMIs such interface may be totally absent, especially whenever the employed channels do not require external references for performing their actions. In this case the body of the performer becomes the interface to some extent. This would be the case for many of the channels discussed in Ch. 3. Some other channels, like Gaze pointing, require visual objects on a screen in order for gaze fixations to occur and be detected. Several of the brain-computer interfaces discussed in Ch. 5 also require external stimulation that elicits the desired brain responses.

If present, the user interface may be represented by physical or virtual objects (e.g. shown on a screen). Furthermore, it may be part of the mapping strategies of the instrument, as physical channel parameters may manipulate static or dynamic objects (keys, sliders, or more complex elements), and this manipulation would reflect on musical parameters.

This component can be also devoted to providing extrinsic feedback to the performer, in addition to the intrinsic (tactile, proprioceptive, kinesthetic) feedback generated by performer's movements [48] and the auditory feedback provided by the sound production unit. Extrinsic feedback may use several modalities (e.g., visual feedback through computer screen, vibrotactile feedback through actuators, etc.). In acoustic instruments, a physical interface has the double function of mapping actions to sound and of providing extrinsic feedback to the performer: as an example, the piano keyboard provides both the mechanical machinery that sets strings into vibration, and a visual and haptic interface for the performer to locate pitches and control the dynamics. In DMIs, rich, multimodal feedback can be introduced to enhance the interaction between player and instrument [24].

### 1.4.2. Target group

The potential target population for HeaDMIs is vast. Sears *et al.* [49] presents an overview of health conditions and related physical impairments that affect the upper body and consequently hinder the use of traditional instruments and musical interfaces:



**(a)** **(b)**

**(c)** **(d)**

**Figure 1.3.:** *Illustrations to explain various diseases which could lead to quadriplegia: (a) vertebrae numbers and classification; (b) axial T2 magnetic resonance imaging of cervical spine with normal cord signal (green circle) and increased signal due to transverse myelitis (red circle); (c) a visual rendering of the effect of multiple sclerosis on myelin; (d) a visual rendering of the polio virus capsid.*
***Source:*** *(a) CNX OpenStax; (b) JasonRobertYoungMD; (c) Stephanie021299; (d) Manuel Almagro Rivas; all licensed under CC BY-SA 4.0, via Wikimedia Commons.*

- High-level spinal cord injuries, especially those related to cervical vertebrae (C1-C7, Fig. 1.3a), can cause loss of motor and/or sensory control. To date, an effective cure for such condition is unknown. In addition, an injury to the first two thoracic vertebrae (T1-T2, Fig. 1.3a) can also lead to a difficulty or an inability to move hands and fingers correctly. Other traumatic injuries can lead to the amputation of one or more limbs;
- Transverse myelitis (Fig. 1.3b) causes an inflammation of the spinal cord, which in turn can cause numbness, and/or a deficit of sensory and motor skills in both the arms and legs;
- Amyotrophic lateral sclerosis is a progressive neurodegenerative disease that affects nerve cells both at brain and spinal cord level; consequently, the affected individual could lose the ability to move, eat, speak, breathe;
- Multiple sclerosis is a disease that results in the loss of insulating myelin which covers nerve cells (Fig. 1.3c). Both the brain and the spinal cord can be damaged as a result. A percentage of people with multiple sclerosis have also sensory problems;
- Polio (Fig. 1.3d) is a viral disease which could lead to muscle weakness and inability to move; it has however been nearly eradicated since 1988 through vaccines;
- Lock-in syndrome describes a condition in which the affected person cannot move any part of the body except eyes and eyebrows, but keeps all their cognitive functions unaffected. Total lock-in syndrome is a variant in which the eyes and eyebrows are also out of control. Such conditions are caused by damage to certain parts of the lower brain and/or brainstem, which can occur following thrombosis, stroke, cancer and/or traumatic injury.
- Muscular dystrophy is an expression that indicates a group of at least 30 different types of congenital disorders, the most common of which is Duchenne muscular dystrophy. These diseases cause progressive weakening of the skeletal muscles. Amelia, that is the absence of one or more limbs, can be given by a congenital disorder.

The size of the affected population can be inferred by epidemiological data. As an example, the incidence (occurrence of new cases) of spinal cord injuries varies between developed countries (13.1-163.4 cases per million) and undeveloped countries (13.0-220.0 cases per million) [50], with 250,000 to 500,000 persons affected every year worldwide [51]. Rates of prevalence (persons affected at a given time) range from 906 per million in the US (highest recorded) to 250 per million in France (lowest recorded) [52]. As a further example, the incidence of amyotrophic lateral sclerosis is 1.0-2.6 cases per 100,000 people every year [53], and is particularly high in Europe, with 15,000 new cases per year [54, 55]. Prevalence ranges from 4 to 9 people per 100,000 [53, 54, 55].

## 1.5. Evaluation

While there is extensive literature dedicated to Digital Musical Instruments in general, one of the main addressed issues among it is their evaluation [56, 57, 58].

Evaluation of DMIs encompasses a broader set of aspects than those typically considered in HCI.

Being these technologies dedicated to artistic expression, it is difficult to define performance expectations and therefore to define evaluation metrics. While one of the most critical aspects of DMIs design is the relationship between the musician and the instrument's interface during performance [21], a number of other design aspects are to be taken into account. Evaluation spins around the concept of musical performance, which involves a number of stakeholders: performers, composers, audiences, designers, manufacturers. O'Modhrain [56] proposed a framework for DMIs evaluation which takes into account the roles of stakeholders in a number of evaluation goals (enjoyment, playability, robustness, compliance to design specification). While every stakeholder is interested in the *enjoyment* and *playability*, the performer and the manufacturer are also interested in the *robustness* of the instrument's hardware and software, while the designer and the manufacturer could be interested in evaluating how they are meeting the design *specifications*. A summary of O'Modhrain's framework is given in Table 1.2. The original table [56, Table 1] is also populated with methods that a given stakeholder might use in order to evaluate a DMI against a given design goal. This has been the basis for various DMI evaluation methodologies. As an example, for their evaluation of *The EyeHarp* instrument Vamvakousis and Ramirez [59] implemented part of this framework by providing a questionnaire to an audience which attended a small organized concert (see also Sec. 5.1.8). Other frameworks, such as the one proposed by Birnbaum and Malloch [30, 44], are aimed at DMIs classification, providing ground for evaluation of design choices.

For DMIs in general, the causal link between the performer's gestures and the sound generation may not always be clearly perceivable by the audience: this may impact negatively on audience's engagement. Augmenting the audience experience through additional sensory channels (e.g., visuals [60] or haptics [61]) can help reestablish such a link by increasing the transparency of the interaction. This issue is even more challenging in the case of HeaDMIs, where mappings have necessarily a limited ecological validity, and transparency is reduced by the lack of apparent exchange of energy between the performer and the instrument. As a consequence, causal relationships between performer's actions and sound are opaque to the audience. The need for additional multimodal cues, able to reestablish the connection between cause and effect, becomes particularly important in order to make a HeaDMI performance convincing, effective, and expressive. A possible solution could be to show the audience (e.g. through projection) the performer interface on a screen in real time [62].

| **Stakeholders** | **Evaluation goals**<br>Enjoyment | Playability | Robustness | Specifications |
|---|---|---|---|---|
| Audience | • | • | | |
| Performer | • | • | • | |
| Designer | • | • | | • |
| Manufacturer | • | • | • | • |

**Table 1.2.:** *DMI evaluation framework: stakeholders, evaluation goals, and connections (based on O'Modhrain [56, Table 1]). A dot represents the interest to evaluate the given goal from the perspective of the given stakeholder.*

# 2

# Dimension space analysis for ADMIs design

In 2020, we've presented an article covering ADMIs design concepts [2]. This paper proposed a formal tool to explore the main design aspects of ADMIs based on *dimension space analysis*, a well established methodology in the *New Interfaces for Musical Expression (NIME)* literature which allows to generate an effective visual representation of the design space. We therefore proposed a set of relevant dimensions, which are based both on categories proposed in recent works in the research context, and on original contributions.

Despite the publication of studies dedicated to reviewing the state-of-the-art of ADMIs [34, 35, 37, 40], there was still a lack of contributions towards a systematic analysis of the most important dimensions of their design.

The goal of our work was to contribute to the advancement of this research direction by proposing a more comprehensive framework for evaluating and classifying a broad range of ADMIs. In doing so, we discussed a set of relevant requirements and design choices for this class of instruments.

An account of previous works on dimension space analysis is provided in Sec. 2.1. Our implementation is instead described in Sec. 2.2. Some examples of its application are provided as case studies in Sec. 2.3. Conclusions and future works are discussed in Sec. 2.4.

## 2.1. Previous works on Dimension Space Analysis

Several approaches have been proposed in the past to classify and evaluate the main design aspects of a broad range of "musical devices", including musical instruments, interactive installations, games, and so on [24, 63, 64]). Among them, the dimension space analysis proposed by Birnbaum *et al.* [44] is particularly appealing both for its applicability to a variety of contexts and for the effectiveness of dimension plots, which allow to visualize and rapidly compare musical devices along a set of design dimensions.

Birnbaum empirically observed that the functionality of a space is not affected in plots with as many as eight axes, and proposed seven dimensions for analyzing musical devices in their broadest meaning: *Required Expertise*, representing the level of practice of the performer; Musical Control, specifying the level of musical control exerted by the performer; *Degrees of Freedom*, indicating the number of input controls available to a user; *Feedback Modalities*, specifying the degree to which a system provides real-time feedback; *Inter-actors*, representing the number of people involved in the musical interaction; *Distribution in Space*, indicating the total physical area in which the interaction takes place; *Role of Sound*, representing the category of sound role (with three main values: artistic/expressive, environmental, and informational).

The flexibility of the dimension space approach lies in the ability to redefine the axes. In fact, alternative representations have been proposed: Magnusson [65] presented an "epistemic" dimension space, as opposed to the more "phenomenological" Birnbaum space. Some other authors proposed more specialized spaces, aimed at evaluating specific categories of musical devices: Hattwick and Wanderley [66] presented a dimension space for evaluating collaborative music performance systems. In a similar fashion, Hödl and Fitzpatrick [67] targeted hand-controlled guitar effects for live music and described a related design space.

Following these examples, here we resort to the dimension space analysis approach to propose a new space dedicated to ADMIs. The aim of this effort is to reflect on the dimensions that define their design space and to offer a tool for labeling, discussing, and evaluating such instruments. As such, the proposed space is not intended as an alternative representation to the one designed by Birnbaum, but rather as a finer layer of description devoted to a specific class of instruments. Therefore, a complete description of a single ADMI could be obtained by classifying it both along the seven-axis Birnbaum space and along the eight-axis space proposed in this work. The two descriptions are complementary, as discussed in the remainder of this chapter.

## 2.2. Our Dimension Space

Following these examples, here we resort to a dimension space analysis approach to propose a new space dedicated to ADMIs. The aim of this effort is to reflect on the dimensions that define their design space and to offer a tool for labeling, discussing, and evaluating such instruments.

As such, the proposed space is not intended as an alternative representation to the one designed by Birnbaum, but rather as a finer layer of description devoted to a specific class of instruments. Therefore, a complete description of a single ADMI could be obtained by classifying it both along the seven-axis Birnbaum space and along the eight-axis space proposed in this work. The two descriptions are complementary.

The eight-axis configuration resulting from our analysis is shown in Fig. 2.1. The eight axes can be conceptually grouped into two subsets: the four axes in the lower part of the plot (labeled in blue) relate to the intended target users and to the use contexts of the instrument, while those in the upper part of the plot (labeled in green) are related to the design choices of the instrument. These groups reflect two distinct phases of the design of an ADMI, where the first one is more related to preliminary definition of requirements, and the second one is concerned with subsequent choices in the design. We discuss these two subset of axes in Sec. 2.2.1 and 2.2.2, respectively.

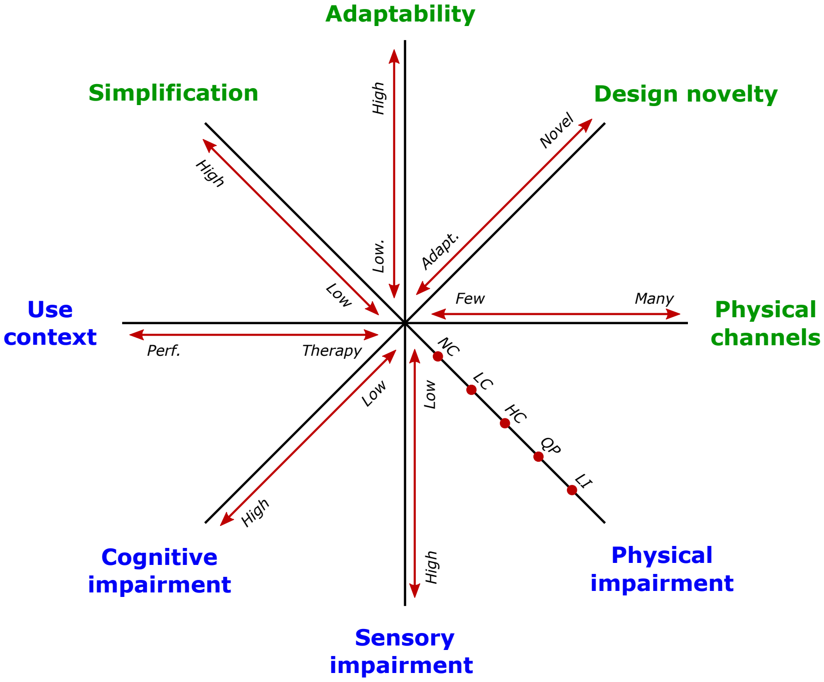


**Figure 2.1.:** *A visual representation of the proposed dimension space. Axes are grouped in two subsets: Target users and use contexts (blue), and Design choices (green).*

### 2.2.1. Target users and use contexts

In this section we discuss the four axes in the lower half of the proposed dimension space (see Fig. 2.1).

Disability is a "complex multidimensional experience" [68] which poses challenges for measurement and classification. In the "International Classification of Functioning, Disability and Health" (ICF) [69], the World Health Organization categorizes problems with human functioning in three interconnected areas: *impairments* are problems in body function or alterations in body structure, and are often identified as symptoms or signs of health conditions; *activity limitations* are difficulties in executing specific activities; *participation restrictions* are problems with involvement in any area of life. *Disability* refers to difficulties encountered in any or all these three areas of functioning.

The Washington Group on Disability Statistics [68, p.26], an international, consultative group of experts aimed at facilitating the measurement of disability, applies an ICF-based approach which covers six functional domains or basic actions: seeing, hearing, mobility, cognition, self-care, and communication.

In the context of human-computer interfaces, Jacko *et al.* [70, Ch.43] proposed a conceptual scheme aimed at defining categories of impairments and their relation to the use of interactive technologies. Specifically the authors considered five broad categories: (a) hearing impairments, (b) mental impairments, (c) physical impairments, (d) speech impairments, and (e) visual impairments. Each is composed of a collection of related clinical diagnoses which, in turn, influence certain functional capabilities that are critical to accessing specific classes of technologies.

For the purposes of this work, categories of impairments are further clustered into three main axes: physical, cognitive, sensory (or perceptual). These three broad categories are often employed in the literature of accessible HCI [70, Ch.41-44]. Previous reviews on ADMIs [34, 37, 40] suggest that target user groups can be classified into these three categories. Moreover, the multi-dimensional character of disability also means that physical, sensory, and cognitive impairments are often intertwined.

**Physical impairment.** Usually, ADMIs aimed at musicians with physical impairments are designed to address a specific degree of motor ability. With a handful of exceptions (the harmonica, the kazoo, and possibly a few more), traditional musical instruments include upper limbs (hands and fingers in particular) among their physical interaction channels, with the possible addition of the feet (used for example to control pedals and foot switches), and breath. Consequently, this axis indicates the level of motor impairment addressed by the instrument along a discrete scale of five points, each representing the minimum motor skill level required by the instrument. The rationale behind this classification is that instruments devoted to higher levels of impairment can potentially be used also for lower levels of impairment. The proposed levels are as follows (see Sears *et al.* [70, Ch. 42] for an overview of related health conditions and traumas):

1. *Uncompromised motor skills (NC)* - Includes instruments dedicated to fully motor skilled users, therefore aimed at other types of impairments.

2. *Lightly compromised motor skills (LC)* - Includes instruments for users who do not have full control of (or have difficulty controlling) limbs. This could be given by cerebral palsy, heart attack, and other conditions.

3. *Heavily compromised motor skills (HC)* - Includes instruments dedicated to people who have at least one limb completely compromised, unable (or almost completely unable) to move. DMIs dedicated to users with situations such as hemiplegic paralysis or paraplegia belong to this category.

4. *Quadriplegic paralysis (QP)* - Includes instruments dedicated to people who no longer control upper and lower limbs. These instruments can take advantage of interaction channels available from the neck upwards (face muscles tension, gaze, brain frequencies, breath, etc.) [71].

5. *Lock-in syndrome (LI)* - This is a condition in which a person is awake and conscious but can only move his eyes. This category includes instruments that use only eye based interaction (e.g. gaze-based or blink-based), or electroencephalogram (EEG) based interaction.

**Sensory impairment.** Frid [34] remarks that a very limited amount of existing ADMIs are specifically designed for users with sensory impairments. As an example, only 3.6% of the 83 instruments reviewed in her work focused on persons with visual impairments, while 6.0% focused on persons with hearing impairments. This is therefore a research direction that needs to be further explored. The corresponding axis in the proposed dimension space indicates the level of sensory impairment addressed by the instrument. Unlike the physical impairment axis, this is a continuous axis ranging in value from low to high levels of impairment. The reason for this choice is that different or multiple types of sensory impairments may be addressed, which makes impossible to define a unique discrete scale of levels.

**Cognitive impairment.** The cognitive impairment axis may include several different target groups, such as children with special educational needs (SEN), learning and developmental difficulties, behavioral disorders, autism spectrum disorder (ASD), as well as elderly people with aging-related losses of cognitive abilities, and persons with severe intellectual deficits lacking conceptual and/or communication skills. In the literature, reference is often made to four levels of intellectual impairment (mild, moderate, severe, profound) [72, 73, 74], ranging from situations in which the person is able to learn practical skills and communicate, to scenarios in which the person does not have any degree of autonomy and independence. However these categories are not easily related to musical abilities: for this reason, in the proposed dimension space this is a continuous axis ranging in value from low to high levels of impairment.

**Use context.** The fourth and last axis in the lower half of the proposed dimension space is related to the use context for which the instrument is intended. Regardless of the type(s) of impairment of the target users, defining the context of use influences all the subsequent design choices. Harrison and McPherson [75] make a distinction between two broad categories of ADMIs. On one side, “therapeutic devices” are meant to provide a means for persons with disabilities to enjoy the health, social and psychological benefits of music making, demonstrated by music therapy practices. Instruments in this class often have a low-barrier to expressive music making, and work particularly well in group music workshop contexts, or as part of music therapy sessions. On the other hand, the category of “performance-focused instruments” refers to instruments which allow the performer to reach high levels of expression and virtuosity, similarly to traditional instruments for performers who live without disabilities. Instruments in this category often require larger amounts of practice and are particularly suited for inclusive music performance contexts, such as accessible orchestras. It has to be noted that therapy and performance are not mutually exclusive use contexts and may co-exist in the design of an ADMI. Therefore, in the proposed dimension space *use context* is a continuous axis ranging in value from therapy to performance, depending on the aims of the instrument.

### 2.2.2. Design choices

Having defined the potential target groups and the use contexts of the instrument, the remaining dimensions relate to fundamental choices in the design of the ADMI. These can strengthen some aspects of inclusion rather than others, influencing the accessible qualities of the instrument. By analyzing previous works in the literature, we extracted four design dimensions which are especially relevant for accessibility. These are represented in the upper half of the proposed dimension space (see Fig. 2.1) and are discussed next.

**Adaptability.** This has long become a key concept in the field of accessible HCI. When designing for persons with disabilities, every user has different and individual requirements and needs, and adaptive interfaces have the potential for accommodating a wide range of users. Jacko *et al.* [70, Ch.43] discuss several adaptive interfaces in the context of sensory impairments. An analysis of the ADMI literature shows that existing instruments generally allow for limited adaptability to the specific needs of an individual user. Thus, this remains one of the ultimate challenges in this context, as acknowledged by various scholars [34, 37, 76]. Some instruments may include the possibility of customizing parts of the interface and instrumental features. However a deeper form of adaptability should be based on user models with respect to their abilities and should consequently include the possibility of modifying the interface, the employed interaction channels, and the musical mappings. An emerging trend amounts to using interactive machine learning techniques in order for an instrument to learn preferred or idiosyncratic gestures of an individual user, and to map the learned gestures to musical parameters. A notable related example is the Wekinator software [77], in which various supervised machine learning approaches are used to build musical mappings through training examples. Interestingly, this sofware has been used in a recent project aimed at building customized musical rehabilitation devices for children with severe motor impairments [78]. It has to be noted that the *adaptability* axis plays a special role in the definition of the space, as it can affect the remaining dimensions related to design choices. As an example, a high level of adaptability may imply that the instrument can address various levels of physical impairments, or be equally suited to music therapy contexts and to performance contexts through ad-hoc setup changes, and so on. Therefore, in the case of an instrument with high *adaptability* it would be recommended to classify it according to the most common use case, or provide a judgment that reflects the maximum level attained for a given axis.

**Design novelty.** A distinction can be drawn between instruments that are designed from scratch having persons with disabilities as target users, and adaptations of existing instruments. The definition “adapted instrument” is often used to refer to a modification of a traditional or pre-existing instrument, obtained through either mechanical, electroacoustic, or digital means: in this case the focus is thus shifted towards the “assistive” facet of technology. On the other hand, in the case of completely novel instruments the focus is shifted towards the “accessible” facet of technology, which calls for cyclical, participatory design approaches that only recently started to enter the mainstream of DMI research [79]. Similarly to some of the axes discussed above, there is a continuum of possibilities in between these two dichotomic alternatives. As an example ADMIs employing existing control metaphors (e.g., the piano keyboard) may be assigned an intermediate rank along this axis. Tending towards one of the two extremes of this axis depends on the target users and contexts of use. Although a completely novel interface may possibly better accommodate the necessities of persons with disabilities, offering the possibility to play a traditional instrument may provide an added value for inclusive music practices.

**Physical channels.** The usability and the expressivity of the instrument are largely affected by the amount of physical channels that the user can employ in the interaction. This axis thus relates to the number of motor skills needed by the user. Examples are finger movements, breath, EEG features, head movements, gaze pointing, etc. These in turn are related to the types and levels of impairments, especially physical ones, but also sensory and cognitive ones. This axis has some relation with the “Degrees of Freedom” dimension indicated in the space for generic musical devices. However, that axis indicates the “number of input controls” and is thus focused on the sound production unit of the instrument; on the other hand, this axis shifts the focus to the perspective of the users and their functioning.

**Simplification.** The fourth and last axis in the lower half of the proposed dimension space is related to the degree of simplification designed into the instrument. The word “simplification” in this context is once more borrowed from the literature of accessible HCI, where it is often used to refer to simplified interfaces aimed at reducing the cognitive load and/or simplifying motor actions required to complete a task [70, Ch.23]. However, here we use this term in a wider sense: the degree of simplification of an ADMI along this axis refers to all the aspects of the instrument design aimed at aiding the user in completing musical tasks. These may include enlarging of elements of the visual interface, but also temporal quantization of musical events to compensate for rhythmic difficulties, simplified gestures to play chords or arpeggios, etc. Related concepts have been investigated in the context of DMIs for novices and non-musicians (beginning with the “low entry fee with no ceiling on virtuosity” claim by Wessel and Wright [80]), and are discussed by McPherson *et al.* [40]. Correspondingly, the Birnbaum space includes the dimension “Required expertise”. Here, however, the focus is not on user expertise, but rather on user abilities and on the related design simplifications aimed at providing an engaging and rewarding experience.

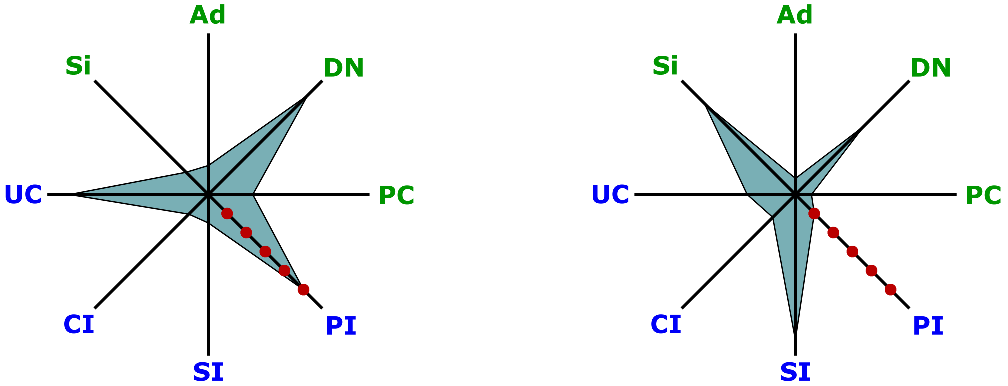
## 2.3. Case studies

In this section a review of 8 ADMIs, including academic projects, commercial products, and non-academic projects funded through charity programs is provided as an example. All the chosen instruments appear in at least one recent review of ADMIs [34, 37, 40, 71]. The purpose is to demonstrate the applicability of the proposed space to the analysis of existing instruments, and to provide further discussion on its dimensions.

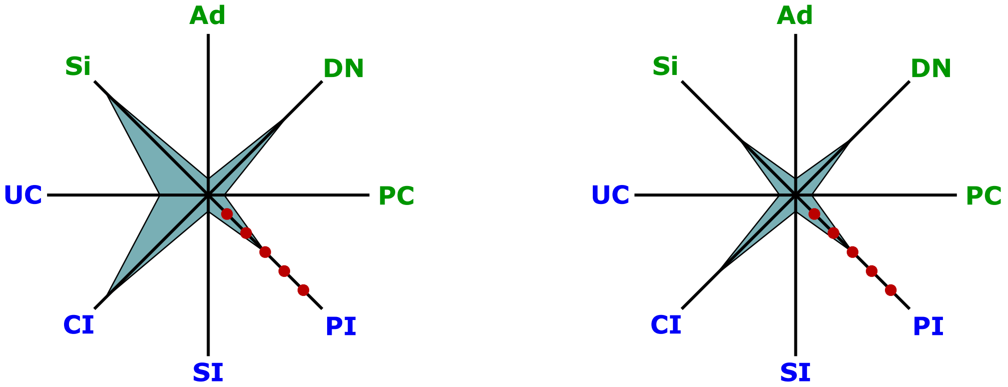
Figure 2.2 presents the visualization of the ADMIs in the dimension space. The plots show that each of the reviewed instruments scores extreme (high or low) values along at least one axis. In fact, each instrument has been chosen because it is especially relevant to discuss at least one of the 8 dimensions. It should be noticed that since three of these instruments are HeaDMIs (namely SSVEP, Clarion and Biomuse, in Fig. 2.2), they are included in the state-of-the-art review in Ch. 5.

Examples of ADMIs focused on *physical impairments* abound in the literature. Miranda and coworkers developed several Brain-Computer Music Interfaces (BCMI). One in particular [81] (Fig. 2.2a) was specifically designed for users with severe motor impairments, and was tested with a patient with Locked-in Syndrome. The instrument allows for real-time generation of melodic lines through a reactive brain-computer interface based on steady-state visual evoked potentials (SSVEPs).

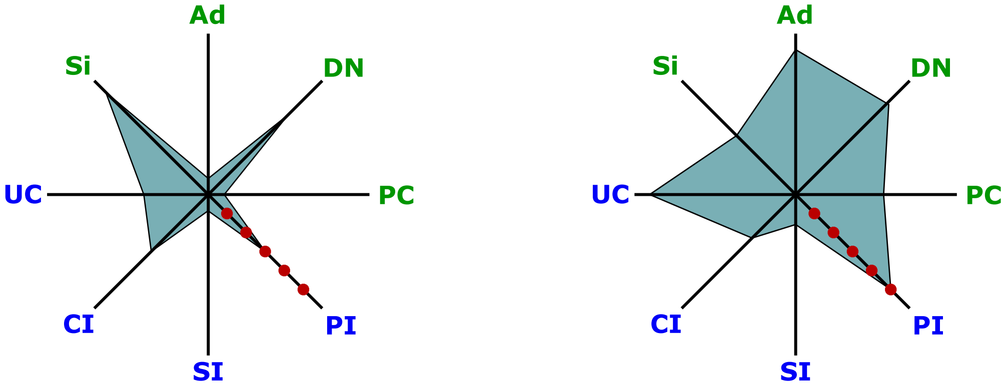
In contrast to the above, as already discussed, relatively little work has been done regarding instruments for persons with *sensory impairments*. One interesting example is provided by Gri-



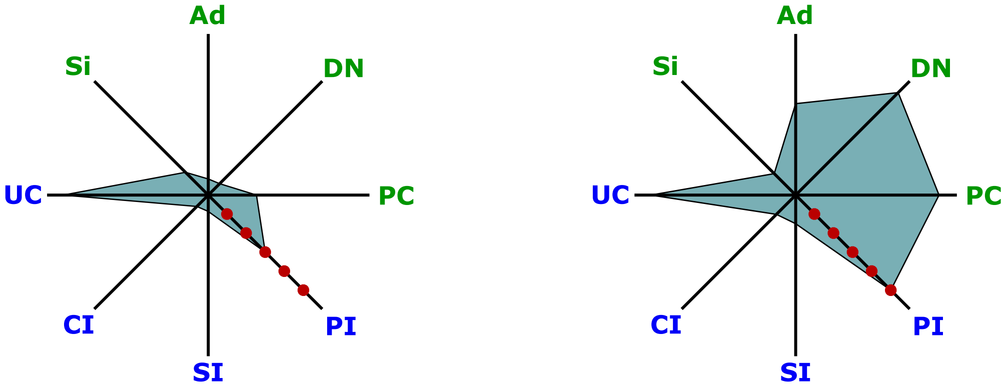
**(a)** *SSVEP* **(b)** *MMWI*



**(c)** *WamBam* **(d)** *(MINWii)*



**(e)** *Skoog* **(f)** *Clarion*



**(g)** *Augmented Bass* **(h)** *Biomuse*

**Figure 2.2.:** *Dimension space analysis of the reviewed ADMIs.*

erson [82], who developed an interactive audiovisual performance system for hearing impaired persons ("Making Music With Images", MMWI hereafter, Fig. 2.2b). The system visualizes sound in real-time to allow hearing impaired individuals to interact with an experiential representation of sound. The author foresees its use in rehabilitation mainly, but collaborative performance are also considered as a possible use context.

One notable example with a strong focus on *cognitive impairments* is provided by WamBam [83] (Fig. 2.2c). This is a self-contained electronic hand-drum meant for severely intellectually disabled users. It is shaped as a dome made of various soft pads with different colors and textures, which provide acoustic and vibrotactile feedback when touched. The authors foresee applications in music therapy sessions mainly, but discuss potential uses also in the context of musical performance.

Unlike the previous instruments, the MINWii project [84] (Fig. 2.2d) is conceived and designed exclusively as a *therapeutic instrument.* It is a music game which lets players improvise or play songs by pointing at a virtual keyboard with color-coded keys, using a Wiimote Pistol. Different implementations of the system were used both with children suffering from behavioral disorders and with elderly patients suffering from mild to moderately severe Alzheimer's disease, also taking into account possible related motor impairments.

The Skoog [85] (Fig. 2.2e) is a commercial DMI which is presented as a "tactile instrument" and consists of tangible interface that can be paired with a compatible mobile device. The interaction metaphor is loosely based on a drum instrument, and provides an example of extreme *simplification*, as potentially complex musical events are produced by pressing on one of five colour-coded buttons. The instrument is said to be used both for performance and therapy, in particular it is used with children with SEN and ASD, as well as individuals with physical impairments.

One of the few examples of ADMIs with a high degree of *adaptability* is provided by Clarion [86] (Fig. 2.2f). This is an instrument developed through a long-term charity program, with the goal of allowing performers with physical impairments to play in an orchestra. It has a strong emphasis on participatory design, interface adaptability to individual needs, and exploitation of off-the shelf assistive technologies used by persons with disabilities in their everyday lives. It can use various alternative physical channels, including gaze pointing and head movements, depending on the type and level of individual impairment, but can also be played with the fingers or the feet.

Harrison and McPherson [75] present a system for adapting the bass guitar for one-handed musicians (Fig. 2.2g). Possibly not a DMI in a strict sense, this may be regarded as an assistive technology that enables bass guitar playing by performers users with upper-limb impairments. Specifically, it enables MIDI-controlled actuated fretting via a foot pedal control. As such, this project provides a notable example of an adapted traditional instrument along the *Design Novelty* dimension.

BioMuse [87] (Fig. 2.2h) is a pioneering project which underwent several implementations, all having at their core a hardware and software setup developed specifically to collect electroencephalographic, electrooculographic, and electromyographic signals from a large number of *physical channels.* These are then processed to extract a set of relevant features and map those to MIDI events. BioMuse has been used to augment traditional musical instrument performance or as a stand-alone DMI, in the latter case being suitable also for performers with physical impairments. The system is described in more detail in Sec. 5.1.1.

## 2.4. Conclusions and future works

We have shown a possible dimension space for ADMIs design analysis, demonstrating its descriptive potential through the analysis of some state-of-the-art instruments.

There is often little information about the availability of technologies in disability related contexts. In addition to being a conceptual design tool, the dimension space we propose could prove to be a classification and categorization aid suitable for searching in catalogs. Assuming to search an instrument suitable for a specific musician's condition into a database of classified ADMIs, the system could allow quick access to a list of proper instruments.

However, the system we propose certainly has limitations in the field of classification: it does not reflect all the possible parameters and all the ADMI classification systems presented in past literature, nor does it provide a complete definition of ADMI design choices. There are some non-orderable categorical variables (e.g. related to sensors choice) that can hardly be representable in the above web charts.

Future works could be aimed at using this system in new case studies to verify its efficiency, as well as cataloging new instruments to highlight other design trends within the literature. New works about the relationship between impairments and musical abilities could lead to a better definition of the three axes *Cognitive impairment, Perceptual impairment* and *Physical impairment*, or to their discretization.

# 3

# Interaction channels for HeaDMIs

Exponential increases of available computational resources, miniaturization, and sensors, are enabling the development of digital musical instruments that use non-conventional interaction paradigms and interfaces. This scenario opens up new opportunities and challenges in the creation of accessible instruments to include persons with quadriplegia into music practice. In Sec. 1.4 we defined HeaDMIs, a particular type of skill-based Accessible Digital Musical Instruments suitable for such target group and dedicated to musical performance activities. Developing a musical instrument that allows control by people with tetraplegic disabilities implies the use of unconventional physical interaction channels: being this type of users paralyzed in large parts of the body, residual movement functions are available from the neck up.

The fact that some groups of muscles can be controlled more finely than others has well-known implications for the design of human-computer interfaces [88]. One determinant – albeit not the only one – of the performance of a group of muscles is the portion of the motor cortex devoted to it. Fig. 3.1 shows the representation of the motor homunculus obtained by Penfield and colleagues [89] from the mapping of the motor cortex: one relevant aspect is that physical channels located on the head occupy the second largest area of the motor cortex, after the hand and fingers (which reflects their evolutionary importance for verbal and non-verbal communication). This provides support to the idea of using these channels (plus the brain cortex itself) as musical controllers [90].

In this chapter such channels are reviewed for HeaDMIs interaction design, analyzing them under both musical interfacing and general Human-Computer Interaction perspectives. Sec. 3.1 provides a description of characterization concepts used to describe channels, as well as a list of sensors which can be used for their detection. Detailed analysis of each channel is provided in Sec. 3.2.

## 3.1. Channels characterization

For the sake of clarity, the analyzed channels are clustered into four groups, namely Eyes, Mouth, Head, and Brain. A summary is provided in Table 3.1.

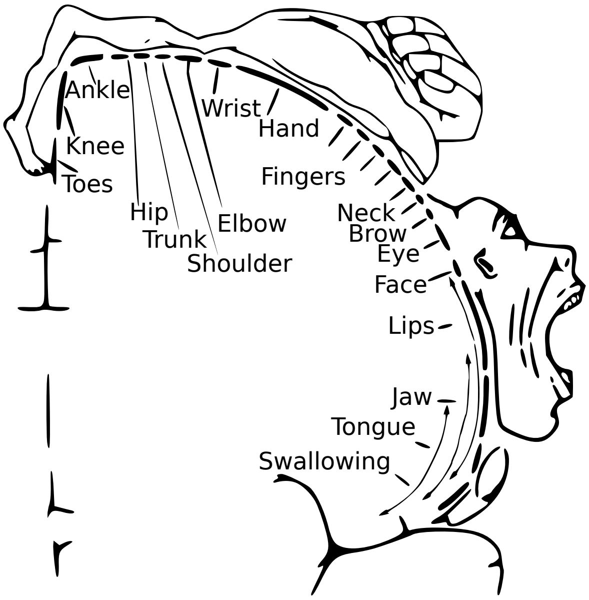


**Figure 3.1.:** *The motor homunculus, a topographic representation of body areas on the motor cortex [89].*
***Source:*** *adapted from a figure licensed under CC-BY-SA-4.0:*
*`https://commons.wikimedia.org/wiki/File:Motor_homunculus.svg`.*

Each analyzed physical channel is characterized in terms of its associated parameters, along with a list of suitable sensors for measuring such parameters, and a number of channel properties.

The properties considered here are visible in Table 3.1 (rightmost 9 columns). These were empirically selected on the basis of their utility to characterize a physical channel in terms of the mappings that it could be used for, with respect to both musical parameters and mapping strategies.

- *Resolution* refers to the number of distinct values that channel parameters can achieve, and relates to the possibility of performing fine motor actions, accessing for example a higher number of discrete pitch values.
- *Fatigue* indicates whether the prolonged use of a channel can easily tire the performer, requiring rest after a performance period.
- *Involuntary movements* can be present and interfere with voluntary use, given the nature of the channels.
- *Stability* indicates the possibility of maintaining a stable value (e.g., a given intensity or pitch) without flickering.
- *Rest* indicates whether the channel possesses a natural, stable and easily accessible rest state in which fatigue is minimized, which can be mapped to special values (e.g., zero intensity or no sound emission).
- *Persistence* indicates whether there is a physiological need to return to the rest state after a period of use, as in the case of breath emission.
- *Smoothness* indicates the degree of fluidity in changes of the channel parameters, as opposed to movement jerkiness, typical for example of gaze movements.
- *Accuracy* refers to the ability of hitting a target value with the smallest possible error, needed for example to move through pitch values.

- *Velocity* refers to the maximum attainable movement speed, regardless of accuracy, needed to move quickly through different parameter values.

A detailed characterization of the above properties would require to collect, analyze, and compare large amounts of physiological data. Moreover, it is known that user models calibrated exclusively on subjects without disabilities are not applicable to persons with various motor impairments [91]. In light of these considerations, for the purpose of this work we resort to a qualitative characterization in which these properties are only given binary values (high-low, yes-no) with reference to persons who live without disabilities, which serve as a best-case reference for motor-impaired persons.

It should be noted that the proposed set of channels could also be extended: as an example, the relatively subtle movements of nose and ears could be considered as a possible source of input control for certain individuals with motor impairments, yet no previous related studies could be retrieved at the time of writing.

### 3.1.1. Sensors

The considered sensors are summarized in Table 3.2. Many of these do not require further comments, while more complex ones merit additional discussion.

Eye trackers detect the 3D position of pupils and the absolute gaze point in the visual scene (e.g., on a screen). Majaranta and Bulling [92] provide a review of eye tracking technologies. In this paper we mainly refer to non-invasive eye trackers equipped with infrared cameras, which require short calibration times and work in natural exposure conditions, some of which are now available at relatively low prices. An alternative sensing technology is electrooculography, in which eye movements are detected by measuring the standing corneal-retinal potential arising from hyperpolarizations and depolarizations. Electrooculographic sensors consist of electrodes (typically 5) placed on the skin around the eye. They have some advantages over video- and infrared-based tracking, namely independence on lighting conditions, lightweight signal processing, mobile implementations.

Electromyographic sensors detect muscle activations by measuring electrical activity. The potential of these sensors for the development of "muscle-computer" interaction has long been recognized [93].

Head trackers detect head rotation angles (pitch, yaw, roll) and possibly translatory degrees of freedom. Hess [94] provides a categorization of current head-tracking technologies. The 3D position of the eyes can be used also for head tracking purposes, and many commercial eye trackers possess this feature, thus enabling the use of two physical channels through one sensor.

Electroencephalographic (EEG) headsets consist of a cap with electrodes. This brings about the more general topic of Brain-Computer Interfaces (BCIs), which are well discussed in several surveys [95, 96, 97]. BCIs are operated by detecting brain signals through more or less invasive sensors, chiefly EEGs. Although alternative techniques exist (magnetoencephalography, functional Magnetic Resonance Imaging, functional Near-Infrared Spectroscopy), EEG sensors have several

| Group | Channel name | Abbr. | Main parameters | Sensors | Resolution | Fatigue | Involuntary | Stability | Persistence | Rest | Smoothness | Accuracy | Velocity |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Eyes** | *Gaze pointing* | gaze | 2D coordinates, Fixation duration | eyet, cam | H | L | X | L | H |  | L | H | H |
|  | *Eye movements* | eymv | 2D angular displacements | eog, eyet, cam | H | L | X | L | H |  | L | H | H |
|  | *Blinking* | blnk | Displacement (boolean), duration | eyet, emg, eeg, acc, cam | L | L | X | L | H | X | L | L | H |
|  | *Eyebrow movements* | eybr | Vertical displacement | emg, eeg, acc, cam | L | H | X | H | L | X | L | L | H |
| **Mouth** | *Voice* | voic | Pitch, Intensity, Timbre, Speech | micr | H | L |  | H | L | X* | H | H | H |
|  | *Whistling* | whis | Pitch, Intensity | micr | H | H |  | H | L | X* | H | H | H |
|  | *Breath* | brea | Pressure, Airflow | brea, micr | H | H | X | L | L | X | H | H | H |
|  | *Mouth-lip movements* | mout | Height, Width, Aspect ratio, Area | emg, acc, cam | H | L |  | H | H | X | H | L | H |
|  | *Tongue* | tong | 3D Tip coordinates, Pressure | hall, piez, btn, acc | H | L | X | H | H | X | H | H | H |
|  | *Teeth* | teet | 3D coordinates, Pressure | piez, btn | H | L | X | H | H | X* | H | L | L |
| **Head** | *Head movements* | head | 6D coordinates | eyet, acc, head, dept, gyro | H | H |  | H | H |  | H | H | L |
|  | *Neck tension* | nkte | Tension | emg | L | H |  | L | L | X | L | L | L |
| **Brain** | *Mental states* | ment | PSA, SCA, HJA, TAV | eeg | L | L | X | L | L |  | L | L | L |
|  | *Attention* | atte | P300, SSVEP, ERNP | eeg | L | H | X | L | L |  | L | L | L |
|  | *Imagery* | imag | ERS/ERD, SCP | eeg | L | H | X | L | L |  | L | L | L |

**Table 3.1.:** *List of physical channels analyzed in Sec. 3.2, with related parameters and sensors suitable for detection. Property values are labeled as H (high), L (low) or X (present). Asterisks in the Rest column indicate that the property applies to a subset of channel parameters only.*

| **Sensor name** | **Abbr.** |
|---|---|
| Accelerometer | acc |
| Breath sensor | brea |
| Button | btn |
| Camera | cam |
| Depth (IR) camera | dept |
| Elecroencephalographic headset | eeg |
| Electromyographic sensor | emg |
| Electrooculographic sensor | eog |
| Eye tracker | eyet |
| Gyroscope | gyro |
| Hall effect sensor | hall |
| Head tracker | head |
| Microphone | micr |
| Piezoelectric pressure sensor | piez |

**Table 3.2.:** *List of sensor names and abbreviations.*

advantages in terms of invasiveness, costs, and temporal resolution. BCIs are often categorized as "passive" (using arbitrary brain activity without the purpose of voluntary control), "reactive" (using brain activity arising in reaction to external stimulation), and "active" (using brain activity that is consciously controlled by the user, independently from external events). The three brain physical channels discussed in Sec. 3.2.4 map one-to-one into these categories.

## 3.2. Analysis

This section provides detailed channels analysis, grouping them in accordance with the classification system proposed in Tab. 3.1. For each channel, physiological characteristics are described, as well as previous work in their use for human-computer interaction purposes and musical interfacing.

### 3.2.1. Eyes

**Gaze pointing.** Movements of the gaze point on an object or surface have some peculiar characteristics [92]. Saccades are rapid and short (~ 30 ms) movements. Involuntary saccades can be stimulated by fast or unexpected objects, and by the absence of reliable reference points. During fixations, the gaze point is still and focused on a narrow area. Fixations between two subsequent saccades have typical durations of 100 – 400 ms. Involuntary angular jitter (~ 0.1°) can occur during a fixation. Finally, smooth pursuits occur when the gaze follows a moving target.

Gaze pointing has well-established applications in HCI, including mouse emulation, gaze-based text entry, web browsing, gaze-controlled games, attention-aware interfaces, user modeling and monitoring [92]. Hornof [98] reviews eye-controlled music performance systems, which in most cases allow interaction with pre-defined compositions (by triggering samples and musical events), or control over music production software. He also proposes an interesting analysis of the capabilities and constraints of this channel in relation to musical expression.

One prominent challenge is the so called "Midas touch" problem [92]: fixations on an interface element may lead to its activation even when the user has no such intention. Moreover, elements crossed by saccadic movements may also be activated. Typical solutions include introducing a dwell-time (a short delay to detect fixations), using specially designed selection areas and gaze gestures, or exploiting a second physical channel to perform activations. For musical instruments, this problem can prevent the completion of basic actions (e.g., jumping between two non consecutive pitches).

Hold strategies may also be employed, exploiting fixation duration. On the other hand, because of the discontinuous character of saccades, Excitation is a less viable strategy. The main parameters are the 2D gaze point coordinates on the screen, as well as fixation duration. Given its properties, gaze pointing is particularly suited for Continuous and Discrete range selection mapping strategies (see Sec. 1.4.1.3).

**Eye movements.** Although strictly related to the previous one, this channel refers to angular displacements of the eyeballs relative to the frontal direction, with no reference to an absolute pointing direction. As such, it does not necessarily require the presence of a screen (or any fixed reference object) and has therefore the advantage of leaving freedom of movement to the subject's head and body. Regarding the properties of channel parameters, the same considerations made for Gaze pointing apply here as well.

Although vision-based eye-tracking techniques may be used, the most common sensing technology for eye movement detection is electrooculography. One specific example on the use of this channel for interacting with a HeaDMI, BioMuse, is discussed in Sec. 5.1.1. We were not able to recover additional examples in the field of assistive interfaces: in this context eye movements are always targeted at fixating points of interest on a screen, and thus fall within the gaze pointing channel discussed previously.

**Blinking.** This physical channel refers to the vertical movement of the eyelids, which is usually impulsive (a blink). The literature makes a distinction among *voluntary* (in response to an identifiable self-initiated or external stimulus), *reflexive* (in response to a potential threat to the organism) or *spontaneous* (dependent on the psychophysical state of the individual) blinks [99]. Winks (movements of a single eyelid) have different characteristics from blinks. In particular, the ability to selectively close a single eyelid seems to be linked to personal abilities, and may be compromised by some forms of motor impairments.

Numerous blink detection algorithms based on image analysis have been developed [100, 101]. In addition eye blinks can be detected through electrooculography, as well as EEG headsets, with extensive uses for assistive technologies [101, 102]. Involuntary blinks pose problems to the interaction design, and require the ability to recognize voluntary ones. Blink duration has been used as a salient feature, where the duration of a spontaneous blink is about 300 – 350 ms [99]. In the field of musical interfaces, this channel is still minimally explored.

The main usable parameter is the boolean (open-closed) vertical displacement of eyelids. Since an event lasts fractions of a second, this makes it suitable for the Trigger, Toggle, and Counter mapping strategies. Using the Hold strategy (associated to blink duration) has the undesirable side effect of occluding the visual channel for a prolonged time and compromising the use of other channels (e.g., gaze pointing).

**Eyebrow movements.** Simultaneous upward and downward movements of both eyebrows are the most straightforward to achieve, although independent movements of one eyebrow can be effectively performed by some subjects. An additional movement is squeezing, which reduces the horizontal distance between eyebrows. Spontaneous movements are present, and are known to be strongly correlated with emotional states, as well as vocal activity (particularly with prosody) [103]. Consequently, using this channel for interaction poses non trivial problems, although some studies suggest that deliberate eyebrow movements may be characterized and recognized from spontaneous ones [104]. The same studies also suggest that this channel has low resolution, and is prone to fatigue.

Correspondingly, studies in the context of accessible interfaces are almost invariably based on boolean detection of low-high eyebrow movements, e.g. to trigger mouse clicks [105]. Movements can be sensed through cameras, but also electromyography [106], and even inertial sensors such as accelerometers attached to the skin.

Similarly to eyelid movements, this channel is still minimally explored in musical interfaces (one specific example regarding a HeaDMI named EyeConductor is discussed in Sec. 5.1.14). Suitable mapping strategies are also similar and amount to Trigger, Toggle, and Counter. The Hold strategy (associated to prolonged displacements) may also be considered.

### 3.2.2. Mouth

**Voice.** The human voice is produced by a complex mechanical and acoustic system, based on the combined action of larynx (with the vocal folds), and the vocal and nasal tracts [107]. The vocal folds act as a sound source which can oscillate to produce pitched sounds, or can stay open letting the airflow through. The vocal tract acts as a filter, whose spectral characteristics are determined by the tract shape and controlled by various articulators, notably the tongue and the lips. The resonances (formants) of the vocal tract are particularly relevant to characterize vocal sounds (e.g., different vowels).

The human voice, specifically singing, is in itself a very expressive and versatile acoustic musical instrument. The use of non-verbal voice input for interactive control has been widely explored [108, 109]. In the context of musical interactions, the versatility of voice may suggest a natural mapping between singing parameters (pitch, intensity, formants, etc.) and analogous musical parameters of the sound production unit. All these parameters are suitable for Continuous range mapping strategies, and can be estimated straightforwardly through a microphone and a palette of well established signal processing techniques. Various approaches have been proposed for general purpose instruments [110, 111, 112].

On a different level, speech input is a well established interaction modality [113]. Speech-based control is less used in the context of DMIs, but may be suitable for Trigger mapping strategies, possibly mapped to musical parameters at the process level.

**Whistling.** Acoustically, a whistle is produced by the action of a Helmholtz resonator consisting of the oral cavity bounded by two orifices [114]. Anatomically, the shape of the resonant cavity responsible for the modulation of the sound is mainly given by tongue movements, rather than by jaw posture, as well as by the presence of "lateral chambers" inside the mouth, during the emission of notes at high frequencies [115].

Although this channel is much less explored than Voice, similar considerations may be made regarding available parameters (pitch, intensity), their related mapping strategies, and their estimation. In addition, anatomical considerations suggest that tongue position may also be used to infer pitch.

"Whistling user interfaces" have been proposed [116]. Musical applications are mostly focused on the use of whistling as an input to query-by-humming music retrieval systems [117]. Shen and Lee proposed a whistle-to-music composition system [118], but to our knowledge this channel has not been used in DMIs.

**Breath.** Breath is a primary interaction channel in many acoustic aerophone instruments. As a consequence it has been widely used also in DMIs, and several commercial interfaces incorporate a breath sensor. The main associated parameters (pressure and airflow) can allow for highly expressive control, as demonstrated by the variety of subtle sound nuances obtainable in acoustic instruments, and can be used in DMIs through Continuous range mapping strategies. The most typical mapping, mutuated from acoustic instruments, is between pressure and Intensity and Note on/off parameters. However different mappings may be explored.

Breath has also been considered in the context of accessibility, chiefly for the control of powered wheelchairs [119], but also for other devices, e.g., digital music players [120]. In this context breath is typically used to trigger changes in a state machine.

**Mouth and lip movements.** The shape of the lips and the mouth can be controlled voluntarily, through the action of facial muscles and the jaw. In "virtual human representation" applications (e.g., generation of avatars), parameters such as mouth aperture and mouth stretch/squeeze are typically estimated through vision-based approaches [121]. EMG sensors may also be used especially for stretch/squeeze associated to articulatory muscles on the cheek. This is demonstrated by EMG-based "silent speech" interfaces for speech disabled people [122]. Jaw movements may be also sensed through Hall effect sensors attached to the teeth.

In the context of musical interactions, vision-based approaches have been used to estimate and map this channel's parameters into musical control [90, 123], mainly at the timbral level. As an example, the Mouthesizer [90] uses aperture to control timbral parameters which are applied as audio effects to instrumental sounds.

Some parameters (particularly height and area) have high resolution due to the fine control over jaw movements, which makes them suitable for Continuous range mapping strategies. Other parameters (particularly those associated to cheek muscles) are more suited for mapping strategies such as Carousel selections, Trigger, or Hold.

**Tongue.** The tongue is capable of very rapid and precise movements [107]. It is customary to divide it into sections: tip, blade, front, back, and root. For simplicity, and in accordance with typical applications found in the literature, we limit our analysis to the 3D position of the tip. Additional parameters, such as pressure against teeth or palate, may be considered.

Tongue pointing devices have been proposed [124]. In the field of assistive technologies, several works have used tongue tip movements for the control of powered wheelchairs [125, 126, 127]. Many use a small magnet positioned on the tongue (glued or installed as a piercing), and a series of magnetic sensors (e.g., Hall effect sensors) placed on the mouth, to detect the distance of the magnet. The estimated tongue tip position is thus relative to the mouth and influenced by the position of the sensor.

Alternative sensing strategies have been proposed: Vaidyanathan *et al.* [128] used a microphone in the ear canal to detect pressure variations due to tongue movements and found that at least 4 tongue gestures could be accurately recognized; Cheng *et al.* [129] used an array of textile pressure sensors attached to the cheek and showed that 5 gestures could be accurately recognized.

In light of its high velocity and resolution, which make it suitable for Continuous range mapping strategies, the tongue is an interesting musical controller. This was proposed already in 1991 [130], however with limited success. Vogt *et al.* [131] developed a music controller based on tongue posture estimation via ultrasounds. More recently, Nam and DiSalvo [132] described an experiment in sonification of tongue movements via a Hall Effect Sensor. Involuntary movements (e.g., swallowing due to salivation) are a possible drawback.

**Teeth.** Lower (mandibular) teeth can be displaced from upper (maxillary) teeth through mandible movements, independently from mouth aperture: vertical, lateral, and – to a lesser extent – longitudinal displacements can be made.

Various studies explored this channel for hands-free interaction. Most of them share a common approach based on detection and recognition of “tooth clicks”, i.e. clenching actions. Typical applications are directed at controlling a pointer, but also include other use cases such as initiating a process (e.g., a phone call), controlling an ongoing process, or responding to a notification.

Employed sensors include EMG sensing on the temporal muscles [133, 134], sensing of vibrations in the jawbone and skull through an accelerometer (typically positioned around the external ear) [135, 136], as well as acoustic detection of tooth clicks using contact microphones on the throat or ear [137, 138]. Some of these studies report accurate recognition of up to seven different clicks [133], which can be used to emulate a mouse. Lateral and longitudinal displacement may also be recognized, through Hall effect sensors. One further parameter, the pressure between lower and upper clenching teeth, may also be measured: an example is provided by “food simulators” [139], which use pressure sensors in between the dental arches.

Such studies suggest that this channel may be used also for musical interactions, using Trigger, Toggle, Continuous range mapping strategies. Pressure may be used for Continuous range mapping strategies. It can be expected that all the parameters have high stability and low fatigue.

### 3.2.3. Head

**Head movements.** Active head movements, especially along the three rotational degrees of freedom (yaw, pitch, roll), are actively used in several everyday interactions: in conjunction with the vestibular and visual systems for postural balance [140], as a support to vision and audition in localization and target reaching tasks [141, 142], and as a mean to convey paralinguistic information in synchrony with speech utterances [143]. The related kinematics have been extensively studied [144].

Head tracking is used ubiquitously in HCI applications, including assistive technologies (as an example, powered wheelchairs operated by head gestures are common [145]). The recent rise of virtual reality technologies include head-mounted displays with integrated head-trackers. The term “virtual reality musical instruments” (VRMIs) is now used to refer to DMIs that include a simulated visual component delivered via a head-mounted display or other forms of immersive visualization [146]. However in this case head tracking is used to provide convincing immersion in the virtual environment, rather than to control musical parameters. On the other hand, some studies have explored the use of head gestures for musical control at the timbral and note levels [147, 148].

Having high resolution, accuracy, and velocity, head movements can be used with a variety of mapping strategies, including Continuous range, Hold, and Switch. Head motion produces relatively high levels of kinetic energy. It is therefore well suited for Excitation mapping strategies, to control e.g. sound intensity. Additional natural mappings may be associated to timbral parameters such as vibrato. However, the issue of fatigue associated to prolonged movements would need further investigation.

**Neck.** This channel is associated to articulation of neck muscles, which can be detected through EMG. Muscular activations leading to changes in head orientation pertain to the previous channel and have already been discussed. However, isometric contractions and relaxations of neck muscles can also be produced, with no associated head movements, and these can also be detected by EMG.

Examples of studies employing this channel for interaction are scarce. One such example is provided in the work by Hands and Stepp [149], who experimented with the use of EMG sensors on the anterior neck and on the submental surface to control the vertical displacement of a pointer in a target reaching task. Specifically, participants were instructed to produce and maintain static EMG activations at different target levels to move the icon, for various time intervals. The suitability of this channel for musical interactions remains to be explored.

### 3.2.4. Brain

**Mental states.** This channel is related to covert aspects of user state, including latent cognitive processes (arousal, workload, etc.) and "cognitive events" (perception of errors, bluffing, surprise, etc.). These can be seen as a secondary communication channel for HCI, that enriches the interaction through implicit user information [150]. Applications include interface evaluation, adaptive systems, and neuro-feedback.

Mental states can be recognized from the EEG signal [151, Ch.7]. Typical parameters are derived from power spectrum analysis, which divides the EEG signal into frequency bands ("rhythms") and uses the power density distribution across bands as a feature set for subsequent classification. A simple related parameter is the spectral centroid. Temporal features are also used, such as Hjorth parameters (activity, mobility, and complexity) [152], which are estimated on successive windows (epochs) from the time-domain signal and its derivatives.

The detection of affective states is particularly relevant for musical applications. "Affective BCIs" [153, 154] are often based on the so-called "valence-arousal" 2D space [155] to define emotional classes of interest. The first axis defines a dimension related to emotion positivity/pleasantness, while the second one defines the degree of engagement/excitement.

Estimated cognitive workload has been used e.g. for intelligent music tutoring systems [156] and automatic accompaniment [157]. Estimated affective states have been used for automatic generation of music, for composition [158], in computer games [159], or for modulating the affective user state [160]. Direct EEG sonification has also been explored as a way of representing mental states using auditory output [161], for monitoring, diagnostics, neuro-feedback, and communication.

The main quality of this channel is that it increases the information flow without requiring conscious effort. Therefore it has low associated fatigue, and no training is required. The associated latency is significant, although variable: as an example, the duration of the epochs used to perform emotion assessment can vary from 0.5 s to 5 minutes [154]. As such, this channel is especially suited to map onto musical parameters at the process level.

**Attention.** This channel comprises brain signals evoked by external stimuli [95, 96], particularly event related potentials (ERPs). As such, it depends on attentional capacity and sensory information to be intact.

One relevant example is the P300 potential, detected in the parietal cortex ~ 300 ms after the occurrence of a significant stimulus interspersed with frequent or routine stimuli. Many BCIs based on P300 use a matrix-like visual interface, operated through an "oddball paradigm" [162], in which rows and columns flash randomly: if the user focuses on a specific matrix element, a P300 peak will be produced when it flashes. The "P300 speller" is perhaps the best known embodiment of this paradigm, and allows to select letters from a matrix [163].

Steady-state visual evoked potentials (SSVEPs) are also widely used, and can be measured from the EEG at the occipital cortex during periodic presentation of visual stimuli [164], with a latency of ~ 100 ms. Error-related negative potentials (ERNPs) instead occur 200 – 250 ms after the detection of an erroneous response in a continuous stimulus-response sequence, and can be used e.g. to identify cursor movements outside a defined visual field or to detect an error in a sequence of target stimuli [165].

This channel has been used in musical interfaces. Grierson [166] proposed a "P300 composer" which allows to select and write notes using an interface based on the oddball paradigm and P300 evoked potentials. Pinegger *et al.* [167] and Chew *et al.* [168] used the P300 to select notes from a matrix. Miranda and coworkers contributed pioneering works on "Brain-Computer Musical Interfaces" (BCMIs) [151], and experimented especially with the use of SSVEPs [169, 170].

ERPs have latencies in the order of hundreds of milliseconds. Moreover their reliable detection requires many repetitions of the stimuli. This makes this channel unsuitable for triggering musical events in real time, but leaves space for process-level control. Since most working applications allow for the detection of a limited number of options, simple mapping strategies (Trigger, Toggle, Discrete range) may be used.

**Imagery.** This channel relates to the active performance of cognitive tasks associated to various types of mental imagery, including geometric shapes, familiar faces, tunes, word associations, calculations, and motor imagery [171]. The latter is the most common and amounts to imagining self movements, which activates primary sensorimotor areas: as an example, imagined movements of left hand cause event-related desynchronization (ERD) and synchronisation (ERS) in the right and left motor cortex, respectively [96]. These can be detected in specific bands.

A related mechanism is provided by slow cortical potentials (SCPs), which measure cortical EEG polarization related to preparation (e.g., readiness/planning to move) or decreased activation. It has been shown that a subject can learn to actively control SCPs by means of various mental tasks, and thus to use them for control [96].

Applications range from prosthetic limbs [172] to the control of quadricopter drones [173]. The most famous interface based on SCPs is probably the "Thought Translation Device" [174], which allows 1D displacement of a cursor. Some musical interfaces make use of this channel. Pham *et al.* [175] used SCPs to control pre-set pitch sequences, although these were merely intended as auditory feedback for SCP training rather than for music generation. Vamvakousis and Ramirez [176] used ERD/ERS parameters for a musical application that lets users move the pitch of a tone up and down in a musical scale. A similar approach is followed by the "encephalophone" [177], where users can generate different notes of a C major musical scale. The "Brain dreams Music" project [178] uses instead music imagery, specifically four chords, which are detected and played back.

This channel requires moderate to extensive training, both for subjects to learn and for the systems to gather sufficient data for classification. It also requires significant cognitive resources, leading to higher fatigue. Velocity is low: as an example, classification of imagined movements may require seconds. Simple mapping strategies (Trigger, Toggle, Discrete range) may be used.

# 4

# Evaluation of interaction channels through experimental testing

This chapter describes an experiment for the evaluation of performance parameters of interaction channels suitable for HeaDMIs development. This work was published in the paper [3].

Some experiments have been carried out in the past to evaluate interaction channels suitable for musical instrument dedicated to users without disabilities. Shima et al. [179] for example propose an empirical evaluation of fingers in rhythmic and musical interaction tasks. Evaluating interaction channels objectively is a complex task, just as it is complex to define which features need to be evaluated. While fingers are an established interaction channel for musical instruments dedicated to non-disabled people, little work was however made in the past to evaluate channels suitable for quadriplegic users.

As remarked in Ch. 3, designing instruments dedicated to quadriplegic users introduces the need to exploit uncommon interaction channels such as gaze pointing, head movement, facial expressions, breath, tongue movement and so on. Those channels are much less consolidated and exploited in the musical context. Tab. 3.1 provided a review of these channels, along with sensors suitable for their detection. Interaction parameters and desirable performance features (i.e. resolution, fatigue, involuntary movements, stability, persistence, rest position, smoothness, accuracy, velocity) have been defined and a qualitative evaluation was provided, leaving ground for a quantitative evaluation to be carried out through experimental and empirical means. Some channels may prove to be more suitable for performing certain tasks within musical interaction, fitting certain mapping strategies rather than others.

This work aims to evaluate, through a simple experiment, three interaction channels (gaze pointing, breath, head rotation) by analyzing two performance features: movement *speed* and *stability*. Results are exposed in Sec. 4.4. In light of these, possible usages of the aforementioned in the context of ADMIs and HeaDMIs design are discussed in Sec. 4.5.

## 4.1. Fitts' Law tests and ISO-9421-9

Our experiment took deep inspiration from McKenzie's formulation of Fitts' Law. As indicated by the author [180], exploring interaction performance limits is a major problem in the human-computer interaction context. Although it is often the machine's performance that is measured, quantifying human performance is also important. A widely used model in this area of research is Fitts' Law. This aims to predict human performance in target acquisition tasks, which is defined as *throughput*. Its most recent formulation, provided by McKenzie, is the following:

$$T = a + b\log_2\left(\frac{D}{W} + 1\right)$$

In this formulation:

- $T$ indicates the movement performance time;
- $D$ and $W$ respectively represent the distance and the width of the target;
- $a$ and $b$ are empirical constants dependent on the input peripheral. These constants can be deduced by performing linear regression operations starting from a valid dataset.

The term $\log_2\left(\frac{D}{W}+1\right)$ is usually referred as $ID$ (Index of Difficulty).

In order to standardize experimental evaluation methodology, the ISO 9421-9 standard was introduced, described by McKenzie [180] and used, for example, by Natapov, Wang *et al.* [58, 181] respectively for the evaluation of video game controllers and multi-touch finger input. This standard prescribes the performance of two target selection tasks, one in one dimension (with the targets arranged along a single line) and one in two dimensions (with targets arranged in a circular manner). The model is often used to evaluate hand based interaction, but McKenzie himself indicates how it can be suitable for different peripherals.

## 4.2. Methodology for our experiment

Given our different objectives, which are related to musical performance, we've chosen to set up the experiment differently than ISO 9421-9, leading to the structure described in this section. In musical performance very high interaction speed and selection stability are required: our experiment is designed to evaluate both characteristics. The differences can be summarized as follows:

*a*) some of the analyzed channels offer only one degree of freedom, hence the two-dimensional extension of the experiment has been removed;

*b*) an additional degree of freedom would be necessary to make a selection (e.g. mouse click), which is not possible with the investigated channels;

*c*) as stated above, we are also interested in detecting selection *stability*: a new section was thus introduced to detect it;

*d*) the target width has been kept fixed to standardize the stability measurements.

Interaction occurring through a physical channel is inextricably linked to the type, setup and quality of the used sensor. Thus it must be specified that channel-sensor pairs are evaluated. Sec. 4.3.1 describes the analyzed pairs in detail providing details about employed sensors peripherals.

The used software GUI is shown in Fig. 4.1. Interaction takes place on the depicted *interaction bar* (dark gray), located in the center of the screen. The *cursor* (in white) can only be moved horizontally within the limits of this bar. A marker line (in red) indicates the center of the *target zone* (light gray).

**Figure 4.1.:** *A screenshot of the interface used to run the experiment.*

### 4.2.1. Procedure

The experiment was conducted using a within-group design. The following two simple instructions were given to the testers:

1. move the cursor to reach the target area as quickly as possible;
2. once reached, keep it in the center of the target as stably as possible until the end of the *trial*.

Each test session, with each tester, followed this procedure:

*a*) each tester performed a *series* of *trials* for each channel-sensor pair. The order of the channel-sensor pairs is randomized to be different for each tester, to avoid as much as possible learning or fatigue effects;
*b*) each *series* included 15 *trials*, of which the first 5 were unrecorded training, while the next 10 were recorded performances;
*c*) when ready to begin a trial, the tester pressed a key to start it;
*d*) a *target* appeared on the far left of the bar. The tester had to position the cursor inside the target for two seconds.

Following this, the test entered two main logged and recorded phases.

**Selection phase.** After those two seconds, an acoustic signal warned the tester and a new target appears to the right inside the bar at a given *distance*. The tester had a maximum of 5 seconds to make a target *selection*: after this time, the trial is declared as *failed*. A *selection* was considered valid only if the cursor remained within the target diameter for at least 1 second (which is not considered in the reported time). The line indicating the center of the target turned green to confirm the user that the cursor is inside the target.

**Stability phase.** Just after the selection had been confirmed, the *stability* analysis started, denoted by a purple target center marker. During this phase the user had to keep the cursor as centered and stable as possible on the target. This phase lasted 2 seconds, after which a new acoustic signal notified the end of the trial, and the target disappeared.

During the 5 training trials, the target distance was fully randomized to avoid learning effects. During the 10 recorded trials, the center of the target cycled through 5 possible distances (repeated twice in a series), denoted as *D200, D300, D500, D700* and *D900*. Numbers indicate the target center distance, as a proportion of a D1000 interaction bar. The order was again randomized.

At the end of each series, one minute break was provided to change the sensor peripheral. The total time needed to carry out a single session with a single tester was around 30 minutes.

At the end of the test session, the tester was asked to fill in a questionnaire. This was useful to investigate other aspects of the interaction (i.e. questions *a, b, d, f*) or to investigate the personal perception of the measured parameters (i.e. questions *c, e*). Answers were provided using a 7-value Likert scale, where (1) always denoted a negative trait and (7) a positive one. The questions, repeated for each sensor, were as follows:

*a*) **Fatigue**. Did you feel fatigue, tiredness, or pain during the performance? [1 = Drastically fatigued; 7 = I felt no fatigue]
*b*) **Usability**. Did you find the interaction easy and comfortable or frustrating? [1 = Intolerably frustrating; 7 = Perfectly handy]
*c*) **Precision**. Did you find the interaction accurate and precise? [1 = Totally imprecise; 7 = Absolutely precise]
*d*) **Involuntary movements**. During the test, did you perceive involuntary movements which affected the position of the cursor? [1 = Involuntary movements have totally compromised the performance; 7 = I didn't feel any involuntary movement]
*e*) **Speed**. Did you find the selection method quick and fast? [1 = Terribly slow; 7 = Very fast]
*f*) **General opinion**. Imagine having only this interaction channels available to use the computer and make very fast and stable selections of elements on the screen in a similar fashion to how you did in the test. Give a general rating of the channel. [1 = Bad; 7 = Excellent]

In order to stimulate testers to perform at their best, the experiment was "gamified" by giving away small prizes to the two testers who would have obtained the highest *selection time* and greatest *stability*.

## 4.3. Resources

In this section we specify hardware details for the employed hardware, as well as software implementation for the test interface. Lastly, details about the testers sample are given.

### 4.3.1. Hardware and sensors

As already indicated in Sec. 4.2, sensors peripherals play a key role in defining the quality of the interaction. The following is a list of technical details for all channel-sensor pairs.

**Hands (Mouse).** This channel was introduced as a touchstone for the other channels. We used a high-end gaming mouse, namely a *Corsair M65 Pro*[1], equipped with a 12,000 DPI resolution optical sensor, operated on a smooth surface. Sensitivity was set to 1500 DPI for the experiment.

**Gaze point, raw (Eye Tracker).** Only the horizontal axis of gaze movement is evaluated. We used a *The Eye Tribe* eyetracker, which has a sampling frequency of 60Hz and an accuracy of about 0.5-1.0° on the visual field[2]. Several studies have analyzed its applicability in research contexts [182, 183]. The cursor followed the gaze point.

**Gaze point, smooth (Eye Tracker).** This setup is the same as the previous, but in this case a smoothing filter made available natively by the eye tracker API was activated[3].

**Breath (Breath sensor).** NeeqBS, an open-source hardware breath sensor peripheral was used. The peripheral is described in Sec. 8.1, but here are summarized the most relevant technical specifications for the experiment. The peripheral is equipped with a *NXP MPX5010DP* low pressure sensor, which detects a pressure range of 0-10KPa $\approx 102cmH_2O$, has a sensitivity of 1mV/mm and a response time of 1ms. The sensor is directly interfaced to the computer with an *Arduino Uno* microcontroller. A rubber tube is connected to the sensor inlet, with a simple interchangeable plastic mouthpiece at the other end. Since there are no holes, the air is not vented and remains compressed inside the tube. At zero pressure, the cursor is positioned to the left end of the *interaction bar*. For the experiment, the useful range was reduced to 5KPa $\approx 51cmH_2O$ (i.e., a pressure of 5KPa is needed to reach the opposite end of the *interaction bar*), to reduce fatigue and make the pressure reachable for all the subjects [184]. A sampling frequency of ~200Hz has been measured averaging the sampling time recorded in the output files.

**Head Yaw, Pitch, Roll (Head Tracker).** NeeqHT, an open-source hardware head tracker was used. The peripheral is described in Sec. 8.2, but again relevant technical specifications are summarized. The peripheral is built using the *MPU-6050 (GY-521)* 6DoF accelerometer and gyroscope integrated sensor, interfaced with the computer through an *Arduino Nano* microcontroller. On the latter a program able to extract the absolute angular position on 3 degrees of freedom was loaded. On all the three rotational axes, the range required to move the cursor from the left to the right side of the interaction bar was 40° with respect to the head rotational axes. The system was

---

[1] Corsair M65 Pro mouse on Corsair website: `https://www.corsair.com/us/en/Categories/Products/Gaming-Mice/FPS-Fast-Action-Mice/M65-PRO-RGB-FPS-Gaming-Mouse-%E2%80%94-Black/p/CH-9300011-NA`

[2] The Eye Tribe eye tracker on IMotions website: `https://imotions.com/hardware/the-eye-tribe-tracker/`

[3] The Eye Tribe API on The Eye Tribe website: `https://theeyetribe.com/dev.theeyetribe.com/dev.theeyetribe.com/api/index.html`

calibrated so that a natural rest position corresponded to the cursor placed in the center of the interaction bar. The measured sampling rate was ~100Hz.

**Computer.** An *Apple MacBook Pro* (2017) with *Windows 10* operating system (installed as dual-boot system through Bootcamp[4]) was used for all the experiments. The laptop was equipped with a dual-core Intel Core i5 CPU at 2.3GhZ, 8GB RAM LPDDR3 at 2133MHz and Intel Iris Graphics 640 GPU with 1536MB VRAM. Sensors and screen were connected via a Thunderbolt/USB+VGA port adapter.

**Screen.** A 21-inch VGA monitor capable of a 1920x1080px resolution was used. The *interaction bar* was 24.7cm long (out of a 47.7cm screen width).

### 4.3.2. Software implementation

The test software was developed in C#, using the .NET Framework 4.8 and WPF. The source code is available online[5] and is provided under an open-source GNU-GPLv3 license to allow anyone to repeat the experiment. In addition to providing the interface in Fig. 4.1 and the automated performance of the experiment, the program records to file all the samples recorded by the sensors. Each new sample loaded on the USB buffer generates an interrupt which causes file writing. The elapsed time for each sample is given by the *Stopwatch*[6] class (belonging to the System.Diagnostics namespace in .NET Framework 4.8). This class provides a very high performance timer usually employed in code execution time micro-benchmarks. Since there could be small differences in timing depending on the CPU, we preferred to carry out all the experiments using the same computer.

### 4.3.3. Participants and sessions

A sample of 16 non-disabled people, aged between 22 and 47, participated in the experiment. None of them had previous e xperiences with the investigated interaction channels except for the mouse and, in one case, the breath (since they had some experience in saxophone playing). Everyone had prior experience in the use of computers. All the sessions were held in the Laboratory of Music Informatics (LIM) at University of Milan, Italy, between December 2019 and January 2020. Raw data resulting from the sessions, for both performance tests and questionnaire, have been published in anonimized form on the aforementioned GitHub repository[7].

---

[4] Bootcamp official web page: `https://support.apple.com/it-it/boot-camp`

[5] GitHub repository for the test suite software: `https://github.com/Neeqstock/HanDMIs-TestSuite`

[6] Documentation for the Stopwatch class: `https://docs.microsoft.com/en-us/dotnet/api/system.diagnostics.stopwatch?view=netframework-4.8`

[7] `https://github.com/Neeqstock/HanDMIs-TestSuite/tree/master/Raw%20data`

### 4.3.4. Limitations

The small number of people in the sample may represent a statistical limitation. The initially planned number was around 25-30 people, but the experiments had to be stopped due to the onset of the COVID-19 epidemic in Italy in February 2020. It was not possible to access a significantly random sample: acquaintances were used to recruit testers. No data was available to verify how much this can impact on the quality of the statistics. All testers didn't have disabilities: the test was performed under the assumption that a quadriplegic user has same level control of the analyzed interaction as a non-disabled user (except for the mouse). Moreover, defining a representative random sample of quadriplegic users, covering every possible condition, could be an impossible task.

## 4.4. Results

### 4.4.1. Tests

Fig. 4.2a and 4.2b show the results for the interaction tests.

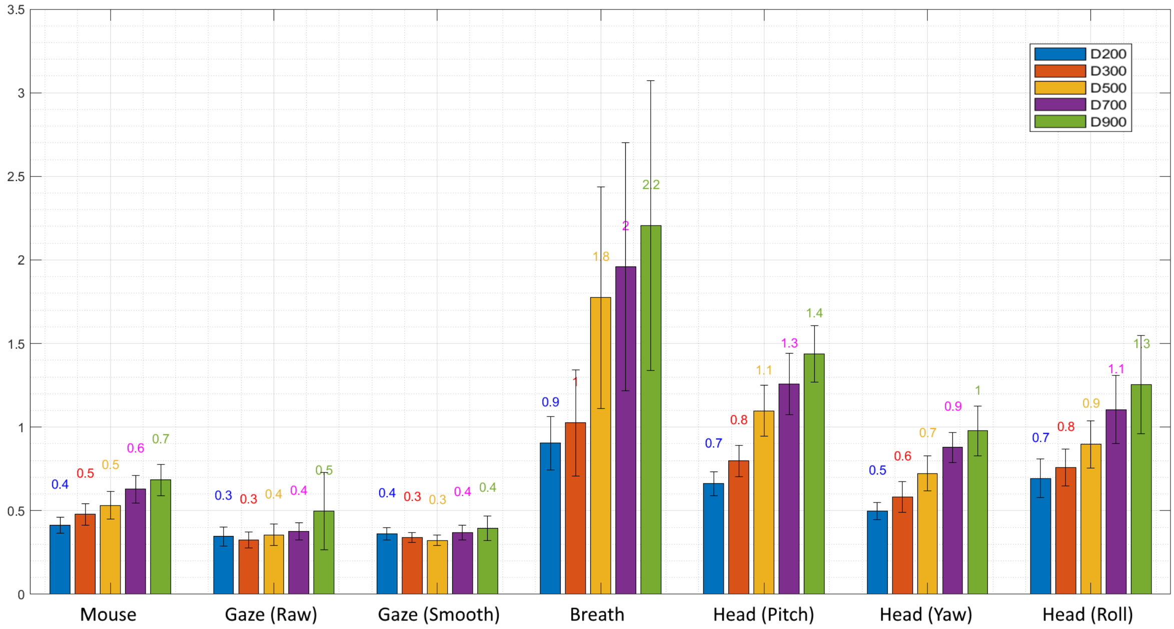
The following types of samples were excluded from the analysis:

*a*) trials failed due to expired time, as indicated in Sec. 4.2;
*b*) trials in which the user performed a wrong cursor movement due to a self-reported misunderstanding of the task;

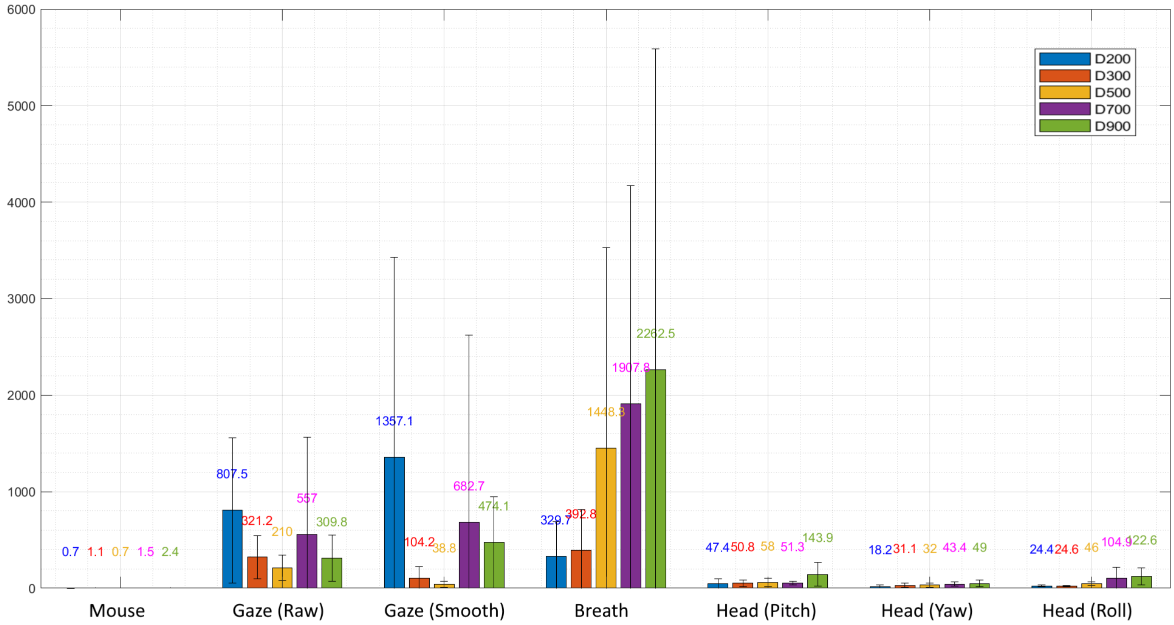
In addition, Once the *selection time* and *stability* measurements were calculated for each trial, values falling outside the 10th and 90th percentiles have been removed as outliers. Such outliers removal method was chosen due to the fact some trials were failed due to events such as hardware malfunctions, testers distraction or non self-reported misunderstandings of the task, and we couldn't reconstruct the exact occurrences a posteriori.

Fig. 4.2a shows the results for *selection time*. These results were obtained, after the outliers removal procedure, by averaging selection times of all the available trials, divided in channel-sensor and distance classes. Error bars represent the standard deviation. The presence of two important factors to influence each value can be noticed: a reaction time (formalized by the $a$ parameter in the Fitts Law model in Sec. 4.1) and a parameter that influences the weight given to the distance (modeled by the $b$ parameter).

A least square linear regression was run to estimate those parameters. An ANOVA test for repeated measures revealed a non-significant difference ($p = .487$) between the $a$ parameters, suggesting comparable reaction times for all channels. In order to obtain a more accurate estimate for the $b$ parameter, the model was then adjusted by imposing the same value for $a$ in all channels, and linear regression on the adjusted model provided the results reported in Table 4.1. A subsequent ANOVA test for repeated measures resulted in statistically significant difference for the $b$ values ($p < .001$). Post-hoc pairwise tests with Bonferroni-Holm correction revealed non-significant differences only in the following pairs: [Mouse - Gaze (Raw)] ($p = .23$), [Gaze (Raw) - Gaze (Smooth)]



**(a)** *Selection time, in seconds. Lower is better.*



**(b)** *Selection stability expressed as samples variance, in milliseconds. Lower is better.*

**Figure 4.2.:** *Results for selection time and selection stability tests. The indicated error measure for error bars is standard deviation. Bar colors refer to the various distances (see legend).*

($p = .21$), [Head (Roll) - Head (Pitch)] ($p = 0.21$). All remaining pairs showed significant differences with $p < .05$.

| | Mouse | Gaze (raw) | Gaze (smooth) | Breath | Head (pitch) | Head (yaw) | Head (roll) |
|---|---|---|---|---|---|---|---|
| $b$ | 0.2304 | 0.1896 | 0.1273 | 0.8637 | 0.5315 | 0.3537 | 0.4690 |

**Table 4.1.:** *Estimates of b parameters for each interaction channel.*

Gaze interaction resulted very quick in both the smooth and the raw setups, and slightly more independent of the target distance (especially in the "Smooth" formulation). The three head channels appear to have comparable performance, although Yaw seems to perform slightly better. This could be explained by: (1) daily habit of turning the head on that axis; (2) greatest rotational range; (3) congruence between rotation direction and cursor movement, resulting in a more natural mapping. Breath seems to perform worse in this task. The variability of the samples also appears to be very high for distances D500, D700, D900.

Fig. 4.2b shows the results for *stability*. This was calculated as follows: (1) for each trial and within the time interval of the *stability phase* described in Sec. 4.2, the variance was calculated with respect to the average cursor position; (2) after outliers removal (with the same procedure as before), the variances thus obtained were averaged within channel-sensors and target distance classes. The error bars indicate the standard deviation of the variances.

All the head related channels seem to have great stability, comparable to that shown by the mouse. The gaze point stability is negatively affected by the jerky nature of the sensor output, as well as fixations natural instability [185]. Curiously, the smoothing algorithm provided by EyeTribe (indicated on the graph with the *Gaze (Smooth)* group) does not seem to have brought significant improvements compared to the unfiltered output (indicated by the *Gaze (Raw)* group). Reasons for the apparent odds for different distances are not clear. Breath is visibly more unstable especially for high distances, with a great trial-to-trial deviation. Testers reportedly found it difficult to keep a stable pressure in the case of long distances, given that the required pressure was greater. However, as indicated in Sec. 4.3.4, different channel configurations could be tried to confirm or deny this quality.

Tab. 4.2 shows the total number of failed trials for expired time, as indicated in Sec. 4.2, divided by channel and distance. There is a high number of failures for the *Breath* channel, increasing with the distance: as an example, for D900, 10 trials out of 32 have failed.

| Channel | D200 | D300 | D500 | D700 | D900 |
|---|---|---|---|---|---|
| Gaze (Raw) | 0 | 0 | 0 | 0 | 1 |
| Breath | 0 | 1 | 2 | 9 | 10 |

**Table 4.2.:** *Total number of trial failures (elapsed time) divided by channels and distances. Only Gaze (Raw) and Breath are reported since the number of failures for other channels was always 0 for any distance.*

### 4.4.2. Questionnaires

Fig. 4.3 shows the results of the proposed questionnaires. The median was used as central tendency measure, while the error bars indicate the interquartile range (IQR). Wilcoxon rank sum tests with continuity correction have been performed to verify statistical relevance of the findings discussed below. The *Mouse* channel have been used as a touchstone for most of the comparisons.

With regards to *Fatigue*, results seem to indicate that head related channels are not or negligibly fatiguing, with performances comparable to those of the mouse, while some fatigue or discomfort related to gaze interaction is indicated (in particular in the unfiltered formulation) as well as with the breath. Results related to breath and gaze channels are significantly different from mouse ($p < .001$)

Testers found the *Usability* of the various channels lower than that of the mouse but generally comparable (differences are not statistically significant), except for the breath, which scored quite low (significantly different from mouse with $p < .001$).

The perceived mouse *Precision* seems to be superior to the other channels, followed by head related channels, gaze and, finally, the breath. This order seems congruent with the stability results from the interaction test. Differences between mouse, head (pitch) ($p = 0.005$) and head (yaw) ($p = 0.003$) have been found less significant.

According to the results of the *Involuntary movements* question, mouse interaction was perceived as free. A slight presence of involuntary movements was noted for all the other channels (with some indecision regarding the breath, highlighted by the higher IQR), becoming more important for unfiltered gaze (significantly different from mouse with $p < 0.001$): this can be explained by the incessant and snappy movement of the cursor, which could attract the attention and, subsequently, gaze movements.

Results regarding *Velocity* seem to confirm the high performance of the eye tracker shown by the performance test, perceived more or less on a par with the Mouse. Among head related channels, a lower speed seems to be perceived with regard to Roll, however not confirmed by the interaction tests results. Breath has ranked lower, in accordance with the interaction tests. However, differences for all channels have been found to be not statistically significant.

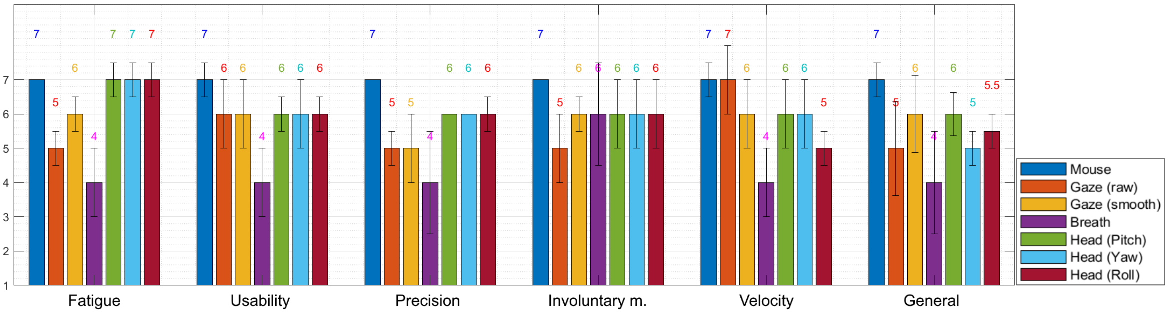


**Figure 4.3.:** *Results for the questionnaire divided by question and channel-sensor pairs (higher is better).*

Excluding mouse which had the highest results, *General* ratings seem to show a preference for *Gaze (Smooth)* and *Head (Yaw)* channels. These could be explained, in the first case, by the high speed and reduced vibration of the cursor compared to Gaze (Raw) and, in the second case, by the greater naturalness compared to other head related channels (as indicated in 4.4.1). However differences between channels excluding mouse have been found to be not statistically significant. Breath got the lowest score (significantly different from the mouse with $p < .001$).

## 4.5. Discussion

In this section, possible use of the channels in ADMIs and music interaction are discussed in light of the obtained results. It should be noticed that (considering *Gaze* as a single channel) these channels are independent: they are based on different degrees of freedom of physical movement, and there are sensors capable of independently detecting all of them.

The high speed of the saccadic movements seems to justify the choice of using the Gaze for note selections in gaze-based DMIs (e.g. The EyeHarp [62], Netytar [7] or EyeJam [186]). However, given the nature of gaze movement and related sensors, *stability* is not particularly high. As indicated by Hornof [98], several solution could be implemented to deal with those stability issues, such as the introduction of filters (e.g. in The EyeHarp [62]) or particular interface designs (e.g. in Netytar [7]). As discussed in Sec. 6.2, there's a low upper limit for the maximum number of saccades per second.

This constraints could lead to a reconsideration of the head related channels (e.g. *Pitch* and *Yaw*) as a note selection method in 2D interfaces. Little exploration has been carried out in this regard, even if some DMIs exploit Pitch and Yaw through the movement of mouthpieces (e.g. Magic Flute [187] and Jamboxx [188]). Gaze-based interfaces are sometimes adapted for head interaction (e.g. The EyeHarp [189]). References to the use of *Roll* movements in DMIs have not been found. This could be mapped to pitch bend, vibrato or other musical performance parameters.

Breath had comparatively lower scores in almost all interaction tests and questionnaire sections. Despite its widespread application in assistive technologies (as discussed in Sec. 3.2.2), it performed badly in precise target selection tasks. However it can reconsidered for musical interaction given the following observations. First, breath is naturally linked to the regulation of sound intensity (e.g. while singing or playing aerophones). Furthermore, the emission of breath inside a tube, with the introduction of an air vent hole, could cause a sensation of kinetic energy movement. This does not happen for the other analyzed channels, with which the cursor can be stable in any position even if there is no movement. Hunt [46] highlights how the sensation of introducing energy in the system can result in a natural mapping with sound intensity.

Finally, it should be noted that other configurations should be explored even for the channel-sensor pairs analyzed in this work: for example, the position of the cursor in this experiment could be mapped to head movement speed in the three rotation axes.

# 5

# State of the art in HeaDMIs

As mentioned in Sec. 1.3 some recent works have provided broad surveys of ADMIs, including both research projects and commercial products. Frid [34] provides an extensive review which encompasses several target groups, including persons with physical, sensory, and cognitive impairments, persons with complex needs or special educational needs, elderly or young children. The review of Larsen *et al.* [37] has a similar broad focus. Graham-Knight and Tzanetakis [35] propose a set of principles for how to work with people with disabilities to develop a new musical instrument. A large percentage of the ADMIs reviewed by Frid focuses on user groups with physical impairments (39.8%). However, many of these are aimed at musicians with partially able limbs, while only 4.8% of the reviewed ADMIs are devoted to persons with quadriplegia.

Building on the framework developed so far, in this chapter we review a list of HeaDMIs previously proposed either in the scientific literature or as commercial products. The literature search was performed on Google Scholar while the search for commercial products was performed on Google. Both were based on free text terms related to DMIs, namely "digital musical instrument", "musical (or music) interface", which were coupled (logical AND) to additional terms related to one of the physical channels discussed in Ch. 3. Additionally, we analyzed the lists of references from previous reviews on ADMIs [34, 37]. The retrieved items were selected based on four inclusion criteria. Specifically, they had to:

*a*) present a skill and performance based DMI, according to the definition of "performance instrument" described in Sec. 1.2. Thus, musical interfaces related to offline composition, sequencing, playback, etc. were not considered;

*b*) make use of one or more physical channels among those discussed in Ch. 3, with no additional ones (thus, neither augmented instruments nor DMIs requiring the use of upper/lower limbs were considered);

*c*) describe a concrete – albeit prototypical – working implementation (thus, theoretical studies and reviews were not considered);

*d*) make explicit reference to users with some form of motor impairment, among the target user groups.

Instruments described in the development part (II) of this thesis have been excluded from the list.

As a result, a relatively small number of HeaDMIs was retrieved: a summary is provided in Tab. 5.1, while Sec. 5.1 presents a structured analysis for each of them, considering the employed physical interaction channels and mappings, as well as sensors and interfaces. The degree and type of instrument evaluation (if any) are also mentioned. Based on this analysis, Sec. 5.2 provides a comparative discussion of the surveyed instruments, while Sec. 5.3 reviews design perspectives and future challenges.

## 5.1. Analysis

In the next pages, instruments are presented in chronological order with respect to the first publication or product release.

| **Instrument** | **References** | **Year** | **Sensors** | **Mapping (phys. channel → musical param.)** |
|---|---|---|---|---|
| BioMuse | [87, 190] | 1990 | eeg, eog, emg | ment→isel ; eymv→pann(stereo) |
| Tongue-Controlled EMI | [191] | 2004 | btn | tong→pitc/hcha,note |
| Hi Note | [192, 193] | 2005 | head, brea | head→pitc ; brea→note |
| Magic Flute | [187] | 2007 | gyro, brea | head→pitc ; brea→note,inte |
| Jamboxx | [188] | 2009 | ?, brea | head→pitc ; brea→note,inte |
| Lumiselo | [194] | 2010 | eyet, brea | gaze→pitc ; brea→note,inte |
| SSVEP BCMI | [81, 195] | 2011 | eeg | atte→pitc |
| EyeHarp | [59, 62, 196] | 2012 | eyet, eeg | gaze→ pitc,note,inte,vibr,hcha ; ment→temp,msel |
| Eye play the Piano | [197] | 2014 | eyet | gaze→pitc,note,hcha |
| P300 Harmonies | [198] | 2014 | eeg | atte→hcha |
| Imitone | [199] | 2014 | micr | [voic,whis]→ pitc,note,inte,brig,glid,vibr |
| Clarion | [86] | 2015 | eyet | [head,gaze]→note,pitch,vibr,brig |
| EyeJam | [186] | 2015 | eyet | gaze→pitc,note |
| Eye conductor | [200] | 2016 | eyet, cam | gaze→pitc,note ; eybr→tran(octave) ; mout→brig |

**Table 5.1.:** *List of surveyed HeaDMIs. Refer to Tables 1.1, 3.1 and 3.2, for musical parameters, physical channels and sensors, respectively.*

### 5.1.1. BioMuse [87, 190]

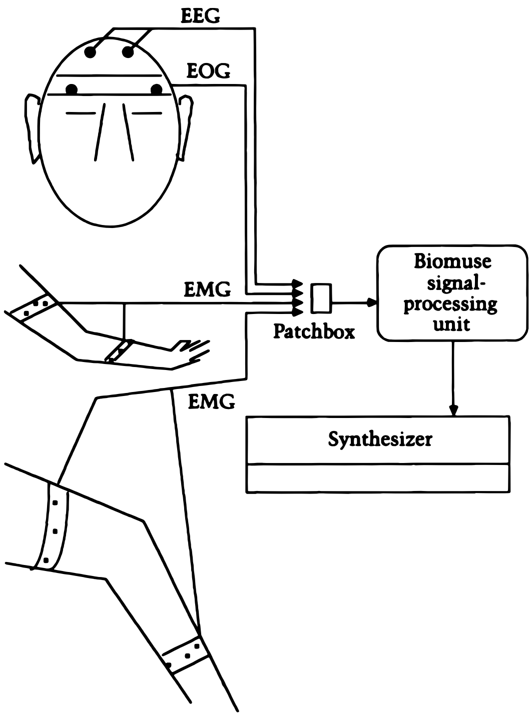


**Figure 5.1.:** *A diagram showing the sensors scheme for Biomuse.*
***Source:*** *[87] (processed and vectorized for better quality).*

Originally defined as a “biocontroller”, this was a pioneering project (1990) and is included in this survey also for historical reasons. The system underwent several implementations, all having at their core a HW/SW developed specifically to collect EEG (electroencephalographic), EOG (electrooculographic), and EMG (electromyographic) signals, extract a set of relevant features, and map those to MIDI signals. Currently, a musical ensemble called “The BioMuse Trio” performs using BioMuse, a violin, and a laptop.

The first implementation used 3 channels: Mental states (Brain), Eye movements (Eyes), and Voice (Mouth, optional). The employed sensors were two EEG electrodes (occipital lobe and frontal area), two EOG electrodes, and a microphone. Bands with disposable snap electrodes were used. A scheme of the instrument is depicted on Fig. 5.1. In addition the instrument included a variable number of EMGs. Although Knapp [87] discusses an example where EMGs are around biceps and forearms, the system is agnostic with respect to EMG positioning.

Possible mappings are exemplified in the first implementation. There, two EMG signals were mapped to the intensity and pitch of a synthesizer, EOG signals to stereo panning, and the Alpha component of the EEG signal to instrumental sound (from violin to glockenspiel). No extrinsic feedback is provided. The first implementation included a GUI which could be used to modify sensors thresholds and channel to MIDI mapping.

### 5.1.2. Tongue-controlled Electro-musical Instrument [191]

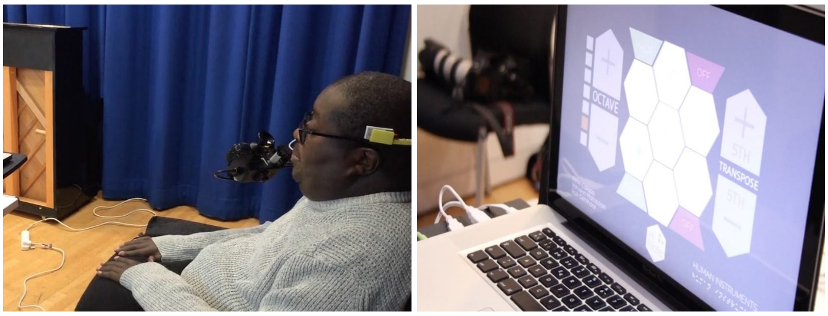
**Figure 5.2.:** *The PET board placed on the palate in Niikawa's Tongue-controlled Electro-musical Instrument.*
***Source:*** *[191].*

This DMI uses exclusively the Tongue (Mouth) physical channel. Tongue position is detected by a PET board extruding from the mouth, with 5 buttons (switches) placed on the palate (Fig. 5.2). By virtue of the mechanical contact of tongue with buttons, tactile and proprioceptive intrinsic feedback is provided to the performer. Button activations are mapped into MIDI events.

The instrument employs a Discrete range mapping strategy with a small number of values, as well as a Counter mapping strategy. Four buttons ("scale control switches") are arranged in a + shape and are mapped to four different chords. The fifth button (the "chord shift switch") is placed below them and changes the pattern of assigned chords along a circular set of available patterns, depending on the tune to be performed. This allows to explore a reasonably wide palette of chords.

The instrument was evaluated with three non-disabled subjects, mainly for the purpose of assessing rhythmic capabilities of the tongue: two subjects with previous musical experience were able to maintain a sufficiently correct rate of button depressions at three different BPMs (60, 120, 180). The song "Twinkle, twinkle, little star" was performed by a subject with no musical experience at 65 BPM.

### 5.1.3. Hi Note [192, 193]



**(a)** *Headset* **(b)** *Software interface*

**Figure 5.3.:** *HiNote depicted in two images.*
***Source:*** `https://mtflabs.net/hi-note/`.

The instrument name refers to the most recent iteration of a commercial HeaDMI developed over several years. Previous iterations included the Headspace and the Typhoon (the year reported in Table 5.1 refers to the first reported release of the Headspace).

The employed channels are Head movements (Head) and Breath (Mouth). The former is mapped to pitch, discretized as a set of notes visible on the screen (Discrete range). Regarding the latter, audio-video documentation shows that the pressure parameter of Breath can be alternatively mapped to Note on/off (Toggle) or to the rate of Note on events (Continuous range), with higher pressure values producing faster "ribattuto" effects. The choice among these mappings is left to the performer. In a sub-section of the graphical interface, the same channels are mapped to additional control at the process level (octave transposition). Head movements are tracked using a 9-axis sensor equipped with a 3DoF Accelerometer, a 3DoF Gyroscope, and a 3DoF Magnetometer, which is claimed to provide accurate and high-resolution tracking. Mouth pressure is detected by a breath sensor. The instrument is shown in Fig. 5.3.

No formal evaluation studies were retrieved. However, the Hi Note is used in public performances and notably it has featured in the 2012 Paralympic Games closing ceremony, with the British Paraorchestra.

### 5.1.4. Magic Flute [187]

**Figure 5.4.:** *The Magic Flute instrument.*
***Source:*** *`https://sites.google.com/site/windcontroller/order`.*

This commercial DMI, depicted in Fig. 5.4 uses Head movements (Head) and Breath (Mouth) as physical channels. The instrument physically consists of a mouthpiece rigidly connected to the main instrument body. The body swivels on a standard camera/microphone mount, and is hinged on the lateral axis. Consequently it follows head rotations (pitch) of the performer holding the mouthpiece. Thus, head position is inferred through mechanical contact with the instrument.

The vertical rotation of the instrument (head rotation along pitch angle) is detected by an embedded gyroscope and is mapped to musical pitch, discretized along the musical scale (Discrete range). Mouth pressure, detected by a breath sensor, controls Note on/off events (Toggle) and intensity (Continuous range). An additional control module allows the performer to choose from a set of predefined instrumental timbres.

The instrument provides no extrinsic feedback to the performer. No formal evaluation studies were retrieved. However, the Magic Flute is notably used in the musical activities of the Dutch foundation “My Breath My Music”, devoted to people with motor impairments.

### 5.1.5. Jamboxx [188]

**Figure 5.5.:** *The Jamboxx instrument.*
***Source:*** *`https://thenextweb.com/news/paralyzed-man-invents-hands-free-instrument-nou`*

This is a commercial DMI which works very similarly to an acoustic mouth harmonica. The instrument is depicted in Fig. 5.5. It shares several features with the Magic Flute and uses Head movements (Head) and Breath (Mouth) as physical channels. It consists of a horizontal body that swivels on a standard camera/microphone mount and is equipped with a movable mouthpiece. This can slide along the lateral axis of the body and consequently it follows head rotations (yaw) of the performer holding the mouthpiece. Additionally the instrument can also rotate vertically along the lateral axis.

Yaw and pitch head rotations are detected through the lateral displacement of the mouthpiece and the vertical rotation of the instrument (details of the sensors are not provided). The former parameter is mapped to pitch discretized along the musical scale (Discrete range), like in a mouth harmonica, while the latter is used to add pitch glides ("pitch bend" effects, Continuous range). Mouth pressure, detected by a breath sensor, controls Note on/off events (Toggle) and intensity (Continuous range).

Tactile feedback is provided by means of "bumps" (crests and troughs) on the instrument face which, in analogy to frets on a guitar, provide information about horizontal position. Vogels [201] evaluated the usability of the instrument, with five non-disabled subjects.

### 5.1.6. Lumiselo [194]

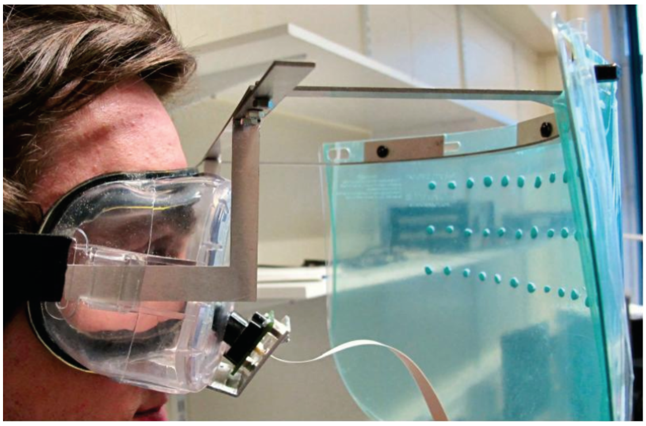


**Figure 5.6.:** *Lumiselo.*
***Source:*** *[194].*

Presented by the authors as an "electronic wind instrument", it makes use of Gaze pointing (Eyes) and Breath (Mouth) as physical channels. It consists of head-mounted goggles equipped with a custom-made eye-tracker (infrared LED and camera), as well as a breath sensor connected to the performer's mouth by a rubber tube, as depicted in Fig. 5.6. The authors emphasize that this design allows free head movements to have no effect on the location of the performers pupil with respect to the visor (unlike gaze pointing interfaces based on external monitors).

A 12 × 3 grid of LEDs on the visor represents three octaves of a chromatic scale and the corresponding 2D coordinates of the gaze point are mapped to pitch (Discrete range). Mouth pressure controls Note on/off events (Toggle) and intensity (Continuous range).

The LED pointed by the performer's gaze changes color on the visor. Additionally, the pressure detected by the breath sensor controls the brightness of the same LED. This provides extrinsic visual feedback, which is claimed to improve performance and engagement (although no formal evaluation is presented).

### 5.1.7. SSVEP BCMI [81, 195]

**Figure 5.7.:** *SSVEP BCMI.*
***Source:*** *[81].*

One of the many BCMIs developed by Miranda and coworkers, this was specifically designed for performers with severe motor impairments, and was tested with a patient with Locked-in Syndrome. The instrument allows for real-time generation of melodic lines using Attention (Brain) as the only channel. Fig. 5.7 depicts an user playing the instrument.

The first implementation employed a visual interface with four icons, and SSVEPs (sensed through EEG) were used to detect the selected icon and the intensity with which that icon was attended. Visual feedback was provided by changing the size of the icon in relation to the magnitude of the SSVEP signal. Various subsequent implementations were released [195].

The employed mapping is a distinguishing feature of this instrument. Within a given frequency band (associated to the flashing rate of one of the icons), the signal magnitude is quantized to five values, each corresponding to a note from a predefined 5-note pattern. In this way the SSVEP parameter is mapped to pitch (Discrete range), allowing the performer to generate melodies.

The authors report [81] on their trial with the Locked-in Syndrome patient was successful in terms of usability and engagement. In particular, response times between attending a target and the corresponding musical event were approximately 1–2 s.

### 5.1.8. EyeHarp [59, 62, 196]

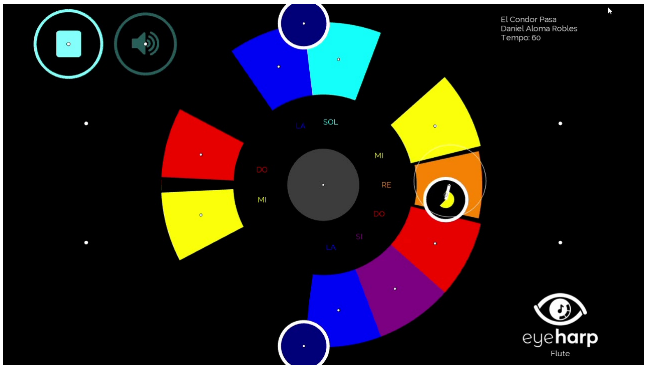


**Figure 5.8.:** *A recent iteration of EyeHarp's software interface.*
***Source:*** `https://www.closingthegap.com/eyeharp-play-music-without-barriers/`.

This instrument is based exclusively on Gaze pointing (Eyes), but realizes a complex interaction by means of a two-layer interface: one layer manages the performance of melodic lines, while the second one allows to compose short accompanying patterns that can be played in loop.

The "melody layer", depicted in Fig. 5.8, employs a pie-shaped visual interface, with slices associated to pitches (Discrete range), and with an inactive area in the center. This layout (together with the introduction of a dwell-time to recognize fixations) alleviates the Midas touch problem. The radial position inside a slice is mapped to intensity and vibrato (Continuous range): an example of a one-to-many mapping.

Visual feedback is minimalistic yet informative, and amounts to the appearing of one or more focus points in the selection area of a selected pitch. The instrument has been extensively evaluated from the perspective of both the performer and the audience [62].

A fork of the project led to a "Brain-Gaze controlled" musical interface augmented with an additional channel, namely Mental states (Brain), where emotional states control the "step sequencer layer". Valence is mapped to three different chord sequences (Discrete range), while arousal is mapped to the relative intensities of four predefined arpeggios (Continuous range). Additional visual feedback is provided in the form of varying colors (associated to valence) and brightness (associated to arousal).

### 5.1.9. Eye Play the Piano [197]

**Figure 5.9.:** *A boy performing with Eye Play the Piano in front of a grand piano, while the interface is projected.*
***Source:*** `https://yukakojima.com/healthcare.`

This DMI was developed as part of a collaboration between a manufacturer of commercial VR headsets and the University of Tsukuba. Despite the lack of accompanying publications, the main characteristics can be deduced from the project web site and audio-video materials.

The distinguishing feature is the use of an actuated acoustic piano, which can then be played without hands, as shown in Fig. 5.1.9. Gaze pointing and Blinking (Eyes) are employed as physical channels, by means of an eye tracker integrated into a HMD. This allows free head movements of the performer.

Two different mappings are implemented for gaze pointing. The "monotone" and "chord" modes map to piano notes pitches, and to a set of chords, respectively: in both cases, a Discrete range mapping strategy is adopted. Blinking is mapped to Note on events (Trigger strategy): a pointed pitch or chord is triggered by a blink, and a corresponding Note off event is produced when a subsequent blink triggers a different pitch or chord.

Visual feedback is provided through the stereoscopic HMD display. Colored keys are placed on a virtual surface and mapped to single notes or chords. Selection of a key is signalled by visual effects. The number and type of keys displayed on the surface can be customized.

No formal evaluation appears to have been conducted. However the instrument has been used in public performances by young performers with motor impairments. At the time of writing the project is supported through charity fundraising.

### 5.1.10. P-300 Harmonies [198]

**Figure 5.10.:** *P-300 Harmonies interface, showing the switches used to interact with the arpeggiator.*
***Source:*** *[198].*

This DMI focuses on real-time generation and modification of arpeggios. The employed physical channel is Attention (Brain), and P300 evoked potentials are detected through a low-cost, 14-channel commercial EEG device.

Arpeggios in the instrument consist of loops of 6 notes, played in random order to trigger the P300 response, with predefined timing and duration. A visual interface shows a $2 \times 6$ matrix, whose columns are “switches”, i.e. vertical pairs of notes which correspond to “up” and “down” switch positions. Each switch flashes when the corresponding note is played. The visual interfaces (Fig. 5.10) then serves both as extrinsic feedback and as a provider of stimuli for P300 responses.

By focusing on a specific switch, the performer can toggle between the up and down positions, thus modifying one note of the arpeggio. The selected note of the switch is highlighted with a different color on the visual interface. The P300 channel parameter is therefore mapped onto pitch, using a Discrete range strategy with a limited number of options.

Preliminary evaluation of the instrument was carried out with 4 subjects, in terms of accuracy in performing the toggle action on each of the 6 switches.

### 5.1.11. Imitone [199]

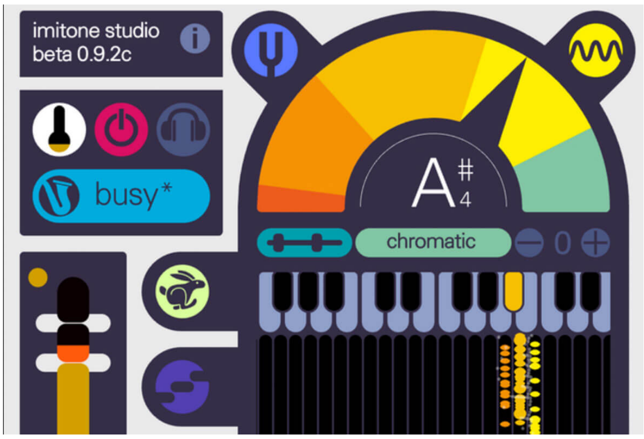


**Figure 5.11.:** *A section of Imitone's software interface.*
***Source:*** `https://www.adsrsounds.com/news/imitone-lets-you-play-instruments-with-yo`

This commercial DMI is a software app, which makes use of Voice (Mouth) as the only physical channel. Whistling (Mouth) is also mentioned as a possible alternative channel. The only required sensor is a microphone. Its interface is shown in Fig. 5.11.

The main advertised feature is the use of an efficient and accurate pitch tracking algorithm, which allows for low-latency conversion of voice parameters into MIDI events and consequently for voice-based real-time control of virtual instruments.

Several parameters are used for Note-level control, including Pitch, Voice activity, and Intensity, which are mapped to corresponding MIDI messages. Pitch in particular allows for either Discrete range (pitch quantization on various scales) or Continuous range strategies (glissando). In addition the instrument allows for control at the Timbral level, by detecting vibrato and glides.

A graphical interface provides extrinsic visual feedback in particular with regard to pitch, displayed through either a "piano roll" visualization or a continuous plot.

One advantage of using the Voice channel and intuitive mappings is that the instrument has minimal requirements in terms of expertise and training. However no formal evaluation studies are found.

### 5.1.12. Clarion [86]

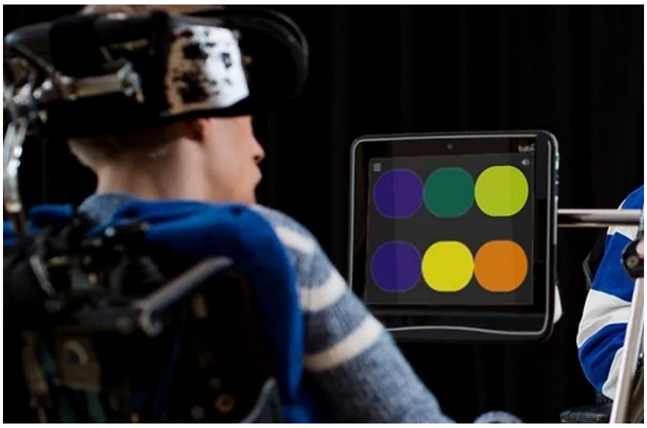


**Figure 5.12.:** *A musician with disabilities playing Clarion.*
***Source:*** `https://carlsonfelicity.wixsite.com/website/post/product-1`.

This is a DMI developed through a long-term charity program, with a strong emphasis on participatory design, adaptability to individual needs, and exploitation of off-the-shelf assistive technologies used by persons with disabilities in their everyday lives. Fig. 5.12 shows a possible setup of the instrument.

It can use various alternative physical channels, including Gaze pointing and Head movements (both detected via a commercial eye tracker), but can also be played with one's fingers or feet. The chosen channel is mapped to Note on/off and to Pitch, via a graphical interface that represents notes on a screen (Discrete range strategy). Additionally, timbral parameters can be controlled by the rate of movement of the physical channel (Excitation strategy) and by the position within the area representing a single note (Continuous range strategy): an example of a one-to-many mapping.

Notes can be arranged into several different layouts and assigned different shapes, sizes, and colours. This, together with the possibility of choosing among a palette of physical channels, allows for high adaptability. Visual effects in the graphical interface provide extrinsic feedback to the performer. Although no formal evaluation studies were retrieved, the instrument is actively used by a large number of performers through the Open Orchestras initiative.

### 5.1.13. EyeJam [186]

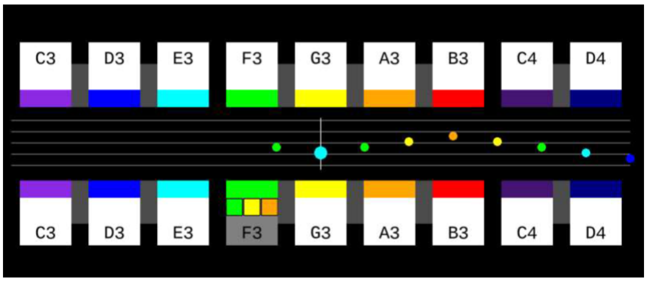


**Figure 5.13.:** *One of EyeJam's possible software interface configurations.*
***Source:*** *[186].*

This DMI allows for the generation of melodic lines using Gaze pointing (Eyes) as the only physical channel. It exploits an interesting "context switching" interaction paradigm, proposed by the same authors, which addresses the Midas touch problem by associating focus and selection to different eye movements.

Specifically, the visual interface is made of two identical keyboards placed at the top and bottom of the screen (the two "contexts") and separated by a narrow empty area (a "bridge"): in order to select a key, the gaze trajectory must cross the bridge and switch context through a saccadic movement, whereas fixations on different keys within the same context do not trigger any event.

The 2D gaze point position on the screen is mapped to pitch of the keyboard keys, using a Discrete range mapping strategy. A low-cost commercial eye tracker is used, whose limitations in terms of accuracy and calibration are dealt with by providing a limited number of large keys: 9, arranged along a major scale. Keys are color coded according to their position on the scale. The visual interface, shown in Fig. 5.13 can also provide additional feedback to help a user follow and learn a predefined melody.

The system was evaluated with 6 subjects without disabilities, who compared the proposed interface with one where selection is based on dwell-time (see Sec. 6.4). Experimental results suggest that the context switching paradigm allows for improved accuracy in rhythmic tasks.

### 5.1.14. Eye Conductor [200]

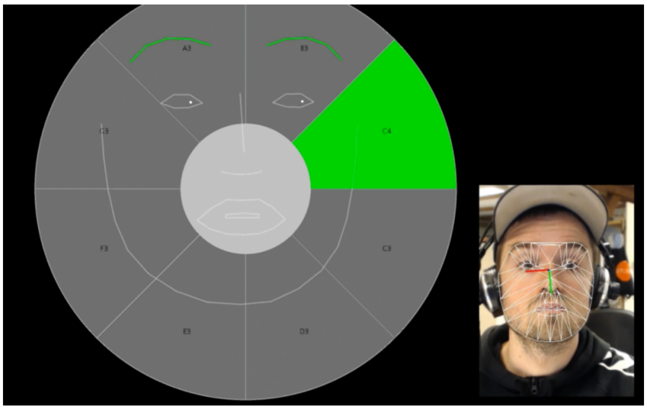


**Figure 5.14.:** *Eye Conductor's software interface, showing the motion capture operated on the musician's face.*
***Source:*** *`https://designawards.core77.com/Design-for-Social-Impact/46641/Eye-Conductor`.*

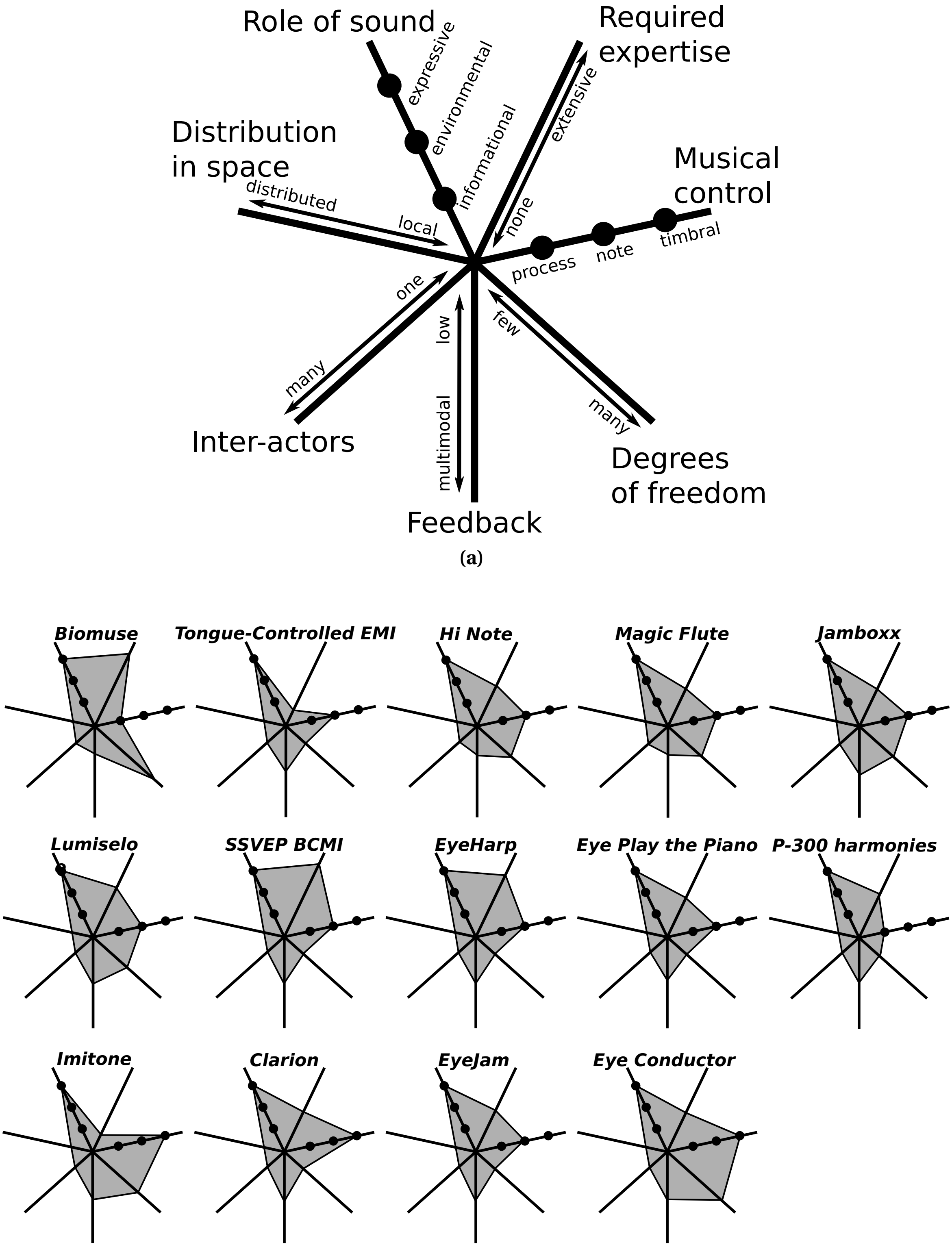
This DMI is notable for its use of several physical channels, including Gaze pointing (Eyes), Eyebrow movements (Eyes), and Mouth-lip movements (Mouth), employed for control at the process, note, and timbral level. Low-cost sensors (a commercial eye-tracker and a webcam) are used.

Gaze pointing is mapped to Pitch (Discrete range) and Note on/off, using a pie-shaped interface in which eight sectors represent an octave, and a central "silent area" corresponds to Note off events (this layout shares some similarities with EyeHarp). Although not specified in the documentation, selection appears to be based on dwell-time. Eyebrow movements are mapped to transposition using a Toggle strategy: raising/lowering eyebrows transposes all pitches by one octave up/down. Mouth aperture is mapped to timbral brightness (other possibilities are envisioned, such as controlling reverb or delay-based effects). Alternative layouts are also proposed, which include control over chords and drum sequencing.

The graphical interface, depicted in Fig. 5.14 provides extrinsic visual feedback to the performer. Circle sectors are colored upon selection of the corresponding notes, and a stylized animated silhouette of the performer's face is rendered in the background.

There are no accounts of formal evaluation. However the accompanying audio-video materials show that several preliminary user tests were conducted with people with various motor impairments, since the early design stages. Notably, the documentation mentions an adaptation step that allows for adjusting thresholds for facial gestures to fit abilities of different users.

## 5.2. Discussion



(a)

(b)

**Figure 5.15.:** *The 7-axis dimension space proposed by Birnbaum et al. [44] for the analysis of musical devices (a), used for dimension space analysis of the reviewed HeaDMIs (b).*

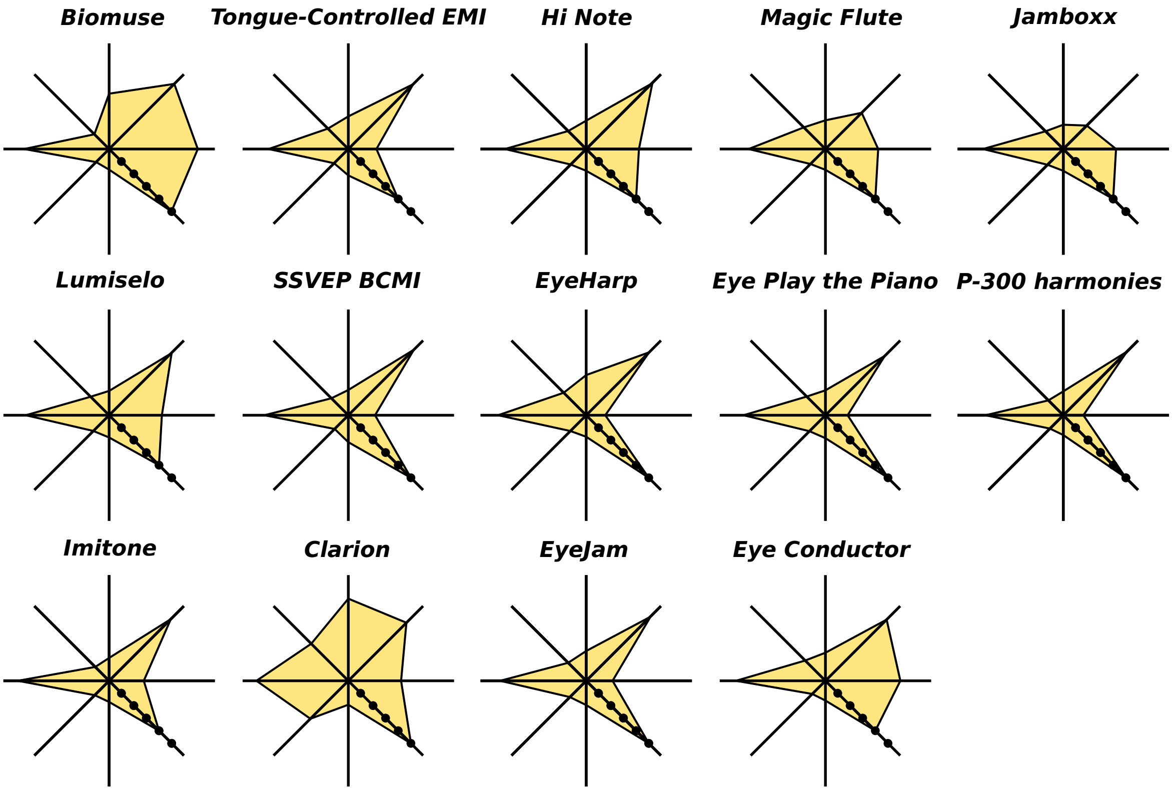


**Figure 5.16.:** *Dimension space analysis of the reviewed HeaDMIs through the design space for ADMIs described in Ch. 2 (refer to Fig. 2.1 for the meaning of the axes).*

Most of the HeaDMIs surveyed in the previous section were developed in recent years, suggesting a growing interest from both the research community and the industry. It is worth noting that a significant portion of the surveyed instruments are either commercial products or non-academic projects funded through charity programs: this suggests that there is a potential for exploitation and a large population of target users. At the same time, the limited number of surveyed instruments, as well as their design choices, shows that the field is still underexplored.

**DMI design space analysis.** In order to provide a structured and visually-oriented comparison of the instruments, we resort to the dimension space analysis approach proposed by Birnbaum *et al.* [44]. This HCI-driven approach has been used to evaluate the main design aspects of a broad range of "musical devices" (musical instruments, interactive installations, games, and so on) along the 7 axes depicted in Fig. 5.15a. Required Expertise represents the level of practice of the performer; Musical Control specifies the level of control exerted by the performer (see Sec. 1.4.1.2); Degrees of Freedom indicates the number of independent available channel parameters; Feedback Modalities indicates the degree to which extrinsic feedback is provided; Interactors represents the number of people involved in the musical interaction; Distribution in Space represents the total physical area in which the interaction takes place; Role of Sound represents the category of sound role.

Fig. 5.15b describes the surveyed instruments on the Birnbaum dimension space. Even though such representation involves qualitative and subjective evaluations, it highlights some relevant points. Values on the three left-hemiplane axes are shared by all the instruments ("expressive" role of sound, “local” distribution in space, “one” inter-actor). These axes are included here for the sake of compliance with the original formulation, which is meant to represent other types of musical devices in addition to skill-based instruments considered here. Instead, relevant differences and common trends may be observed along the four axes in the right hemiplane.

Most of the instruments have few degrees of freedom, corresponding to low numbers of employed physical channel parameters. This can be also appreciated from the data reported in Tab. 5.1, according to which the explored channels are Gaze pointing (7 instruments), Breath (5), Head movements (4), Mental states (2), Attention (2), Voice (1), Tongue (1), Eye movements (1), Mouth and lip movements (1), EyeBrows (1), Whistling (1).

The musical control exerted by the performer is generally confined at the note level or even at the process level (generation of chords, arpeggios, etc.). This correlates with the generally low number of degrees of freedom. The only exceptions in this respect are Imitone, Clarion, and Eye Conductor, in which control at the timbral level is achieved to some extent, also by means of a larger number of degrees of freedom.

The variety of mappings is also limited. As an example, all the instruments using gaze pointing or head movements map parameters of these channels into pitch, typically through a Discrete range strategy. Similarly, breath (pressure) is always mapped into intensity through a Continuous range strategy. Brain channels are always mapped to musical parameters at the process level (with the notable exception of SSVEP BCMI). Also, all the mappings are of the one-to-one type, in which a single channel parameter influences a single musical parameter, whereas more complex cardinalities are rarely explored. Notable exceptions are Eyeharp and Clarion, which both provide examples of one-to-many mappings.

All the instruments score low values along the Feedback axis. Apart from intrinsic and auditory feedback, additional extrinsic feedback is absent or very limited (most typically, 2D visuals on a screen). Moderately higher levels of feedback are provided by Lumiselo and Eye Play The Piano (3D visual feedback on helmet or Head-Mounted Display), while Jamboxx (tactile bumps) and Tongue-Controlled EMI include forms of tactile and proprioceptive feedback.

Several instrumental designs are lacking extensive and structured evaluation based on frameworks commonly used for DMIs (e.g. those described in Sec. 1.5). In the absence of structured evaluation, the levels of required expertise plotted in Fig. 5.15b are estimated qualitatively based on our subjective judgement, and vary considerably depending on the employed parameters, mappings, and interfaces.

**ADMI design space analysis.** Ch. 2 provided a dimension space suitable for the analysis of ADMIs, complementary to the one proposed by Birnbaum *et al.*. This provides different axes which address characteristics of the instruments related to the accessibility context. Fig. 5.16 provide an evaluation of the same instruments through this tool. Although values present similarities given the rather restrictive design specifications and target, some different characteristics provide the ground for further discussion.

From the *simplification* point of view, a low score is reported for all the instruments, since no particular aids are introduced for the musician. Clarion represents a slight difference as the highly customizable interface allows to draw simplified layouts depending on the required type of musical performance.

The *Adaptability* of the instruments is also generally low. The provided systems do not allow an high degree of customization to include musicians with different type of disabilities, modeling themselves on users needs. An exception is again represented by Clarion, which offers the possibility to change input methods and layouts. Biomuse would also seem to offer a minimum degree of adaptability, as the different sensors are repositionable, and the mappings seem to be arbitrary.

Since HeaDMIs require particular design solutions to exploit the residual physical interaction channels of their target users, most of the reviewed instruments present an high *design novelty*. A lower value is reported for Magic Flute and Jamboxx since their design is more inspired to traditional instruments (especially the latter, which resembles an harmonica).

The number of employed *physical channels* differs greatly from instrument to instrument, ranging from lower values (represented for example by gaze-only based instruments like EyeHarp, Eye Play The Piano and EyeJam) to higher values (e.g. Eye Conductor, which exploits gaze and a number of facial expressions and muscles). This confirms the wide range of control possibilities offered by physical channels available from the neck up.

Addressed *motor impairment*s range between two values: quadriplegic disability and lock-in syndrome. While most of the instruments are only able to address the first category, exceptions are represented by instruments which support gaze-only based interaction (EyeHarp, Eye Play the Piano, Clarion and EyeJam) and brain controlled instruments (Biomuse, SSVEP BCMI, P-300 Harmonies).

No instruments actively addresses *cognitive* nor *sensory* impairments. Clarion's score on cognitive impairments have been reported as slightly higher since the degree of simplification offered could be potentially useful to address this kind of disabilities.

By definition, the *use context* of all the listed HeaDMIs is oriented to musical performance rather than music therapy settings.

## 5.3. Challenges and Perspectives

The survey and the structured comparison reported in the previous section provide the ground for discussing current limitations in the design and development of HeaDMIs, presenting related open challenges, and proposing future research directions.

### 5.3.1. Channels, mappings, feedback

Only a small subset of potentially available physical channels is used in the surveyed instruments. Some are rarely considered, and some are completely ignored (Blinking, Teeth, Neck Tension, Imagery). A more comprehensive exploration of alternative channels is an endeavour for future work. In addition, physical channels and related parameters need to be characterized in terms of properties providing useful indicators for musical interactions. We proposed a set of such properties (Table 3.1, 9 rightmost columns), which may be reconsidered or expanded. However this issue remains largely untouched in DMI-related research, with few exceptions reported in Ch. 3): whereas the importance of characterizing sensors for musical applications is well recognized [47], the same cannot be said for intrinsic characteristics of physical channels, especially unconventional ones discussed in this work.

Concerning control and mappings, our survey shows that musical control rarely extends to the timbral level, which limits the expressive possibilities of the instruments to a great extent. The predominance of one-to-one mappings is also a major limitation for instrumental expressivity. As already discussed, mappings with higher cardinalities (many-to-one, one-to-many, many-to-many) are typical of most acoustic instruments and have the potential to be more rewarding and to provide more expressive control. Finally, the generally limited (or absent) extrinsic feedback impacts negatively especially on the transparency of the interaction, for both the performer and the audience. All these aspects should be considered together in the design of future HeaDMIs.

### 5.3.2. Adaptation and Intelligence

Most of the reviewed instruments allow for a limited degree of adaptation to different needs of various groups of users. Some include the possibility of customizing parts of the interface and musical features (e.g. range). For an instrument designed for users with different types of motor impairments, however, a key asset would be the possibility of adapting the employed channel parameters and the mappings: an example in this direction is provided by Clarion, which allows the use of gaze pointing or touch, for musicians.

A recent trend amounts to using machine learning techniques in order for an instrument to learn preferred or idiosyncratic gestures of a user, and to map these to musical parameters. A notable example is the Wekinator software [77], in which various supervised machine learning approaches are used to build musical mappings through training examples. Interestingly, this software has been used in a recent project aimed at building customized musical rehabilitation devices for children with severe motor impairments [78].

Further insights can be found in the related emerging field of Smart Musical Instruments (SMIs), which can be defined as instruments equipped with embedded intelligence and able to communicate with external devices. Specifically, the five abilities of a SMI identified by Turchet [202] comprise in particular:

*a*) context awareness, including models of the performer (needs, goals, state, etc.),
*b*) reasoning, including sensor fusion approaches to define control mappings,
*c*) learning, including learning from the way a performer interacts with the SMI, and
*d*) adaptation and proactivity, e.g. exploiting knowledge about the performer to adapt its function or behavior.

Related design principles include in particular personalization and embedded intelligence, which however maintains the performers sense of control.

It is argued that the main current obstacle to the creation of SMIs is the lack of hardware and software tools able to guarantee low-latency in conjunction with all activities related to knowledge management, reasoning, and learning (feature extraction from audio and sensors, other forms of sensor signal processing, sensor fusion and machine learning, etc.). This latter remark may be extended to the fields of ADMIs and HeaDMIs, and may explain why the issue of adaptation is largely disregarded in almost all of the instruments reviewed in this work.

### 5.3.3. Design and evaluation

We have reasoned in Sec. 1.5 about the multifaceted nature of evaluation in the context of DMIs. This is counterpointed by the lack of structured evaluation for most of the surveyed instruments. The development of evaluation frameworks specifically devised for ADMIs and HeaDMIs is certainly a challenge for future research. One further open issue is a general lack of musical pedagogies and repertoire for these instruments, which not only hinders their adoption and longevity [203], but also limits the possibility of conducting longitudinal studies targeted at long-term evaluation (learning, retention, and so on).

A related point is about the approach to instrument design. All stakeholders should be involved in the design process since the early stages, using a cyclical, participatory design approach [204] in which mock-ups and early prototypes are evaluated and redesigned based on stakeholders' feedback. This is even more needed in the case of ADMIs and HeaDMIs, where target users have specific and individual needs and requirements.

Principles of participatory design have only recently started to enter the mainstream of DMI research [79]. Although some of the surveyed instruments mention the involvement of one or more subjects in the design cycle, developing participatory design approaches specific to HeaDMIs is yet another challenge for future research.

# 6

# Design of gaze controlled instruments

The use of gaze as an interaction method for musical interfaces, reviewed in Sec. 3.2.1 and with the experiment described in Ch. 4, can be a particularly simple and suitable solution for HeaDMIs. This is also demonstrated by the amount of gaze-based HeaDMIs found in the literature. Acoustic musical instruments interfaces are usually designed to exploit the peculiarities of hands and finger movements. Similarly, a gaze-based musical interface should consider the characteristics of eye movements to guarantee comfortable and effective interaction. In fact, a simple imitation of the layout of a traditional musical instrument may be unsuitable for gaze interaction, thus requiring the implementation of specific solutions.

This section serves as a theoretical background for some of the development choices described in Part II of this thesis, by providing more insight on gaze interfaces design, as well as a collection of related design cues. Many of these have been considered for the development of the instruments Netytar (Ch. 9) and Netychords (Ch. 10).

Sec. 6.1 provides an insight on the state of the art. Sec. 6.2 describes how the eyes move from a physiological point of view. Sec. 6.3 lists a series of design cues and techniques which could be used to enhance interaction. Finally, Sec. 6.4 addresses the Midas Touch problem, a known issue in gaze-based interfaces design, and some possible solutions to tackle it.

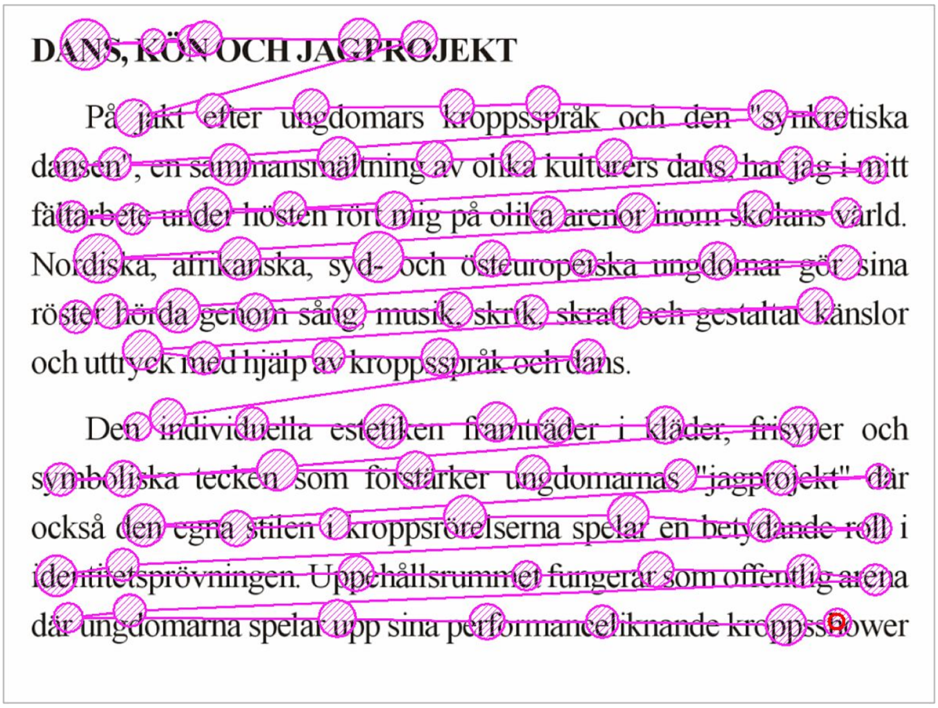
## 6.1. Related works

Relatively few gaze-based methods for playing music have been developed to date. Interesting analyses of strengths and weaknesses of these approaches, as well as limits and challenges that future solutions should address, can be found in the works by Hornof et al. [98, 205]. The EyeMusic system and related performances [206] are a first attempt at creating tools for generating sounds with the eyes, although they cannot be strictly considered real musical instruments. Vamvakousis' Ph. D. thesis [207] provides an important source for gaze-based instruments research. Eye Play The Piano (Sec. 5.1.9) allows to select notes and chords by looking at hexagonal graphical shapes that control the keys of a real piano. EyeJam (Sec. 5.1.13) proposes a method for note selection called “context-switch”, where sound is produced only when the gaze crosses a horizontal line. Lumiselo (Sec. 5.1.6), is probably the first to propose a hybrid method involving both gaze and breath (through a sip-and-puff controller): a note is selected by gaze, and then its actual playing occurs by blowing into a breath detecting sensor. EyeConductor (Sec. 5.1.14), and The EyeHarp (Sec. 5.1.8) introduce pie-shaped interfaces in which the central area is mapped to silence (pauses). EyeConductor also exploits facial expressions, such as raising eyebrows or opening the mouth to change octaves or to control filters. The EyeHarp is a complete musical instrument that allows to play notes on several octaves and to control sound dynamics. Its interface contain gaze sensitive buttons with white dots used to guide the gaze, and the central area is exploited for both pauses and note repetition, through a dynamically mapped button.

## 6.2. Characteristics of gaze movements

*Gaze point* is the point in space (or, in software applications, the point on the screen) where the user looks at.

Eyes generally move through *saccades*, which are jerky movements, lasting about 30 ms, during which the gaze point moves from one discrete point to another. These are interspersed with *fixations*, where the gaze point remains, indeed, almost fixed on a position. Usually a fixation lasts from 100 ms to 400 ms. Fig. 6.1 shows a visual representation of gaze point moving through saccades and fixations while reading a text.



DANS, KÖN OCH JAGPROJEKT

På jakt efter ungdomars kroppsspråk och den "synkretiska dansen", en sammansmältning av olika kulturers dans, har jag i mitt fältarbete under hösten rört mig på olika arenor inom skolans värld. Nordiska, afrikanska, syd- och östeuropeiska ungdomar gör sina röster hörda genom sång, musik, skrik, skratt och gestaltar känslor och uttryck med hjälp av kroppsspråk och dans.

Den individuella estetiken framträder i kläder, frisyrer och symboliska tecken som förstärker ungdomarnas "jagprojekt" där också den egna stilen i kroppsrörelserna spelar en betydande roll i identitetsprövningen. Uppehållsrummet fungerar som offentlig arena där ungdomarna spelar upp sina performanceliknande kroppsshower

**Figure 6.1.:** *A scanpath, namely a visual representation of saccades (straight lines) interspersed with fixations (circles) while reading a text*
***Source:*** *Lucs-kho at English Wikipedia, Public domain, via Wikimedia Commons*

That said, the eye is unable to perform fluid movements unless it has a target to lock on: this is called *smooth pursuit*, a fluid movement which follows the movement of a target.

*Blinks* are sometimes not recommended as an interaction channel due to their potentially involuntary nature [208], but are listed as one of the residual movement abilities for a quadriplegic person (Sec. 3.2.1). As they are very fast, blinks employed in some accessible applications and HeaDMIs like Netytar (Ch. 9), using some filter or rule to discriminate voluntary and involuntary blinks.

Finally, even during a fixation eyes are not perfectly still but make small random movements within 0.1° of the visual angle, called *jitter*.

These movements can be activated voluntarily, but many can occur involuntarily and unconsciously. Involuntary saccades, for example, occur on a regular basis even during fixations [185]. Those may preclude musical performance, which requires very precise control. There is evidence for gaze anticipating physical movement [209] and interactions in virtual environments [210], a behavior which the performer must learn to avoid during gaze-controlled musical performance. In gaze controlled instruments, those may lead to the anticipated performance of notes with respect to the prescribed tempo, unless the introduction of filters to compensate by creating latency. Such behavior was also noticed during the evaluation of The EyeHarp [62, Sec. 2.2.2, 'Melody layer evaluation']. Netytar and Netychords as a counterexample does not use filters in order to improve the precision at higher tempos [7], thus not providing any aid to avoid anticipations.

Rhythmic capabilities of eye movements are limited. [98] reports an eye-tapping experiment which shows that eyes are unable to deliberately perform more than 4 saccades per second (approximately one saccade every 250 ms). According to the author, this seems to be an upper limit which cannot be overcome, not even through training. In systems like Netytar where notes are selected through gaze pointing, this translates into a maximum limit in note changing speed. Experimenting with Netytar we however make the hypothesis, through direct observations, that more trained people could manage to maintain sustained tempos with greater precision. In Sec. 9.4 we discussed and proposed a training method to possibly reach this goal through exercising.

## 6.3. Visual cues and techniques

The following are some techniques which can be considered and combined.

**Color.** When using a gaze-based musical interface, an eye movement can result in a involuntary interaction. This leaves little space for the user to explore the interface, and usually the performer needs to know in advance where the next gaze movement should happen. While many musical interfaces employ differently shaped keys (e.g. a normal piano keyboard) or spatialization (e.g. The EyeHarp, Sec. 5.1.8) to help note localization, color can be used strategically to enhance interaction and partially solve this problem. It has been proven that the areas of sight outside the fovea (the central area of human vision), corresponding to peripheral vision, are particularly sensitive to contrasting color variations [211].

**Cursors.** Although showing a cursor is a classical way to give a visual feedback to the user for the current pointing position, its usage in gaze-based interfaces could be problematic since it could distract the user. It has been shown, through experiments on primates, that involuntary gaze movements can be caused by moving objects [212]. Furthermore, given the general imperfect accuracy and precision of eye tracker data, even a slightly different position of the cursor with respect to the fixation point could frustrate the user and be unnatural. It can be argued that the use of visual feedback may not be necessary to indicate the user's gaze position. When using a pointing device such as a mouse, a cursor is required as the pointed position would otherwise not be known to the user. In the case of gaze, the position is already known since the user who knows where their gaze point lies. There are however alternatives to cursors to return visual feedback on selecting items: one of these is to highlight the selected element through a change of color, a flash or a shape change when the gaze point enters its area. Alternatively, it is possible to implement "discrete cursors" like the one introduced in Netytar (Ch 9), which instead of moving in a continuum can only assume a limited number of positions (e.g. centered on gaze sensitive elements).

**Visual elements to enhance precision.** Given the aforementioned jittering nature of eye movements and gaze detection by eye trackers, some visual elements can be introduced to enhance interaction precision. Gaze sensitive elements can be equipped with "visual hooks", such as dots, to help the user concentrate fixations on the center of their area. The EyeHarp (Sec. 5.1.8), for example, presents a series of points on keys and external areas used for pausing sound. In the first versions of Netytar (Ch. 9), keys had a dot in the center. The solution was however abandoned in later versions to lighten and simplify the interface, reducing the number of elements displayed. Abandoned for the same reason, the first versions of Netytar also featured several lines on the interface which connected the keys, acting as potential guides for gaze movement.

**Auto-scrolling.** Netytar (Ch. 9) introduces an auto-scrolling feature and approach. The view on the virtual keyboard moves automatically, and “gently” places the key corresponding to the just played note at the center of the visualization area. The speed at which the interface moves is proportional to the square of the distance between the observed point and the center of visualization area on screen. This allows to have a theoretically infinite playing region available, regardless of screen size. This solution also could increase accuracy, as gaze detection provided by eye trackers based on Near Infrared technology is usually more accurate in the central screen area [213]. This solution is particularly suitable if applied to 2D key layouts similar to those used by Netytar or Netychords (Ch. 10), but could potentially be introduced in others.

## 6.4. The Midas touch problem

One of the best known issues to be addressed in the creation of gaze based interfaces, present also while designing other types of interfaces based for example on gestural controls, is the so-called "Midas Touch" problem [92, 214, 215]. It consists in the fact that, if the act of passing gaze point through the area of on an interface element triggers any event such as its activation (as often happens in gaze-based musical interfaces), any exploratory or involuntary gaze movement can potentially cause an unwanted interaction. Jacob [208] summarizes the problem with the following sentence:

*"Everywhere you look, another command is activated; you cannot look anywhere without issuing a command."*

One very important consequence in musical interfaces is that keys layout design is a non-trivial problem which requires an additional effort. Traditional acoustic musical instrument layouts may not be suitable for gaze-based interaction. Let us take as an example a piano keyboard. In order to perform any given musical interval which requires a jump between two non-adjacent keys, other keys should be crossed (Fig. 6.2). Even if a saccadic movement is very fast, the sampling frequency of modern eye trackers is high enough to detect intermediate positions, causing an involuntary activation of intermediate keys. While fingers can be lifted from a keyboard, it is not possible to control the musical performance in the same way with gaze.

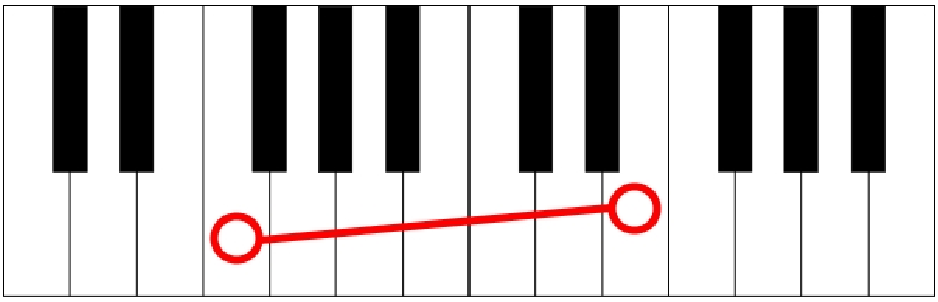


**Figure 6.2.:** *Gaze scanpath on a piano keyboard. When gaze moves from the F key to the E key, intermediate keys are crossed.*

Various solutions have been proposed to this problem in the literature, for both musical and general purpose gaze-based interfaces. The following is a review of the main ones.

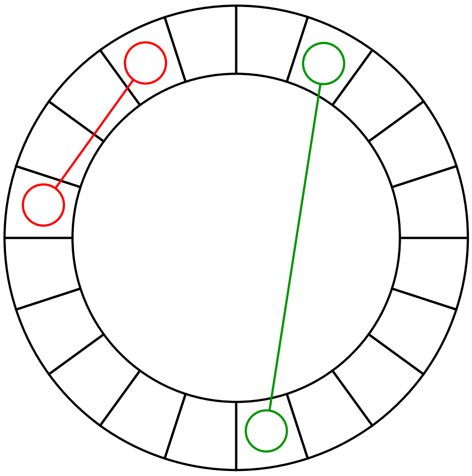
**Dwell time.** A possible solution to this problem is to apply a delayed selection method. Using dwell time, an interface element is selected by gaze entering its area, but activated after the expiration of a given time interval [216, 217]. In musical terms, however, this might not be a very efficient solution since it introduces a Delayed Audio Feedback (DAF) between the action of the physical input and the generation of the related sound. A DAF may alter the quality of a musical performance, impeding correct play of rhythmic pieces [218]. According to Wessel and Wright [80], ten milliseconds are an acceptable upper bound for a delay on audible system reactions during live computer music performances. This is the reason why, to allow fast musical executions and “virtuosities”, in the gaze based instruments Netytar (described in Ch. 9) and Netychords (described in Ch. 10) we tried to avoid the use of dwell time.

**Filtering.** Another solution is applying a filter to discriminate saccadic movements from fixations, and enable activations only when a fixation occurs. An implementation is provided by The EyeHarp (Sec. 5.1.8). However, even in The EyeHarp a DAF was observed which could preclude the performance of rapid sequences of notes [7].

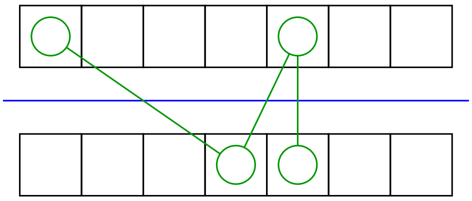
**Hybrid interaction.** Using an extra physical channel in addition to gaze allows to decouple the note selection from its performance. As an example, Lumiselo (Sec. 5.1.6) and Netytar (Ch. 9) exploit breath to control note onsets and sound intensity: when no breath is emitted, gazed keys are not activated.

**Keys displacement.** Passing through intermediate keys during the performance of different musical intervals can be avoided, in part or completely, through an adequate keys positioning. Various solutions have been proposed in literature, all having pros and cons, being partially capable of solving the problem.

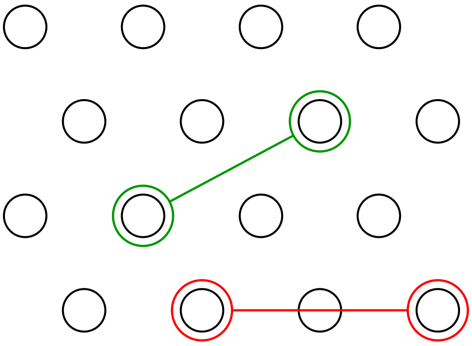
- **Pie shaped layouts**. A layout where the keys are arranged in a circular fashion, as illustrated in Fig. 6.3a, can partially solve the problem. As shown in the figure, ideally many musical intervals do not require crossing keys in between. However, with finite-sized keys, some intervals may still require passing through intermediate keys. One drawback of using such a solution is that the space is not fully exploited and, since the eye tracker detection limitations require the use of large keys, it is not possible to represent more than a given number of notes on the screen. It is also not easy to implement solutions such as auto-scrolling (Sec. 6.3) for this type of layout. The EyeHarp (Sec. 5.1.8) and Eye Conductor (Sec. 5.1.14) are two examples of instruments which make use of pie-shaped layouts.
- **Context switching**. A particular and original solution, called "context switching", has been proposed by the instrument EyeJam (Sec. 5.1.13). The solution, summarized by Fig. 6.3b, consists in placing two rows (or columns) of keys mapped to the same notes. The two rows should be separated by an area where no mapping is done from gaze movement to sound. Any activation should be preceded by the act of crossing this area. Any gaze movement which does not follow this rule is substantially ignored. In this way, an up-and-down motion of the gaze is required, but intermediate keys crossing is avoided.
- **Spacing between keys**. Another solution to tackle this problem is using a 2D layout where keys are interspersed with non-sensitive areas and placed in a strategic way to avoid intermediate keys crossing for common musical intervals. This is obtained by reducing the size of keys and/or setting the gaze sensitive area associated with each key (often called "occluder") to have a different dimension than the key itself, as for example happens in Netytar (Ch. 9). Fig. 6.3c summarizes the key concepts behind its layout. While some key jumps require to cross intermediate keys, the layout strives to provide alternate paths for the same pitch interval.



**(a)** *Pie-shaped layout.*



**(b)** *Context switch based layout.*



**(c)** *Layout with spacing between keys.*

**Figure 6.3.:** *Three possible keys layouts to tackle the Midas Touch problem. In a pie-shaped layout (a), the red colored scanpath crosses intermediate keys, while the green colored one does not. In a context switch based layout (b), the green colored scanpath does not cross intermediate keys, since it passes through the blue rows separator in the middle. In a layout with spacing between keys, like Netytar's layout (c), the red scanpath cross intermediate keys, while the green does not.*

# Part II.

# Development

# 7

# NeeqDMIs: a C-sharp HeaDMIs development framework

HeaDMIs all share a common structure in a general sense. Some components, described in Sec. 1.4 and by the structural diagram in Fig. 1.2 are possibly common to any HeaDMI. All of them need sensors to detect the user's interaction channels, have a mapping logic to transform interaction into musical features through mapping and have to communicate those features to a *sound production unit* through some protocol. Some of them provide extrinsic feedback through different modalities, some possess a *performer interface*.

*NeeqDMIs* is a library written to facilitate and speed up the development of software HeaDMIs by implementing some basic common elements among instruments. It contains a series of classes which can be used for fast HeaDMI prototyping and development, and can be easily integrated into the code. The library is written in C♯ and is therefore useful for developing software instruments with the same language, within the .NET Framework[1] by Microsoft. NeeqDMIs is released under the GNU-GPLv3 open-source license and its source code and release files (in the form of a Windows *.dll* library) are downloadable from its GitHub repository[2].

The library has not yet been presented in a scientific paper at the time of writing this thesis. However, it was used as a basis for the development of all the HeaDMIs and peripherals described in this chapter.

NeeqDMIs is structured in macro-groups of classes and tools. Although the library is expanded with new classes at every new update, a general structure can already be defined. The library was developed using an Object Oriented Programming (OOP) approach, and encourages the use of the same approach for HeaDMI development.

An overview of similar works found in the literature is provided in Sec. 7.1. Sec. 7.2 outlines some concepts and design metaphors, while in Sec. 7.3 a general description of the library's features is provided. The structure of this last section matches the library directory tree structure and the namespaces hierarchy. An example of its usage to implement a simple HeaDMI is provided in

---

[1] .NET Framework official website: `https://dotnet.microsoft.com/`

[2] GitHub repository for NeeqDMI: `https://github.com/Neeqstock/NeeqDMIs`

Sec. 7.4. Future work and directions are discussed in Sec. 7.5.

## 7.1. Similar works

Although an ADMI designer can make use of libraries dedicated to the development of DMIs in a general sense, no other prototyping libraries for software ADMIs have been found in the literature. Libraries for the design of DMIs dedicated to people without disabilities may not be adequate to design instruments for people with quadriplegic disabilities. That is the case for example if the supported interaction methods and controls require the use of hands. However, some of the recovered libraries and toolkits share some aspects of NeeqDMIs' philosophy. Some examples are given below.

The library is aimed to support rapid prototyping. Making prototypes and sketches is a felt necessity within the design of DMIs, as noted by Calegario et al. [219]: *"prototypes help identify flaws, redirect and adjust decisions, improve the understanding of context, and generate new ideas".* As they furtherly remark, in order to evaluate an instrument design idea it is necessary to have the possibility to play it. The authors proposal to address the problem is the *Probatio* toolkit. This consists of a series of physical "blocks", which are housings or modules containing sensors (for example keys, levers, cranks, touch sensitive strips). An instrument can thus be assembled following a "morphological chart" to choose the building blocks.

The *OSC-XR* software library, proposed by Johnson et al. [220] offers a similar concept in the context of Virtual and Extended Reality instruments. It is a library for Unity[3], a well-known 3D graphics engine. It offers a series of controllers and scripts for rapid instrument prototyping. The work of the instrument designer is simplified, since they can thus implement OSC-XR elements without having to build from scratch three-dimensional graphic components. The authors propose *Hyperemin* as an example, an augmented theremin with extended reality elements which offer additional control over audio processing.

Roberts et al. [221, 222] proposed *Gibber*, a toolkit comprising a highly simplified programming language for prototyping and creating web instruments which work via browser. Gibber provides a ready-made interface elements collection, along with a series of event handlers for interacting via touch, mouse, and keyboard. Gibber is also oriented towards sharing ideas and sketches, as it allows the publication of instruments within a centralized database.

Matthew's "The instrument-maker" [223] is a library for the prototypation of Assistive Music Technologies for the Pure Data programming language. Instruments made with that library can be run on Bela, Arduino and Raspberry boards, enabling affordable embedded and standalone technologies development.

Lastly, following the idea of simplification, Gonçalves and Schiavoni [224] proposed Mosaicode, a Visual Programming Language (VPL) with a high level of abstraction and high specificity, being a Domain-Specific Language (DSL) dedicated to sound design and DMIs development. According to the authors, the use of a visual language where programming is done by diagrams could be a

---

[3] Unity's official website: `https://unity.com/`

simpler approach for instruments designers without previous programming experiences.

## 7.2. Framework concepts and paradigms

The library proposes a general design paradigm for the implementation of the software HeaDMIs developed with it. Specifically, NeeqDMIs introduces a physical metaphor for implementing musical instruments, given by the *Rack* and *DMIBox* classes.

The static class *Rack* class will contain elements and modules that can be referenced globally within the project. This class symbolizes one of the typical "racks" used as a shelf for analog modules like synthesizers, amplifiers, an other electrical audio equipment (an example is shown in Fig. 7.1). The main container-module will be the *DMIBox*, a class to be extended and integrated into the HeaDMI project. An extension (through inheritance) of the DMIBox will contain the interaction logic of the instrument. Submodules can be added to both the Rack or the DMIBox. DMIBox class and Rack examples are contained in the *Templates* namespace.

**Figure 7.1.:** *An audio rack with caster wheels*
***Source:*** *Stephan N, CC BY-SA 3.0, via Wikimedia Commons*

Most of the sensor modules described in Sec. 7.3 include one or more *Lists* of *Behaviors* specific to that sensor in the instrument's operating logic. Behaviors are actually implementable C♯ interfaces. The programmer is expected to implement the specific Behaviors for each sensor by extending these interfaces. Once the Behaviors have been added to the corresponding lists in the related sensor modules, these will be triggered (called) each time a new data sample is communicated by the sensor.

## 7.3. Library contents

NeeqDMIs contains classes and tools to facilitate the introduction of sensor peripherals to detect the user's physical interaction channels.

Some of the sensors which the library is dedicated are already available on the mass market. In this case, NeeqDMIs aims to extend their official API in order to facilitate their use in the ADMIs development context. Some additional sensors introduced in the framework are part of the DIY sensors inventory described in Ch. 8.

### 7.3.1. ATmega

This namespace contains tools for the implementation of generic sensors built using Arduino microcontrollers[4].

***SensorModule*** defines a class for writing output or receiving input strings from a sensor connected via USB port (also called *"COM"*). Each time an input string is received by the sensor, the data is reported to all the Behaviors contained in the homonymous list. *SensorModule* contains methods to check if the connection to the sensor was successful, and possibly disconnect the sensor and stop receiving input data. The *ISensorBehavior* interface allow to implement new Behaviors for *SensorModule*.

### 7.3.2. Eyetracking

This and the following sub-namespaces are dedicated to eye tracking sensors.

***Tobii*** namespace contains classes and tools related to Tobii eye trackers[5], which can enable the instrument to detect gaze point, eye position in three-dimensional space, blinks and head movement. These classes include references to the official *Tobii.Interaction* framework library, available as a NuGet[6] package. *TobiiModule* is the main class to implement in the instrument code. It contains an instance of *Tobii.Interaction.Host* capable of detecting and receiving input from a Tobii eye tracker connected via USB port. *TobiiModule* also contains the following sub-modules:

- An instance of *TobiiBlinkProcessor*, a sub-module capable of detecting user blinks. Blinks detection occurs by considering the number of samples within which an eye is not detected by the sensor. Blinks performed with one or both eyes are detected, and it is possible to associate different behaviors to these events. The detection thresholds are adjustable.
- An instance of *MouseEmulatorModule*, which is described further in this section.

*TobiiModule* contains three Behaviors lists: *GazePointBehaviors*, which receive as input gaze point data; *EyePositionBehaviors*, which receive as input the position of the user's eyes within the three-

---

[4] Arduino official website: `https://www.arduino.cc/`

[5] Tobii official website: `https://www.tobii.com/`

[6] NuGet is .NET framework official package manager. NuGetOfficial website: `https://www.nuget.org/`

dimensional space; *HeadPoseBehaviors*, which receive as input the user's head position and rotation (detectable by some eye trackers produced by Tobii such as the 4C[7]). Three interfaces, namely *ITobiiEyePositionBehavior, ITobiiGazePointBehavior, ITobiiHeadPoseBehavior* allow to implement the related Behaviors to be included in the lists.

***EyeTribe*** contains classes and tools related to the EyeTribe eye tracker, to integrate the sensor into a DMI. The peripheral, whose specifications have been tested and analyzed by Ooms et al. [183], Popelka et al. [225] and Funke et al. [226] is now out of production. Compared to recent Tobii eye trackers, the EyeTribe only allows gaze point detection. The namespace contains the main module *EyeTribeModule*, which works in a similar way to the aforementioned *TobiiModule*, except for having a single list of Behaviors which receive the user's gaze point as input. An instance of *MouseEmulatorModule*, included in the module, can also be set to receive as input gaze point as raw unfiltered data, or data automatically filtered by the sensor hardware. The *IEyeTribeGazePointBehavior* should be used for implementing gaze point related Behaviors.

***MouseEmulator*** contains the *MouseEmulatorModule* mentioned above. It allows to control mouse cursor through gaze point, using the raw data provided by the eye tracker or a filtered version through the *PointFilters* described in Sec. 7.3.10.

***Common*** contains utility classes referenced by the modules described above. *Eyes* is an enum which simply lists left and right eyes, while *EyeTrackerModels* provides a list of eye trackers supported by the library.

### 7.3.3. Headtracking

This and the following sub-namespaces contain classes and tools for implementing head tracking support into an instrument.

**NeeqHT** contains classes and tools for interfacing a DMI with *NeeqHT*, the DIY head tracker described in Sec. 8.2. Currently, *NeeqHT* is the only head tracker supported by the framework. *NeeqHTModule* is the main module to be placed in the instrument code. Among other objects and methods, it contains a list of Behaviors which are triggered each time a new input is received from the head tracker. The latest input is stored inside an *HeadTrackerData* object, which contains head rotation data (Yaw, Pitch, Roll). The object also contains methods to calibrate the head tracker by setting a central position for each rotation axis, corresponding to a neutral position of the head. *TranspYaw, TranspPitch* and *TranspRoll* properties will return the head rotation angle with reference to the calibrated center. The namespace also contains the *INeeqHTbehavior* interface, for Behaviors implementation.

---

[7] Tobii 4C specifications on Tobii official website: `https://help.tobii.com/hc/en-us/articles/213414285-Specifications-for-the-Tobii-Eye-Tracker-4C`

### 7.3.4. Keyboard

This namespace contains useful tools for implementing computer keyboard controls into the instrument. The implemented classes exploit and have a reference to the *RawInput*[8] package. This framework allows to perform advanced operations with keyboards such as discriminating input from multiple keyboards connected to the computer (e.g. via different USB ports), or detecting keystrokes even when the application window is placed in the background.

*KeyboardModule* is the main module to be placed inside the DMIBox of the instrument to implement keyboard controls. Whenever a keystroke is detected by the module, from any keyboard connected to the computer, the event is signalled to all the Behaviors contained in the homonymous list. The *IKeyboardBehavior* interface is useful for implementing such Behaviors. *VKeyCodes* enumerates keycodes associated to all the possible types of keys detectable and which can be processed by the module.

### 7.3.5. MIDI

This namespace contains modules and utilities to send MIDI messages. These are useful if the musical instrument is a MIDI controller. The modules refer to the MIDI output ports detected by the computer. Windows automatically provides a list of detected MIDI ports.

The *IMidiModule* interface allows to declare a new type of MIDI Module. This module will implement, in addition to a MIDI channel selector, a method for checking MIDI connection and methods for sending the following types of MIDI messages: Note On/Off, Expression, Modulation, Pitch bend (and reset), and Channel Pressure. Furthermore, the *SendMessage()* method should allow to send generic MIDI messages by manually setting each byte.

*MidiModuleNAudio* is a MIDI module (which implements the *IMidiModule* interface just described) created using the NAudio[9] library.

*MidiDeviceFinder* is a class which takes as input an *IMidiModule* and sets the MIDI device to the last available and active from the Windows MIDI devices list. This is due to the fact an external MIDI device (opposed to an internal one, such as the Windows pre-built MIDI synth) is usually set as last on the devices list.

### 7.3.6. Mouse

This namespace contains the *MouseFunctions* static class. This references Windows *DLL*[10] system libraries to allow the following functions: (a) detecting the position of the mouse cursor with reference to its coordinates on the screen; (b) setting mouse cursor position using the same coordinates system; (c) hiding the mouse cursor from the user. The latter is useful, for example, to implement applications which work by detecting the user's gaze point, as showing a cursor could

---

[8] RawInput library on NuGet: `https://www.nuget.org/packages/RawInput/`

[9] NAudio package on NuGet: `https://www.nuget.org/packages/NAudio/`

[10] Dynamic-link library (or DLL) is a Microsoft implementation of shared library concept in Windows systems.

be a source of distraction for the user's gaze (a design choice better argued in Sec. 6.3).

### 7.3.7. Music

This namespace contains tools and classes which refer to notes, scales, MIDI pitches and other musical concepts. Notes refer to the Equal Temperament[11]. Microtones and other temperaments are not supported by the library at the moment.

*MidiNotes* enumerates all musical notes and associates them to the relative MIDI pitches. Notes are listed ranging from octave number 0 to octave number 8. Accidentals are reduced to sharp notes only, indicated with an *'s'* as first character in their name. Extended enum methods allow to get the frequency associated to each note and to convert the note in other note formats included in the namespace.

*MidiNotesUtils* and *PitchUtils* are static classes which contain other useful methods to achieve such conversions.

*AbsNotes* enumerates musical notes without the octave information associated.

The *Scale* class describes a musical scale. Scale types are defined in *ScaleCodes*, while *ScalesFactory* (which implements the *Factory* OOP Design Pattern[12]) can generate Scale objects given a root note and a scale code. *Scale* also contains methods to determine if a given note belongs to the scale, and to determine if two notes are consequent in the given scale.

*TempoUtils* contains static methods to convert musical tempos between two measurement units: milliseconds and Beats Per Minute (BPM).

### 7.3.8. Mappers

This namespace contains tools and classes useful to make conversions between values, domains and more.

*AngleBaseChanger* allows to perform transformations between reference systems for angles having different bases. The angle reference systems must be expressed in degrees within the space $(-180°, +180°)$.

*ValueMapper* and *SegmentMapper* allow to make transpositions from one numeric domain to another through proportions.

---

[11] Equal Temperament is a known concept in music. It consists in subdividing the frequency spectrum into octaves consisting of 12 basic intervals or half-tones, and it's generally used in western music. It has been described by Dirk de Klerk in a jorunal article on *Acta Musicologica* [227].

[12] Factory design pattern according to the website "C♯ Corner": `https://www.c-sharpcorner.com/article/factory-method-design-pattern-in-c-sharp/`

### 7.3.9. Audio

This namespace contains classes and tools useful for implementing audio processing operations.

The only class currently included is *AudioFormatFft* which describes data useful for performing Fast Fourier Transform (FFT). An AudioFormatFft object receives the following information as input: sampling rate, bit rate, number of channels, size of the input buffer (expressed in milliseconds), and Zero Padding modality[13]. It is therefore able to: calculate the dimensions of the FFT output sample, considering or not the applied zero padding technique, and find the frequency associated with each specific *bin* of the FFT array, given the described format.

### 7.3.10. Filters

The following namespaces contain collections of numerical filters. They can be useful for performing tasks such as smoothing sensor inputs.

***PointFilters*** contains geometric filters for two-dimensional points. Filters are described by the *IPointFilter* interface. The following filters are included in the library in the current version while writing:

- *Moving Average Array*. Input points are stored in an array having finite size, and the output is given by their geometric mean;
- *Moving Average Array with Outsiders*. Similarly to the previous one, input points are stored in an array of finite size. However, a system of identification of outsider values is introduced which excludes from the calculation of the mean values that deviate from the previous average by a certain radius;
- *Exponentially Decaying Moving Average*. Filter output is described by the following formulas, which are a classic mathematical formulation [229] for obtaining a moving average.

$$O_x(t) = \alpha \cdot I_x(t) + (1-\alpha) \cdot O_x(t-1) \tag{7.1}$$

$$O_y(t) = \alpha \cdot I_y(t) + (1-\alpha) \cdot O_y(t-1) \tag{7.2}$$

where $t$ and $t-1$ represent the current and previous time instants, while $O_{x,y}$ and $I_{x,y}$ represent the X and Y coordinates of the output and input points, respectively.

***ValueFilters*** contains one-dimensional numerical filters, described by their respective interfaces. The proposed formulations are similar to *PointFilters*, but adapted to the one-dimensional case. ***ArrayFilters*** contains filters which take an array of values as input, and apply a *ValueFilter* to each entry.

---

[13] Zero Padding is a known technique in computational signal analysis theory. It consists in adding zeroes to the sample to be transformed, to obtain a larger number of bins in the transformed sample. It is well described by Donelle and Rust [228]

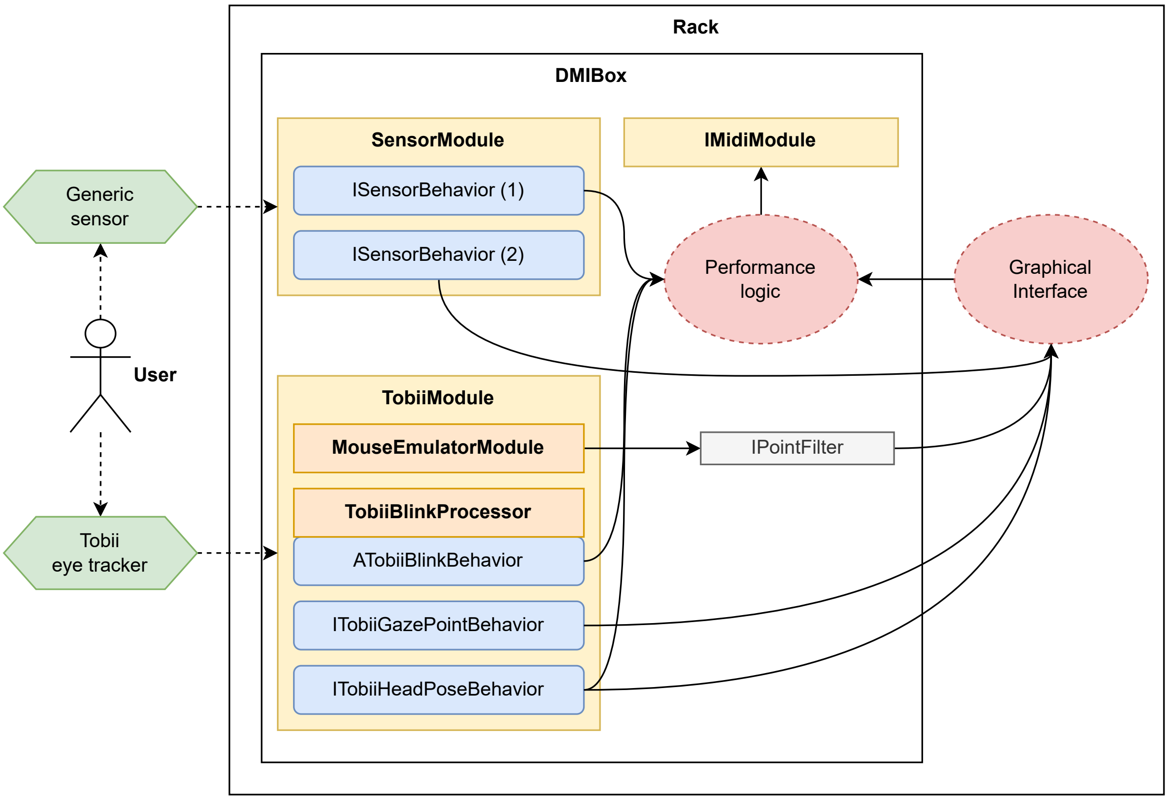


**Figure 7.2.:** *A diagram representing the main components of a sample HeaDMIs implemented with NeeqDMIs library. Hardware components (sensor peripherals) are colored in green; modules and submodules are colored respectively in yellow and orange; behaviors are colored in blue; generic code sections are colored in red. Arrows represent data flow between components.*

## 7.4. Example

The schematic of a simple software HeaDMI controlled through a Tobii eye tracker and a generic sensor is shown in Fig. 7.2. We chose to abstract on the performance logic of the instrument and its graphical interface, as well as the implementation of all behaviors, the MIDI module, and a filter in order to propose a generic example.

User action is detected by the two sensor peripherals, whose data streams are passed as input to *SensorModule* and *TobiiModule*. Two different behaviors associated to *SensorModule* are used respectively to control parameters of the performance logic and to generate feedback on the graphical user interface: the two code sections are voluntarily separated into two *ISensorBehavior*s. In this way, any of the behaviors can be added or removed (enabled or disabled) flexibly at runtime. The gaze point detected by *TobiiModule* is used both to control elements of the graphical interface through a *ITobiiGazePointBehavior*, and to control the mouse cursor through the *MouseEmulatorModule*. The latter's position is filtered through a *IPointFilter*, and also causes an effect on the graphical interface. Similarly, head position is used to affect both the graphical interface and performance logic through a *ITobiiHeadPoseBehavior*. Eye blinks are detected by *TobiiBlinkProcessor*. Through a *ATobiiBlinkBehavior* they contribute to the performance logic of the instrument. Eyes position in the 3D space is not exploited. The graphical interface elements in turn cause an effect on the performance logic. Finally, performance parameters are transformed into MIDI events by an *IMidiModule*.

Other library elements such as classes within the *Music* namespace (e.g., *MidiNotes* and *Scale*s definitions) could actively contribute to performance logic or elements displayed on the interface. Sensor values could be remapped through *ValueMapper*s or *SegmentMapper*s.

It should be noted that all components are placed inside the *DMIBox*, while the graphical interface (which could consist of an application window, for example) is referenced inside the *Rack* class and separated from the instrument implementation.

## 7.5. Future work

The library, at the time of writing, is constantly being updated by introducing new features. Possible additions that will be introduced in the future are discussed below.

At the current version, white this thesis is written, the library supports only some types of sensors, capable of detecting only a part of the proposed physical interaction channels in Ch. 3: only sensors to detect gaze, blink, eye position and head rotation are currently supported by the library. Tools to detect and analyze voice, whistle, vocal tract resonances through spectral analysis of microphone input could be introduced (for example through a generalization of those used to develop the instrument *Resin*, described in Ch. 11). Electroencephalographic and myographic sensors could be supported by future versions of the library, for the detection of brain and muscular activity.

Some controls and graphical elements for designing graphical user interfaces for virtual instruments (discussed in Sec. 1.2) could be introduced, with reference to C♯ graphical frameworks such as Windows Presentation Foundation[14].

The library currently supports only MIDI as an output protocol to communicate with sound generation units such as synthesizers. MIDI only officially supports the aforementioned Equal Temperament note to frequency mapping, which could be a major limitation for some types of musical instruments. OSC[15] or other communication protocols could be supported by the library in the future.

The library is written in C♯ and references Windows operating system libraries, which binds its usage to software instruments that can be run under such operative systems. This is due to the availability of some of the employed Application Programming Interfaces (APIs), such as *Tobii.Interaction*, for Windows operating systems only. A new library could be written in different languages without such bindings to work under different systems.

---

[14] Windows Presentation Foundation (WPF) is a User Interface framework for building Windows desktop applications, part of Microsoft .NET Framework. Repository on GitHub: `https://github.com/dotnet/wpf`

[15] OSC protocol on

# 8

# DIY sensor peripherals

This chapter describes two sensor peripherals developed ad-hoc to detect breath pressure and head rotation respectively. As stated in the preface of this thesis work, the philosophy of the research project includes economic affordability in addition to accessibility. Accessible and assistive technologies often have very high costs [230]. This, as well as lack of communication between developers and end users, may lead products to reach only a small part of the target group. High costs could depend on the small size of the user base, therefore on the absence of large-scale production. In addition to using free and open-source licenses for the distribution of the developed software HeaDMIs, the two peripherals described in this chapter aim at economic accessibility through the following principles:

- *Ease of construction.* The sensors can be replicated through DIY (Do-It-Yourself) practices, following simple and clear instructions which do not require specific electronics and manufacturing skills. In this way it is potentially possible to "distribute" them to the target users without the need for production;
- *Use of open-source microprocessors.* The construction plans of the employed Arduino microprocessor boards are available under open-source licenses[1]. This makes it possible for different manufacturers to replicate those boards thus keeping costs relatively low;
- *Use of easily available materials.* The materials used (e.g. pipes, boxes, cables) are easily found in DIY stores, while the transducers are easily obtainable through e-commerce;
- *Hardware reproducibility.* The peripheral construction projects and schemes are published under Creative Commons licenses;
- *Open-source control software.* The software to be loaded in the Arduino microcontrollers to allow the peripherals to function properly, as well as the libraries and APIs which serve as drivers are released under open-source licenses.

These principles are coupled with a general strive for design simplicity, both on the software and hardware sides.

It can be stated that the two peripherals agree with the concept of Open-Source Hardware (OSH). Niezen [231] defines OSH as *"hardware whose design is made publicly available so anyone can*

---

[1] Introduction to Arduino on Arduino website: `https://www.arduino.cc/en/guide/introduction`

*study, modify, distribute, make and sell the design or the hardware based on that design"*. Fisher and Gould [232] indicate OSH devices as products where *"designs, software and development efforts are made freely available to all"*. Pearce [233] assumes that OSH is the transposition to the hardware context of Free Open Source Software (FOSS) principles. As Bonvoisin et al. [234] point out, however, there's no agreement on the definition of OSH, and the term is interpreted in different ways and sometimes misused. While a license is often sufficient to determine whether a software product is open-source, the issue becomes more complex in the case of hardware products. The authors emphasize the need for comprehensive documentation, which allows for replication.

Bonvoisin et al. [234] report the definition proposed by the Open-Source Hardware Association [235] which indicates the basic principles which characterize OSH:

- Freedom to study the hardware project and schematics, having sufficient informations to interpret them;
- Freedom to modify the project, for example by "forking" the project (developing another independent branch);
- Freedom to make, produce, manufacture, replicate the components;
- Freedom to distribute, give or in certain cases sell the product documentation or manufactured product units.

Balka et al. [236] propose another wording for similar principles, namely "transparency, accessibility and replicability".

Open source documentation is often used to bolster community-based hardware development and collaboration processes [234], as well as fast innovation and cost reduction [231]. A review on how OSH could be used for the development of scientific hardware is proposed by Pearce [233], who states that an increasing number or peripherals is available nowadays, including 3D printers. OSH is also a source for medical instrumentation, which could be useful especially in developing countries [231].

As stated, the two peripherals share the use of an Arduino programmable microprocessor[2] for sensors and computer interfacing. Arduino is arguably one of the best known and most successful OSH development platforms [231]. It is easily expandable via add-on boards, and has great potential for the development of different types of instrumentation, even for scientific and research purposes [232].

It can be said that open-source software and hardware principles could also enhance scientific transparency, empowering for example the evaluation of musical instruments which exploit them.

In the next sections a description of the two peripherals is provided. Although they could be useful for different accessibility and human-computer interaction purposes, here they are used for musical interaction purposes in Netytar, Netychords and Resin, the three HeaDMIs developed as part of my research project and described in their dedicated chapters (Ch. 9, Ch. 10 and Ch. 11).

---

[2] Arduino official website: `https://www.arduino.cc/`

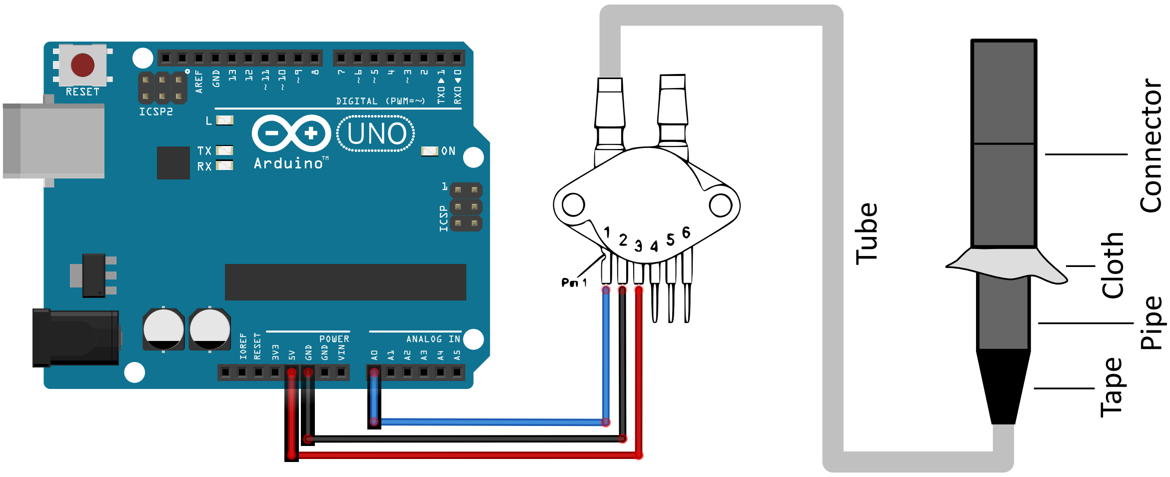


**Figure 8.1.:** *Implementation scheme of NeeqBS breath sensor peripheral. The image include an Arduino Uno board, the MXP5010DP sensor, the tube and the mouthpiece. The latter is built using a pipe, a pipe connector, a cloth filter and insulating tape.*
***Source:*** *schematics drawn using Fritzing software.*

## 8.1. NeeqBS

NeeqBS (abbreviation of "Neeq Breath Sensor") is a device for the detection of breath pressure. The device is depicted in Fig. 8.1.

The peripheral is built using the following materials:

- Arduino microcontroller (e.g. Uno or Nano);
- USB Type A to Type B cable (to connect Arduino to the PC);
- MPX 5010DP low pressure sensor (or an equivalent 10Kpa low pressure sensor);
- At least 3 male to female (or female to female if using Ardino Nano) jumper wires for microelectronics, to connect Arduino to the sensor;
- 70-80cm long, 5mm wide rubber tube (e.g. irrigation tubes found in DIY stores);
- A short (around 10cm long) PVC tube, and a set of small compatible pipe junctions (to make changeable mouthpieces);
- A small (3x3cm) piece of cloth (you can cut it from an old t-shirt or something like);
- Strong insulating tape;
- (Optionally) a small box to contain the electronics.

These materials have an estimated total cost of 30€/35$ at the time of writing.

The pressure range detectable by the peripheral is between 0 and 10 KPa, and depends on the use of the low pressure sensor MPX-5010 DP[3]. According to the data sheet, the sensor has a sensitivity of 1 mV/mm and a response time of 1 ms. The sampling rate, calculated through experimental evaluation (Sec. 4.3.1) is ~200 Hz.

Unlike a normal wind instrument, air does not come out of the tube, since the pipe is capped

---

[3] NXP Freescale Semiconductor MPX-5010 DP data sheet: `https://www.nxp.com/docs/en/data-sheet/MPX5010.pdf`

by the sensor at its end, and no vents are included. While the interaction may be less natural in this way, particularly for a user accustomed to a traditional wind instrument, the most immediate advantage is the ability to breathe through the nose while pressing with the mouth. In this way the duration of the notes is potentially unlimited.

Interaction through NeeqBS was evaluated with the experiment described in Ch. 4. It has been shown that pointing stability is not very high, however the device can be suitable for controlling sound intensity through breath, like a wind instrument.

The software which allows the peripheral to work is divided into two sections: an Arduino code, and a C♯ library which acts as a driver. The Arduino code is contained in the GitHub repository *Neeqstock/NeeqSensors*[4]. The peripheral output is very simple: each line is a string consisting of a numerical value corresponding to the breath pressure detected by the sensor. Interfacing the peripheral to a computer involves using any library capable of reading a series of inputs from the serial port to which the peripheral is connected. An option is to use the *NeeqDMIs* library, specifically the *SensorModule* described in Sec. 7.3. It is sufficient to implement a class and an instance of *ISensorBehavior* to be included in the Behaviors list of the *SensorModule* object. The Behavior will be noticed for every new input generated by the peripheral. Through the Behavior it is possible for example to map a breath intensity change to a musical performance parameter within the instrument logic.

## 8.2. NeeqHT

NeeqHT (an abbreviation for Neeq Head Tracker) is a device suitable for detecting head rotation. This information is returned in the form of an absolute position.

The peripheral is built using the following materials:

- Arduino microcontroller (e.g. Uno or Nano);
- USB Type A to Type B cable (to connect Arduino to the PC);
- MPU-6050/GY-521 integrated accelerometer/gyroscope;
- At least 5 male to female (or female to female if using Ardino Nano) jumper wires for microelectronics, to connect Arduino to the sensor;
- Strong insulating tape;

Using these materials results in an estimated cost of 25€/29$ at the time of writing. The electronic components must be mounted on a wearable support, such as a pair of audio headphones, in order to be integral with the head. The accelerometer/gyroscope chip can be interlocked without requiring welding, but using a soldering iron is highly preferable. The MPU-6050/GY-521 MEMS integrated sensor[5] is capable to detect acceleration in six degrees of freedom, produced by movement along rotational axes (Yaw, Pitch, Roll) and positional movement into 3D space.

At start-up, or at each subsequent reset of the peripheral (using the appropriate button included

---

[4] NeeqSensors repository on GitHub: `https://github.com/Neeqstock/NeeqSensors`

[5] Invensense MPU-6050 datasheet: `https://tinyurl.com/4p5cpmde`

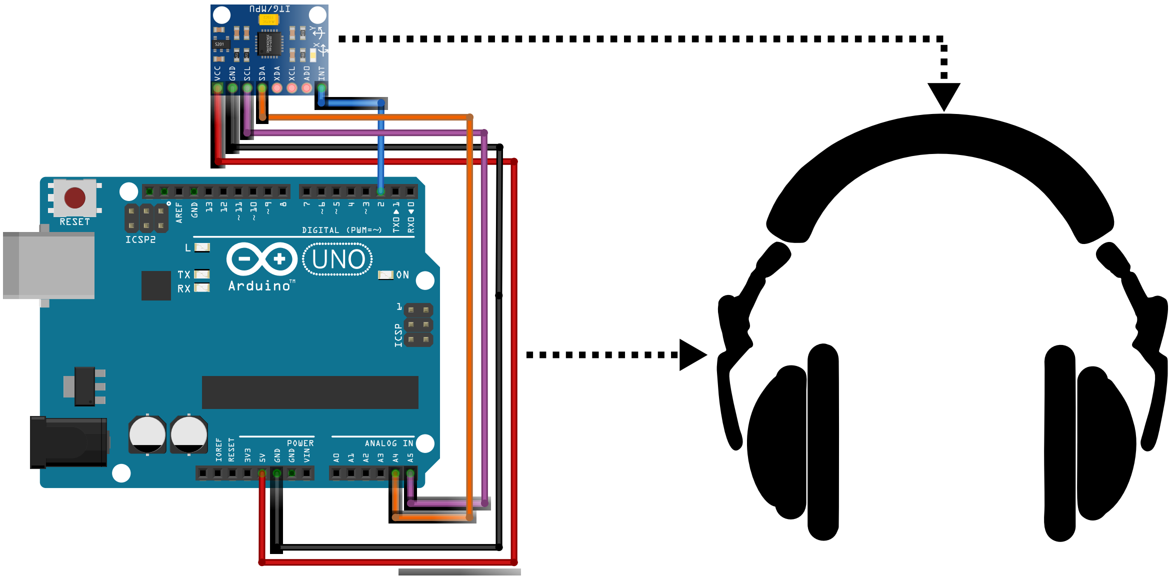


**Figure 8.2.:** *Implementation scheme for NeeqHT, mounted on a pair of headphones. The image include an Arduino Uno board and the MPU6050/GY521 chip.*
***Source:*** *headphones image shared by `https://freesvg.org/` under CC public domain license. Schematics drawn using Fritzing software.*

in the Arduino board), it is necessary to keep the peripheral at rest in a stationary position for about 30 seconds, in order to let it calibrate automatically.

As for NeeqBS, the software is divided into two sections: an Arduino code and a C♯ library which acts as a driver. The Arduino code is available from the aforementioned GitHub repository *Neeqstock/NeeqSensors*. The program executed by Arduino translates acceleration data into absolute rotation data with respect to the inertial system in which the sensor was calibrated. Each string output from the device contains information on head positioning. Values are expressed in degrees, with a resolution of one hundredth of a degree, using the following format:

```
$Yaw!Pitch!Roll
```

A programming API is provided by the *NeeqDMIs* library, in the *Headtracking* namespace described in Sec. 7.3.3. The *NeeqHTModule* class described in the same section allows to manage the head tracker generated output. The class also allows for position centering with respect to a new reference system, for example while the user's head is on a "neutral" and relaxed position.

The experiment reported in Ch. 4 used NeeqHT to detect head rotation. According to the discussed conclusions, head movement could be useful for managing sound intensity, or for navigating an interface by moving a cursor, as an alternative to gaze tracking.

## 8.3. Future work

An evaluation of the peripherals for different purposes should be carried out. Both NeeqBS and NeeqHT have been evaluated through the experiment described in Ch. 4 for general and music oriented purposes. More domain specific evaluations, for example for other accessibility purposes, could be carried out.

Further development could improve the replicability of the peripherals. For example, utilizing a chip which does not require soldering in substitution for the MPU-6050/GY-521 used to build NeeqHT could make the process easier for people who do not have the required skill or do not possess the required equipment. Documentation for building and replicating the peripherals, which is actually provided online through a website[6], could be improved introducing variations and substitutions in the case the required materials are not available.

---

[6] NeeqSensors documentation on Neeqstock website: `https://neeqstock.github.io/sensors/`

# 9

# Netytar

This chapter presents and describes the design, development and testing of Netytar. It is an isomorphic, monophonic and virtual HeaDMI controlled through gaze detection (used mainly for notes selection) and breath (used to control playing intensity and dynamics). A first version of the instrument was described in the paper [7], while [4] provided greater insight on the instrument, as well as the learning method discussed in Sec. 9.4.

The main motivation behind the development of Netytar is to experiment a new layout for gaze-based musical interfaces, described in Sec. 9.1. The instrument is mainly designed for motor impaired people, who cannot play a standard musical instrument, however it can also be an interesting instrument for users without disabilities for playing two instruments at the same time since it does not require the use of hands.

Netytar's design was originated from the author's own ideas, but was subsequently informed by collecting feedback and requests from the involved target group. This kind of iterative design brought to the introduction of refinements in the interaction scheme and performer aids. However the keys layout was almost completely unmodified since the first version of the software HeaDMI.

Although Netytar was conceived to be partially controlled using breath, other interaction methods to control playing intensity and note dynamics have been developed subsequently, as described in Sec. 9.8. The breath sensor could be substituted with a simple "switch" (for example, a button, a sip-and-puff tool, a pedal, or even eye blinking), as indicated in Sec. 9.3.

Netytar is a MIDI controller, thus it requires connection to an external or internal (i.e. software) device such as a synthesizer in order to produce any sound.

Netytar is released under the Open-Source GNU GPL-v3 license, and its source code is available on its GitHub repository[1].

This chapter is structured as follows. Sec. 9.1 describes Netytar's virtual keyboard layout and gaze interaction characteristics; Sec. 9.2 describes breath interaction and notes performance control; Sec. 9.3 provides informations on sidebar controls and settings; Sec. 9.4 proposes a learning

---

[1] Netytar repository on GitHub: `https://github.com/LIMUNIMI/Netytar`

**Figure 9.1.:** *The latest version of Netytar running on a laptop having NeeqBS breath sensor peripheral connected.*

method for both musicians and people without prior musical experience; Sec. 9.5 discusses a possible musical notation to write music to be played with Netytar; Sec. 9.6 provides a dimension space evaluation of the instrument through the framework described in Ch. 2; Sec. 9.7 describes a pre-test for the experimental evaluation of the instrument, compared to another gaze-based HeaDMI; finally Sec. 9.8 discusses planned future improvements of the instrument.

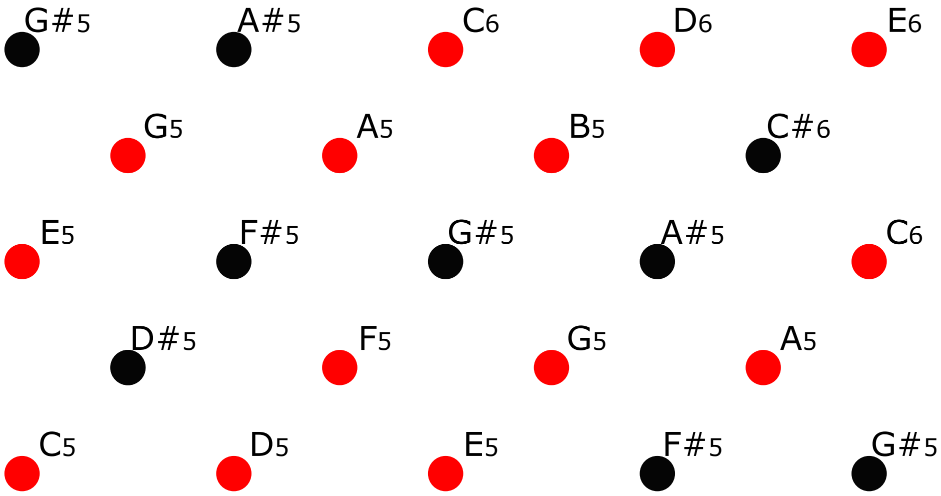
## 9.1. Notes layout and gaze interaction

Netytar's keys layout consists in an array of round colored keys, each corresponding to a note, arranged in a slanted squared grid.

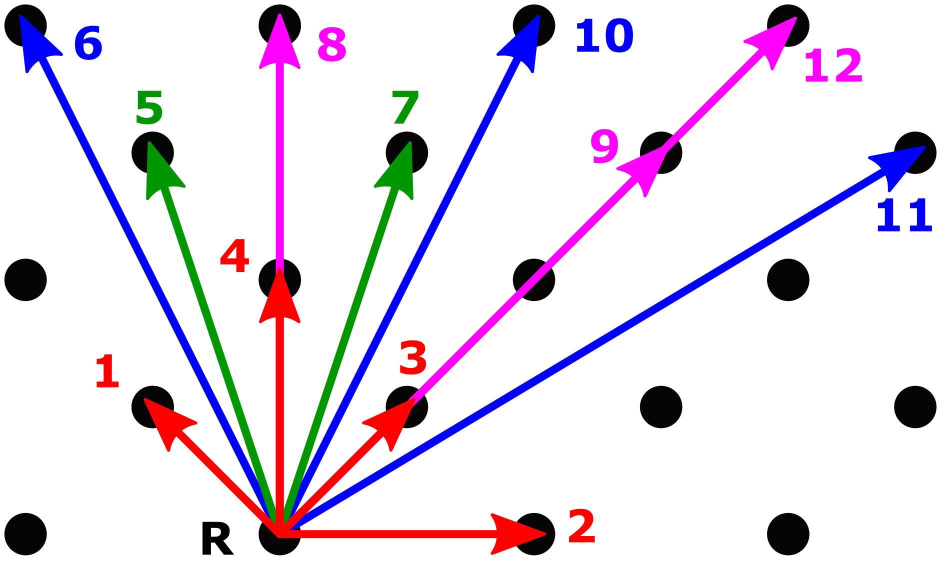
Netytar's interface is shown in Fig. 9.3. Its layout have been designed to provide a new solution for the Midas Touch problem (described in Sec. 6.4).

Musically speaking, it is *isomorphic*. Isomorphism, applied to keys layouts, means that keys have a geometrical displacement which brings the following property: the transposition of any musical piece, scale, or chord to a different musical key does not change the "shape" of the corresponding path. To give a counter-example, in a non-isomorphic instrument like the piano, different combinations of white and black keys would be necessary to transpose a musical piece to a different key/tonality.

Isomorphisms in musical interfaces have been studied by Maupin [237], which provided a review of various squared grid isomorphic layouts for fingered instruments, highlighting musical benefits and disadvantages from a melodic and harmonic point of view. As noted by Maupin, isomorphism, in a squared grid layout, can be obtained simply by assigning a fixed rule with respect to the musical intervals corresponding to a vertical or horizontal movement between adjacent keys.

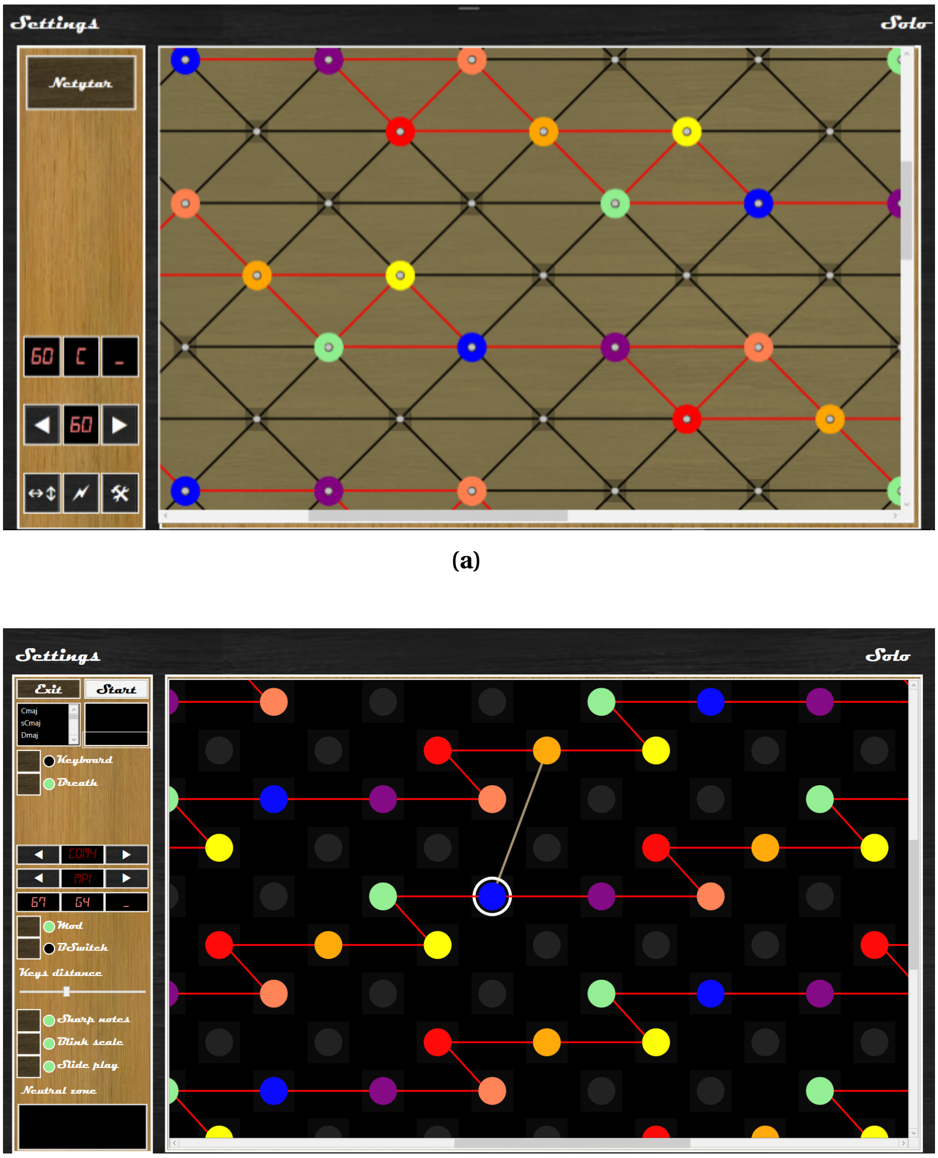


**(a)** *Netytar's keyboard layout, explained indicating the position of absolute notes, in a section ranging from C5 to E6. Keys assigned to notes of the C major scale are indicated in red, while black keys denote accidentals.*



**(b)** *Vectors indicating possible ways to trace intervals on Netytar's keyboard. Numbers indicate half-tones from the starting note (indicated with an R). Colors follow the interval groups defined in Sec. 9.4.2 (exercise T1), and reflect playing difficulty. From the least to the most difficult: adjacent group is colored red; distant easy green; distant hard blue; obstructed purple.*

**Figure 9.2.:** *Two schemes to explain Netytar's note layout.*



**(a)**

**(b)**

**Figure 9.3.:** *Two screenshots of Netytar application in one of its first versions, developed in late 2017 (a), and in its latest version at the time of writing (b).*

| Grade in scale | Color |
|---|---|
| 1st | Red |
| 2nd | Orange |
| 3rd | Yellow |
| 4th | Green |
| 5th | Blue |
| 6th | Purple |
| 7th | Peach |

**Table 9.1.:** *Color code for keys on Netytar's keyboard. Colors represent grades on the selected scale (major or minor), and not absolute note values.*

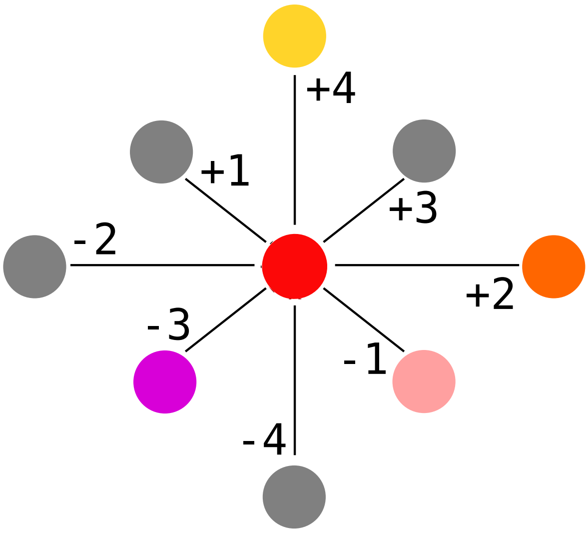


**Figure 9.4.:** *Isomorphic rule for adjacent keys in Netytar. Numbers indicate half-tone intervals from the center key (C), in red.*

Netytar's layout arranges keys on a rhomboid grid, thus having the same properties as a squared grid rotated by 45 degrees. Fig. 9.4 shows which intervals correspond to a shift between adjacent keys, which can be summarized as follows:

- A horizontal shift to the right corresponds to an increment of 2 half-tones; a shift to the left corresponds to a decrease of 2 half-tones;
- A vertical upward shift upwards corresponds to an increment of 4 half-tones; a vertical downwards shift to a decrease of 4 half-tones;
- A movement on the top-right/bottom-left diagonal produces positive (towards top-right) and negative (towards bottom-left) jumps of 3 half-tones;
- Moving on the top-left/bottom-right diagonal produces positive (towards top-left) and negative (towards bottom-right) jumps of 1 half-tone.

All remaining intervals can be obtained as a vector sum of the previous ones. For example, combining a shift of 3 and 4 semitones, a 7 semitones shift can be obtained (which musically speaking corresponds to a perfect fifth interval). A more complete vision on the intervals within an octave range described from different movements is given by Fig. 9.2b.

Another consequence of this layout is that different keys can be used to play the same note, and therefore there are several "paths" to play the same sequence. One path could be simpler and more comfortable than others while playing, as happens while playing a stringed instrument like the guitar.

The evaluation method described in [237] consists in evaluating the melodic and harmonic possibilities offered by a given layout. Following the same scheme, it is possible to highlight the melodic capabilities of Netytar, showing how simple the performance of chromatic and diatonic scales is. Fig. 9.5a shows the path required to play a diatonic major C scale on the Netytar layout. As regards an evaluation of the harmonic capabilities, it is useful to show the paths necessary to play basic arpeggios found in music. Maupin frames them as the major, a minor, and dominant 7th arpeggios. Fig. 9.5b shows the paths required to play such arpeggios on the Netytar layout.

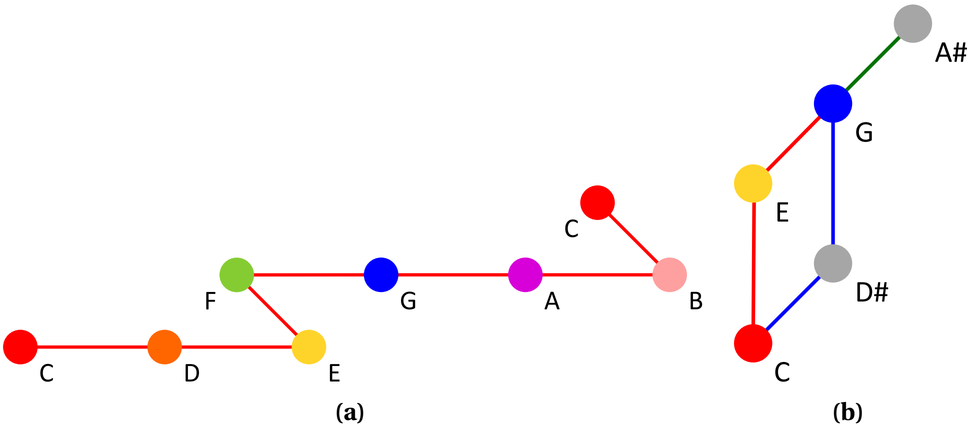


**Figure 9.5.:** *Path required to play a C major scale (a); and paths required to play a C major arpeggio (in red), a C minor arpeggio (in blue) and a C dominant seventh arpeggio (combining red and green) (b).*

The use of an isomorphic layout should make the transposition of musical sequences on different keys easy. Moreover, it could make the relationships between notes clearer and more immediate, due to geometrical consistency. Among other isomorphic layouts, this one has been chosen in order to avoid as much as possible intermediate key crossings for the most common musical intervals (illustrated in Fig. 9.2b), proposing a partial solution to the aforementioned "Midas Touch" problem. Sequences of notes can be described as paths or geometric shapes composed of broken lines, potentially making memorization easier. The aforementioned study from Maupin [237] investigate in greater detail the learnability of various isomorphic layouts. Effects on learnability and differences between isomorphic and non-isomorphic layouts have been studied also by Stanford et al. [238]. There is evidence that isomorphic layouts have benefits among musicians, but results among non-musicians are mixed, leaving home for further experimentation.

In the latest version of Netytar[2] keys are arranged in accordance with the so called *SMARC* effect: a layout in which the highest notes are found in top-right and the lower notes in the bottom-left should be more natural and immediate [239]. Figures 9.2b and 9.2a provide a graphical explanation of keys arrangement in terms of intervals and notes displacement.

As mentioned in Sec. 6.3, color can be used as a visual cue to enhance gaze-based interfaces. In Netytar, colored lines connect keys belonging to the currently selected musical scale. Red is used to highlight a diatonic major scale, while blue is used for a minor one. Keys color highlight the position of the corresponding notes within the currently visualized scale or, musically speaking, the degree of the corresponding note within the scale. For example, red denotes the root note (or 1st degree), and blue is used to indicate the the 5th degree. Table 9.1 provides a complete list of key colors with the corresponding degree. Accidentals (flat and sharp notes) are colored gray.

Scales can be selected through blinking. A blink with the left eye will make the interface draw new colored connectors and dots to highlight the minor scale having as root note the currently gazed key while blinking. A right eye blink will instead highlight a major scale.

Gaze trace is displayed by means of white flashing lines that connect keys to each other, which

---

[2] `https://github.com/Neeqstock/Netytar`. Accessed on: 29/02/2020.

gradually disappear after each transition. These provide a "history" of the movements just performed. In the first versions of Netytar, a flash appeared in the center of the key providing a visual feedback of note performance (or note onset). In recent versions of the instrument the flash was dismissed, as it could provide an element of distraction to the user, and has been substituted by a persistent indicator consisting of a white circumference which encircles the gazed key. The described elements correspond to an implementation of the "discrete cursor" described in Sec. 6.3. This way, the performance of a music piece can be associated with the act of describing paths on the surface and between keys, thus offering a mnemonic help to the performer.

Unlike other gaze-controlled musical instruments, and given the characteristics of its layout, Netytar is not characterized by Delayed Audio Feedback (DAF) introduced by software filtering (i.e. latency between the action of the physical input and the generation of the related sound, which may reduce the quality of the musical performance, as described in Sec. 6.3). Some interfaces based on gaze interaction (e.g. The EyeHarp, described in Sec. 5.1.8) employ algorithms, fixation-discrimination systems and other kinds of spatial filters to alleviate problems such as movement inaccuracy, noise introduced by the eye tracker sensor and, most importantly, Midas Touch problem (Sec. 6.4). Since Netytar does not employ such filtering techniques, latency in the feedback is minimized, making the instrument more reactive but potentially more challenging to learn.

To reduce gazing errors, each key is associated to an actual (non-visible) gaze-sensitive area (or *occluder*, as described in Sec. 6.4) that is larger than the key itself. The dimension of the occluder is configurable to meet different performance needs.

As a solution to provide a virtual keyboard with a potentially unlimited number of keys, Netytar implements the autoscrolling feature described in Sec. 6.3. This smoothly moves the surface on both vertical and horizontal axes, in such a way that the point which falls under the gaze point is always scrolling to the center of the screen. In such a way, Netytar keyboard dimensions can be larger than the size of the screen.

The appearance of the interface has changed through the various iterations of the instrument's development. Fig. 9.3a shows an early version of the instrument's interface. The squared grid was highlighted by black or colored connectors, to serve as a guide for gaze to perform the closest musical intervals. White dots were placed in the center of the key to serve as guide for the gaze, following a design cue listed in Sec. 6.3.

Fig. 9.3b shows the latest version of the instrument. The surface background was colored black to improve contrast and reduce eyes strain due to light. The interface was simplified to remove all possible distraction elements which could cause involuntary gaze movements. White dots were removed from the center of the keys and the grid connectors only connect the keys of the currently selected musical scale.

## 9.2. Notes performance

Breath is used to control playing intensity and dynamics. Netytar was designed to work with the *NeeqBS* breath sensing peripheral described in Section 8.1. The performance takes place in a similar way to a wind instrument, where the intensity of the sound is proportional to breath pressure. In order for the onset of a note to take place it is necessary to reach a minimum pressure threshold.

As discussed in Sec. 6.4, using a pie-shaped interface like the one used in the EyeHarp (described in [62] and in Sec. 5.1.8) or a context-switch based interface like in EyeJam (described in [186] and in Sec. 5.1.13) requires to designate one or more areas dedicated to pauses (i.e. stopping sound). If gaze is the only mean to control the performance, gaze should be able to control performance dynamics and pauses. In Netytar, the separation of note selection, controlled through gaze, and dynamics, controlled through breath, eliminates the need for pause areas on the instrument's virtual surface, since silence simply corresponds to no breath pressure. It is also technically possible to avoid the unwanted performance of notes (e.g. intermediate keys) through pausing, thus avoiding the Midas Touch problem. The solution is similar to the one proposed by Lumiselo (described in [194] and Sec. 5.1.6), which introduced the same mapping separation strategy.

Gazing two notes consecutively while breathing will result in the performance of a *legato*, i.e. the consecutive performance of both notes without intermediate pauses. This feature is called "Slide Play" and can be disabled, thus requiring breath pressure interruption to play any new note.

A common concern of eye-controlled musical interfaces is the repetition of the same note. Different instruments in the state-of-the-art have addressed the problem in various ways. In The EyeHarp [62], a key placed in the center of the interface is mapped dynamically to the last played note. Gazing the center key, the last note played is repeated. In EyeJam [186], the same note is mapped to two different keys. Netytar implements two alternative solutions for the repetition of the same note:

- the performer can interrupt the breath flow then exhale again;
- the performer can blink both eyes at the same time to repeat the gazed note, without the need to interrupt their breath.

## 9.3. Sidebar controls

Interface controls placed in the left side bar (shown in Fig. 9.3b) provide some additional customization possibilities to the instrument.

- The **Keyboard/Breath** selector allows to switch between breath and keyboard note performance control. The latter implies note onset happens by pressing the spacebar key. This is useful both for testing purposes and for users with partial control over fingers movement, such as people having hemiplegic paralysis;
- **Mod** switch allows to activate modulation control. In this modality, MIDI modulation is added proportionally to breath intensity over a certain threshold;
- **BSwitch** control switches between two breath control modes. In the default mode, breath controls sound intensity proportionally to pressure, while in *BSwitch* mode the breath controller acts as a switch: only two intensity values are detected (zero or maximum);
- **Keys distance** slider allows to adjust spacing between keys;
- **Sharp notes** allow to show or hide keys corresponding to accidentals (colored in gray);

- **Blink scale** activates/deactivates an interaction mode in which left/right eye blinks can be used to make the virtual keyboard highlight different scales (as described in Sec. 9.1);
- **Slide play** allows to activate/deactivate the "Slide play" feature described in Sec. 9.2.

## 9.4. Learning method

In this section a method which we have developed to learn playing music with Netytar is proposed and discussed. Although there are several other gaze operated ADMIs available in market and literature, a formal method for studying music with them has not yet been proposed. This can be useful for approaching musical performance with Netytar, but can also be potentially generalized for learning other similar instruments. The exercises are illustrated, discussed and explained in view of an improvement. At the end of a learning cycle, a user is expected to be able to perform simple melodies, and have a basis with which to learn other new ones. In the future, the method will be tested with the target users.

Despite the abundance of ADMIs, and gaze operated instruments in particular, there is a general lack of teaching methods for them in literature[3]. This and numerous other factors may discourage their use both by private users and by centers for rehabilitation or hospitalization with music teaching or music therapy departments. As an example, as highlighted by [203], the lack of repertoires and communities dedicated to a specific Digital Musical Instrument (DMI) in general could negatively affect its diffusion. Ward *et al.* [240] outline a group of guidelines for the development of musical instruments dedicated to Special Educational Needs (SEN) contexts, highlighting that technology is often overlooked, being seen as too complex, useless, or “geeky”.

The proposed method is designed to cover some aspects of gaze-based musical interaction. This consists of a series of exercises dedicated to non-musicians who approach music for the first time using the instrument. Such exercises are aimed at covering different aspects of a first experience with a musical instrument, both gaze based and in a general sense. Exercises should make the musician able to explore the instrument by themselves and deepen its practice, having gained a certain familiarity with the movements and understanding the rationale of the interface. Exercises are designed for simplicity, and are given in order of difficulty: some are preparatory to others and should be performed in the prescribed order, at most mixing them up between categories and going back to the previous ones from session to session. It is assumed that at the end of a certain number of iterations, the user will be able to perform simple melodies, as well as to learn new ones independently. Broader aspects of musical theory are not addressed in this chapter, given that adequate literature already exists. The focus is instead on performance aspects and on the use of the instrument. Nonetheless, it may be useful to combine the proposed exercises with pure music theory provided by other sources.

The method consists of three categories of exercises, which correspond to three related sections:

- **Musical calisthenics**, i.e. exercises designed to train motor skills of the eyes and breath in view of the required performance;

---

[3] One notable exception is the MUSA project, which involved teaching music to users with disabilities using the EyeHarp. See `https://www.upf.edu/web/musa` (Accessed on: 29/02/2020).

- **Musical techniques**, performed using the instrument;
- **Musical practice**, where the performer applies the acquired skills for musical purposes. This part is particularly important to provide motivation for the student.

For each exercise, a sentence is provided to describe its aim and discuss the expected improvement. Although the method is focused on Netytar, some sections (particularly *Musical calisthenics*, Sec. 9.4.1) could be easily adaptable to other gaze based musical interfaces.

### 9.4.1. Musical calisthenics

Eyes are governed by muscles. Constant training could improve or stabilize their rhythmic performance, as well as reducing fatigue. People with physical disabilities may have reduced coordination in the residual movement channels, as well as a lack of rhythmic ability. This part of the training therefore consists of a series of exercises aimed at improving these aspects: improve sensitivity in the awareness of eye movements, noting and bringing to consciousness involuntary movements, jitter and other peculiarities; perform muscular stretching, so as to accustom the eye to perform large movements while keeping the head still, without suffering fatigue and pain; accustom the eye to make saccadic movements rhythmically, perceiving muscle tension. Similar exercises are given for breath as well, which is the second interaction channel employed by Netytar. Exercises are as follows, divided by those dedicated to the eyes (with prefix *CE-*) and those dedicated to the breath (with prefix *CB-*):

**(CE1) Arhythmic stretching and smooth pursuit.** An assistant places themselves in front of the student, holding two colored objects. The student is instructed to move with the gaze from one object to another with saccadic movements, at a moderate pace while keeping the head still. The objects are initially placed close to each other, and their distance is slowly increased (horizontally, then vertically in the subsequent iteration), until the limits of the visual field are reached. Then, the distance is gradually decreased again. The exercise must be interrupted if the user experiences excessive discomfort or pain, especially near the limits of the visual field. A short session, in which the student is asked to concentrate and observe an object moving smoothly in front of them in the most precise way may be added (bringing the *smooth pursuit* movement to consciousness). **Aim:** an improvement in eyes mobility is expected after this exercise, as well as reduced fatigue while making long saccades.

**(CE2) Rhythmic blinking.** Using a metronome, starting from slow tempos then repeating the exercise at faster ones, the student is asked to blink both eyes in time, at every tick. It is possible to introduce rhythmic dictation exercises to introduce complex rhythms, possibly in combination with simple notions of rhythm theory. **Aim:** this exercise is designed to familiarize the student with the concept of rhythm, before moving on to more difficult eye-tapping exercises.

**(CE3) Rhythmic eye-tapping.** An assistant places themselves in front of the student in the same way as *CE1*. The student performs *CE1* in a timed manner with a metronome (one saccade per tick). The assistant can also provide feedback on the correct timing through direct observation. This exercise is potentially more difficult than *CE2*, given the anticipatory characteristic of the gaze movement discussed in Ch. 6. Since when playing Netytar the new note will sound exactly at the end of the saccadic movement, the student must become accustomed to this characteristic, as well as to perceive objects outside the fovea before even performing the movement. **Aim:** an improvement in anticipatory movement reduction is expected as a consequence of this exercise.

**(CE4) Rhythmic fixation.** The student is asked to perform *CE3*, but instead of performing a saccade for each tick, they will perform one saccade every four, concentrating on keeping the fixation as stable as possible on the object between the saccades. **Aim:** this exercise aims to bring to a conscious level the involuntary movements of the eye, which can preclude a voluntarily stable fixation. A precision improvement is hence expected.

**(CE5) Rhythmic color tapping.** Several objects with different (possibly highly contrasting) colors are placed in front of the student. A sequence is established a priori (e.g. *"red, yellow, blue, green"*). The student is then asked to perform timed eye-tapping (like in *CE3*) by fixating on the objects in turn, cycling along the predetermined sequence. After a few cycles, the student is asked to close their eyes: the objects are re-positioned randomly, then they repeats the exercise. **Aim:** this exercise could be useful to strengthen the ability to find objects outside the fovea, taking advantage of the color sensitivity discussed in Sec. 6.3.

**(CE6) Rhythmic mixture.** This consists of a variant of *CE5* (i.e., with two or more objects) where the student performs a mixed sequence of eye-taps (corresponding to note changes), blinks (corresponding to repeated notes) and fixations (corresponding to holding a note), timed by metronome ticks. Possible repeated sequences could be, for example: *tap, blink, tap, blink...* or *tap, fix, blink, fix, tap, fix, blink, fix...*. By introducing a simple symbolic notation, more complex sequences can be outlined, to be read and played in real time, while increasing the difficulty (as happens with *solfeggio* in traditional music education contexts). **Aim:** this exercise is aimed at strengthening the independence between saccadic movements (useful for selecting a new note) and blinking (useful for performing a repeated note).

**(CB1) Stabilizing breath.** The student is asked to blow into the breath sensor's mouthpiece with as constant and stable intensity as possible for a few seconds. In subsequent iterations, the level of breath intensity to be achieved is varied. **Aim:** this exercise should improve breath stabilization.

**(CB2) Breath crescendo.** The student is asked to perform a "crescendo", i.e. a continuous increase of intensity, to reach a peak, and then gradually decrease to a resting position, all in the smoothest possible way. This should be performed at different speeds at each iteration. **Aim:** this exercise aims to strengthen control over the change in intensity.

**(CB3) Breath tapping.** Once the metronome is set, the student emits breath with constant intensity for a predetermined number of ticks, then stops the emission for as many ticks. They will repeat the exercise in a continuous cycle. An example would be: *two ticks blowing, two pause ticks, two ticks blowing, two pause ticks, etc.*. **Aim:** this exercise should improve rhythmical breath control.

**(CB4) Breath tapping with short bursts.** Again with a metronome, the student will perform breath emission impulses at each tick (at slow rhythms) or interspersed with a variable number of ticks (at more sustained rhythms), estabilished a priori. **Aim:** this exercise could be useful to gain confidence with the timed release of breath, as well as to strengthen the required muscles (i.e. diaphragm).

### 9.4.2. Musical techniques.

Once the rhythmic control of the eyes has been strengthened with exercises in the previous section, this next set of exercises should be performed directly on the Netytar's interface. These aim to transfer the acquired motor skills to simple technical musical sequences, which are preparatory to melody performance. Exercises in this category are noted with the prefix *T-*.

**(T1) Interval tracing.** Playing Netytar, the difficulty associated to performing different musical intervals while avoiding the activation of intermediate keys is uneven: with respect to some intervals, distances between keys are large and paths narrow (sometimes obstructed). Intervals, with reference to the chosen isomorphic layout, could be roughly divided into 4 ranges of difficulty: **adjacent**, or in the immediate vicinity of the key (1, 2, 3 and 4 half-tones); **distant easy**, i.e. not adjacent but not obstructed by other keys, therefore rather simple to perform (5 and 7 half-tones, corresponding to perfect 4th and 5th); **distant hard**, or described by unobstructed but narrow, distant or difficult paths (6, 10 and 11 half-tones); **obstructed**, or described by paths obstructed by other keys (8, 9 and 12 half-tones), however playable through a rapid saccadic movement or breath interruption. These 4 groups are shown in Fig. 9.2b. Having established this classification, the proposed exercise consists in performing, in both directions, in turns and in repetition, intervals with difficulty *adjacent* and *distant easy*. Notes can be played as quarter notes with a metronome. A variant can be introduced by performing a repeated note with a blink between each note change. **Aim:** this exercise should improve the association of geometric movements within the keyboard with given musical intervals, in addition to improving the confidence with the keys layout.

**(T2) Scales tracing.** Major and minor diatonic scales are performed using a subset of the *adjacent* group described in *T1*. It should therefore not be difficult for the student, once *T1* has been trained, to perform this next exercise: the major and minor scale are performed in ascending and descending directions with a metronome, one note per tick. It is possible to introduce repeated notes as indicated for *T1*. As a variant, it could be useful to introduce also major and minor pentatonic scales. **Aim:** this exercise should increase the performer's knowledge of the keyboard and

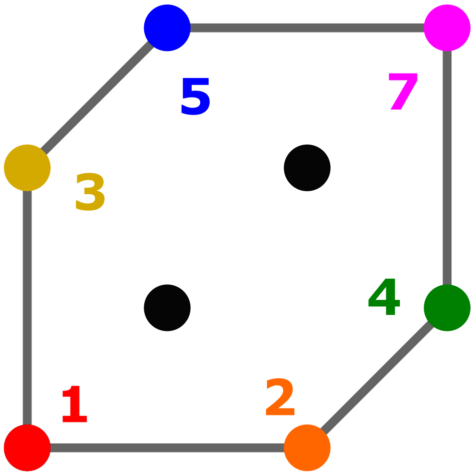


**Figure 9.6.:** *A possible closed shape for exercise T4 in Sec. 9.4.2. Numbers next to keys indicate the degree in the major scale of the corresponding note. Color code for keys is as in Tab. 9.1.*

its melodic capabilities, and improve the performer's playing precision.

**(T3) Arpeggio tracing.** In this exercise, the student plays various arpeggios using a metronome, one note per tick. Although it is advisable to start from simple major and minor arpeggios with single triads, major or dominant 7th arpeggios could be introduced. These arpeggios trace very short and simple paths on Netytar's virtual keyboard, and should be repeated transposing them to other keys, so that the user becomes familiar with its isomorphic properties (described in Sec. 9.1) and the concept of transposition. **Aim:** the expected improvement given by this exercise is comparable to the previous, with respect to arpeggios.

**(T4) Complex shape tracing.** This exercise consists in defining arbitrary shapes and trace them with gaze on Netytar's keyboard, playing the notes following the metronome. Shapes can consist of open shapes (to be performed in an ascending or descending direction), or closed shapes (to be performed both clockwise and counter-clockwise). Examples of these shapes could be given by a complex, multiple octave chord arpeggio, or by the closed shape made by the 1st, 3rd, 5th, 6th, 4th and 2nd degrees of the major scale (as illustrated in Fig. 9.6). The student could be stimulated to invent and perform new shapes and "test" them. **Aim:** in addition to precision improvements, this exercise should introduce the performer to keyboard exploration.

**(T5) Shapes with returns.** Many traditional musical instruments study methods involve "repeated pattern" exercises. An example could be given by this sequence constructed on the major scale: *C D E, D E F, E F G, F G A, G A B, A B C* (to be played in both ascending and descending manner). **Aim:** this exercise could be preparatory to performing less linear and more complex phrases.

**(T6) Complex rhythms.** All the previous exercises are revisited, adding complex rhythms instead of the "one note per tick" pattern. Examples could be given by the execution of 2/4 notes followed by 1/4 notes, with or without the introduction of repeated notes. Sequences should be estabilished and given a priori. **Aim:** this should be the final introduction to melodic phrasing. The following section consists indeed in the execution of actual music pieces.

### 9.4.3. Musical practice

This section of the training consists of giving the student simple tunes to be played with Netytar, to put into practice the improvements given by previous exercises. Focus is not addressed on providing a list of melodies, given that there are already several texts and advice on the subject, dedicated to other instruments but still suitable. As an example, the *My Breath My Music* foundation is active in the music education field within SEN contexts (teaching people with disabilities in the upper limbs how to play the *Magic Flute*[4] instrument), and offers a training program composed of simple melodies freely available on their website[5]. It should also be noted that it is probably simpler for the student to play an already known melody than learning a new one. The musical tradition however varies from culture to culture. In different contexts it is possible to identify different pieces to propose. The following lend themselves to be useful guidelines for identifying simple tunes for Netytar.

- Identifying which types of musical intervals the performance requires helps to determine their difficulty. A difficulty classification is indicated in Sec. 9.4.2, for exercise *T1*.
- Tracing and transcribing passages using the notation described in Sec. 9.5 can help in the process, highlighting also the amount and localization of the required eye movement.
- The upper speed boundary imposed by the nature of saccadic movements, discussed in Sec. 6.2, should be taken into account, providing some constraints for the tempo.
- A good difficulty progression should take into account the rhythmic complexity of the piece. A homogeneous rhythm should be simpler.

## 9.5. Notation

In order to propose a simple way to write down music or exercises for the proposed learning method (in Sec. 9.4), which could be developed by a possible teacher or assistant, a simple notation is introduced. This does not aim at being as complete as traditional staff notation, but relies on the idea of indicating the “geometric shape” described by a short musical sequence and therefore provide simple mnemonic support that does not require previous knowledge in reading notes on the music staff. It can be described using the following rules:

- Notes that make up the sequence are connected by a broken line, following Netytar’s virtual keyboard layout and colors. The line can also be "folded on itself" to indicate to go over the same interval several times.
- Only a small number of bars should be drawn in a single image (1 bar or few more, depending on the complexity).

---

[4] `https://mybreathmymusic.com/en/magic-flute`. Accessed on: 29/02/2020.

[5] `http://mybreathmymusic.com/en/liedjes-spelen-voor-beginners`. Accessed on: 29/02/2020.

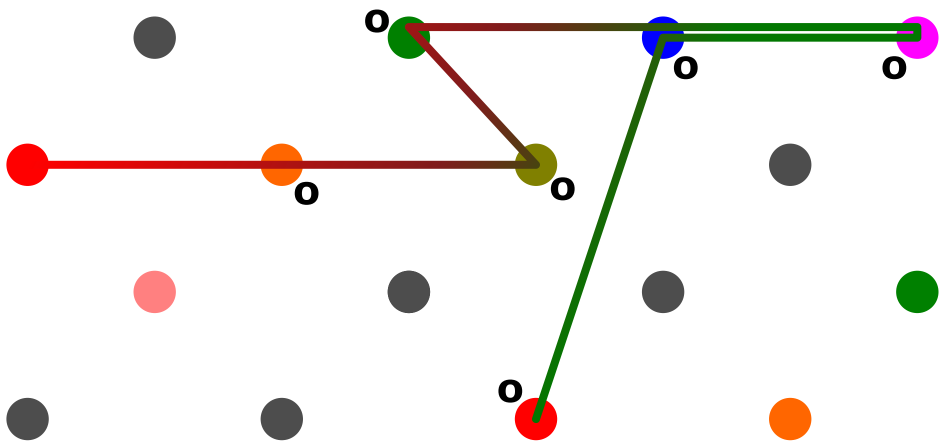


**Figure 9.7.:** *The first bars of the popular tune Twinkle, Twinkle, Little Star, drawn using the notation explained in Sec. 9.5*

- The temporal progress of the sequence is indicated by a color gradient along the line: a color (e.g. green) indicates the beginning of the sequence, another color (e.g. red) the end. Otherwise, if not possible, only the two endpoints could be noted down with color.
- A repeated note is indicated by single or multiple symbols (e.g. an **O**) placed next to the keys. This information could be otherwise omitted for visual simplicity.

An example is given by Fig. 9.7, which represents the first bars of the song "Twinkle, Twinkle, Little Star" *(C, C, G, G, A, A, G, F, F, E, E, D, D, C)*.

It should be noted that while playing an instrument that requires the performer to use their gaze as an interaction channel, music cannot be read at the same time using traditional staff notation. A future development of Netytar could implement the simple notation described above so that the score can be displayed directly on the keyboard while playing, providing a significant aid to the performance.

## 9.6. Design space analysis

An evaluation carried out using the framework proposed in Ch. 2 can help to point out and define some specific characteristics of the musical instrument. Fig. 9.8 shows a design space analysis through the aforementioned approach.

The *adaptability* of the instrument could be considered to be slightly higher than the other instruments presented in Part II of this thesis. In light of recent developments, in fact, although gaze tracing is the main input method for selecting notes, several new interaction channels for controlling the dynamics of the performance and the onset of notes have been introduced. These new channels and methods, described in Sec. 9.8, allow Netytar to adapt, for example, the potential of a user with hemiplegic paralysis (who can use a key and therefore a hand to play), or of a quadriplegic user with breathing difficulties (who can use the eyes to control pitch selections, notes onsets and dynamics).

Netytar's *design novelty* is considered high in light of the fact that its design is not inspired by any

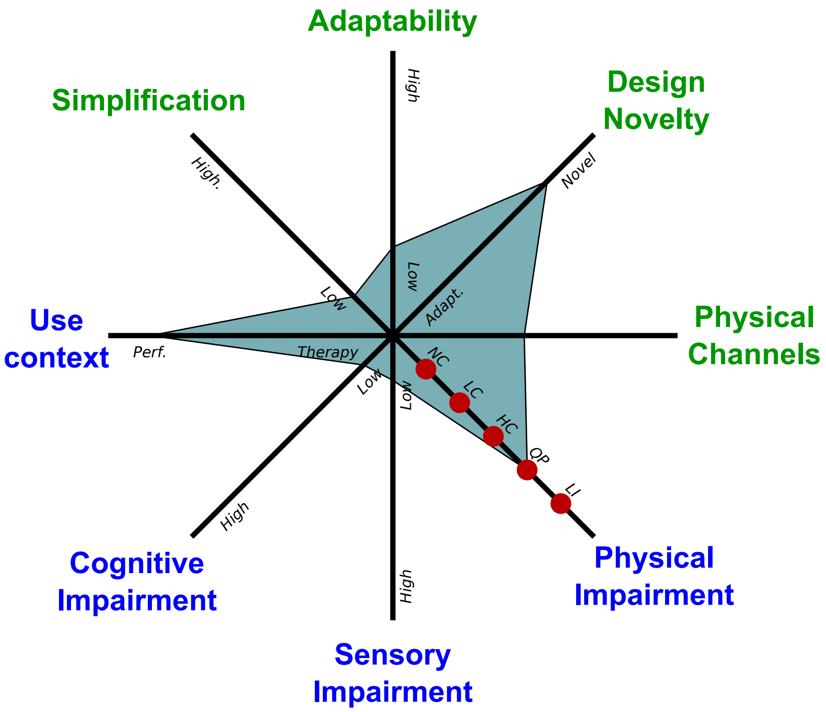


**Figure 9.8.:** *Netytar's design space evaluation using the framework described in Ch. 2.*

acoustic instrument, neither for the interaction methods, nor for the proposed layout. Breath is mapped to notes onsets and dynamics, as in wind instruments, but unlike those Netytar's breath sensor does not allow air to escape from the sensor pipe.

The number of *physical channels* employed simultaneously to control the instrument is average, at most two (in its gaze + breath configuration).

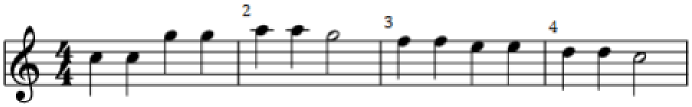
Netytar, as already stated, can adapt to the abilities of a user with partial limb control, however the highest degree of *physical impairment* addressed is quadriplegic paralysis.

*Cognitive* and *sensory impairments* are not addressed by Netytar, which does not introduce any aid to support these categories of disabilities.

Netytar's *use context* is strongly oriented towards performance, much like all the instruments presented in Part II of this thesis. A small number of *simplifications* are introduced in the recent versions of Netytar (such as the possibility to remove accidentals from the note layout described in Sec. 9.3), nonetheless Netytar is a performance instrument (using the definition in Sec. 1.2), which requires time and skill to be mastered.

## 9.7. Experimental evaluation

A preliminary test has been conducted on Netytar through experimental means [7] with the participation of eight non-disabled expert musicians. Four of them were pianists, three were guitarists, and one was a saxophonist. As stated in Sec. 1.5, designing an objective test to evaluate a musical instrument is a complex task, starting with the selection of the parameters to be measured



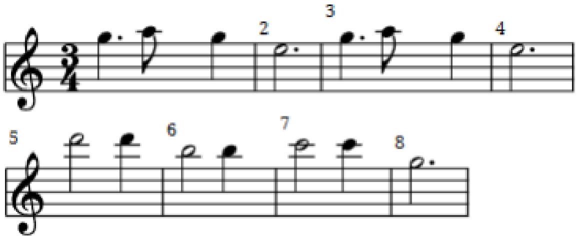
**(a)** *Twinkle*

**(b)** *Silent Night*

**Figure 9.9.:** *Music sheets for the musical exercises performed during Netytar test sessions.*

for a tool intended for an artistic activity. It is difficult to set specific objectives, requirements, thresholds and other objective components. For this reason, we preferred to compare Netytar to an instrument found in literature that we considered at the state-of-the-art: The EyeHarp (described in Sec. 5.1.8). All the testers had no prior experience with Netytar, The EyeHarp nor any gaze-based instrument. In order to limit emotional bias as much as possible, no tester was told the authorship of both instruments nor which laboratory developed them. Tests were conducted in Computer Vision Lab at the University of Pavia (Italy), between November and December 2017.

### 9.7.1. Test procedure

The test session was subdivided into three parts. The following procedure was repeated with each tester for both Netytar and The EyeHarp. Four testers performed at first with Netytar and then with EyeHarp, while the other four performed at first with EyeHarp and then with Netytar.

- In an initial training phase, the tester could play each of the two instruments for half an hour, performing free play and guided exercises. During the last ten minutes, a metronome set to 80 bpm was also used, to accustom the tester to precise rhythmic performance with the instrument;
- The second part of the test was a timed performance. Using a metronome at two different tempos for each exercise (70 and 100 bpm), the tester played the first bars of two well-known melodies, “Twinkle, Twinkle, Little Star” and “Silent Night”, while the MIDI output was recorded. The music scores of the two exercises are reported respectively in Fig. 9.9a and 9.9b. The tester could try an exercise for five times: only the trial they thought was the best was considered for the analysis;
- The third part of the test required the tester to answer a usability questionnaire, providing a rating about the characteristics of each of the two instruments.

Testers were also asked to "think aloud", and their most relevant comments were transcribed.

### 9.7.2. Usability questionnaire

In the usability questionnaire, the same sentences were repeated for both Netytar and The EyeHarp. The questionnaire was split in two sections. In the first part, scores were provided using a five-level Likert scale to express the agreement level with the given sentences: (1) I do not agree at all; (2) I do not agree; (3) I neither agree nor do not; (4) I agree; and (5) I totally agree. A high agreement score corresponded to a positive quality of the instrument.

The complete sentences of the first section were:

- **Feeling**: In general, I had a good feeling with [. . . ];
- **Scales**: It was simple to play scales with [. . . ];
- **Arpeggios**: It was simple to play arpeggios with [. . . ];
- **Complex melodies**: It was simple to play complex melodies with [. . . ];
- **Fatigue**: It was more tiring (ocular effort) to use [. . . ];
- **Frustration**: It was more frustrating to use [. . . ];
- **Improvisation**: It was simple to improvise melodies with [. . . ];
- **Visual FB**: I appreciated the visual feedback of [. . . ];
- **Learning simplicity**: It was easy to learn how to play with [. . . ].

Fig. 9.11a shows the result for this first section of the questionnaire.

The second part of the questionnaire did not employ Likert scale but simply shows “judgments” in a scale from 1 (lowest) to 5 (highest). The full sentences were:

- Timing difficulty: Express the difficulty you had in playing in time using [. . . ];
- Eyes vs. hands: Express how hard it seemed to you to play in time using your eyes, compared to playing with your hands, using [. . . ];
- Scrolling nuisance: I found the Netytar grid scrolling annoying.

Fig. 9.11b shows the result for the second section of the questionnaire.

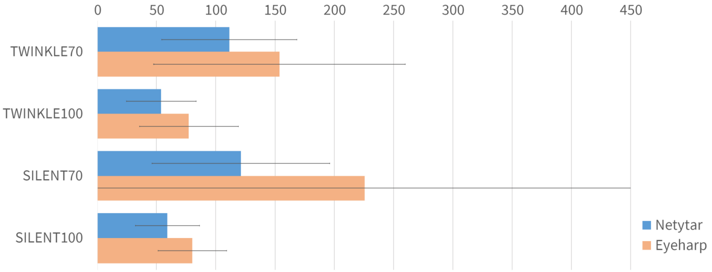
### 9.7.3. Equipment

Tests were performed using the Tobii EyeX eye tracker[6] for gaze tracing. Each subject initially performed a calibration procedure using the standard tool supplied with the official Tobii driver for the EyeX, which consists in fixating a few circles appearing in various positions of the screen. Calibration accuracy was checked through an interface (containing nine target points) showing gaze position in real-time. If, during the different test phases, the tester perceived a decreased accuracy, the calibration was repeated.

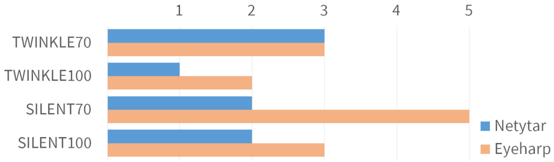
The MIDI recording software had a resolution of 192 PPQ (Pulses Per Quarter note). EyeHarp was tested in its “standard” configuration at the moment of download, using the latest version available (December 2017). In order to focus on the comparison of gaze-related characterstics

---

[6] Specifications for Tobii EyeX on Tobii Support website: `https://help.tobii.com/hc/en-us/articles/212818309-Specifications-for-EyeX`



**(a)** *Average timing error per note, in milliseconds*



**(b)** *Number of testers which played at least one wrong note in each exercise*

**Figure 9.10.:** *Results for the MIDI recorded tests for Netytar and EyeHarp. Error bars refer to standard deviation.*

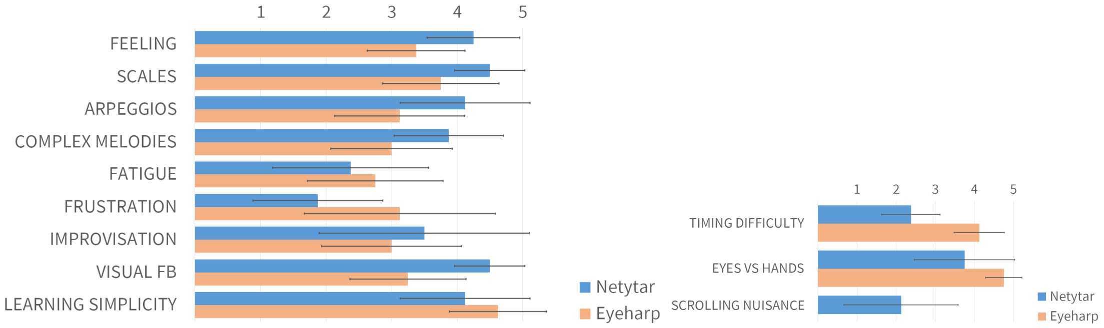
of the instrument, Netytar was tested using a switch (i.e. the spacebar key of the computer) in the place of the breath sensor. Key pressure corresponded to breathing at full pressure, while key released corresponded to no breath. Testers exploited however the "eye slide" characteristic described in Sec. 9.2 to avoid strumming as much as possible. The instrument was at an early development version, showing the interface depicted in Fig. 9.3a.

Both instrument were coupled to the same synthesizer, which provided the combined sound of a piano and a flute (to provide an accurate feedback on note timing and length).

### 9.7.4. Results

In the timed performance, preliminary test results suggest that Netytar performed better than EyeHarp in the tested musical exercises. Fig. 9.10a shows the average timing error per note (number of notes divided by total timing error) expressed in milliseconds. Fig. 9.10b depicts, for each exercise, the number of testers who played at least one wrong note. As can be seen, with The EyeHarp the maximum number of wrong notes was 4, while with Netytar it was 2. Regarding the usability survey, our system received almost always higher scores compared to EyeHarp. The results shown in Fig. 11 seem to confirm the lower difficulty of playing in time with Netytar ("Timing difficulty", "Eyes vs Hands"). Also, the automatic scrolling system ("Scrolling nuisance") does not appear to bother the performer.

The test and the questionnaire seem to confirm that the absence of delayed audio feedbacks in Netytar is positively perceived by the performer and can enhance the rhythmic performance. Probably, the use of a hybrid eye tracker and switch solution also played a key role. Given the pre-testing nature of the experiment, however, the number of obtained data was too low to assume a normal probability distribution of the samples and to confirm statistical significance of the comparison. The same issue applies to the questionnaire: we conducted a sign test which resulted in no statistical significance. More in-depth tests with a larger number of participants



**(a)** *First part of the questionnaire (Likert scake)* **(b)** *Second part of the questionnaire (plain rating)*

**Figure 9.11.:** *Results for the usability questionnaire for Netytar and EyeHarp. Error bars refer to standard deviation.*

should be conducted to confirm the hypotheses above.

## 9.8. Conclusions and future work

Several features included in Netytar's interface and described in Sec. 9.1 can be traced to well-known usability principles, that guided our design process. The increased size of the gaze pointing area for notes, in compliance with Fitts's Law [180], aims to improve the speed and the accuracy of the system with respect to other similar instruments. The isomorphic layout described in Sec. 9.1, while different from traditional keyboards, makes the transposition of sequences of notes more immediate (thus reducing the mental load) and allows to memorize musical phrases as visual paths. In addition, the isomorphic layout offers a certain degree of flexibility: indeed, the performer can choose among different paths to play the same sequence. The main advantages of Netytar compared to other state-of-the-art alternatives can be summarized as follows:

*a*) Absence of filters, i.e. delayed audio feedbacks;
*b*) Minimization of gaze shifts needed to play the most common musical intervals;
*c*) Strategic use of color, to reduce "exploration gaze movements", which may lead to involuntary note activations;
*d*) Use of an automatic-scrolling system, which provides a potentially unlimited playing area regardless of screen size (without the need to reduce the size of interface elements to expand the tonal range of the instrument).

A preliminary test suggest the hypothesis that Netytar could have better performance compared to an instrument at the state-of-the-art in all the measured metrics (execution accuracy, user satisfaction, octave extension, etc.).

As shown in Fig. 9.2b the actual implementation of Netytar's layout doesn't allow to play an octave interval without traversing several other keys. Possible solutions include the introduction of a breath sensor capable to detect mouth gesture which could be mapped to octave changes allows to modify the octave.

Additional interaction methods can substitute breath in order to provide different interaction channels and mappings for note performance. Some methods could provide control through eyes over both note selection and sound intensity control. Those are under testing phase and have been removed from the current release version of the instrument, and include:

- Control through **eyes height** in the 3D space. This can usually be detected by near-infrared camera based eye trackers. Height of the eyes is proportional to sound intensity. Under a certain height threshold, sound will stop (pause). In this way, the musician can control performance dynamics through head rotation on the pitch (vertical) axis;
- Control through **eyes movement velocity**. The faster the user will rotate his head, thus moving their eyes in the 3D space, the greater will be sound intensity. Keeping still corresponds to a silence (pause). Head rotation within both pitch (vertical) and yaw (horizontal) axes can be thus used.
- Control through **two eyes blink**. When the user makes a blink with both eyes, a note will play at maximum intensity for a given amount of time. The musician doesn't have control on sound duration and envelope; however, this interaction method is particularly simple. Since the onset of a note takes place only following a blink, unwanted interactions due to Midas Touch problem are also limited.

Such interaction methods can be thoroughly tested and included in future versions of the instrument.

More in-depth tests could be conducted to evaluate some features of Netytar. The preliminary experiment described in Sec. 9.7 need to be expanded through case studies conducted with the participation of people with motor disabilities.

Most of the exercises of the learning method described in Sec. 9.4 should be performed with a human assistant, which provides visual elements and gives feedback. However, a simple additional software interface could be created as a replacement, thus making the user autonomous in practising, providing also more precise and objective feedback. Visual objects indicated in Sec. 9.4.1 (*CE1, CE3, CE4, CE5,* and *CE6*) can be easily replaced by virtual objects on screen, providing also auditory or vibrotactile feedback (using an actuator) upon successful gaze selections of each item. Breath-related exercises in Sec. 9.4.1 (*CB1, CB2, CB3* and *CB4*) could be more effective if supported by an intensity indicator on screen. The use of other gaze controlled applications unrelated to the musical purpose could strengthen eyes control abilities and confidence with gaze interaction (e.g. eye controlled text writing software such as the freely available *GazeSpeaker*[7]). The proposed learning methodology can be tested with the target users, performing case studies and user experience assessment, measuring also possible improvements in users musical performance using objective methods. The method should be validated by experimental observations, which are deferred to future publications.

Finally, further learning aids could be developed. A gamification of instrument learning can for example be achieved by introducing an automated system which shows subsequent notes to be played directly on the interface (e.g. exploiting the notation described in Sec. 9.5).

---

[7] `https://www.gazespeaker.org/`. Accessed on: 29/02/2020.

# 10

# Netychords

This chapter describes Netychords, an HeaDMI developed and presented in 2021 [5]. Netychords is a Netytar extension developed to allow a user to play chords. The instrument is controlled through gaze and head movement. While gaze controls note selection through eye tracking, an head tracking sensor is used to control chord strumming. Interaction methods and mappings are discussed, along with a series of experimental note layouts.

The main point which motivated the development of Netychords is the lack of HeaDMIs in the state-of-the-art dedicated to the performance of chords. As we will see in Sec. 10.1, a small amount of instruments dedicated to quadriplegic users found in literature allow this kind of interaction, thus Netychords could fill that niche.

The chapter is structured as follows: Sec. 10.1 will compare Netychords to past literature; Sec. 10.2 discusses the implementation of the instrument; Sec. 10.3 provides an evaluation of its characteristics through the dimension space proposed in Ch. 2; finally Sec. 10.5 discusses further developments of the instrument after its publication, as well as possible future work and improvements.

## 10.1. Previous works and considerations

In Ch. 5, fifteen musical instruments operable by people with quadriplegic disability are analyzed. Only 6 of these seem to allow the performance of chords, with various degrees of versatility. Head tracking and eye tracking are widely consolidated as interaction channels for accessible applications.

Their use in selection tasks have been evaluated and compared through Fitts' Law tests, with the experiment described in Ch. 4, showing that gaze pointing is particularly fast and fairly stable, especially if the eye tracker data stream is filtered, while head movement is slightly slower but has excellent stability. However, it has also been mentioned in Sec. 6.2 that a maximum limit of physiologically possible saccadic eye movements per second can be obtained [98], which potentially hinders the performance of fast sequences. This limit is potentially more important if gaze is used for notes selection while performing melodic lines, whereas for chord changes it could be sufficiently high.

While precise eye tracking requires a dedicated peripheral, head tracking has been exploited for accessibility purposes (e.g. to navigate tablets) also through integrated cameras [241], as already mentioned in Sec. 3.2.3. However, for Netychords we preferred to use the *NeeqHT* wearable peripheral in order to improve tracking precision, which has been developed ad-hoc through DIY materials. Such peripheral was described in Sec. 8.2.

Some of the HeaDMIs reviewed in Ch. 5 allow chords performance.

Tongue-Controlled Electro-Musical Instrument [191], described in Sec. 5.1.2, consists of a PET board mounted on the palate. Using the tongue it is possible to press one of the buttons, arranged in a cross shape, to play the corresponding chord.

Eye Play The Piano [197], described in Sec. 5.1.9, is a gaze-based interface which allows to control a real piano through actuators placed on the keyboard. Although the instrument is not described in any scientific publication, the material available on the official website [197] shows that it is possible to customize the interface to play chords.

Jamboxx [188], described in Sec. 5.1.5, is an ADMI bearing similarities to a digital harmonica: a cursor moved using the mouth along a continuous horizontal axis is mapped to pitch selection. Although no scientific publication on the instrument is available, as with the previous one, it seems that it is possible to play chords.

The EyeHarp [62], described in Sec. 5.1.8, is controlled entirely by gaze. Keys are arranged in a pie shape. Through the described prototype it is possible to build arpeggios through a sequencer layer, then use some of the circularly arranged keys to trigger chord changes. Those will be played continuously in the background following the defined arpeggio pattern at a fixed tempo.

P300 Harmonies [198], as described in Sec. 5.1.10, is an electroencephalogram based interface that allows to generate and edit arpeggios live in a simplified way, by editing a 6-note loop.

From the same list, Clarion and Hi-Note are potentially controllable through head-tracking. The two instruments have been extensively described in Sec. 5.1.12 and 5.1.3 respectively. In both Clarion and Hi Note, head movement is mapped to note selection through a virtual interface.

Netychords is somehow related to a guitar: in both instruments, notes selection and strumming are controlled by two different channels (the two hands in the guitar, gaze and head movement in Netychords). Some guitar-inspired ADMIs, in the form of augmented instruments or novel interfaces, are already present in the literature, though not operable by a quadriplegic user. The Actuated Guitar [242, 243], for example, consists of a guitar adapted for by people with hemiplegic paralysis. The able hand is placed on the neck, while a pedal controls an actuator capable of plucking the strings. Strummi [244] is an instrument relatively similar to a digital guitar, which allows the performance of chords, designed for partial motor disabilities.

## 10.2. Implementation

Netychords interface, in its latest version, is depicted in Fig. 10.1a.

As said, the idea for its implementation stems from a general lack in the literature of polyphonic instruments (i.e., able to play chords) dedicated to users with severe motor disabilities.

As an example, the aforementioned EyeHarp [62] allows the performance of chords only in diatonic logic and using predefined rhythmic patterns.

Netychords shares with Netytar the use of the gaze point to perform notes selection. However, while Netytar exploits breath to control note dynamics, Netychords exploits head movement to perform note strumming events, which actually trigger a group of notes at the same time.

The instrument is operable through low cost sensors. It has been developed using a Tobii 4C eye tracker, which features 90Hz image sampling rate through near infrared illuminators (NIR 850nm)[1]. Technical specifications of the *NeeqHT* DIY head tracker are provided in Sec. 8.2. An implementation of the device, next to a laptop running Netychords, is depicted in Fig. 10.1b. Netychords source code is available[2] under the open-source GNU GPLv3 license. A demo video of the instrument is linked in the GitHub repository *Readme*.

### 10.2.1. Chord selection

Gaze point is used to navigate a virtual keyboard, having differently colored keys to indicate different chords. Each color corresponds to a different root note, while different color shades indicate different chord types.

Six different layouts have been implemented in the current iteration, allowing the user to choose the most suitable for their performance. Each key is assigned a square shaped gaze sensitive area (*occluder*). Given the noisiness of the eye tracker signal, there is a trade-off between selection accuracy and length of movement required. For this reason, most of the implemented layouts have two display modes having different distances between keys: square grid (Fig. 10.1b, 10.2a, 10.2b and 10.2c) or slanted. The reader can use Table 10.1 as a reference for the chords named below. The implemented layouts are the following:

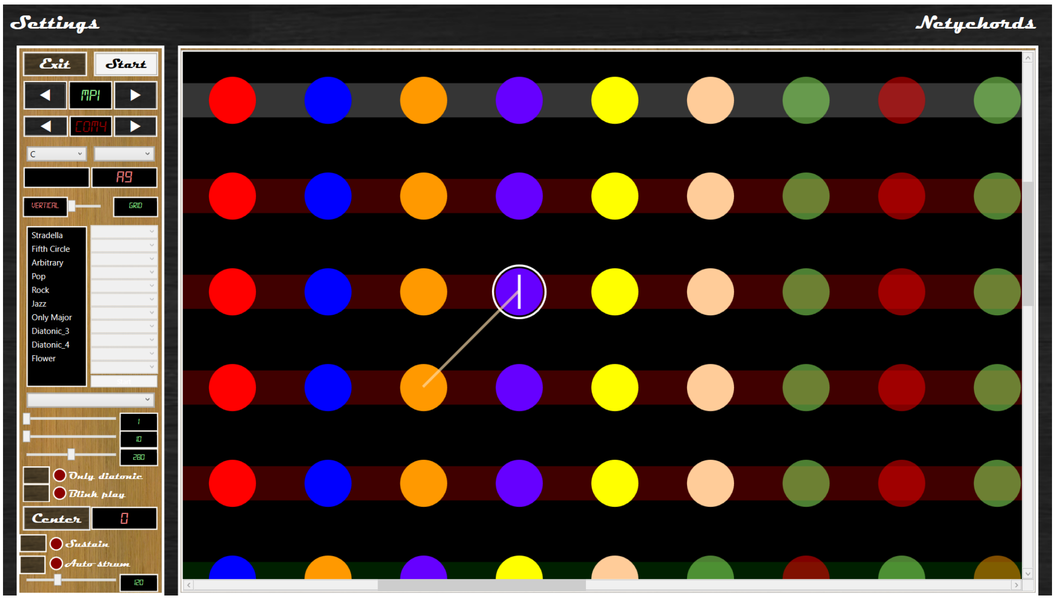
- **Stradella.** It is inspired by the Stradella bass system [245], used in some italian accordions, which arranges the chords using the circle of fifths.

  While each column of keys corresponds to the same root note, each row corresponds to a different family of chords, as shown in Fig. 10.2a. The original Stradella includes 4 chord families. In Netychords these have been extended to 11: major, minor, dominant 7th, diminished 7th, major 7th, minor 7th, dominant 9th, dominant 11th, suspended 2nd, suspended 4th and half-diminished 7th.

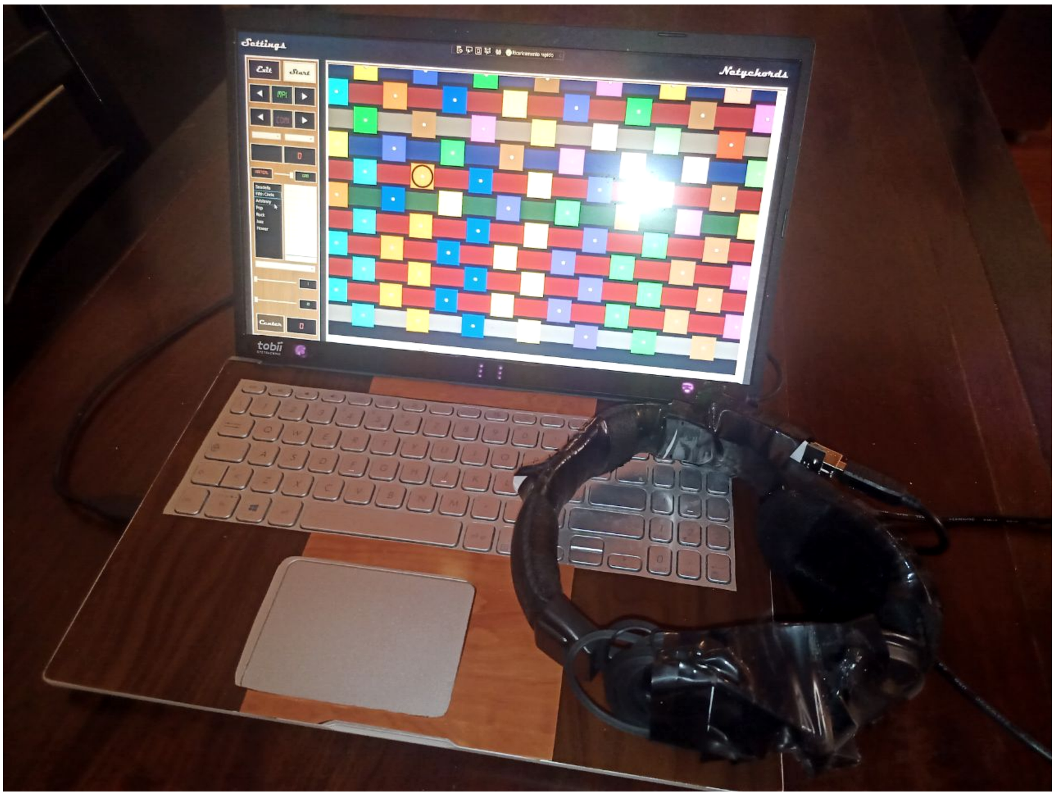
---

[1] Tobii 4C eye tracker on Tobii website: `https://help.tobii.com/hc/en-us/articles/213414285-Specifications-for-the-Tobii-Eye-Tracker-4C`

[2] Netychords on GitHub: `https://github.com/LIMUNIMI/Netychords/`



**(a)** *Netychords interface in the latest version of the instrument*



**(b)** *An early version of Netychords running on a laptop equipped with Tobii 4C and the ad-hoc built head tracker NeeqHT.*

**Figure 10.1.:** *Two images representing Netychords implementation.*

- **Simplified Stradella.** Since the original Stradella bass system is designed for a fingered keyboard, the distance between the keys could be disadvantageous for gaze based interaction. In this simplified version, all the chords from Tab. 10.1 have been grouped into 5 families (major, minor, dominant, diminished and half-diminished). While maintaining the circle of fifths rule for horizontal movements (each key is incremented by seven half-tones from the previous one), the first chord of each row is different according to the chord family. Taking as fundamental the first major chord (for example, a C chord), the minor chords row starts from the VI degree of the major scale (relative minor, therefore in the example an A chord), the dominant sevenths start from the V degree (in our example, a G chord), the diminished rows follows the same arrangement as the major one while the half-diminished rows starts from the VII degree (in the example, a B chord). Fig. 10.2b shows how, with this arrangement, chords belonging to an harmonized diatonic scale are kept close together, resulting in less eye movement required to play musical pieces in a single key. Diatonic scale harmonization is resumed in Tab. 10.2.

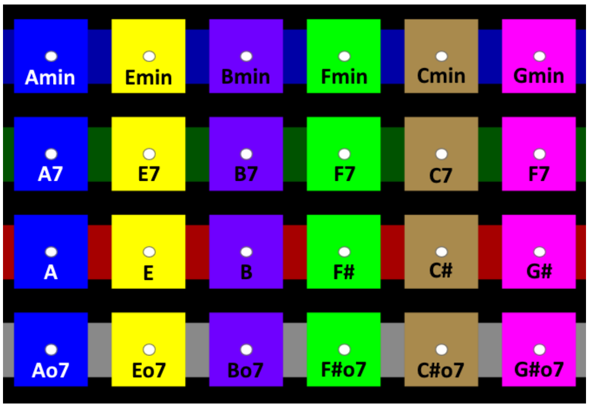
  By removing rows from this layout, three simplified genre-specific presets have been obtained, potentially useful for playing pop, rock and jazz music. In the *jazz* preset, for example, the major and minor chords are replaced with major 6th and minor 6th chords, following the indications of Pino Jodice, jazz pianist [246], thus keeping close the grades listed in Tab. 10.2 (right half). The *rock* preset instead contains only major, minor (thus keeping close the degrees described in the left half of Tab. 10.2), dominant 7th, suspended 2nd and suspended 4th chords.

  The user is also able to create a new custom layout selecting which chord rows to include, and in which order.

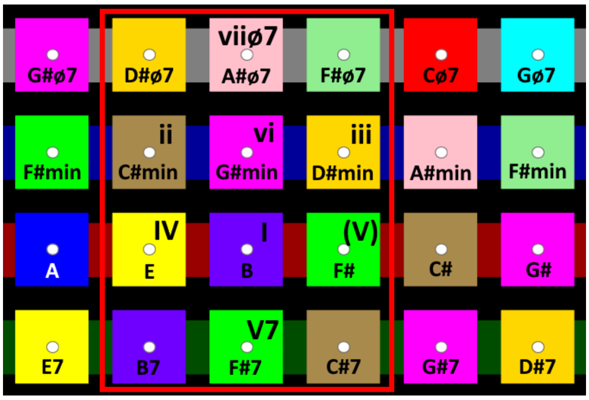
- **Flowerpot.** This layout has a completely different structure from the previous ones. The keyboard is divided into groups of 5 keys arranged in a cross, called *flowers*. A *major flower* is red; the middle key is mapped to a natural major chord, while the other keys are mapped to dominant 7th, major 7th, major 6th and suspended 4th chords. A *minor flower* is blue; the central key is mapped to a natural minor chord, while the other keys are mapped to minor 7th, minor 6th, diminished 7th and half-diminished chords. Flowers are grouped in proximity to each other, obtaining a square grid tessellation without empty spaces. After selecting the tonal center, the central flower will correspond to the I degree of the harmonized diatonic major scale, while the adjacent flowers, arranged in a circle, will cover the other degrees (again according to the scheme shown in Fig. 10.2c). This layout could therefore be practical for playing songs without key changes.

For most of the layouts, keys cannot all be shown within the application window, due to the size of the screen and because too small keys would be difficult to select using gaze pointing. Netychords therefore implements the same autoscrolling system as Netytar (described in Sec. 6.3 and Sec. 9.1), which smoothly moves the fixated key to the center, taking advantage of the “smooth pursuit” capabilities of the eyes [92].

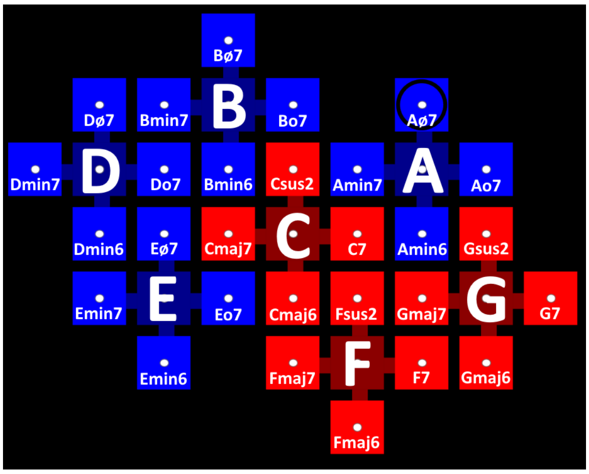
Octave choice is based on "reed" selectors (recalling again the Stradella accordion), to reduce the number of keys drawn on the screen.



**(a)** *A four rows/six chords detail of the Stradella layout implementation. Chord labels are not visible while playing.*



**(b)** *A four rows/six chords detail of the Simplified Stradella layout implementation. Keys belonging to the diatonic harmonization of the B major scale (labels indicate the various degrees) are enclosed in the red square. All the labels are not visible while playing.*



**(c)** *Current implementation of the Flowerpot layout with C as root note. Labels are not visible while playing.*

**Figure 10.2.:** *The three main layouts implemented in Netychords.*

| Suffix | Chord name | Intervals (degrees) | Sample C- chord |
|---|---|---|---|
| - | Major | 1, 3, 5 | C-E-G |
| min | Minor | 1, ♭3, 5 | C-E♭-G |
| maj6 | Major 6th | 1, 3, 5, 6 | C-E-G-A |
| min6 | Minor 6th | 1, ♭3, 5, 6 | C-E♭-G-A |
| maj7 | Major 7th | 1, 3, 5, 7 | C-E-G-B |
| min7 | Minor 7th | 1, ♭3, 5, ♭7 | C-E♭-G-B♭ |
| 7 | Dominant 7th | 1, 3, 5, ♭7 | C-E-G-B♭ |
| o7 | Diminished 7th | 1, ♭3, ♭5, 𝄫7 | C-E♭-G♭-A |
| ø7 | Half-dimin. 7th | 1, ♭3, ♭5,♭7 | C–E♭–G♭–B♭ |
| sus2 | Suspended 2nd | 1, 2, 5 | C-D-G |
| sus4 | Suspended 4th | 1, 4, 5 | C-F-G |
| 9 | Dominant 9th | 1, 3, 5, ♭7, 9 | C-E-G-B♭-D |
| 11 | Dominant 11th | 1, 3, 5, ♭7, 9, 11 | C–E–G–B♭–D–F |

**Table 10.1.:** *Intervals describing each type of chord present in Netychords. Colors reflects the chord families subdivision implemented in the Simplified Stradella layout: major (red); minor (blue); dominant (green); diminished (orange); half-diminished (gray).*

| 3 notes / chord | | 4+ notes / chord | |
|---|---|---|---|
| **Degree** | **Example** | **Degree** | **Example** |
| I | C | I | Cmaj7 |
| ii | Dmin | ii | Dmin7 |
| iii | Emin | iii | Emin7 |
| IV | F | IV | Fmaj7 |
| V | G | V7 | G7 |
| vi | Amin | vi | Gmin7 |
| vii | Bmin | vii$^{ø7}$ | B$^{ø7}$ |

**Table 10.2.:** *Harmonized diatonic major scale pattern used in Netychords. On the left, harmonization with 3 notes per chord; on the right, harmonization with 4 or more notes per chord. Degrees are provided using jazz notation.*

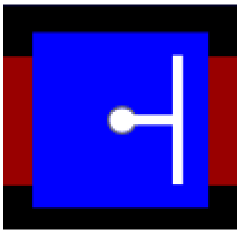


**Figure 10.3.:** *Head position feedback handle (in white).*

### 10.2.2. Strumming

Chords strumming (translated into MIDI note on/off events and velocities) is controlled through head tracking. Here is discussed the strumming modality implemented for Netychords, which to our knowledge has not been previously proposed in the literature of digital musical instruments.

Head rotations on the horizontal axis (yaw) are tracked. A strum occurs when a change in rotation direction is detected. A MIDI velocity value (which in turn determines the resulting sound intensity) is generated as a proportional value to the angle described by the head with respect to a center position (calibrated before playing), where the proportionality factor is adjustable through a slider. In order to actually trigger a new strum (i.e., to generate a MIDI note on event), it is necessary to pass through a central zone called *deadzone*, defined around 0°. The deadzone has an adjustable size, within which changes in direction are not detected.

Visual feedback of head rotation is given directly on the key that is being fixated, through a white handle whose width corresponds to the head rotation angle with respect to the center, which is indicated by a white dot, as depicted in Fig. 10.3.

A known problem to face when designing gaze-based interfaces is that of *Midas Touch* [92], namely the involuntary activation of interface elements (e.g. keys) when these are crossed by gaze trace. The problem has been described in detail in Ch. 6. In previous works, different gaze controlled instruments addressed the problem in different ways. In Netychords the Midas Touch is to an extent solved by design, as notes are triggered by strumming and head movements, rather than fixations. Therefore, notes are let ring until a new strum action is detected regardless of gaze moving to other keys. In addition, neutral areas between occluders are also placed.

To trigger chord stops (pauses), eye blinking is exploited. By closing both eyes for a sufficiently long period (corresponding to 4-5 samples of the eye tracker data stream, adjustable) the sound is stopped.

## 10.3. Design space analysis

In this section some notable properties of the instrument design are discussed, as well as the rationale behind the outlined design choices, through the design space analysis framework described in Ch. 2.

The *simplification* axis indicates which degree of simplification was introduced to enable less

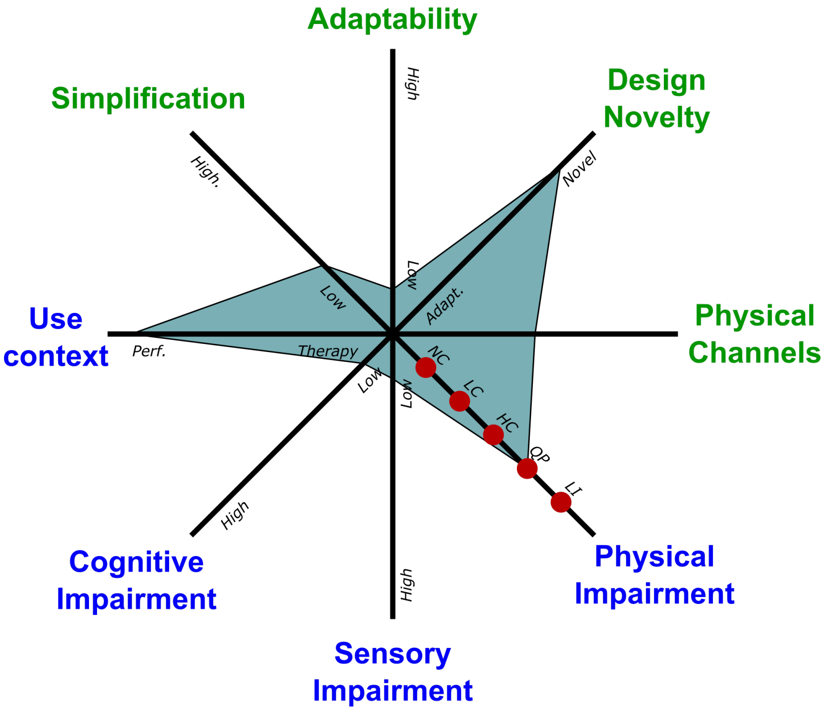


**Figure 10.4.:** *The main design aspects of Netychords, visualized on the dimension space for the evaluation of ADMIs described in Ch. 2.*

trained musicians (or to make up for some deficits) through the introduction of specific aids. In Netychords, no layout allows the selection of the single notes in a chord, differently from most polyphonic acoustic instruments. This however translates into a potential gain in ease of use and could be considered an aid. Apart from this, no other aids have been introduced. Playing aids like strumming temporal quantization could be introduced for facilitating the learning process.

Hence, Netychords' *use context* is more oriented towards performance, as it does not offer particular performance aids. It is a complex instrument requiring training to be mastered.

The *design novelty* axis indicates whether an ADMI resembles the design of a traditional musical instrument or departs from such traditional designs. Although Netychords differs from any acoustic instrument, some similarities can be found with accordions, with particular regard to the octave management system, the reed system and the Stradella Layout.

The usability and the expressivity of an instrument are influenced by the amount of *physical channels* that the user can employ in the interaction. Netychords uses two channels, namely head movement and gaze pointing. This choice is the result of a trade-off between expressivity (as an example, no continuous control on the emitted chord is possible after it has been triggered) and usability (with particular regard to the Midas Touch problem discussed earlier). Although the two interaction channels are largely independent from each other, the eye tracker can only tolerate a certain degree of head rotation. We found that a head rotation angle of about 30 degrees is sufficient to strum at different intensities without compromising gaze detection.

The dimension space clusters categories of impairments into three main axes: physical, cognitive, sensory (or perceptual), and classifies target user groups along these three categories, also considering that physical, sensory, and cognitive impairments are often intertwined due to the multidimensional character of disability. Although Netychords is potentially usable by a quadriplegic user, the head movement required makes it incompatible with the maximum degree of the discrete *physical impairment* scale (*LI*, or Lock-In syndrome, characterized by the possibility of move your eyes only). Instead it falls into the *QP* (quadriplegic Paralysis) level.

Future iterations of the instrument may include the choice for different and new input methods. Blink based strumming could be tested for users with difficulties in rotating the head. This could influence and improve the actually low *adaptability* of the instrument, namely the possibility to adapt to the individual user needs. *Sensory* and *cognitive impairments* are not addressed by Netychords.

## 10.4. Further developments

After the publication of [5] where the instrument is described, further developments of the instrument took place.

The instrument interface and virtual keyboard has been finished so as to be visually more simple and less fatiguing for the eyes. Keys color codes are now compatible with the color scheme proposed by Netytar for keys notation (Tab. 9.1).

A small case study was performed with a young user having hemiplegic paralysis, having therefore a partial motor disability of the upper limbs. The user was unable to control his fingers, but they were partly able to move an arm, albeit with imprecision. A first attempted approach was to introduce an oversized key with which strumming was performed, giving up the possbility to control sound intensity and chord expressiveness. It was however observed that, in addition to not allowing a sufficient level of control, the movement of the arm involved an involuntary and spasmodic movement of the head, which complicated or sometimes compromised gaze detection by the eye tracker. A second approach consisted in making available fully eye-based interaction methods for the instrument. A first method consists in chords strummming through a blink with both eyes. A second method consists in the introduction of an automatic strumming system, with adjustable time. Both methods have been found useful and enabling for the user under study, and could be revised in future versions of the instrument, introducing for example different rhythm patterns for the automatic strumming system.

Some simplifications useful for musical teaching have been introduced following the case study. A learning method for Netychords has not yet been developed (unlike Netytar, for which a learning method has been described in Sec. 9.4). Selectors to reduce the number of keys displayed on the screen have been introduced. New predefined layouts that allow to show only a chords row (e.g. major), or only the seven keys corresponding to the harmonization of a diatonic scale have been introduced. The publication of these updates, after refinement, will be deferred to future articles.

## 10.5. Future work

Future work will be mainly addressed to the evaluation with target users, in order to guide subsequent iteration of the instrument design.

One primary element of evaluation concerns an in-depth study of new key layouts and a comparative study of existing ones. Representing a large amount of chord families on the screen requires a great number of keys. We plan to experiment with separating, among different interaction channels, root note selection and chord type selection. Different head rotation axes (or different angles) could for example correspond to different chords (e.g. major, minor).

The proposed strumming modality also needs thorough evaluation, and may be further extended to allow for more expressive interaction. In the current prototype, chords notes are all played simultaneously while strumming. A method for sequential note strumming (arpeggio) could be implemented, in a similar fashion as for guitar strings, by subdividing the head rotation interval. Strumming can be good for simulating plucked instruments or piano. Continuous intensity detection could be implemented, suitable for playing strings, for example evaluating head rotation velocity for each sample.

Evaluating a musical instrument from an objective point of view is a complex task. The already cited framework from O'Modhrain ([56], Ch. 1.5) proposes a framework for the evaluation of Digital Musical Instruments from the perspective of the various stakeholders (performer/composer, designer, manufacturer and audience), consisting in a set of parameters to be evaluated. Vamavakousis and Ramirez [62], as an example, implemented this framework for the evaluation of The EyeHarp from the point of view of the audience through questionnaires submitted to the attendees of a concert. As O'Modhrain highlights, being able to trace a link between the performer's motion and the perceived sound is a very important element for the audience to appreciate a live performance, and this is a concern for gaze based interfaces since movements are very subtle. Head rotation in Netychords could be a way to convey the expressive intention.

Interaction in Netytar has been evaluated quantitatively, also from a precision and accuracy point of view, through the recording of simple musical exercises in order to measure timing errors and number of wrong notes. The experiment, which compared Netytar to another instrument in the state-of-the-art, is described in Sec. 9.7. We feel however that a comparative evaluation between Netychords and another instrument could be difficult to make since all the similar instruments found in literature and listed in Sec. 10.1 offer a different degree of control on chords performance.

### 10.5.1. Testing

The COVID-19 pandemic has so far prevented testing Netychords with target users during its first development period. This section thus concludes by discussing a test procedure for Netychords we intend to carry out in the future. This is similar to the one used for Netytar's evaluation (leaving aside the comparison phase). A sample of at least 25-30 individuals, possibly with musical experience, will be recruited.

- A *training phase* of at least 20 minutes will provide a minimum of familiarity with the instrument and the proposed interaction methods;
- A *practical test* will involve the performance of musical exercises (or songs), and the performance will be recorded as a MIDI track and analyzed later. We intend to detect elements such as error rates, strumming and chord change speed, as well as the flexibility of the various layouts in allowing chord changes between distant keys;
- A *qualitative test* will include a questionnaire with general questions on the usability of the system. Elements such as perceived fatigue, degree of naturalness and simplicity of interaction and interface clarity will be detected. Case studies should be also carried out with musicians having physical (quadriplegic) disabilities.

Another session could be devoted solely to testing the proposed head-based interaction method. While the movement precision and stability have already been discussed in other experiments, such as the evaluation experiment described in Ch. 4, it would be useful to detect through recorded exercises the head's rhythmic capabilities, namely the relationship between precision and speed/frequency of head strums.

# 11

# Resin

This chapter describes Resin, an HeaDMI which exploits two interaction channels to control musical performance parameters: head movements and the shape of the vocal tract, detected through the corresponding acoustic resonances. The instrument has been described in the article [6]. Like Netytar (described in Ch. 9) and Netychords (described in Ch. 10), Resin is a MIDI controller.

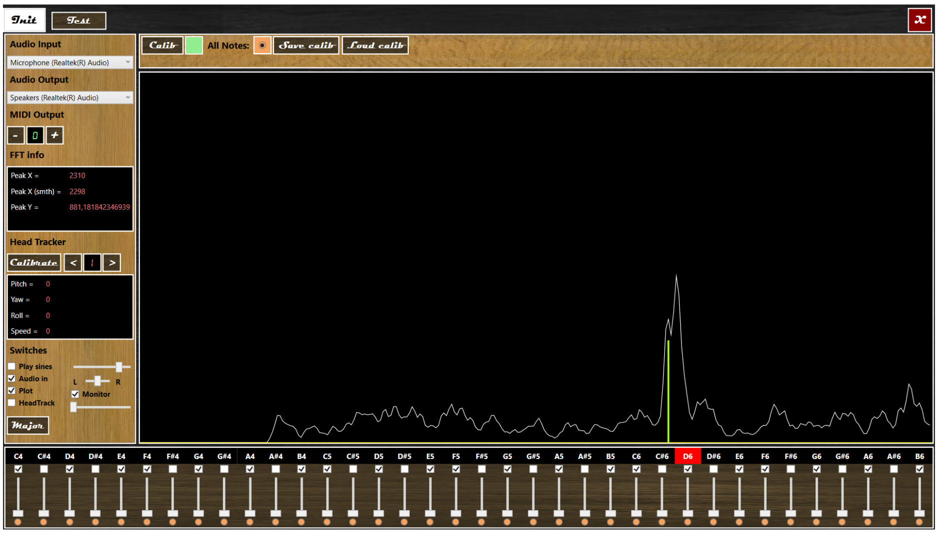
The two physical interaction channels which the instrument exploits are still so far underexplored in ADMIs design. Head movement along the horizontal axis (yaw) is used to control notes onset and dynamics (MIDI *Pressure*, *Velocity*, *Note On* and *Note Off* parameters), while resonances within the vocal tract are used to control the note *Pitch*. Resin consists of a software and a hardware part, the latter consisting of both DIY and pre-built components. Its software interface is depicted in Fig. 11.1.

Resin software is available for download from its GitHub Repository,[1] licensed under the Open Source GNU GPLv3 license.

In this chapter the structure of the instrument is discussed, from both hardware and software points of view. Sections are subdivided as follows: Sec. 11.1 provides a comparison with related works found in the state of the art; Sec. 11.2 describes the two employed physical interaction channels and their mapping to musical performance parameters; Sec. 11.3 discusses the instrument implementation. Feature extraction algorithms for both channels are explained, particularly focusing on the vocal tract resonances interaction paradigm; Sec. 11.4 provides an evaluation of design choices for the instrument in the ADMIs context through the dimension space analysis tool described in Ch. 2; finally, in Sec. 11.5 a discussion on future works and improvements to be included in the next iterations of the instrument is provided.

---

[1] Resin's repository on GitHub: `https://github.com/LIMUNIMI/Resin`



Audio Input
Microphone (Realtek(R) Audio)
Audio Output
Speakers (Realtek(R) Audio)
MIDI Output
- 0 +
FFT info
Peak X = 2310
Peak X (smth) = 2298
Peak Y = 881,181842346939
Head Tracker
< 1 >
Pitch = 0
Yaw = 0
Roll = 0
Speed = 0
Switches
☐ Play sines
☑ Audio in
☑ Plot
☐ HeadTrack
☑ Monitor

**Figure 11.1.:** *Resin's graphical user interface.*

## 11.1. Related work

Regarding Resin's vocal interaction paradigm, no similar systems have been found in the literature, to the best of the authors' knowledge.

In Ch. 3 a list of possible channels suitable for the development of HeaDMIs is reported. The use of vocal tract resonances is not included in the above list, but can be associated with other physical channels. It can be stated that the internal shape of the oral cavity, which is the terminal part of the vocal tract, varies in accordance with tongue movements. Various systems have been tested to detect the tongue position for interaction purposes.

An analogy can be then traced with tongue controlled interfaces. In Tongue Music [132], an interface part of a sonification experiment, the detection happens through hall-effect sensors coupled with magnets. In Niikawa's Tongue Controlled Electro-Musical Instrument (described in Sec. 5.1.2) the tongue presses some buttons positioned on the palate to play chords. Other detection methods, both related to musical contexts and other interaction contexts, are described in Sec. 3.2.2. Those include the use of textile pressure [129], ultrasound [131, 247], magnetoresistive [126] or optical sensors [248, 249].

Since Resin exploits pitch detection techniques, an analogy can be traced with Imitone (described in Sec. 5.1.11). It is a software interface able to detect the pitch of voice or whistle through a microphone, and convert the input in MIDI messages in real time.

An analogy can be traced also with Silent Speech Interfaces (SSI). These are interfaces aimed at recognizing facial and buccal movements in order to reproduce speech without the need for the user to emit any sound. According to [122], such interfaces usually exploit electromagnetic articulography, electromyography, ultrasounds, microphones, electroencelography or neural cortex implants.

A comparison could also be drawn with interfaces that exploit mouth shape detection to control sound filters. The *Talkbox* is a common analog electric guitar effect that consists of a speaker channeling the guitar sound into a rubber tube, having the other end placed in the musician's mouth. The sound comes out of the mouth to be picked up by microphone. Several interfaces, such as Mouthesizer [90], use cameras and computer vision techniques to detect mouth movement and use it to control sound filters. Eye Conductor (described in Sec. 5.1.14) is an HeaDMI which exploits the same paradigm.

Head tracking techniques related to musical performance have already been extensively described in Sec. 3.2.3 and were already exploited to control chord strumming dynamics in Netychords, as described in Sec. 10.2.2.

## 11.2. Interaction and mapping

In this section the two physical interaction channels exploited by the instrument are discussed: vocal tract resonances and head rotation.

### 11.2.1. Vocal tract resonances

The process responsible for the production of vocalized sounds in humans is often modeled by considering two main components. This can be exemplified in the practice of singing. In the first one, vocal cords vibrate creating a "pulse train", whose frequency defines the voice pitch. In the second one, various components of the vocal tract, such as the mouth, act as a filter enhancing some frequencies in the spectrum of the vocalized signal. The corresponding resonance frequencies (formants) characterize different vowels [107]. Some singing styles, such as tuvan throat singing, exploit the resonances (or overtones) inside the vocal tract to combine them into complex melodies [250]. Similarly, vocal tract and mouth can be shaped into a Helmholtz resonator to whistle [115].

In Resin, the vocal tract is stimulated through a synthesized sound conveyed to an ad-hoc built hardware component, which we will refer as *sound tube*. This consists of a speaker, properly muffled on the sides, which emits a synthesized sound (which we will refer to as *sine pad*) into a rubber tube. The musician puts the tube end in their mouth and grasps it with their teeth as they play, keeping their mouth slightly open. The sound, produced by Resin's software, is a linear combination of different sinusoidal components, whose frequencies are tuned to successive halftones in the equal temperament (A5 tuned to 440Hz). The interface allows the musician to select which notes (which we will refer as *playable notes*) are included. By varying the mouth shape, some of these sinusoidal components resonate louder. We've seen from our tests that an adult man's mouth, for example, can resonate notes approximately between the fourth and sixth octave (from C4 to C6). A small Lavalier microphone is placed in the mouth next to the sound tube, and picks up the filtered sound. Resin's software therefore recognizes the resonating frequency/note. Resin's mouthpiece is depicted in Fig. 11.2.

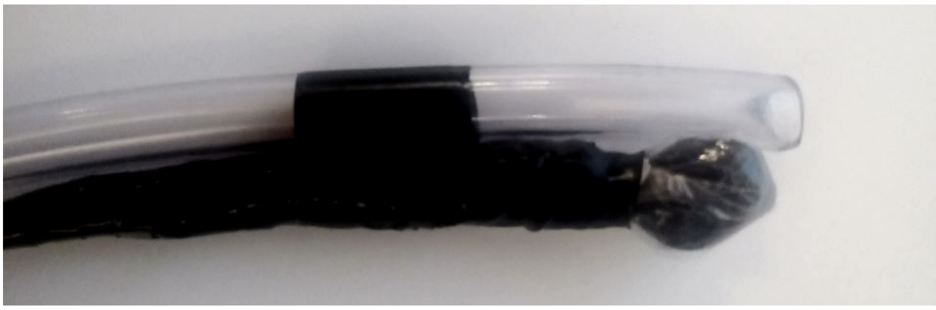


**Figure 11.2.:** *Resin's mouthpiece, consisting of the sound tube and the Lavalier microphone.*

### 11.2.2. Head tracking

Head movement is detected in a range of 40 degrees [-20°; +20°] in the horizontal plane. Head movements are used to perform attack and release actions on the instrument. A new note (attack) is triggered when an inversion of the head motion is detected, prior passing through the central position (0°).

Two different operation modes can define strum intensity. It can be chosen to be proportional to the distance from the center at the inversion point, when the strum action is triggered, or to the average movement speed in the previous instants. Movement speed determines also the *channel pressure*. The latter is calculated as the distance between head position in the current sample and in the previous one, filtered by an exponentially moving average filter.

### 11.2.3. Performance logic

Musical performance takes place in the following way. The system continuously detects the resonating note. An indicator on screen highlights the detected note $n_r$; as soon as a head strumming action is detected, a MIDI *note-on* message is sent for $n_r$, with a MIDI velocity determined by one of the two approaches for strum intensity mentioned above; the note remains on-set even if the detected resonating note changes, until a subsequent strumming action occurs. In this case, the old $n_r$ is stopped by a *note-off* event, and is replaced by the new $n_r$; *channel pressure* varies continuously in proportion to head movement speed, contributing to note dynamics.

## 11.3. Implementation

We now discuss the actual implementation of Resin, at its current version. Interaction, sound processing and generation are summarized in Fig. 11.3.

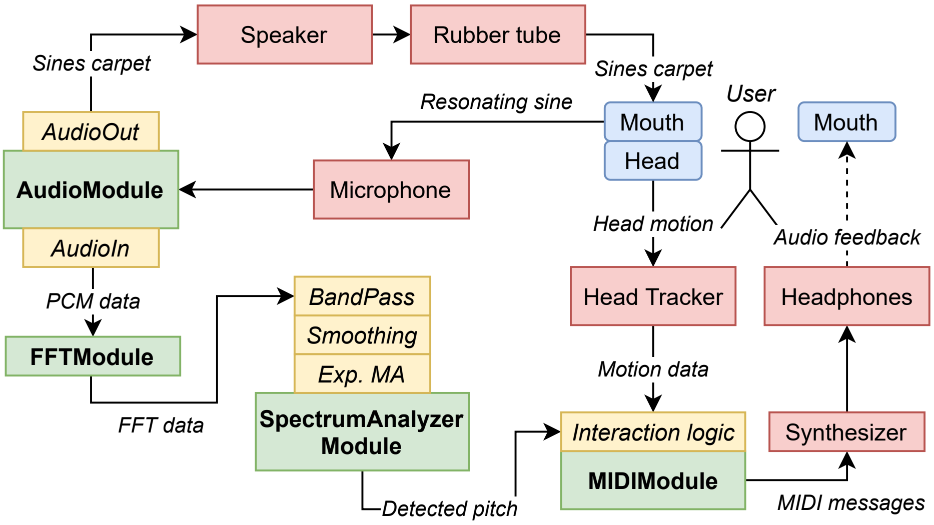


**Figure 11.3.:** *Functional diagram for Resin*

### 11.3.1. Hardware

The employed microphone is a Røde smartLav+[2], with a frequency response of 20Hz-20kHz and a sensitivity of -32.0dB re 1 Volt/Pascal, equipped with a pop filter. The microphone is enclosed in a cellophane layer to prevent water infiltration.

The system was tested using an Alesis iO2[3] sound card, plugging the microphone into an XLR input via a Røde VXLR+ adapter, capable of converting the +48 V phantom power supplied by the sound card into the 3-5 V voltage required by the microphone. The Alesis iO2 sound card is able to sample sound at a frequency 48 KHz with 24 bits resolution.

The hardware required to operate Resin includes the low cost DIY head tracker *NeeqHT* described in Sec. 8.2. The head tracker is capable to translate acceleration into absolute position data.

### 11.3.2. Software, audio generation and processing

Resin software is coded in C# language, using the Windows Presentation Foundation graphical framework, part of the .NET 4.8 framework. Components of the diagram in Fig. 11.3 resemble the main classes and components structure implemented through Object-Oriented Programming paradigms. The *NeeqDMIs* library described in Sec. 7 was used as a basis for the development of the instrument.

Prior to playing, the user selects which are the *playable notes*, using the interface. Therefore the *AudioModule* class generates the corresponding sine pad (see Sec. 11.2.1) where sinusoidal frequencies are the fundamental frequencies of the selected notes, and all the components have the same amplitude. The resulting sound is conveyed to the performer's mouth through the sound tube.

---

[2] Rode Smartlav+ on Rode website: `https://it.rode.com/microphones/smartlav-plus`

[3] Alesis iO2 sound card on Alesis website: `https://www.alesis.com/products/legacy/io2`

After the musician's mouth has filtered the sound, the *AudioModule* class receives the PCM signal from the microphone. The *FftModule* class performs then a Fast Fourier Transform of an audio buffer. In accordance with the sound card specifications, the FFT is performed considering a 48 KHz/24 bit sampled signal (according to the capabilities of the sound card used in the tests) and an audio buffer of 43 ms, resulting in an array of 2064 samples (= $48000KHz \cdot 0.043ms$) for each sampling cycle. After applying a Hamming Window, the audio buffer is zero-padded to 4096 values, in order to improve frequency resolution. This results in a greater distance between the sinusoidal components of the sound spectrum, at the expense of a greater "spreading" of the energy for each spectrum component. The number of FFT bins obtained is therefore 2048 (= 4096/2).

Some filters are then applied to the resulting FFT magnitudes array. A bandpass filter clears the frequency bins placed outside the range of the playable notes selected by the user. A smoothing filter then performs spectral smoothing. Each element of the output spectrum is calculated as in Eq. (11.1):

$$O_i = \frac{I_{i-1} + I_i + I_{i+1}}{3}, \tag{11.1}$$

where $O$ and $I$ are respectively the output and input short-time spectra, while $i$ denotes the $i$-th bin.

An exponentially moving average filter is applied to successive short-time spectra, causing the energy of each bin to vary more smoothly over time, in order to prevent sudden oscillations due to noise. The filter is in the form described by Eq. (11.2):

$$O_i(t) = \alpha \cdot I_i(t) + (1-\alpha) \cdot O_i(t-1), \tag{11.2}$$

where $t$ is the discrete time of the current sample, *t-1* the time of the previous sample, while $\alpha$ is an arbitrary constant set at 0.9.

The *SpectrumAnalyzerModule* class then determines the pitch of the resonant note. Each playable note $n$ is associated to the bin $B(n)$ where its fundamental frequency falls. The spectrum is divided into different groups of bins $G(n)$, each centered around the corresponding $B(n)$. Specifically, the upper and lower boundaries of $G(n)$, defined as $B_U[G(n)]$ and $B_L[G(n)]$, respectively, are defined by Eqs. (11.3) and (11.4):

$$B_U[G(n)] = \frac{B(n) + B(n+1)}{2}, \tag{11.3}$$

$$B_L[G(n)] = \frac{B(n) + B(n-1)}{2}. \tag{11.4}$$

Fig. 11.4 graphically summarizes this groups division approach.

The *SpectrumAnalyzerModule* then proceeds by determining the mean quadratic spectral energy

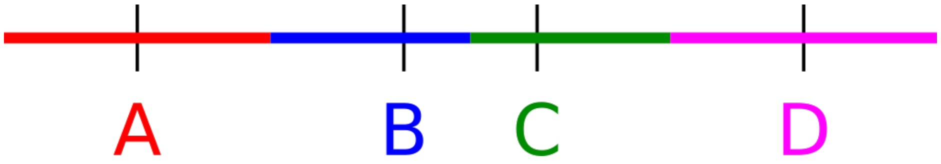


**Figure 11.4.:** *FFT spectrum division (horizontal axis) for notes A, B, C and D. Different bin groups are highlighted with different colors. Since B and C have a distance of a single halftone, their associated group is smaller.*

$E(n)$ for each $G(n)$, defined as:

$$E(n) = \frac{1}{N_{bin}(G(n))} \sum_{b \in G(n)} E(b)^2. \tag{11.5}$$

The estimated resonant note is the one corresponding to the group with the greatest energy. Finally, the *MIDIModule* generates the corresponding MIDI messages.

### 11.3.3. Graphical User Interface

The *FFTPlot* class draws the short-time spectrum together with two indicators Fig. 11.1. A bar denotes the single bin with the highest energy, while a second bar indicates the $G(n)$ with the highest energy. Several selectors allow to select the audio input/output channels for the sound tube. The lower part of the interface highlights the detected pitch. Some buttons and sliders allow to select the playable notes, as well as to perform a manual calibration of the volume of each note.

## 11.4. Design space analysis

A further insight on Resin's characteristics can be given by assessing the instrument through the design space tool described in Ch. 2. Fig. 11.5 shows a graphical rendering of the analysis conducted with such framework.

The *adaptability* of the instrument to various needs of different users is very low at the current version. The use of vocal tract resonances and head rotation is mandatory to control Resin and mappings do not offer any degree of customization, with the only exception of head rotation which could be mapped to note dyamics control in two different ways, as stated in Sec. 11.2.2.

*Design novelty* in Resin is particularly high. Its design does not imitate any traditional acoustic instrument. Vocal tract resonances to our knowledge have never been exploited to control a digital musical instrument, and even if head rotation have been exploited before, the proposed mapping and interaction paradigm is novel.

The employed *physical channels* are two, which is arguably on the average compared with other musical instruments. The value on the graph has been set to a middle position.

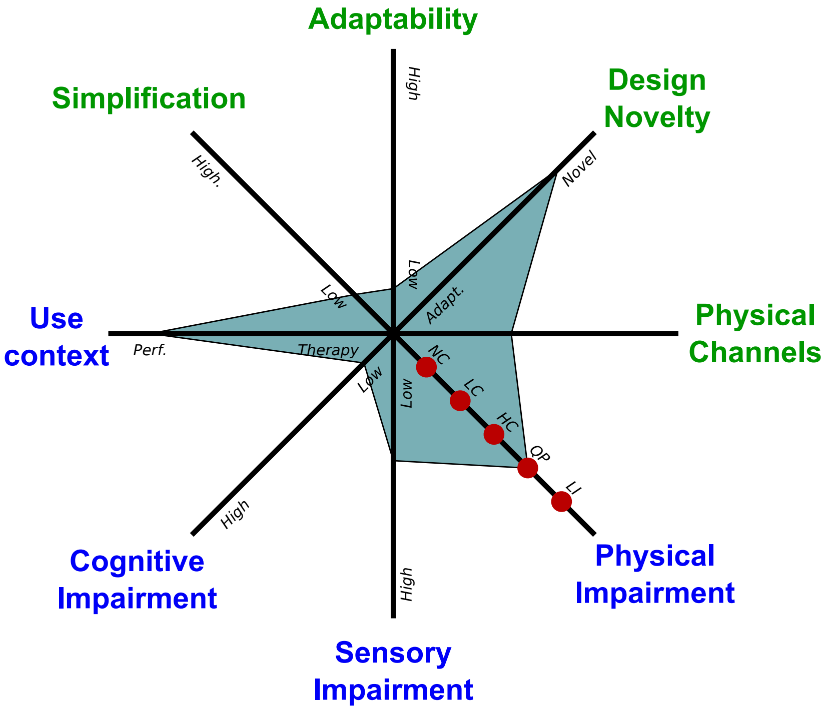


**Figure 11.5.:** *Resin's design space evaluation using the design space analysis tool described in Ch. 2*

With regards to the addressed *physical impairment*, as the other HeaDMIs covered in this thesis, the addressed target group are users with quadriplegic paralysis.

Although Resin was not designed to address any specific *sensory impairment*, it has to be noticed that the use of sight is not required to play. A blind user can potentially use Resin without any particular disadvantage compared to a sighted one, except for not being able to perceive the graphical feedback offered by the GUI to check if the vocal tract resonances are in tune with the desired note.

*Cognitive impairments* are not addressed by Resin. It could be difficult to introduce cognitive aids in the instrument, which learning curve could be particularly steep.

Resin is thought for a performance *use context*, according to the definition proposed in Sec. 1.2. It requires training to be mastered, and we suspect that the entry barrier for musicians with no prior knowledge of the instrument and low intonation skills could be high.

The instrument does not include any particular aid thus the degree of *simplfication* is very low. It could be however stated that reducing the number of playable notes could result in the instrument being more easy to play, requiring the user to be less in tune and precise while shaping their mouth to produce the required resonances.

## 11.5. Discussion and future works

Accuracy in note recognition already allows to play melodies with different complexity levels but could still be improved. Sinusoidal energy spreads through multiple FFT bins, due to the algorithm's nature, resulting in some imperfections. This can cause the energy of a note to "invade" an adjacent note's associated group of bin $G(n)$.

In the current version of Resin, each sine wave in the pad has the same frequency as its associated MIDI note. This results in a natural interaction: the mouth shape associated with each note is similar to the one required to whistle that note. In an attempt to improve detection precision, we tested the use of larger distances between sine frequencies (thus mapped to notes having different frequencies). However this solution resulted in the interaction being less natural thus more difficult, so it was abandoned. Instead, we opted for the possibility of choosing which notes are "playable" (e.g. only those belonging to the C major scale, or pentatonic scale). With a smaller set of notes, each $G(n)$ (group of bins associated with each note) become larger, thus improving recognition.

The speaker/tube/microphone system does not have a flat frequency response. To (at least partially) mitigate this problem, an automatic calibration system was implemented, which proceeds to "flatten" the detected spectrum by adjusting the volume of each sinusoidal component while not performing. Sliders placed in the bottom part of the interface also allow to adjust the volume of each sine manually.

The sines pad could be audible from the outside and cause nuisance. While it is possible to greatly reduce its volume while preserving the instrument functionality, below a certain value the system may be sensitive to ambient noises, as well as to the MIDI synthesizer feedback. This could be improved by using a more directional microphone, or headphones.

The head-based "strumming" metaphor implemented in Resin may be suitable for simulating some types of instruments such as strings, while being less suitable for others.

The entry barrier in learning Resin could be very high. A user able to whistle in tune is probably facilitated in learning the instrument. However, the initial learning curve could make it very difficult to evaluate the instrument objectively (e.g. by recording the performance of musical exercises) with an adequate number of testers.

Planned future developments include an evaluation of the instrument through case studies and the performance of pre-established exercises, as well as refinement and tuning of the detection algorithms for both vocal resonances and head movements. Evaluating the pitch detection algorithm could be a difficult task, since it is difficult to distinguish system imprecision from lack of user skill (e.g. intonation). Apart from this issue, an evaluation could be carried out using similar modalities as those described in Sec. 10.5.1 for future evaluation of Netychords.

Lastly, a more complete graphical user interface should be developed, introducing support for eye tracking, or cursor movement and aiming via head tracking to make the software fully and autonomously operable by the target user group.

# Conclusions

This thesis provided and explored elements of design, development and evaluation of accessible digital musical instruments for quadriplegic users, defining and focusing on HeaDMIs, a particular subset of skill-based instruments dedicated to live and real-time performance.

Part I provided an overview of theoretical aspects. After framing HeaDMIs in the relevant context (Ch. 1) and providing an ADMIs design evaluation tool (Ch. 2), a complete review of physical interaction channels available from the neck upwards was provided (Ch. 3), along with analysis of their use in general HCI applications and expected performance parameters. An experiment for the evaluation of the latter was proposed (Ch. 4). A review of the state of the art in HeaDMIs was provided (Ch. 5), analyzing design, mapping strategies and future perspectives. Finally, theory fundamentals for the design of gaze-based musical interfaces were reviewed (Ch. 6). Part II described HeaDMIs development and evaluation work. A software library for the prototyping and development of software instruments was introduced (Ch. 7), as well as two open-source hardware sensor peripherals for the detection of breath and head rotation (Ch. 8). The realization of three HeaDMIs was described (Ch. 9, 10 and 11), providing an overview on their implementation, design characteristics and choices, interaction methods and mapping strategies. For one of them an empirical evaluation and a learning method was provided.

The main achievements of this work include the analysis of uncommon and underexplored interaction channels suitable for people with limiting motor disabilities, alternative to the use of hands. Interaction modalities for such channels have been studied and developed, demonstrating that they can be successfully used for musical interfacing. The achievable degree of control, precision and expressiveness was demonstrated by an experimental evaluation of three channel-sensor pairings, as well as the realization of three skill-based digital musical instruments which exploit these channels to obtain musical performance in real time, in a similar fashion to a traditional acoustic instrument. A dimension space analysis framework provided a tool for framing these instruments in the context of ADMIs in general, as well as providing further insights on their design choices. Focus was given on economic affordability of the employed sensor peripherals and openness of software and hardware projects. It is the author's hope that such achievements will stimulate the growth of the research niche dedicated to accessible instruments as well as community development by providing inspiration and design elements.

## Current and future directions

As stated in the Preface and in the respective chapters, the research project was adversely affected by the COVID-19 pandemic, which prevented an objective evaluation of two of the developed instruments (Netychords and Resin, respectively in Ch. 10 and Ch. 11) through experiments and case studies in the field, involving users with quadriplegic disabilities. Although an evaluation experiment has been carried out for Netytar (Ch. 9), case studies involving the targeted user group can lead to greater insights into their actual unique and diverse needs. In the relevant chapters, methodologies have been proposed to perform such tests, which could be carried out in the future. One such case study is currently being carried out in collaboration with the Dept. of Education and Psychology of the University of Aveiro, in which Netychords is being used by a child with cerebral palsy.

Subsequent participatory design and development could be carried out to increase instruments adaptability. Case studies and community works involving the target group can help disseminate knowledge to end users. To this end, we are establishing a collaboration with the Spazio Vita Association at the Niguarda Hospital in Milano[4] to develop a joint project on a music performance laboratory involving quadriplegic patients.

The experimental evaluation test for Netytar in Ch. 9, as well as the interaction channels evaluation experiment described in Ch. 4 were carried out with a limited number of users. We plan to perform a new iteration of the first with more testers, in order to better assess the advantages provided by our keys layout solution. In the second, only some of the channels suitable for HeaDMI development described in Ch. 3 have been evaluated. New experiments could be designed to evaluate them as well as other musical interaction performance parameters described in Tab. 3.1.

The software library *NeeqDMIs* described in Ch. 7 requires refinement as well as the addition of classes useful for interfacing with more sensor peripherals, useful for detecting different physical interaction channels. Some of these could contribute to the expansion of the open-source hardware peripherals collection described in Ch. 8. A more complete evaluation of the latter could be carried out.

The developed instruments offer room for extensions, customizations and integrations. Different interaction channels can be introduced to ensure a greater degree of adaptability to the needs of the musician. Some work has already begun to introduce them in Netytar and Netychords. For the former, fully ocular interaction methods, which do not require the use of the breath, are being designed. New mapping strategies and layouts can be explored, to ensure a more complete control over different parameters of musical expressiveness (e.g. vibrato and glissando). Teaching tools and exercise automation can be introduced for all instruments, expanding the potential audience to include music novices. Given their software nature, a web porting of the musical interfaces could be developed to provide greater portability and version control.

The offered design framework can be the basis for the realization of new HeaDMIs which make use of other interaction channels available to quadriplegic users: as indicated in Ch. 5 some are under-exploited. The introduction of pitch control through the detection of resonances within the upper vocal tract in Resin (Ch. 11) demonstrates that new interaction channels or alternative reworkings of already known ones can be explored.

---

[4] Spazio Vita Niguarda's website: *https://spaziovitaniguarda.it/*

# Bibliography

[1] Nicola Davanzo and Federico Avanzini. “Hands-Free Accessible Digital Musical Instruments: Conceptual Framework, Challenges, and Perspectives”. In: *IEEE Access* 8 (2020), pp. 163975–163995 (cit. on p. iii).

[2] Nicola Davanzo and Federico Avanzini. “A Dimension Space for the Evaluation of Accessible Digital Musical Instruments”. In: *Proc. 20th Int. Conf. on New Interfaces for Musical Expression (NIME ’20)*. NIME ’20. July 2020 (cit. on pp. iii, 19).

[3] Nicola Davanzo and Federico Avanzini. “Experimental Evaluation of Three Interaction Channels for Accessible Digital Musical Instruments”. In: *Proc. ’20 Int. Conf. on Computers Helping People With Special Needs*. Online Conf.: Springer, Cham, Sept. 2020, pp. 437–445 (cit. on pp. iii, 41).

[4] Nicola Davanzo and Federico Avanzini. “A Method for Learning Netytar: An Accessible Digital Musical Instrument:” in: *Proceedings of the 12th International Conference on Computer Supported Education*. Prague, Czech Republic: SCITEPRESS - Science and Technology Publications, 2020, pp. 620–628 (cit. on pp. iii, 103).

[5] Nicola Davanzo, Matteo De Filippis, and Federico Avanzini. “Netychords: An Accessible Digital Musical Instrument for Playing Chords Using Gaze and Head Movements”. In: *In Proc. ’21 Int. Conf. on Computer- Human Interaction Research and Applications (CHIRA ’21)*. Online conf., 2021 (cit. on pp. iii, 125, 134).

[6] Nicola Davanzo and Federico Avanzini. “Resin: A Vocal Tract Resonances and Head Based Accessible Digital Musical Instrument”. In: *Proceedings of the 2021 AudioMostly Conf.* Trento, Italy (online conf.), Sept. 2021 (cit. on pp. iii, 137).

[7] Nicola Davanzo, Piercarlo Dondi, Mauro Mosconi, and Marco Porta. “Playing Music with the Eyes through an Isomorphic Interface”. In: *Proc. of the Workshop on Communication by Gaze Interaction*. Warsaw, Poland: ACM Press, 2018, pp. 1–5 (cit. on pp. iii, 51, 79, 81, 103, 118).

[8] Andrew Crossan, Mark McGill, Stephen Brewster, and Roderick Murray-Smith. “Head Tilting for Interaction in Mobile Contexts”. In: *Proc. 11th Int. Conf. on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI ’09. Bonn, Germany: Association for Computing Machinery, Sept. 2009, pp. 1–10 (cit. on p. 3).

[9] Raymond A. R. MacDonald. “Music, Health, and Well-Being: A Review”. In: *International Journal of Qualitative Studies on Health and Well-being* 8 (Aug. 2013), 10.3402/qhw.v8i0.20635 (cit. on p. 3).

[10] Rudi Črnčec, Sarah J. Wilson, and Margot Prior. “The Cognitive and Academic Benefits of Music to Children: Facts and Fiction”. In: *Educational Psychology* 26.4 (Aug. 2006), pp. 579–594 (cit. on p. 4).

[11] Henkjan Honing, Olivia Ladinig, Gábor P. Háden, and István Winkler. "Is Beat Induction Innate or Learned? Probing Emergent Meter Perception in Adults and Newborns Using Event-Related Brain Potentials". In: *Annals of the New York Academy of Sciences* 1169.1 (2009), pp. 93–96 (cit. on p. 4).

[12] Eugenia Costa-Giomi. "Effects of Three Years of Piano Instruction on Children's Academic Achievement, School Performance and Self-Esteem". In: *Psychology of Music* 32.2 (Apr. 2004), pp. 139–152 (cit. on p. 4).

[13] Sandra L. Siedliecki and Marion Good. "Effect of Music on Power, Pain, Depression and Disability". In: *Journal of Advanced Nursing* 54.5 (2006), pp. 553–562 (cit. on p. 4).

[14] Kate E. Williams, Donna Berthelsen, Jan M. Nicholson, Sue Walker, and Vicky Abad. "The Effectiveness of a Short-Term Group Music Therapy Intervention for Parents Who Have a Child with a Disability". In: *Journal of Music Therapy* 49.1 (Mar. 2012), pp. 23–44 (cit. on p. 4).

[15] Karette Stensæth. ""Musical Co-Creation"? Exploring Health-Promoting Potentials on the Use of Musical and Interactive Tangibles for Families with Children with Disabilities". In: *International journal of qualitative studies on health and well-being* 8 (Aug. 2013), p. 20704 (cit. on p. 4).

[16] Alexa Sheppard and Mary C. Broughton. "Promoting Wellbeing and Health through Active Participation in Music and Dance: A Systematic Review". In: *International Journal of Qualitative Studies on Health and Well-being* 15.1 (Jan. 2020), p. 1732526 (cit. on p. 4).

[17] Martina de Witte, Anouk Spruit, Susan van Hooren, Xavier Moonen, and Geert-Jan Stams. "Effects of Music Interventions on Stress-Related Outcomes: A Systematic Review and Two Meta-Analyses". In: *Health Psychology Review* 14.2 (June 2020), pp. 294–324 (cit. on p. 4).

[18] Adnan Marquez-Borbon and Paul Stapleton. "2015: Fourteen Years of NIME: The Value and Meaning of 'Community' in Interactive Music Research". In: *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression*. Ed. by Alexander Refsum Jensenius and Michael J. Lyons. Current Research in Systematic Musicology. Cham: Springer International Publishing, 2017, pp. 465–481 (cit. on p. 4).

[19] Davide Andrea Mauro, Federico Avanzini, Adriano Baraté, Luca Andrea Ludovico, Stavros Ntalampiras, Smilen Dimitrov, and Stefania Serafin. "Sixteen Years of Sound and Music Computing: 17th Sound and Music Computing Conference, SMC 2020". In: *Proceedings of the 17th Sound and Music Computing Conference*. Ed. by Simone Spagnol and Andrea Valle. Proceedings of the Sound and Music Computing Conferences. Axea sas/SMC Network, 2020, pp. 3–10 (cit. on p. 4).

[20] Alexander Refsum Jensenius and Michael J Lyons. *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression*. Springer, 2017 (cit. on p. 5).

[21] Patrick McGlynn. "Interaction Design for Digital Musical Instruments". PhD thesis. Maynooth: National University of Ireland, May 2014 (cit. on pp. 5, 11, 16).

[22] Mark T. Marshall. "Physical Interface Design for Digital Musical Instruments". PhD thesis. Montreal, QC, Canada: McGill University, Apr. 2008 (cit. on p. 5).

[23] Erich M. von Hornbostel and Curt Sachs. “Classification of Musical Instruments: Translated from the Original German by Anthony Baines and Klaus P. Wachsmann”. In: *The Galpin Society Journal* 14 (1961), pp. 3–29 (cit. on p. 5).

[24] Eduardo Reck Miranda and Marcelo Wanderley. *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard*. 1st edition. Middleton, Wis: A-R Editions, Inc., July 2006 (cit. on pp. 5, 8, 13, 19).

[25] Thor Magnusson and Enrike Hurtado. “2007: The Acoustic, the Digital and the Body: A Survey on Musical Instruments”. In: *A NIME Reader*. Ed. by Alexander Refsum Jensenius and Michael J. Lyons. Vol. 3. Cham: Springer International Publishing, 2017, pp. 317–333 (cit. on p. 6).

[26] Axel Mulder. “Virtual Musical Instruments: Accessing the Sound Synthesis Universe as a Performer.” In: *Oc. First Brazilian Symposium on Computer Music*. axambu, Minas Gerais, Brazil, 1994, p. 10 (cit. on p. 6).

[27] Joel Ryan. “Some Remarks on Musical Instrument Design at STEIM”. In: *Contemporary Music Review* 6.1 (Jan. 1991), pp. 3–17 (cit. on p. 6).

[28] Lizzie Muller. “Collecting Experience: The Multiple Incarnations of Very Nervous System”. In: *New Collecting: Exhibiting and Audiences after New Media Art*. Routledge, 2014 (cit. on p. 7).

[29] H. Morita, S. Hashimoto, and S. Ohteru. “A Computer Music System That Follows a Human Conductor”. In: *Computer* 24.7 (July 1991), pp. 44–53 (cit. on p. 7).

[30] Joseph Malloch, David Birnbaum, Elliot Sinyor, and Marcelo M Wanderley. “Towards a New Conceptual Framework for Digital Musical Instruments”. In: *Proc. 9th Int. Conf. on Digital Audio Effects*. Montreal, Canada, September 18-20, 2006, 2006, pp. 49–52 (cit. on pp. 7, 16).

[31] Jens Rasmussen. “Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models”. In: *IEEE Trans. Systems, Man, and Cybernetics* SMC-13.3 (1983), pp. 257–266 (cit. on p. 7).

[32] Alex Lubet. *Music, Disability, and Society*. Temple University Press, 2011 (cit. on p. 7).

[33] Youth Music. *Reshape Music*. Tech. rep. Oct. 2020, p. 60 (cit. on p. 8).

[34] Emma Frid. “Accessible Digital Musical Instruments—A Review of Musical Interfaces in Inclusive Music Practice”. In: *Multimodal Technologies and Interaction* 3.3 (July 2019), p. 57 (cit. on pp. 8, 19, 22, 24, 25, 53).

[35] Kimberlee Graham-Knight and George Tzanetakis. “Adaptive Music Technology: History and Future Perspectives”. In: *Proc. Int. Computer Music Conf. (ICMC2015)*. Denton, 2015, pp. 416–419 (cit. on pp. 8, 19, 53).

[36] Samuel Thompson, Hugo Scurto, and Rebecca Fiebrink. “Sound Control: Supporting Custom Musical Interface Design for Children with Disabilities”. In: *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME’19)*. Federal University of Rio Grande do Sul, Porto Alegre, Brazil, June 2019, p. 6 (cit. on p. 8).

[37] Jeppe Veirum Larsen, Dan Overholt, and Thomas B. Moeslund. “The Prospects of Musical Instruments For People with Physical Disabilities”. In: *Proc. 16th Int. Conf. on New Interfaces for Musical Expression*. NIME ’16. Griffith University, Brisbane, Australia: NIME, July 2016, pp. 327–331 (cit. on pp. 8, 19, 22, 24, 25, 53).

[38] Brendan McCloskey, Brian Bridges, and Frank Lyons. “Accessibility and Dimensionality: Enhanced Real Time Creative Independence for Digital Musicians with Quadriplegic Cerebral Palsy”. In: *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME’15)*. Louisiana State University, Baton Rouge, LA, USA, May 2015 (cit. on p. 8).

[39] Matti Luhtala, Tiina Kymäläinen, and Johan Plomp. “Designing a Music Performance Space for Persons with Intellectual Learning Disabilities”. In: *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME’11)*. Oslo, Norway, May 2011, pp. 429–432 (cit. on p. 8).

[40] Andrew McPherson, Fabio Morreale, and Jacob Harrison. “Musical Instruments for Novices: Comparing NIME, HCI and Crowdfunding Approaches”. In: *New Directions in Music and Human-Computer Interaction*. Ed. by Simon Holland, Tom Mudd, Katie Wilkie-McKenna, Andrew McPherson, and Marcelo M. Wanderley. Springer Series on Cultural Computing. Cham: Springer International Publishing, 2019, pp. 179–212 (cit. on pp. 9, 19, 22, 25).

[41] Birgitta Cappelen and Anders-Petter Andersson. “Musicking Tangibles for Empowerment”. In: *Proc. Conf. on Computers Helping People with Special Needs*. Ed. by Klaus Miesenberger, Arthur Karshmer, Petr Penaz, and Wolfgang Zagler. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 254–261 (cit. on p. 9).

[42] Christopher Small. *Musicking: The Meanings of Performing and Listening*. Wesleyan University Press, July 1998 (cit. on p. 9).

[43] W. Andrew Schloss. “Recent Advances in the Coupling of the Language MAX with the Matthews/Boie Radio Drum.” In: *Proc. Int. Computer Music Conf. (ICMC1990)*. ICMC 1990. Glasgow, 1990, pp. 398–390 (cit. on p. 10).

[44] David Birnbaum, Rebecca Fiebrink, Joseph Malloch, and Marcelo M. Wanderley. “Towards a Dimension Space for Musical Devices”. In: *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME’05)*. Vancouver, BC, Canada, 2005, pp. 192–195 (cit. on pp. 10, 16, 19, 70, 71).

[45] Sidney Fels, Ashley Gadd, and Axel Mulder. “Mapping Transparency through Metaphor: Towards More Expressive Musical Instruments”. In: *Organised Sound* 7.2 (2002), pp. 109–126 (cit. on p. 12).

[46] Andy Hunt, Marcelo M. Wanderley, and Matthew Paradis. “The Importance of Parameter Mapping in Electronic Instrument Design”. In: *Journal of New Music Research* 32.4 (Dec. 2003), pp. 429–440 (cit. on pp. 12, 13, 51).

[47] Marcelo M Wanderley and Philippe Depalle. “Gestural Control of Sound Synthesis”. In: *Proc. of the IEEE* 92.4 (2004), pp. 632–644 (cit. on pp. 13, 74).

[48] Richard A. Magill and David I. Anderson. *Motor Learning and Control: Concepts and Applications*. Eleventh. Vol. 11. McGraw-Hill New York, 2017 (cit. on p. 13).

[49] Andrew Sears, Mark Young, and Jinjuan Feng. "Physical Disabilities and Computing Technologies: An Analysis of Impairments". In: *The Human-Computer Interaction Handbook*. Ed. by Andrew Sears and Julie A. Jacko. Second. Lawrence Erlbaum Associates, 2008. Chap. 42, pp. 829–852 (cit. on p. 13).

[50] Yi Kang, Han Ding, Hengxing Zhou, Zhijian Wei, Lu Liu, Dayu Pan, and Shiqing Feng. "Epidemiology of Worldwide Spinal Cord Injury: A Literature Review". In: *Journal of Neurorestoratology* 2018.6 (Dec. 2017), pp. 1–9 (cit. on p. 15).

[51] World Health Organization. *Spinal Cord Injury Report*. Tech. rep. World Health Organization, Nov. 2013 (cit. on p. 15).

[52] Anoushka Singh, Lindsay Tetreault, Suhkvinder Kalsi-Ryan, Aria Nouri, and Michael G Fehlings. "Global Prevalence and Incidence of Traumatic Spinal Cord Injury". In: *Clinical Epidemiology* 6 (Sept. 2014), pp. 309–331 (cit. on p. 15).

[53] E.O. Talbott, A.M. Malek, and D. Lacomis. "The Epidemiology of Amyotrophic Lateral Sclerosis". In: *Handbook of Clinical Neurology*. Vol. 138. Elsevier, 2016, pp. 225–238 (cit. on p. 15).

[54] A Chiò, G Logroscino, BJ Traynor, J Collins, JC Simeone, LA Goldstein, and LA White. "Global Epidemiology of Amyotrophic Lateral Sclerosis: A Systematic Review of the Published Literature". In: *Neuroepidemiology* 41.2 (2013), pp. 118–130 (cit. on p. 15).

[55] Orla Hardiman, Ammar Al-Chalabi, Carol Brayne, Ettore Beghi, Leonard H. van den Berg, Adriano Chio, Sarah Martin, Giancarlo Logroscino, and James Rooney. "The Changing Picture of Amyotrophic Lateral Sclerosis: Lessons from European Registers". In: *Journal of Neurology, Neurosurgery & Psychiatry* 88.7 (July 2017), pp. 557–563 (cit. on p. 15).

[56] Sile O'Modhrain. "A Framework for the Evaluation of Digital Musical Instruments". In: *Computer Music Journal* 35.1 (Mar. 2011), pp. 28–42 (cit. on pp. 15–17, 135).

[57] Jerônimo Barbosa, Filipe Calegario, Veronica Teichrieb, Geber Ramalho, and Patrick McGlynn. "Considering Audience's View Towards an Evaluation Methodology for Digital Musical Instruments". In: *Proc. 12th Int. Conf. on New Interfaces for Musical Expression (NIME'12)*. NIME'12. Ann Arbor, Michigan, United States, May 2012 (cit. on p. 15).

[58] Feng Wang and Xiangshi Ren. "Empirical Evaluation for Finger Input Properties in Multi-Touch Interaction". In: *Proc. 27th Int. Conf. on Human Factors in Computing Systems (CHI'09)*. Boston, MA, USA: ACM Press, 2009, pp. 1063–1072 (cit. on pp. 15, 42).

[59] Zacharias Vamvakousis and Rafael Antonio Márquez Ramírez. "Temporal Control In the EyeHarp Gaze-Controlled Musical Interface". In: *Proceedings of the 12th International Conference on New Interfaces for Musical Expression*. NIME 2012. Ann Arbor, Michigan, United States, 21, n.d. (Cit. on pp. 16, 55, 63).

[60] Olivier Capra, Florent Berthaut, and Laurent Grisoni. "Toward Augmented Familiarity of the Audience with Digital Musical Instruments". In: *Proc. Int. Symp. on Computer Music Multidisciplinary Research (CMMR17)*. Matosinhos, 2017, pp. 558–573 (cit. on p. 16).

[61] Luca Turchet, Travis West, and Marcelo M Wanderley. "Touching the Audience: Musical Haptic Wearables for Augmented and Participatory Live Music Performances". In: *Personal and Ubiquitous Computing* (2020), pp. 1–21 (cit. on p. 16).

[62] Zacharias Vamvakousis and Rafael Ramirez. “The EyeHarp: A Gaze-Controlled Digital Musical Instrument”. In: *Frontiers in Psychology* 7 (2016), article 906 (cit. on pp. 16, 51, 55, 63, 79, 110, 126, 127, 135).

[63] Atau Tanaka. “Musical Performance Practice on Sensor-based Instruments”. In: *Trends in Gestural Control of Music*. IRCAM, Paris: M. Wanderley and M. Battier, 2000, pp. 389–405 (cit. on p. 19).

[64] Garth Paine. “Towards a Taxonomy of Realtime Interfaces for Electronic Music Performance”. In: *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME’10)*. Sydney, Australia, 2010, pp. 436–439 (cit. on p. 19).

[65] Thor Magnusson. “An Epistemic Dimension Space for Musical Devices”. In: *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME’10)*. Sydney, Australia, 2010: University of Technology, Sydney, Jan. 2010, pp. 43–46 (cit. on p. 20).

[66] Ian Hattwick and Marcelo M Wanderley. “A Dimension Space for Evaluating Collaborative Musical Performance Systems”. In: *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME’12)*. University of Michigan, Ann Arbor, May 2012, p. 4 (cit. on p. 20).

[67] Oliver Hödl and Geraldine Fitzpatrick. “Exploring the Design Space of Hand-Controlled Guitar Effects for Live Music”. In: *Proc. Int. Computer Music Conf. (ICMC 2013)*. Perth, Australia, 2013, p. 8 (cit. on p. 20).

[68] World Health Organization. *World Report on Disability*. 2011 (cit. on p. 21).

[69] World Health Organization. *International Classification of Functioning, Disability and Health*. 2001 (cit. on p. 21).

[70] Andrew Sears and Julie A. Jacko, eds. *The Human-Computer Interaction Handbook*. Second. Lawrence Erlbaum Associates, 2008 (cit. on pp. 21, 22, 24, 25).

[71] Nicola Davanzo and Federico Avanzini. “Digital Musical Instruments for Quadriplegic Performers: Design Principles, Challenges and Perspectives”. In: *Int. J. Human-Computer Studies* (2020) (cit. on pp. 22, 25).

[72] Thomas F. Boat and Joel T. Wu. “Clinical Characteristics of Intellectual Disabilities”. In: *Clinical Characteristics of Intellectual Disabilities*. Washington (DC): National Academies Press (US), Oct. 2015 (cit. on p. 23).

[73] Susan L. Hyman. “Mental Retardation”. In: *Pediatric Clinical Advisor (Second Edition)*. Ed. by Lynn C. Garfunkel, Jeffrey M. Kaczorowski, and Cynthia Christy. Second Edition. Philadelphia: Mosby, Jan. 2007, pp. 369–370 (cit. on p. 23).

[74] C. B. Mervis. “Mental Retardation: Cognitive Aspects”. In: *International Encyclopedia of the Social & Behavioral Sciences*. Ed. by Neil J. Smelser and Paul B. Baltes. Oxford: Pergamon, Jan. 2001, pp. 9700–9704 (cit. on p. 23).

[75] Jacob Harrison and Andrew P. McPherson. “Adapting the Bass Guitar for One-Handed Playing”. In: *J. New Music Research* 46.3 (2017), pp. 270–285 (cit. on pp. 23, 27).

[76] Barry Farrimond, Duncan Gillard, Doug Bott, and Douglas Lonie. *Engagement with Technology in Special Educational & Disabled Music Settings*. Tech. rep. Youth Music, Dec. 2011, p. 40 (cit. on p. 24).

[77] Rebecca Fiebrink, Dan Trueman, and Perry R. Cook. “A Meta-Instrument for Interactive, on-the-Fly Machine Learning”. In: *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME’09)*. Pittsburgh, 2009, pp. 280–285 (cit. on pp. 24, 74).

[78] Simon Katan, Mick Grierson, and Rebecca Fiebrink. “Using Interactive Machine Learning to Support Interface Development through Workshops with Disabled People”. In: *Proc. ACM Conf. Human Factors in Computing Systems (CHI2015)*. ACM. Seoul, 2015, pp. 251–254 (cit. on pp. 24, 74).

[79] Adnan Marquez-Borbon and Paul Stapleton. “Fourteen Years of Nime: The Value and Meaning of ’community’ in Interactive Music Research”. In: *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME’15)*. Baton Rouge, 2015, pp. 307–312 (cit. on pp. 24, 75).

[80] David Wessel and Matthew Wright. “Problems and Prospects for Intimate Musical Control of Computers”. In: *Computer Music J.* 26.3 (2002), pp. 11–22 (cit. on pp. 25, 81).

[81] Eduardo R Miranda, Wendy L Magee, John J Wilson, Joel Eaton, and Ramaswamy Palaniappan. “Brain-Computer Music Interfacing (BCMI): From Basic Research to the Real World of Special Needs”. In: *Music & Medicine* 3.3 (2011), pp. 134–140 (cit. on pp. 25, 55, 62).

[82] Michael S. Grierson. “Making Music with Images: Interactive Audiovisual Performance Systems for the Deaf”. In: *Int. J. on Disability and Human Development* 10.1 (Jan. 2011), pp. 37–41 (cit. on p. 27).

[83] Arvid Jense and Hans Leeuw. “Wambam: A Case Study in Design for an Electronic Musical Instrument for Severely Intellectually Disabled Users.” In: *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME’15)*. Louisiana State University, Baton Rouge, LA, USA, May 2015, pp. 74–77 (cit. on p. 27).

[84] Mélodie Boulay, Samuel Benveniste, Sandra Boespflug, Pierre Jouvelot, and Anne-Sophie Rigaud. “A Pilot Usability Study of MINWii, a Music Therapy Game for Demented Patients”. In: *Technology and health care* 19.4 (2011), pp. 233–246 (cit. on p. 27).

[85] Skoog. *Skoogmusic: Accessible Musical Instrument for Everyone.* https://skoog.media/skoog2-0/ (cit. on p. 27).

[86] OpenUpMusic. *The Clarion.* http://openupmusic.org/the-clarion/. 2021 (cit. on pp. 27, 55, 67).

[87] R. Benjamin Knapp and Hugh S. Lusted. “A Bioelectric Controller for Computer Music Applications”. In: *Computer Music Journal* 14.1 (1990), pp. 42–47 (cit. on pp. 27, 55, 56).

[88] Stuart K Card, Jock D Mackinlay, and George G Robertson. “A Morphological Analysis of the Design Space of Input Devices”. In: *ACM Transactions on Information Systems* 9.2 (1991), pp. 99–122 (cit. on p. 29).

[89] Wilder Penfield and Theodore Rasmussen. *The Cerebral Cortex of Man; a Clinical Study of Localization of Function.* New York: Macmillan, 1950 (cit. on pp. 29, 30).

[90] Michael J. Lyons, Michael Haehnel, and Nobuji Tetsutani. “Designing, Playing, and Performing with a Vision-based Mouth Interface”. In: *Proc. 3rd Conf. on New Interfaces for Musical Expression.* NIME ’03. McGill University, Montreal, Canada: NIME, 2003, pp. 116–121 (cit. on pp. 29, 36, 139).

[91] Simeon Keates, John Clarkson, and Peter Robinson. "Investigating the Applicability of User Models for Motion-Impaired Users". In: *Proc. Int. ACM SIGACCESS Conf. on Computers and Accessibility*. Arlington, 2000, pp. 129–136 (cit. on p. 31).

[92] Päivi Majaranta and Andreas Bulling. "Eye Tracking and Eye-Based Human–Computer Interaction". In: *Advances in Physiological Computing*. Ed. by Stephen H. Fairclough and Kiel Gilleade. Human–Computer Interaction Series. London: Springer, 2014, pp. 39–65 (cit. on pp. 31, 33, 34, 80, 129, 132).

[93] T Scott Saponas, Desney S Tan, Dan Morris, and Ravin Balakrishnan. "Demonstrating the Feasibility of Using Forearm Electromyography for Muscle-Computer Interfaces". In: *Proc. SIGCHI Conf. on Human Factors in Computing Systems*. Florence, 2008, pp. 515–524 (cit. on p. 31).

[94] Wolfgang Hess. "Head-Tracking Techniques for Virtual Acoustics Applications". In: *Proceedings of the 133rd AES Convention*. Audio Engineering Society. San Francisco, CA, USA: Audio Engineering Society, Inc., Oct. 2012 (cit. on p. 31).

[95] Ujwal Chaudhary, Niels Birbaumer, and Ander Ramos-Murguialday. "Brain–Computer Interfaces for Communication and Rehabilitation". In: *Nature Reviews Neurology* 12.9 (2016), pp. 513–525 (cit. on pp. 31, 39).

[96] Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, and Theresa M. Vaughan. "Brain-Computer Interfaces for Communication and Control". In: *Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology* 113.6 (June 2002), pp. 767–791 (cit. on pp. 31, 39, 40).

[97] Christoph Guger, Brendan Z Allison, and Natalie Mrachacz-Kersting. "Brain-Computer Interface Research: A State-of-the-Art Summary 7". In: *Brain-Computer Interface Research*. Springer, 2019, pp. 1–9 (cit. on p. 31).

[98] Anthony J Hornof. "The Prospects For Eye-Controlled Musical Performance". In: *Proc. 14th Int. Conf. on New Interfaces for Musical Expression (NIME'14)*. NIME 2014. Goldsmiths, University of London, UK, July 2014 (cit. on pp. 33, 51, 77, 79, 125).

[99] Frans VanderWerf, Peter Brassinga, Dik Reits, Majid Aramideh, and Bram Ongerboer de Visser. "Eyelid Movements: Behavioral Studies of Blinking in Humans Under Different Stimulus Conditions". In: *Journal of Neurophysiology* 89.5 (May 2003), pp. 2784–2796 (cit. on p. 34).

[100] Kohei Aarai and Ronny Mardiyanto. "Comparative Study on Blink Detection and Gaze Estimation Methods for HCI, in Particular, Gabor Filter Utilized Blink Detection Method". In: *Proc. 8th Int. Conf. on Information Technology: New Generations (ITNG 2011)*. May 2011, pp. 441–446 (cit. on p. 34).

[101] Kyung S. Park and Kyung T. Lee. "Eye-Controlled Human/Computer Interface Using the Line-of-Sight and the Intentional Blink". In: *Computers & Industrial Engineering*. IE in Korea 30.3 (July 1996), pp. 463–473 (cit. on p. 34).

[102] A. Krolak and P. Strumillo. "Vision-Based Eye Blink Monitoring System for Human-Computer Interfacing". In: *Proceedings of the 2008 Conference on Human System Interactions*. May 2008, pp. 994–998 (cit. on p. 34).

[103] Isabelle Guatella, Serge Santi, Benot Lagrue, and Christian Cavé. “Are Eyebrow Movements Linked to Voice Variations and Turn-Taking in Dialogue? An Experimental Investigation”. In: *Language and speech* 52.2-3 (2009), pp. 207–222 (cit. on p. 35).

[104] Karen Schmidt, Sharika Bhattacharya, and Rachel Denlinger. “Comparison of Deliberate and Spontaneous Facial Movement in Smiles and Eyebrow Raises”. In: *Journal of nonverbal behavior* 33 (Mar. 2009), pp. 35–45 (cit. on p. 35).

[105] K. Grauman, M. Betke, J. Lombardi, J. Gips, and G.R. Bradski. “Communication via Eye Blinks and Eyebrow Raises: Video-Based Human-Computer Interfaces”. In: *Universal Access in the Information Society* 2.4 (Nov. 2003), pp. 359–373 (cit. on p. 35).

[106] Ville Rantanen, Pekka-Henrik Niemenlehto, Jarmo Verho, and Jukka Lekkala. “Capacitive Facial Movement Detection for Human–Computer Interaction to Click by Frowning and Lifting Eyebrows”. In: *Medical & biological engineering & computing* 48.1 (2010), pp. 39–47 (cit. on p. 35).

[107] Ingo R. Titze and Daniel W. Martin. “Principles of Voice Production”. In: *J. of the Acoustical Society of America* 104.3 (Sept. 1998), pp. 1148–1148 (cit. on pp. 35, 37, 139).

[108] Takeo Igarashi and John F. Hughes. “Voice as Sound: Using Non-Verbal Voice Input for Interactive Control”. In: *Proc. ACM Symp. on User Interface Software and Technology*. UIST ’01. Orlando, 2001, pp. 155–156 (cit. on p. 35).

[109] Susumu Harada, James A. Landay, Jonathan Malkin, Xiao Li, and Jeff A. Bilmes. “The Vocal Joystick: Evaluation of Voice-Based Cursor Control Techniques”. In: *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*. Assets ’06. New York, NY, USA: Association for Computing Machinery, Oct. 2006, pp. 197–204 (cit. on p. 35).

[110] Jordi Janer. “Singing-Driven Interfaces for Sound Synthesizers”. PhD thesis. Universitat Pompeu Fabra, Barcelona: Dept. of Information and Communication Technologies, 2008 (cit. on p. 35).

[111] Dan Stowell. “Making Music through Real-Time Voice Timbre Analysis: Machine Learning and Timbral Control”. PhD thesis. Queen Mary University of London: School of Electronic Engineering and Computer Science, 2010 (cit. on p. 35).

[112] Stefano Fasciani. “Voice-Controlled Interface for Digital Musical Instruments”. PhD thesis. National University of Singapore: Citeseer / Grad. School for Integrative Sciences and Engineering, 2014 (cit. on p. 35).

[113] Douglas O’Shaughnessy. “Interacting with Computers by Voice: Automatic Speech Recognition and Synthesis”. In: *Proc. of the IEEE* 91.9 (2003), pp. 1272–1305 (cit. on p. 36).

[114] T. A. Wilson, G. S. Beavers, M. A. DeCoster, D. K. Holger, and M. D. Regenfuss. “Experiments on the Fluid Mechanics of Whistling”. In: *The Journal of the Acoustical Society of America* 50.1B (July 1971), pp. 366–372 (cit. on p. 36).

[115] Alba Azola, Jeffrey Palmer, Rachel Mulheren, Riccardo Hofer, Florian Fischmeister, and W. Tecumseh Fitch. “The Physiology of Oral Whistling: A Combined Radiographic and MRI Analysis”. In: *J. of Applied Physiology* 124.1 (Jan. 2018), pp. 34–39 (cit. on pp. 36, 139).

[116] Urko Esnaola and Tim Smithers. “Whistling to Machines”. In: *Ambient Intelligence in Everyday Life: Foreword by Emile Aarts*. Ed. by Yang Cai and Julio Abascal. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 198–226 (cit. on p. 36).

[117] William Birmingham, Roger Dannenberg, and Bryan Pardo. "Query by Humming with the VocalSearch System". In: *Communications of the ACM* 49.8 (Aug. 2006), pp. 49–52 (cit. on p. 36).

[118] Hung-Che Shen and Chung-Nan Lee. "An Interactive Whistle-to-Music Composing System Based on Transcription, Variation and Chords Generation". In: *Multimedia Tools and Applications* 53.1 (May 2011), pp. 253–269 (cit. on p. 36).

[119] I. Mougharbel, R. El-Hajj, H. Ghamlouch, and E. Monacelli. "Comparative Study on Different Adaptation Approaches Concerning a Sip and Puff Controller for a Powered Wheelchair". In: *Proc. of the 2013 Science and Information Conf.* London, UK, Oct. 2013, pp. 597–603 (cit. on p. 36).

[120] Michael Jones, Kevin Grogg, John Anschutz, and Ruth Fierman. "A Sip-and-Puff Wireless Remote Control for the Apple iPod". In: *Assistive Technology* 20.2 (June 2008), pp. 107–110 (cit. on p. 36).

[121] T.K. Capin, Hansrudi Noser, Daniel Thalmann, Igor Pandžić, and Nadia Thalmann. "Virtual Human Representation and Communication in VLNET". In: *IEEE Computer Graphics and Applications* 17 (Apr. 1997), pp. 42–53 (cit. on p. 36).

[122] Bruce Denby, T. Schultz, K. Honda, T. Hueber, J. M. Gilbert, and J. S. Brumberg. "Silent Speech Interfaces". In: *Speech Communication (Special Issues)* 52.4 (Apr. 2010), pp. 270–287 (cit. on pp. 36, 138).

[123] Davide Bonafede, Luca A. Ludovico, and Giorgio Presti. "A Proposal for the Interactive Sonification of the Human Face". In: *Proc. 2nd Int. Conf. on Computer-Human Interaction Research and Applications (CHIRA '18).* Vol. 1. Seville, Spain, 2018, pp. 163–169 (cit. on p. 36).

[124] Chris Salem and Shumin Zhai. "An Isometric Tongue Pointing Device". In: *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems.* Atlanta, 1997, pp. 538–539 (cit. on p. 37).

[125] Xueliang Huo, Jia Wang, and Maysam Ghovanloo. "A Magneto-Inductive Sensor Based Wireless Tongue-Computer Interface". In: *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society* 16.5 (Oct. 2008), pp. 497–504 (cit. on p. 37).

[126] Hangue Park, Mehdi Kiani, Hyung-Min Lee, Jeonghee Kim, Jacob Block, Benoit Gosselin, and Maysam Ghovanloo. "A Wireless Magnetoresistive Sensing System for an Intraoral Tongue-Computer Interface". In: *IEEE transactions on biomedical circuits and systems* 6.6 (Dec. 2012), pp. 571–585 (cit. on pp. 37, 138).

[127] L. N. S. Andreasen Struijk. "An Inductive Tongue Computer Interface for Control of Computers and Assistive Devices". In: *IEEE Transactions on Biomedical Engineering* 53.12 (Dec. 2006), pp. 2594–2597 (cit. on p. 37).

[128] R. Vaidyanathan, Beomsu Chung, L. Gupta, Hyunseok Kook, S. Kota, and J. D. West. "Tongue-Movement Communication and Control Concept for Hands-Free Human&#x2013;Machine Interfaces". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* 37.4 (July 2007), pp. 533–546 (cit. on p. 37).

[129] Jingyuan Cheng, Ayano Okoso, Kai Kunze, Niels Henze, Albrecht Schmidt, Paul Lukowicz, and Koichi Kise. “On the Tip of My Tongue: A Non-Invasive Pressure-Based Tongue Interface”. In: *Proc. 5th Augmented Human Int. Conf.* AH ’14. New York, NY, USA: Association for Computing Machinery, Mar. 2014, pp. 1–4 (cit. on pp. 37, 138).

[130] David Wessel. “Instruments That Learn, Refined Controllers, and Source Model Loudspeakers”. In: *Computer Music J.* 15.4 (1991), pp. 82–86 (cit. on p. 37).

[131] Florian Vogt, Graeme McCaig, Mir Adnan Ali, and Sidney S. Fels. “Tongue’n’Groove: An Ultrasound Based Music Controller”. In: *Proc. 2nd Int. Conf. on New Interfaces for Musical Expression.* NIME 2002. Dublin, Ireland: NIME, 2002, pp. 60–64 (cit. on pp. 37, 138).

[132] Hye Yeon Nam and Carl DiSalvo. “Tongue Music: The Sound of a Kiss”. In: *Proc. 28th Int. Conf. Ext. Abs. on Human Factors in Computing Systems.* Atlanta, Georgia, USA: ACM Press, 2010, p. 4805 (cit. on pp. 37, 138).

[133] Hyuk Jeong, Jong-Sung Kim, and Jin-Seong Choi. “A Study of an EMG-controlled HCI Method by Clenching Teeth”. In: *Computer Human Interaction.* Ed. by Masood Masoodian, Steve Jones, and Bill Rogers. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004, pp. 163–170 (cit. on p. 37).

[134] Youngbum Lee and Myoungho Lee. “SMS Application Using EMG Signal of Clenching Teeth for E-Health Communication”. In: *Telemedicine and e-Health* 14.6 (2008), pp. 593–597 (cit. on p. 37).

[135] Tyler Simpson, Colin Broughton, Michel JA Gauthier, and Arthur Prochazka. “Tooth-Click Control of a Hands-Free Computer Interface”. In: *IEEE Trans. Biomedical Engineering* 55.8 (2008), pp. 2050–2056 (cit. on p. 37).

[136] Tomás Vega Gálvez, Shardul Sapkota, Alexandru Dancu, and Pattie Maes. “Byte.It: Discreet Teeth Gestures for Mobile Device Interaction”. In: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems.* CHI EA ’19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–6 (cit. on p. 37).

[137] K. Kuzume. “Evaluation of Tooth-Touch Sound and Expiration Based Mouse Device for Disabled Persons”. In: *2012 IEEE International Conference on Pervasive Computing and Communications Workshops.* Mar. 2012, pp. 387–390 (cit. on p. 37).

[138] Daniel Ashbrook, Carlos Tejada, Dhwanit Mehta, Anthony Jiminez, Goudam Muralitharam, Sangeeta Gajendra, and Ross Tallents. “Bitey: An Exploration of Tooth Click Gestures for Hands-Free User Interface Control”. In: *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services.* MobileHCI ’16. New York, NY, USA: Association for Computing Machinery, Sept. 2016, pp. 158–169 (cit. on p. 37).

[139] H. Iwata, H. Yano, T. Uemura, and T. Moriya. “Food Simulator: A Haptic Interface for Biting”. In: *IEEE Virtual Reality 2004.* Mar. 2004, pp. 51–57 (cit. on p. 38).

[140] Lewis M. Nashner and Gin McCollum. “The Organization of Human Postural Movements: A Formal Basis and Experimental Synthesis”. In: *Behavioral and Brain Sciences* 8.1 (Mar. 1985), pp. 135–150 (cit. on p. 38).

[141] B. Biguer, M. Jeannerod, and C. Prablanc. “The Coordination of Eye, Head, and Arm Movements during Reaching at a Single Visual Target”. In: *Experimental Brain Research* 46.2 (1982), pp. 301–304 (cit. on p. 38).

[142] Willard R. Thurlow and Philip S. Runge. “Effect of Induced Head Movements on Localization of Direction of Sounds”. In: *The Journal of the Acoustical Society of America* 42.2 (Aug. 1967), pp. 480–488 (cit. on p. 38).

[143] Carlos Toshinori Ishi, Hiroshi Ishiguro, and Norihiro Hagita. “Analysis of Relationship between Head Motion Events and Speech in Dialogue Conversations”. In: *Speech Communication* 57 (Feb. 2014), pp. 233–243 (cit. on p. 38).

[144] Sarig-Bahat. “Evidence for Exercise Therapy in Mechanical Neck Disorders”. In: *Manual Therapy* 8.1 (Feb. 2003), pp. 10–20 (cit. on p. 38).

[145] J. Leaman and H. M. La. “A Comprehensive Review of Smart Wheelchairs: Past, Present, and Future”. In: *IEEE Transactions on Human-Machine Systems* 47.4 (Aug. 2017), pp. 486–499 (cit. on p. 38).

[146] Stefania Serafin, Cumhur Erkut, Juraj Kojs, Niels C. Nilsson, and Rolf Nordahl. “Virtual Reality Musical Instruments: State of the Art, Design Principles, and Future Directions”. In: *Computer Music J.* 40.3 (Sept. 2016), pp. 22–40 (cit. on p. 38).

[147] David Merrill. “Head-Tracking for Gestural and Continuous Control of Parameterized Audio Effects”. In: *Proc. 3rd Int. Conf. on New Interfaces for Musical Expression.* NIME ’03. McGill University, Montreal, Canada: NIME, May 2003, pp. 218–219 (cit. on p. 38).

[148] Ajay Kapur, Ariel J Lazier, Philip Davidson, R Scott Wilson, and Perry R Cook. “The Electronic Sitar Controller”. In: *Proceedings of the 4th International Conference on New Interfaces for Musical Expression.* NIME 2004. 2004, pp. 7–12 (cit. on p. 38).

[149] Gabrielle L Hands and Cara E Stepp. “Effect of Age on Human–Computer Interface Control via Neck Electromyography”. In: *Interacting with Computers* 28.1 (2014), pp. 47–54 (cit. on p. 38).

[150] Thorsten O. Zander, Christian Kothe, Sabine Jatzev, and Matti Gaertner. “Enhancing Human-Computer Interaction with Input from Active and Passive Brain-Computer Interfaces”. In: *Brain-Computer Interfaces: Applying Our Minds to Human-Computer Interaction.* Ed. by Desney S. Tan and Anton Nijholt. Human-Computer Interaction Series. London: Springer, 2010, pp. 181–199 (cit. on p. 39).

[151] Eduardo Reck Miranda. “Brain–Computer Music Interfacing: Interdisciplinary Research at the Crossroads of Music, Science and Biomedical Engineering”. In: *Guide to Brain-Computer Music Interfacing.* Ed. by Eduardo Reck Miranda and Julien Castet. London: Springer London, 2014, pp. 1–27 (cit. on pp. 39, 40).

[152] Bo Hjorth. “EEG Analysis Based on Time Domain Properties”. In: *Electroencephalography and Clinical Neurophysiology* 29.3 (Sept. 1970), pp. 306–310 (cit. on p. 39).

[153] Mandeep Singh, Mooninder Singh, and Surabhi Gangwar. “Emotion Recognition Using Electroencephalography (EEG): A Review”. In: *International Journal of IT & Knowledge Management (IJITKM)* 7.1 (Dec. 2013), pp. 1–5 (cit. on p. 39).

[154] Christian Mühl, Brendan Allison, Anton Nijholt, and Guillaume Chanel. "A Survey of Affective Brain Computer Interfaces: Principles, State-of-the-Art, and Challenges". In: *Brain-Computer Interfaces* 1.2 (2014), pp. 66–84 (cit. on p. 39).

[155] James A Russell. "A Circumplex Model of Affect". In: *J. of Personality and Social Psychology* 39.6 (1980), pp. 1161–1178 (cit. on p. 39).

[156] Beste F Yuksel, Kurt B Oleson, Lane Harrison, Evan M Peck, Daniel Afergan, Remco Chang, and Robert JK Jacob. "Learn Piano with BACh: An Adaptive Learning Interface That Adjusts Task Difficulty Based on Brain State". In: *Proc. ACM Conf. Human Factors in Computing Systems (CHI2016)*. CHI '16. San Jose, 2016, pp. 5372–5384 (cit. on p. 39).

[157] Beste Filiz Yuksel, Daniel Afergan, Evan Peck, Garth Griffin, Lane Harrison, Nick Chen, Remco Chang, and Robert Jacob. "BRAAHMS: A Novel Adaptive Musical Interface Based on Users' Cognitive State". In: *Proceedings of the 15th Conference on New Interfaces for Musical Expression*. NIME 2015. Baton Rouge, Louisiana, USA: The School of Music and the Center for Computation and Technology (CCT), Louisiana State University, 2015, pp. 136–139 (cit. on p. 39).

[158] Fabio Morreale and Antonella De Angeli. "Collaborating with an Autonomous Agent to Generate Affective Music". In: *ACM Trans. Computers in Entertainment (CiE2016)* 14.3 (2016) (cit. on p. 39).

[159] Audrey Girouard, Erin Treacy Solovey, and Robert JK Jacob. "Designing a Passive Brain Computer Interface Using Real Time Classification of Functional Near-Infrared Spectroscopy". In: *Int. J. of Autonomous and Adaptive Communication Systems* 6.1 (2013), pp. 26–44 (cit. on p. 39).

[160] Ian Daly, Duncan Williams, Alexis Kirke, James Weaver, Asad Malik, Faustina Hwang, Eduardo Miranda, and Slawomir J Nasuto. "Affective Brain–Computer Music Interfacing". In: *J. Neural Engin.* 13.4 (2016) (cit. on p. 39).

[161] A. Väljamäe, T. Steffert, S. Holland, X. Marimon, R. Benitez, S. Mealla, A. Oliveira, and S. Jordà. "A Review of Real-Time EEG Sonification Research". In: *Proc. Int. Conf. on Auditory Display 2013 (ICAD 2013)*. Lodz, Poland, July 2013, pp. 85–93 (cit. on p. 39).

[162] F. Aloise, P. Aricò, F. Schettini, A. Riccio, M. Risetti, S. Salinari, D. Mattia, F. Babiloni, and F. Cincotti. "A New P300 No Eye-Gaze Based Interface: Geospell". In: *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing*. Rome, Italy: SciTePress - Science and and Technology Publications, 2011, pp. 227–232 (cit. on p. 39).

[163] Emanuel Donchin, Kevin M Spencer, and Ranjith Wijesinghe. "The Mental Prosthesis: Assessing the Speed of a P300-Based Brain-Computer Interface". In: *IEEE Trans. Rehabilitation Engineering* 8.2 (2000), pp. 174–179 (cit. on p. 39).

[164] Danhua Zhu, Jordi Bieger, Gary Garcia Molina, and Ronald M. Aarts. "A Survey of Stimulation Methods Used in SSVEP-Based BCIs". In: *Computational Intelligence and Neuroscience* (2010) (cit. on p. 39).

[165] Ricardo Chavarriaga and José del R Millán. "Learning from EEG Error-Related Potentials in Noninvasive Brain-Computer Interfaces". In: *IEEE Trans. Neural Systems and Rehabilitation Engineering* 18.4 (2010), pp. 381–388 (cit. on p. 40).

[166] M. Grierson. “Composing With Brainwaves: Minimal Trial P300 Recognition As An Indication Of Subjective Preference For The Control Of A Musical Instrument”. In: *Proc. ’08 Int. Computer Music Conference (ICMC)*. Vol. 2008. ICMC 2008. Aug. 2008 (cit. on p. 40).

[167] Andreas Pinegger, Hannah Hiebel, Selina C. Wriessnegger, and Gernot R. Müller-Putz. “Composing Only by Thought: Novel Application of the P300 Brain-Computer Interface”. In: *PLoS ONE* 12.9 (Sept. 2017) (cit. on p. 40).

[168] Yee Chieh (Denise) Chew and Eric Caspary. “MusEEGk: A Brain Computer Musical Interface”. In: *Proc. ’11 Ann. Conf. Ext. Abs. on Human Factors in Computing Systems*. Vancouver, BC, Canada: ACM Press, 2011, p. 1417 (cit. on p. 40).

[169] Eduardo Reck Miranda. “Brain-Computer Music Interface for Composition and Performance”. In: *International Journal on Disability and Human Development* 5.2 (Jan. 2006) (cit. on p. 40).

[170] Eduardo Miranda and Andrew Brouse. “Toward Direct Brain-Computer Musical Interfaces”. In: *Proceedings of the 5th Conference on New Interfaces for Musical Expression*. NIME 2005. Vancouver, BC, Canada, May 2005, p. 4 (cit. on p. 40).

[171] Elisabeth V. C. Friedrich, Reinhold Scherer, and Christa Neuper. “The Effect of Distinct Mental Strategies on Classification Performance for Brain–Computer Interfaces”. In: *International Journal of Psychophysiology* 84.1 (Apr. 2012), pp. 86–94 (cit. on p. 40).

[172] Leigh R. Hochberg et al. “Reach and Grasp by People with Tetraplegia Using a Neurally Controlled Robotic Arm”. In: *Nature* 485.7398 (May 2012), pp. 372–375 (cit. on p. 40).

[173] Karl LaFleur, Kaitlin Cassady, Alexander Doud, Kaleb Shades, Eitan Rogin, and Bin He. “Quadcopter Control in Three-Dimensional Space Using a Noninvasive Motor Imagery-Based Brain-Computer Interface”. In: *Journal of Neural Engineering* 10.4 (Aug. 2013) (cit. on p. 40).

[174] Niels Birbaumer et al. “The Thought Translation Device (TTD) for Completely Paralyzed Patients”. In: *IEEE Transactions on Rehabilitation Engineering* 8 (July 2000), pp. 190–193 (cit. on p. 40).

[175] Mirko Pham, Thilo Hinterberger, Nicola Neumann, Andrea Kübler, Nadine Hofmayer, Anke Grether, Barbara Wilhelm, Jean-Jacques Vatine, and Niels Birbaumer. “An Auditory Brain-Computer Interface Based on the Self-Regulation of Slow Cortical Potentials”. In: *Neurorehabilitation and Neural Repair* 19.3 (2005), pp. 206–218 (cit. on p. 40).

[176] Vamvakousis. “Towards a Low Cost Mu-Rhythm Based BCI”. In: *Proc. Fifth International Brain-Computer Interface Meeting 2013*. California: Graz University of Technology Publishing House, 2013 (cit. on p. 40).

[177] Thomas A. Deuel, Juan Pampin, Jacob Sundstrom, and Felix Darvas. “The Encephalophone: A Novel Musical Biofeedback Device Using Conscious Control of Electroencephalogram (EEG)”. In: *Frontiers in Human Neuroscience* 11 (Apr. 2017) (cit. on p. 40).

[178] Takayuki Hamano, Tomasz M Rutkowski, Hiroko Terasawa, Kazuo Okanoya, and Kiyoshi Furukawa. “Generating an Integrated Musical Expression with a Brain–Computer Interface”. In: *Proceedings of the 13th Conference on New Interfaces for Musical Expression*. NIME 2013. KAIST, Daejeon, Korea, May 2013, p. 6 (cit. on p. 40).

[179] Keisuke Shima, Toshio Tsuji, Akihiko Kandori, Masaru Yokoe, and Saburo Sakoda. “Measurement and Evaluation of Finger Tapping Movements Using Log-linearized Gaussian Mixture Networks”. In: *Sensors* 9.3 (Mar. 2009), pp. 2187–2201 (cit. on p. 41).

[180] Scott McKenzie. “Fitts’ Law”. In: *Handbook of Human-Computer Interaction*. Vol. 1. Hoboken, NJ: K. L. Norman & J. Kirakowski, 2018, pp. 349–370 (cit. on pp. 42, 122).

[181] Daniel Natapov, Steven J. Castellucci, and I. Scott MacKenzie. “ISO 9241-9 Evaluation of Video Game Controllers”. In: *Proc. Graphics Interface (GI’09)*. GI’09. Kelowna British Columbia Canada: Canadian Information Processing Society, May 2009, pp. 223–230 (cit. on p. 42).

[182] Gregor Burger, Jože Guna, and Matevž Pogačnik. “Suitability of Inexpensive Eye-Tracking Device for User Experience Evaluations”. In: *Sensors* 18.6 (June 2018), p. 1822 (cit. on p. 45).

[183] Kristien Ooms, Lien Dupont, Lieselot Lapon, and Stanislav Popelka. “Accuracy and Precision of Fixation Locations Recorded with the Low-Cost Eye Tribe Tracker in Different Experimental Setups”. In: *Journal of Eye Movement Research* 8.1 (2015) (cit. on pp. 45, 91).

[184] Christopher G Lausted, Arthur T Johnson, William H Scott, Monique M Johnson, Karen M Coyne, and Derya C Coursey. “Maximum Static Inspiratory and Expiratory Pressures with Different Lung Volumes”. In: *BioMedical Engineering OnLine* 5 (May 2006), p. 29 (cit. on p. 45).

[185] Dale Purves, George J. Augustine, David Fitzpatrick, Lawrence C. Katz, Anthony-Samuel LaMantia, James O. McNamara, and S. Mark Williams. “Types of Eye Movements and Their Functions”. In: *Neuroscience. 2nd edition* (2001), pp. 361–390 (cit. on pp. 49, 79).

[186] Carlos H. Morimoto, Antonio Diaz-Tula, José A. T. Leyva, and Carlos E. L. Elmadjian. “Eyejam: A Gaze-Controlled Musical Interface”. In: *Proceedings of the 14th Brazilian Symposium on Human Factors in Computing Systems*. IHC ’15. Salvador, Brazil: ACM, 2015, 37:1–37:9 (cit. on pp. 51, 55, 68, 110).

[187] Housemate. *Magic Flute*. http://web.archive.org/web/20210506034818/http://housemate.ie/magic-flute/. 2021 (cit. on pp. 51, 55, 59).

[188] Jamboxx. *Jamboxx*. http://web.archive.org/web/20211216125104/https://www.jamboxx.com/. 2021 (cit. on pp. 51, 55, 60, 126).

[189] Alper Kaya. *ALS & Assistive Music*. https://www.youtube.com/channel/UCDtDrroZ09_jRv-4O6geT4w. 2021 (cit. on p. 51).

[190] Eric Lyon, R. Benjamin Knapp, and Gascia Ouzounian. “Compositional and Performance Mapping in Computer Chamber Music: A Case Study”. In: *Computer Music Journal* 38.3 (Sept. 2014), pp. 64–75 (cit. on pp. 55, 56).

[191] Takuya Niikawa. “Tongue-Controlled Electro-Musical Instrument”. In: *Proc. 18th Int. Congr. on Acoustics*. Vol. 3. International Conference Hall, Kyoto, Japan: Acoustical Society, Apr. 2004, pp. 1905–1908 (cit. on pp. 55, 57, 126).

[192] Vahakn Matossian and Rolf Gehlhaar. “Human Instruments: Accessible Musical Instruments for People with Varied Physical Ability”. In: *Annual Review of Cybertherapy and Telemedicine*. Studies in Health Technology and Informatics 219 (2015), pp. 202–207 (cit. on pp. 55, 58).

[193] HumanInstruments. *Human Instruments*. https://web.archive.org/web/20211027211234/https://www.huma
2021 (cit. on pp. 55, 58).

[194] Sam Bailey, Adam Scott, Harry Wright, Ian M. Symonds, and Kia Ng. "Eye.Breathe.Music: Creating Music through Minimal Movement". In: *Proc. Conf. Electronic Visualisation and the Arts (EVA 2010)*. London, UK, July 2010, pp. 254–258 (cit. on pp. 55, 61, 110).

[195] Joel Eaton. "Brain-Computer Music Interfacing: Designing Practical Systems for Creative Applications". PhD thesis. Faculty of Arts, University of Plymouth, 2016 (cit. on pp. 55, 62).

[196] Zacharias Vamvakousis and Rafael Ramirez. "A Brain-Gaze Controlled Musical Interface". In: *Proc. Berlin BCI Workshop 2012 - Advances in Neurotechnology*. Berlin, Germany, Sept. 2012 (cit. on pp. 55, 63).

[197] Fove Inc. *Eye Play the Piano*. http://eyeplaythepiano.com/en/. 2020 (cit. on pp. 55, 64, 126).

[198] Zacharias Vamvakousis and Rafael Ramirez. "P300 Harmonies: A Brain-Computer Musical Interface". In: *Proc. 2014 Int. Computer Music Conf./Sound and Music Computing Conf.* Athens, Greece: Michigan Publishing, Sept. 2014, pp. 725–729 (cit. on pp. 55, 65, 126).

[199] Evan Balster. *Imitone: mind to melody*. https://web.archive.org/web/20211203061925/https://imitone.com/. 2021 (cit. on pp. 55, 66).

[200] Andreas Refsgaard. *Eye Conductor*. https://andreasrefsgaard.dk/projects/eye-conductor/. 2021 (cit. on pp. 55, 69).

[201] Julian Vogels. "Harmonica-Inspired Digital Musical Instrument Design Based on an Existing Gestural Performance Repertoire". PhD thesis. Montreal: Schulich School of Music, McGill University, 2015 (cit. on p. 60).

[202] Luca Turchet. "Smart Musical Instruments: Vision, Design Principles, and Future Directions". In: *IEEE access : practical innovations, open solutions* 7 (2019), pp. 8944–8963 (cit. on p. 74).

[203] Adnan Marquez-Borbon and Juan Pablo Martinez Avila. "The Problem of DMI Adoption and Longevity: Envisioning a NIME Performance Pedagogy". In: *Proc. of the 18th Int. Conf. on New Interfaces for Musical Expression (NIME'18)*. Blacksburg, Virginia, USA: Virginia Tech Libraries, June 2018 (cit. on pp. 75, 111).

[204] Douglas Schuler and Aki Namioka, eds. *Participatory Design: Principles and Practices*. CRC Press, 1993 (cit. on p. 75).

[205] Anthony Hornof, Troy Rogers, Jeffrey Stolet, and Tim Halverson. *Bringing to Life the Musical Properties of the Eyes*. Technical Report 08-05. University of Oregon Department of Computer and Information Science, 2008, p. 10 (cit. on p. 77).

[206] Anthony J. Hornof, Troy Rogers, and Tim Halverson. "EyeMusic: Performing Live Music and Multimedia Compositions with Eye Movements". In: *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*. NIME '07. New York, NY, USA: Association for Computing Machinery, June 2007, pp. 299–300 (cit. on p. 77).

[207] Zacharias Vamvakousis. "Digital Musical Instruments for People with Physical Disabilities". PhD thesis. Universitat Pompeu Fabra Barcelona, 2016 (cit. on p. 77).

[208] Robert J. K. Jacob. “Eye Tracking in Advanced Interface Design”. In: *Virtual Environments and Advanced Interface Design*. USA: Oxford University Press, Inc., 1995, pp. 258–288 (cit. on pp. 79, 81).

[209] Benno Gesierich, Angela Bruzzo, Giovanni Ottoboni, and Livio Finos. “Human Gaze Behaviour during Action Execution and Observation”. In: *Acta Psychologica* 128.2 (June 2008), pp. 324–330 (cit. on p. 79).

[210] Jeremy B. Badler and Alessandro Canossa. “Anticipatory Gaze Shifts during Navigation in a Naturalistic Virtual Environment”. In: *Proc. of the 2015 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY ’15)*. London, United Kingdom: Association for Computing Machinery, Oct. 2015, pp. 277–283 (cit. on p. 79).

[211] Chin Ian Lou et al. “Object Recognition Test in Peripheral Vision: A Study on the Influence of Object Color, Pattern and Shape”. In: *Proc. Int. Conf. on Brain Informatics*. Ed. by Fabio Massimo Zanzotto, Shusaku Tsumoto, Niels Taatgen, and Yiyu Yao. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 18–26 (cit. on p. 79).

[212] Kun Guo and Philip J. Benson. “Involuntary Eye Movements in Response to First- and Second-Order Motion”. In: *NeuroReport* 9.15 (Oct. 1998), pp. 3543–3548 (cit. on p. 80).

[213] Kenneth Holmqvist, Marcus Nyström, and Fiona Mulvey. “Eye Tracker Data Quality: What It Is and How to Measure It”. In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. ETRA ’12. New York, NY, USA: Association for Computing Machinery, Mar. 2012, pp. 45–52 (cit. on p. 80).

[214] Boris Velichkovsky, Mikhail A. Rumyantsev, and Mikhail A. Morozov. “New Solution to the Midas Touch Problem: Identification of Visual Commands Via Extraction of Focal Fixations”. In: *Procedia Computer Science* 39 (Dec. 2014) (cit. on p. 80).

[215] Boris Velichkovsky, Andreas Sprenger, and Pieter Unema. “Towards Gaze-Mediated Interaction: Collecting Solutions of the “Midas Touch Problem””. In: *Human-Computer Interaction INTERACT ’97: IFIP TC13 International Conference on Human-Computer Interaction, 14th–18th July 1997, Sydney, Australia*. Ed. by Steve Howard, Judy Hammond, and Gitte Lindgaard. IFIP — The International Federation for Information Processing. Boston, MA: Springer US, 1997, pp. 509–516 (cit. on p. 80).

[216] John Paulin Hansen, Anders Sewerin Johansen, Dan Witzner Hansen, Kenji Ito, and Satoru Mashino. “Command Without a Click: Dwell Time Typing by Mouse and Gaze Selections”. In: *Proc. Int. Conf. on Human-Computer Interaction (INTERACT ’03)*. Ed. by Matthias Rauterberg, Marino Menozzi, and Janet Wesson. Zurich, Switzerland: IOS Press, 1 (cit. on p. 81).

[217] Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. “Fast Gaze Typing with an Adjustable Dwell Time”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’09. New York, NY, USA: Association for Computing Machinery, Apr. 2009, pp. 357–360 (cit. on p. 81).

[218] P. Pfordresher and C. Palmer. “Effects of Delayed Auditory Feedback on Timing of Music Performance”. In: *Psychological Research* 66.1 (Feb. 2002), pp. 71–79 (cit. on p. 81).

[219] Filipe Calegario, Marcelo M. Wanderley, Stéphane Huot, Giordano Cabral, and Geber Ramalho. “A Method and Toolkit for Digital Musical Instruments: Generating Ideas and Prototypes”. In: *IEEE MultiMedia* 24.1 (Jan. 2017), pp. 63–71 (cit. on p. 88).

[220] David Johnson, Daniela Damian, and George Tzanetakis. "OSC-XR: A Toolkit for Extended Reality Immersive Music Interfaces". In: (), p. 8 (cit. on p. 88).

[221] Charles Roberts, Graham Wakefield, Matthew Wright, and JoAnn Kuchera-Morin. "Designing Musical Instruments for the Browser". In: *Computer Music Journal* 39.1 (Mar. 2015), pp. 27–40 (cit. on p. 88).

[222] Charles Roberts, Matthew Wright, JoAnn Kuchera-Morin, and Tobias Höllerer. "Rapid Creation and Publication of Digital Musical Instruments". In: (), p. 4 (cit. on p. 88).

[223] Charles Matthews. *Instrument-Maker*. https://instrumentmaker.org/. 2022 (cit. on p. 88).

[224] Luan Luiz Gonçalves and Flávio Luiz Schiavoni. "Creating Digital Musical Instruments with Libmosaic-Sound and Mosaicode". In: *Revista de Informática Teórica e Aplicada* 27.4 (Dec. 2020), pp. 95–107 (cit. on p. 88).

[225] Stanislav Popelka, Zdeněk Stachoň, Čeněk Šašinka, and Jitka Doležalová. "EyeTribe Tracker Data Accuracy Evaluation and Its Interconnection with Hypothesis Software for Cartographic Purposes". In: *Computational Intelligence and Neuroscience* 2016 (Mar. 2016), e9172506 (cit. on p. 91).

[226] Gregory Funke, Eric Greenlee, Martha Carter, Allen Dukes, Rebecca Brown, and Lauren Menke. "Which Eye Tracker Is Right for Your Research? Performance Evaluation of Several Cost Variant Eye Trackers". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 60.1 (Sept. 2016), pp. 1240–1244 (cit. on p. 91).

[227] Dirk de Klerk. "Equal Temperament". In: *Acta Musicologica* 51.1 (1979), pp. 140–150 (cit. on p. 93).

[228] D. Donnelle and B. Rust. "The Fast Fourier Transform for Experimentalists. Part I. Concepts". In: *Computing in Science Engineering* 7.2 (Mar. 2005), pp. 80–88 (cit. on p. 94).

[229] S. W. Roberts. "Control Chart Tests Based on Geometric Moving Averages". In: *Technometrics* 1.3 (Aug. 1959), pp. 239–250 (cit. on p. 94).

[230] Jim Tobias. "Information Technology and Universal Design: An Agenda for Accessible Technology". In: *Journal of Visual Impairment & Blindness* 97.10 (Oct. 2003), pp. 592–601 (cit. on p. 97).

[231] Gerrit Niezen, Parisa Eslambolchilar, and Harold Thimbleby. "Open-Source Hardware for Medical Devices". In: *BMJ Innovations* 2.2 (Apr. 2016) (cit. on pp. 97, 98).

[232] Daniel K. Fisher and Peter J. Gould. "Open-Source Hardware Is a Low-Cost Alternative for Scientific Instrumentation and Research". In: *Modern Instrumentation* 2012 (Apr. 2012) (cit. on p. 98).

[233] Joshua M. Pearce. "Building Research Equipment with Free, Open-Source Hardware". In: *Science* 337.6100 (Sept. 2012), pp. 1303–1304 (cit. on p. 98).

[234] Jérémy Bonvoisin, Robert Mies, Jean-François Boujut, and Rainer Stark. "What Is the "Source" of Open Source Hardware?" In: (2017) (cit. on p. 98).

[235] Open Source Hardware Association. *Open Source Hardware - Definition*. https://www.oshwa.org/definition/. 2021 (cit. on p. 98).

[236] Kerstin Balka, Christina Raasch, and Cornelius Herstatt. “The Effect of Selective Openness on Value Creation in User Innovation Communities”. In: *Journal of Product Innovation Management* 31.2 (2014), pp. 392–407 (cit. on p. 98).

[237] Steven Maupin, David Gerhard, and Brett Park. “Isomorphic Tessellations for Musical Keyboards”. In: *Proc. of 2011 Sound and Music Computing Conf.* Conservatorio Cesare Pollini, Padova, Italy, July 2011, pp. 471–478 (cit. on pp. 104, 107, 108).

[238] Sophia Stanford, Andrew J. Milne, and Jennifer MacRitchie. “The Effect of Isomorphic Pitch Layouts on the Transfer of Musical Learning †”. In: *Applied Sciences* 8.12 (Dec. 2018), p. 2514 (cit. on p. 108).

[239] Elena Rusconi, Bonnie Kwan, Bruno L. Giordano, Carlo Umiltà, and Brian Butterworth. “Spatial Representation of Pitch Height: The SMARC Effect”. In: *Cognition* 99.2 (Mar. 2006), pp. 113–129 (cit. on p. 108).

[240] Asha Ward, L. Woodbury, and Tom Davis. “Design Considerations for Instruments for Users with Complex Needs in SEN Settings.” In: *Proc. of the 17th Int. Conf. on New Interfaces for Musical Expression (NIME’17).* Copenhagen, Denmark, May 2017 (cit. on p. 111).

[241] Maria Francesca Roig-Maimó, Cristina Manresa-Yee, Javier Varona, and I. Scott MacKenzie. “Evaluation of a Mobile Head-Tracker Interface for Accessibility”. In: *Computers Helping People with Special Needs.* Ed. by Klaus Miesenberger, Christian Bühler, and Petr Penaz. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 449–456 (cit. on p. 126).

[242] Jeppe Veirum Larsen, Daniel Overholt, and Thomas B. Moeslund. “The Actuated Guitar: A Platform Enabling Alternative Interaction Methods”. In: *Proceedings of the Sound and Music Computing Conference 2013 (SMC 2013).* Logos Verlag Berlin, July 2013, pp. 235–238 (cit. on p. 126).

[243] Jeppe Veirum Larsen, Daniel Overholt, and Thomas B. Moeslund. “The Actuated Guitar: Implementation and User Test on Children with Hemiplegia”. In: *Proceedings of the International Conference on New Interfaces for Musical Expression.* Goldsmiths, University of London, 2014, pp. 60–65 (cit. on p. 126).

[244] Jacob Harrison, Alan Chamberlain, and Andrew P. McPherson. “Accessible Instruments in the Wild: Engaging with a Community of Learning-Disabled Musicians”. In: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems.* Glasgow Scotland Uk: ACM, May 2019, pp. 1–6 (cit. on p. 126).

[245] Donald Balestrieri. *Registers of the Standard Stradella Keyboard.* http://www.accordions.com/articles/strad
1979 (cit. on p. 127).

[246] Pino Jodice. *Composizione, Arrangiamento e Orchestrazione Jazz.* Vol. 1. Morlacchi Editore, 2017 (cit. on p. 129).

[247] Tokihiko Kaburagi and Masaaki Honda. “An Ultrasonic Method for Monitoring Tongue Shape and the Position of a Fixed Point on the Tongue Surface”. In: *J. of the Acoustical Society of America* 95.4 (Apr. 1994), pp. 2268–2270 (cit. on p. 138).

[248] T. Scott Saponas, Daniel Kelly, Babak A. Parviz, and Desney S. Tan. “Optically Sensing Tongue Gestures for Computer Input”. In: *Proc. 22nd ACM Symp. on User Interface Software and Technology*. UIST ’09. New York, NY, USA: Association for Computing Machinery, Oct. 2009, pp. 177–180 (cit. on p. 138).

[249] Thomas Hueber, Elie-Laurent Benaroya, Gérard Chollet, Bruce Denby, Gérard Dreyfus, and Maureen Stone. “Development of a Silent Speech Interface Driven by Ultrasound and Optical Images of the Tongue and Lips”. In: *Speech Communication* 52.4 (Apr. 2010), pp. 288–300 (cit. on p. 138).

[250] Theodore C. Levin and Michael E. Edgerton. “The Throat Singers of Tuva”. In: *Scientific American* 281.3 (1999), pp. 80–87 (cit. on p. 139).