Musical Interaction Based on the Conductor Metaphor

Álvaro Sarasúa Berodia

TESI DOCTORAL UPF / 2017

Dirigida per:

Dra. Emilia Gómez i Dr. Enric Guaus

Departament de Tecnologies de la Informació i les Comunicacions

Universitat
Pompeu Fabra
Barcelona

A mi tío Javier.

Acknowledgements

The first people I want to thank are my thesis directors Emilia Gómez and Enric Guaus. Not only for their valuable guidance, but also and specially for their trust and support (and patience) during these years. I will always be grateful for giving me the chance to come back to the MTG.

This dissertation is written in first person of plural for a good reason. Multiple parts in it are co-authored by other researchers whose work I want to explicitly acknowledge. The system for control of articulation was developed together with Baptiste Caramiaux and Atau Tanaka during my stay at the EAVI group at Goldsmiths University of London. Mark Melenhorst helped designing and analyzing the *Becoming the Maestro* evaluation. Carles F. Julià has advised me on software development issues at various points in this work, and has been directly involved in developing the code for *Becoming the Maestro*. Julián Urbano and Perfecto Herrera have always been willing to help with advice for designing studies and experiments and analyzing their results. I have made extensive use of the *Repovizz* platform, and Óscar Mayor's help has been vital in this regard. Agustín Martorell has greatly helped providing annotations for symphonic music recordings. I am deeply grateful to all of them, since this work would not have been possible without their collaboration.

I would also like to thank all the rest of colleagues of the PHENICX project. Working with all of them has been a pleasure and a privilege. In particular, I want to thank the people at ESMUC, where I worked during the project, who collaborated to make this research possible.

If the MTG is such a great place it is because of the incredibly talented people who work and have worked there. I thank Xavier Serra and all the people I met here for such a great working environment and for making this PhD an enriching experience at the human level. I also worked at EAVI during three months; to Baptiste, Atau, and the rest of the great people I met there, my biggest gratitude for making my stay at London a great experience.

This thesis involved several studies and experiments where many people volunteered to participate. To all of them, my gratitude for their collaboration and valuable feedback.

It also involved some paperwork which was easier thanks to Lydia Garcia's great help and good vibes.

After PHENICX, I also found a great working environment at Voctro Labs. I want to thank Jordi Janer and Óscar Mayor for their trust and flexibility during these last months writing this dissertation.

Finally, I want to thank some people at a more personal level:

All my friends: the Sanse *crew* (Sergi and Guille, ¡al local!), my EUITT colleagues (Alberto, Dani, Esteban, Javi, Cerci, Otegui), and my brothers and sisters in Barcelona (Andrés, Zuriñe, Panos, Sebas, Juanjo, Marius, Fred, Sergio, John, Tim...).

My family: in particular my cousins Diego, Martín and J. Luis, my grandfather Julio, my sister Patricia and her husband David, and the newcomer Pablo.

My partner Marián for her support in moments of doubt, but specially for filling these years with joy.

And above all, my deepest gratitude goes to my parents, Andrea and José Antonio, for their unconditional love and support.

This work was supported by the European Union Seventh Framework Programme FP7 / 2007-2013 through the PHENICX project (grant agreement no. 601166).

Abstract

Interface metaphors are often used in Human Computer Interaction (HCI) to exploit knowledge that users already have from other domains. A commonly used one in Digital Musical Instruments (DMIs) is the conductor metaphor. The simple idea behind it is to turn the computer into an orchestra that the user *conducts* with movements that resemble those of a real orchestra conductor. So far, many refinements have been proposed to provide more accurate or expressive control over different musical parameters. However, even though the orchestra conducting metaphor offers a good case for investigating several aspects of gesture-based interaction, the way in which users interact with these interfaces has not been explored in depth to improve their usability. The availability of commercial depth-sense cameras, which has stimulated the apparition of new DMIs based on this metaphor, also makes this missing in-depth exploration easier. This dissertation offers such analysis.

We theorize that part of the knowledge that users have from the domain that the interface metaphor replicates is user-specific. In this context, we argue that systems using an interface metaphor can see their usability improved by adapting to this user-specific knowledge. We propose strategies to design motion-sound mappings for DMIs that draw upon the conductor metaphor by adapting to personal nuances that can be automatically computed from spontaneous conducting movements.

For this, we first analyze the performance of a professional conductor in a concert, identifying descriptors than can be computationally extracted from motion capture data and that describe the relationships between the movement of the conductor and specific aspects of the performance potentially controllable in an interactive scenario. Then, we use these same techniques to build two systems that adapt to user-specific tendencies in two contexts. The first one allows to control tempo and dynamics with adaptations learned from analyzing conducting movements performed on top of fixed music. The second one provides control over articulation through gesture variation, the mapping being defined by each user through gesture variation examples. In both cases, we perform observation studies to guide the interface design and user studies with participants of different musical expertise to evaluate the usability of the systems.

In addition to the above, we study the potential of the conductor metaphor in a gaming context as a mean to raise interest for classical music. We developed *Becoming the Maestro*, a game that exploits state-of-the-art technologies that allow to interact with symphonic music content in new ways. We also perform a user study which shows the potential of the game to increase curiosity for classical music.

In summary, this thesis offers an in-depth exploration of interaction with interfaces based on the conductor metaphor, proposing strategies to improve their usability that span to other interface metaphor cases and to gesture-based interaction in general. These contributions are complemented by the data collected in all observation studies, which is made publicly available to the community.

Sinopsis

Las metáforas de interfaz se utilizan habitualmente en Interacción Persona-Ordenador (HCI, por sus siglas en inglés) para explotar el conocimiento que los usuarios ya tienen de otros dominios. Una comúnmente utilizada en la construcción de Instrumentos Musicales Digitales (DMIs, por sus siglas en inglés) es la metáfora del director. Básicamente, la idea en este caso es convertir el ordenador en una orquesta que el usuario dirige utilizando movimientos similares a los de un director de orquesta real. Hasta ahora, se han propuesto distintas mejoras para proveer de control más preciso y expresivo sobre distintos parámetros musicales. Sin embargo, a pesar de que la metáfora del director ofrece un buen caso de uso para investigar distintos aspectos de la interacción basada en gestos, la manera en que los usuarios interactúan con estas interfaces no ha sido aún explorada en profundidad para mejorar su usabilidad. La disponibilidad actual de cámaras con sensores de profundidad, que han estimulado la aparición de DMIs basados en esta metáfora, también hace más sencillo realizar esta exploración en profundidad. Esta disertación ofrece tal análisis.

Teorizamos que parte del conocimiento que los usuarios tienen del dominio replicado por la metáfora es específico para cada usuario. En este contexto, sugerimos que un sistema que hace uso de una metáfora puede mejorar su usabilidad si se adapta a dicho conocimiento específico del usuario. Proponemos estrategias para diseñar mapeos entre movimiento y sonido para DMIs construidas sobre la metáfora del director mediante la adaptación a los matices personales que se pueden analizar a partir de movimientos de dirección espontáneos, hechos sin instrucciones concretas.

Para esto, primero analizamos la actuación de un director profesional en un concierto, identificando descriptores que se pueden obtener a partir de datos de captura de movimientos y que describen la relación entre el movimiento del director y aspectos específicos de la actuación potencialmente controlables en un escenario interactivo. A continuación, utilizamos estas mismas técnicas para constuir dos sistemas que se adaptan a tendencias específicas de los usuarios en dos contextos. El primero permite controlar el tempo y la dinámica con adaptaciones aprendidas de movimientos de dirección realizados sobre una música fija. El segundo permite controlar la articulación musical mediante variaciones expresivas de gesto, siendo el usuario quien explícitamente define el mapeo mediante ejemplos de sus gestos. En ambos casos, comenzamos con estudios observacionales que guían el diseño de la interfaz y realizamos estudios de usuario para evaluar la usabilidad de los sistemas propuestos.

Además de lo anterior, estudiamos el potencial de la metáfora del director en un contexto de juego como un medio para incrementar el interés por la música clásica. Hemos desarrollado *Becoming the Maestro*, un juego que explota las últimas tecnologías desarrolladas en el ámbito de este tipo de música para interactuar con ella de nuevas maneras. En este caso también realizamos un estudio de usuario que muestra el potencial del juego para incrementar el interés por la música clásica.

En resumen, esta tesis ofrece un estudio en profundidad de la interación con interfaces basadas en la metáfora del director, proponiendo estrategias para mejorar su usabilidad que son de aplicabilidad en otras metáforas de interfaz y, en general, en interaccción basada en gestos. Estas contribuciones se complementan con los datos recopilados en los estudios observacionales, que se ponen a disposicion pública para la comunidad.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

ACM	Association for Computing Machinery
ANN	Artificial Neural Networks
BPM	Beat Per Minute
CCA	Canonical Correlation Analysis
DMI	Digital Musical Instrument
EM	Expectation-Maximization
ESMUC	Escola Superior de Música de Catalunya
fps	frames per second
GUI	Graphical User Interface
GVF	Gesture Variation Follower
HCI	Human Computer Interaction
HMM	Hidden Markov Models
IML	Interactive Machine Learning
IR	infrared
ISMIR	International Society for Music Information Retrieval
LDA	Linear Discriminant Analysis
LMA	Laban Movement Analysis
ML	Machine Learning
MoCap	Motion Capture

NIME	New Interfaces for Musical Expression
OSC	OpenSound Control
PCA	Principal Component Analysis
PHENICX	Performances as Highly Enriched aNd Interactive Concert Experiences (European Research Project)
SDK	Software Development Kit
SVM	Support Vector Machines
TSV	Tab-Separated-Values (file format)
UI	User Interface
UTAUT	Unified Theory of Acceptance and Use of Technology

Mathematical Symbols

$\mathbf{a}$	acceleration
a	acceleration magnitude
$\mathbf{b}^a$	beat annotation
$\mathbf{b}^p$	beat prediction
CI	Contraction Index
$\mathbf{CoM}$	Center of Mass
Δt	sampling period
ϵ	error
f_s	sampling rate
$\mathbf{j}$	jerk
j	jerk magnitude
μ	mean
$\mathbf{p}$	position
QoM	Quantity of Motion
R^2_{adj}	Adjusted coefficient of determination

σ	standard deviation
$\mathbf{v}$	velocity
v	velocity magnitude
$\mathbf{x}$	body position

Chapter 1

Introduction

1.1 Motivation

The notion of *metaphor* is central in the human-computer interaction (HCI) discipline. An *interface metaphor* is a representation created to help the user understand the abstract operation and capabilities of the computer (Blackwell, 2006). In this representation, elements in the user interface (UI) mimic elements from a real-world scenario from which the user can transfer her knowledge. Probably, the most ubiquitous example is the *desktop* metaphor, which has been present in almost all PC operating systems since its apparition in the Xerox Alto and Star models developed in Xerox Palo Alto Research Center, and its popularization with Apple's Macintosh in 1984. In this metaphor, the representation and behavior of elements in the Graphical User Interface (GUI) mimic elements in an office desktop. For example, files are represented by paper icons that can be *moved* into directories represented by folder icons or deleted if *moved* to a paper bin icon. While files are not actually *moved* from one place to another, the metaphorical representation offers the user a way to identify the affordances (understood as the possible ways in which an element can be used) of elements represented in the metaphor[1]. In this sense, a good interface metaphor must give the user expectations that indeed correspond with the functioning of the system. As Erickson (1995) summarizes:

> *To the extent that an interface metaphor provides users with realistic expectations about what will happen, it enhances the utility of the system. To the extent it leads users astray, or simply leads them nowhere, it fails.* Erickson (1995, p. 73)

The way in which a system using an interface metaphor is expected to work is represented

[1] The "metaphor" and "affordance" concepts, central to HCI, are vastly discussed in the literature. It is out of the scope of this dissertation to discuss the terms in depth. Readers interested in such discussion may refer to Gaver (1991), Erickson (1995), McGrenere and Ho (2000) or Blackwell (2006).

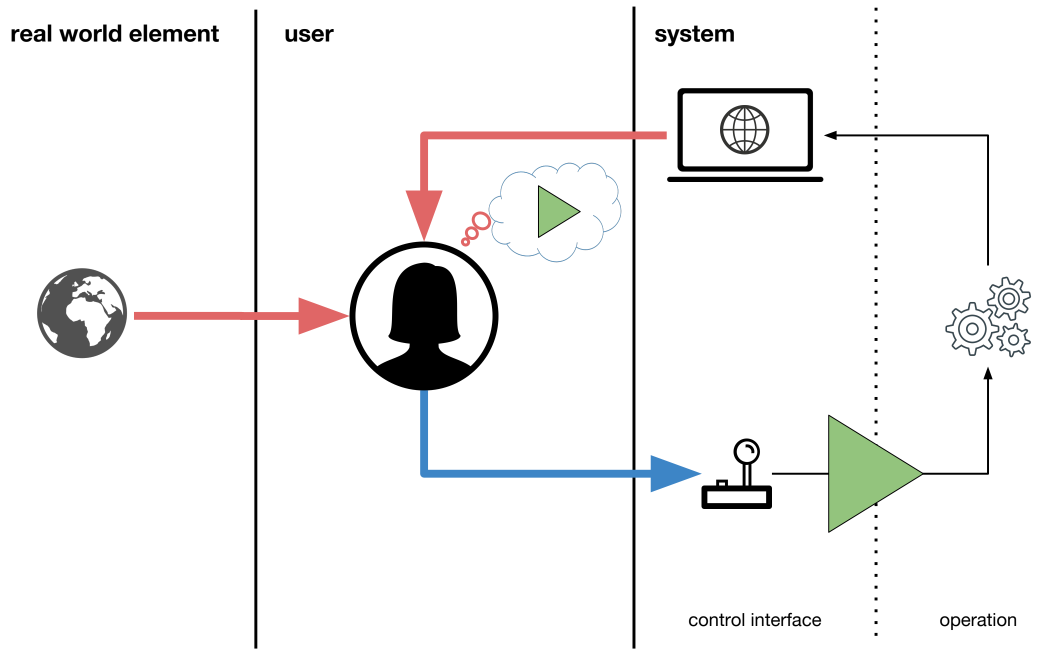


Figure 1.1: Representation of interaction with a system presenting an interface metaphor.

in Figure 1.1. The user interacts with the system through the control interface. This interface includes a representation of the state of the system (represented in the figure by a screen) and an input device (illustrated in the figure by a joystick). The actions that the user performs through the interface have an effect on the operation of the system, which in turn is reflected in the representation seen by the user. The effect that the user's actions have on the system is represented by the triangle that connects both parts. When an interface metaphor is used, the representation displayed to the user imitates elements of a real-world scenario, with the aim that the user can transfer her knowledge of this real-world scenario to the interaction with the system. The behavior that the user expects is represented by the triangle inside the bubble. In this representation, the interface metaphor succeeds when the shape of both triangles coincide, i.e. when the expected and actual behaviors match.

Interface metaphors are also extensively used in the design of digital musical instruments (DMIs). In traditional musical instruments, particularly before the piano, it is many times difficult to separate the control interface from the sound-generating mechanism (Wanderley, 2001; Jordà, 2005). For example, in wind instruments, the vibrating air that generates the sound is produced by the performer. In a violin, the performer directly excites the string to make it vibrate and produce sound. DMIs completely break this physical dependency. The control interface and the sound-generating system are separated, and the possibilities in both sides become unlimited. This separation makes *mapping* a central concept in DMIs. This term is used to define how the *input* (the movements or actions performed by the user) and *output* (the control parameters for the resulting sound) are connected (Paradiso, 1997; Rovan et al., 1997). In Figure 1.1, mapping would be illustrated by the shape of the triangle connecting the control interface and the inner functioning. In the context of DMIs, then, the goal of an interface metaphor is to correctly communicate the mapping to the user, i.e. the sonic results that her actions will have. Imagine, for example, a DMI consisting on a control interface with two sliders controlling an oscillator. The position of one of them is *mapped* to the frequency of the oscillator, and the velocity of the other is *mapped* to the amplitude. While this mapping is of course easy to learn through experimentation, we could expect a user to have an immediate intuition of it if we indicate her that the instrument is controlled *as a violin*: the first slider corresponding to the position of the left hand on the fingerboard, the second one replicating the action of the bow.

The possibilities of mapping in DMIs are however not limited to replicate the schemes of traditional instruments. Actions can also be mapped to control high-level properties of music (e.g. key, tempo, timbre, instrumentation...). In addition, actions can have different effects depending on the moment, either because the instrument allows the user to switch between different control modes, or because it reacts differently depending on the musical context. Considering this, the possible ways to interact with DMIs go beyond what traditionally has been understood as "playing an instrument" (Jordà, 2007). Pressing (1990) identifies four traditional music making metaphors that can describe these different ways to interact with DMIs:

- playing a musical instrument
- conducting an orchestra
- playing together with a machine
- acting as a one man band

Note that the "metaphors" Pressing refers to do not correspond to interface metaphors as defined above. These metaphors describe different ways to interact with DMIs in terms of what the user controls. For example, the aforementioned example of the DMI consisting on two sliders controlling an oscillator would be equivalent to playing a musical instrument. Another DMI where the user controlled the tempo and dynamics of a performance with a predefined score would fall in the "conducting an orchestra" metaphor category. The reason why we mention these metaphors introduced by Pressing is precisely because, in many occasions, these traditional music making metaphors greatly influence interface metaphors used in the design of control interfaces for DMIs. Coming back again to the toy example of the DMI with two sliders and one oscillator, a possible interface metaphor could be as simple as labeling each slider as "fingerboard" and "bow". In this case, the interface metaphor would work, similarly to the case of the desktop metaphor, through iconic representations. A paper bin icon indicates the user that elements moved there will be eliminated; two sliders labeled as "fingerboard" and "bow" indicate that moving the second one will produce sounds whose pitch depends on the position of the first. The interface metaphor can be however more explicit, mimicking the real-world element. In the case of the example, we could imagine a violin-shaped object with both sliders placed in the locations corresponding to the fingerboard and the bridge. Here, we would not only expect the user to have an intuition on the effect of both sliders, but also to grab the instrument in a specific way. As we see, the purpose of interface metaphors is to work as a sort of implicit instructions manual.

The design of new musical controllers is the central interest of the New Interfaces for Musical Expression (NIME[2]) community, gathered around the homonymous annual international conference since 2001. So far we have talked about interface metaphors inspired in traditional music making activities. However, before moving forward and focusing on the conductor metaphor, we would like to point out that there are other possible uses of interface metaphors for DMIs. NIME researchers have, for example, used everyday objects as control interfaces. This option not only helps the user to have an intuition on how to interact with the instrument, it also opens novel music performance paradigms. Browsing through the proceedings of the NIME conference, we can find control interfaces that use real-world objects as different as soap bubbles (Berthaut and Knibbe, 2014) or sponges (Marier, 2014), as well as interfaces inspired by other real-world activities such as drawing (Barbosa et al., 2013).

1.1.1 Why the conductor metaphor?

In this dissertation, we focus on the specific case of systems using the conductor interface metaphor. And, more specifically, we deal with the case where the control interface allows the user to perform conducting movements (i.e. the metaphor is not presented iconically, but explicitly). There are different reasons why we consider worth studying this case in depth.

[2] http://www.nime.org/

First of all, it provides a scenario where most users have some knowledge that they can turn into expectations about how to interact with the system. For the metaphor to be successful, these "expectations" should match the actual functioning of the system. A challenge appears when part of the knowledge from the real-world domain that guides the user's expectations is user-specific. We believe that systems based on the conductor metaphor are precisely a good use case to tackle this problem.

Second, there is a growing interest for using state-of-the-art technologies to enrich experiences around classical music. This interest was at the heart of the PHENICX[3] project, within which most of the work of this thesis was carried out. The project focused on symphonic classical music, where the figure of the conductor is essential. In this context, exploring the conductor interface metaphor becomes particularly relevant "to devise ways of directly interacting with performances via gestures", as introduced in the paper presenting the project (Gómez et al., 2013).

Third, the apparition of new easily-accessible sensors for motion tracking, and particularly depth-sense cameras (popularized by Microsoft Kinect), makes an in-depth exploration of interaction with this kind of systems easier. In fact, these sensors have stimulated the apparition of new systems based on this metaphor, which also reinforces the relevance of this work.

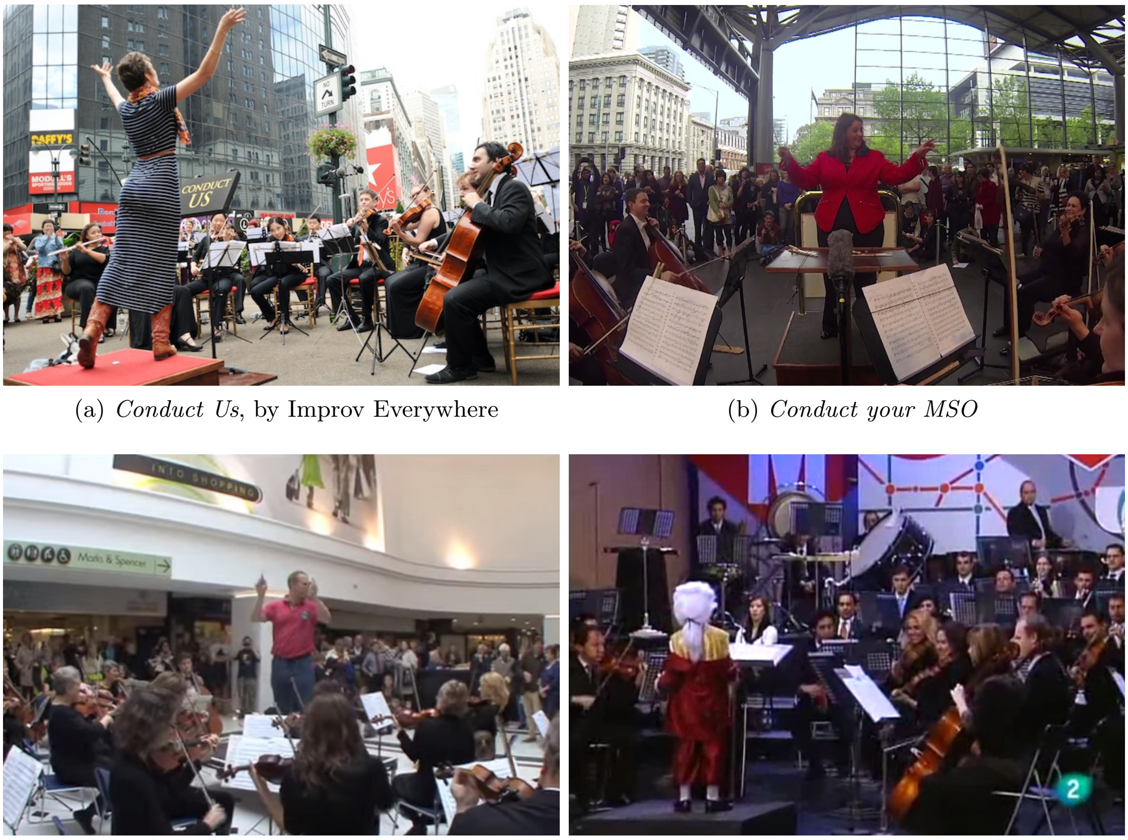
In the following, we discuss these three ideas in greater depth.

"Anyone can conduct"

This claim, without any nuances, would be highly controversial and, surely, highly inaccurate. Of course, musical conducting is a very complex art that requires years of training. However, the idea that "anyone can conduct" is sometimes implicit in popular culture. One of the most popular recent examples of this might be the 2013 viral video *Conduct Us*[4] by the comedy collective Improv Everywhere. A picture from this action is shown in Figure 1.2a. In the video, the Carnegie Hall orchestra appears in the middle of the street in New York with an empty podium in front of the musicians reading "Conduct Us". Then, different people (clearly not professional conductors) take the baton and just start conducting. The orchestra, which had previously made some specific rehearsals for the shooting, is interestingly able to play in a way that, even if probably not musically interesting, is coherent with the performed movements and fun for the *conductor*. It is not the only example. The exact same idea appears in actions

[3] Performances as Highly Enriched aNd Interactive Concert Experiences. http://phenicx.upf.edu/

[4] https://improveverywhere.com/2013/09/24/conduct-us/



(a) *Conduct Us*, by Improv Everywhere

(b) *Conduct your MSO*

(c) *Conduct Us*, by Gloucestershire Symphony Orchestra

(d) *El conciertazo*

Figure 1.2: Different examples of promotional activities by orchestras or institutions based on allowing random people to *conduct* a real orchestra.

by some orchestras such as the Melbourne Symphony Orchestra[5] or the Gloucestershire Symphony Orchestra[6], as shown in Figures 1.2b and 1.2c, respectively. As a more local reference, we also mention the Spanish public TV educational show for children *El conciertazo*[7] ("The great concert"), which aired 2000-2009. This show, presented by the late classical music disseminator Fernando Argenta, often had a section where children were allowed to conduct the orchestra[8]. Figure 1.2d shows a child conducting during the show.

[5] http://www.producermike.com/conduct-your-mso/

[6] http://www.gloucestershiresymphony.org.uk/conduct-us-gso-flashmob/

[7] http://www.rtve.es/alacarta/videos/el-conciertazo/

[8] As a matter of fact, the show previously existed as a school activity in Madrid National Auditorium. I attended once and was not chosen to conduct even though I firmly raised my hand.

In 2008, BBC Two started airing *Maestro*[9], a talent-show with celebrities competing during eight weeks to conduct the BBC Concert Orchestra at the *Proms in the Park* concert after a week-long preparation in a "Baton Camp". The show had new versions in Sweden and the Netherlands, with particularly great success in the latter.

In all these cases, the underlying idea that "anyone can conduct" (either immediately or with little practice) is somehow present. This notion is what, in our opinion, makes systems based on the conductor metaphor a particularly interesting use case to investigate aspects about interaction with systems that use interface metaphors. The idea that "anyone can conduct", in this context, can be nuanced and developed in more detail. What is relevant to the case at hand is the fact that most people, if given the chance to conduct, have some intuitive idea on what actions they can perform and what effects they can expect from them. Accordingly, systems that replicate the conducting activity (i.e. where the user guides the performance of a virtual orchestra through conducting movements) will succeed at providing a large amount of users with expectations about how it works.

Now, as we have discussed above, these expectations must correspond to the actual functioning of the system for the metaphor to succeed in improving usability. A challenge arises when these expectations can have specific nuances for each user. For example, we can expect that most users interpret that they must indicate the tempo to the orchestra with their movements. But will they use exactly the same gesture to communicate the beat? Or, if they want to give indications for loudness variations, will they do it in the same way? Putting it in the terms mentioned above in the context of interaction with DMIs: will the expected mapping vary across users? Throughout this dissertation we show specific scenarios with systems using the conductor metaphor where this is the case, and we investigate strategies to tackle the problem.

Attracting new audiences to classical music: this thesis in the context of the PHENICX project

The PHENICX project (Gómez et al., 2013) (2013-2016) aimed at enriching traditional concert experiences by using state-of-the-art multimedia and internet technologies. The focus was on Western classical music in large ensemble settings, and the main motivation was to make this kind of music appealing to broader audiences. Classical music suffers from very strong audience stereotypes, and is usually perceived as a complex and possibly boring genre. The idea of the project, in this context, was to use state-

[9] http://www.bbc.co.uk/musictv/maestro/

of-the-art technologies to fight these stereotypes by providing appealing experiences to broader audiences. There were mainly four areas of development: multimodal musical piece analysis, multimodal musical performance analysis, profiling and personalization, and exploration and interaction. The main focus of this thesis is the interaction area, dealing with "interactive systems for conductor impersonation" (Gómez et al., 2013, p. 801). In addition, this thesis also contributes to multimodal performance analysis since, as we see in Chapter 3, we analyze conducting movements during performance.

Precisely because one of the main motivations of PHENICX was to attract new audiences to classical music, the work developed in this thesis is influenced by the concrete use cases envisioned for this purpose: mainly public installations for museums or concert halls and games to interact with classical music content via conducting movements.

Other outcomes of PHENICX can be explored in the project's academic and commercial websites: http://phenicx.upf.edu/ and http://phenicx.com/.

The irruption of the Kinect depth sensor

Microsoft Corp. released the Kinect sensor to the market in November 2010 as an input device for the Xbox 360 console. It soon became a great success, selling over 8 million units in its first two months in the market[10]. It was the first successful commercial device that allowed to interact with video games without the need to touch a controller. The popularity of the Kinect comes, in great part, from its ability to perform human skeletal tracking, i.e. tracking the 3-D position of several body joints.

The Kinect was not only a great success in the gaming industry. It has also been extensively used in a wide variety of domains such as language recognition (Zafrulla et al., 2011), therapy (Cornejo et al., 2012; Abdur Rahman et al., 2013; Huang and Jun-Da, 2011), remote learning (Trajkova and Cafaro, 2016) or traditional computer vision tasks. Jungong Han et al. (2013) offer a comprehensive review of the use of Kinect in this domain. It is also commonly used for artistic purposes, either as a sensor for performance analysis (Alexiadis et al., 2011; Hadjakos and Grosshauser, 2013) or as an input device (Rodrigues et al., 2013; Lewis et al., 2012; Diakopoulos et al., 2015; Bacot and Féron, 2016). To illustrate the impact of the Kinect in research, we checked the number of entries that a search for the term "kinect" produces in the Association for Computing Machinery (ACM) Digital Library[11]. At the moment of publication of this dissertation,

[10] "Microsoft's Kinect Selling Twice As Fast As The iPad". http://www.huffingtonpost.com/2010/11/30/kinect-selling-twice-as-fast-as-ipad_n_789752.html, accessed April 3rd 2017

[11] http://dl.acm.org/

the search returns 695 papers (397 within the "interaction" category). Searching for "kinect" in Google Scholar[12] gives 77900 results.

The Kinect also made an impact on the NIME community, and has been present in its annual conference through different works since 2011. For example, it has been used in ways that resemble the interaction with video games: facing a screen and using the tracked hand positions to move elements represented on it. One of the earliest examples of this is *Crossole,* by Sentürk et al. (2012), where the performer controls the music by manipulating crossword blocks represented on the screen. It is also common to use the Kinect for augmenting traditional instruments. For instance, Trail et al. (2012) track the position of mallets for an augmented pitched percussion instrument, while Yang and Essl (2012) augment a piano keyboard tracking hand positions away from the keys. Finally, there are cases where the user can interact with imaginary "objects", either following the air-instruments scheme (Fan and Essl, 2013) or creating more abstract virtual objects (Jensenius, 2007).

In the following Chapter, where we provide a comprehensive review of systems using the conductor metaphor (Section 2.5.2), we see how the Kinect has led to the emergence of new systems of this kind.

In summary, not only the availability of this kind of sensor makes an in-depth exploration of interaction with systems based on the conductor metaphor easier. Also, its influence on the emergence of such systems reinforces the relevance of this exploration.

During the course of this work, Microsoft released the second version of the Kinect, for the Xbox One console. This new version provides some improvements in terms of resolution, tracking accuracy and latency. We used both devices during this time, and in this dissertation we indicate which one is used in each case referring to them as Kinect V1 and Kinect V2.

1.2 Objectives and methodology of the thesis

The objectives of this thesis can be divided into three areas: learning from real conductors, adapting to user-specific expectations, and attracting new audiences to classical music. In the following, we discuss in more detail the objectives of each area and we introduce the overall methodology followed in each case.

[12] https://scholar.google.com/

1.2.1 Learning from real conductors

A detailed analysis of music conducting is out of the scope of this thesis. However, we must not lose sight of the fact that this is the activity that inspires the metaphor that is the center of our research. For this reason, one of our objectives is to establish which computational analysis of conductor task during a performance can be relevant and useful for interactive systems based on this metaphor. In addition, by performing this analysis with the same device used during the interaction, we can identify relevant descriptors that can be computed from the data it provides in a conducting scenario.

During the PHENICX project, this work was developed at ESMUC (*Escola Superior de Música de Catalunya* - Catalan Higher School of Music). This allowed us to attend rehearsals and lessons, as well as to interview conducting teachers and students.

Taking advantage of this, we follow two steps:

- First, we conduct an interview with professional conductors and students to get their impression in this regard. More specifically, we ask them about the causal relationships that can be established and analyzed during a performance between the conductor's movements and the resulting music.
- Then, we carry out a multimodal recording of a real performance, including the movements of the conductor captured by a Kinect. Based on the conclusions of the interview, we computationally analyze the recording focusing on two aspects: the musicians' synchronization with conductor's hand movements and the relationship between some body movement descriptors and the overall loudness of the performance.

1.2.2 Adapting to user-specific expectations

As discussed in the previous section, one of the central problems addressed in this thesis is that of user-specific expectations that may arise when an interface makes use of a metaphor. DMIs based on the conductor metaphor offer a good use case for investigating such problem.

We illustrate this problem in Figure 1.3. In the right-hand side, a system with a control interface using an interface metaphor is represented. This system has a given mapping from user's actions to the outcome. As in Figure 1.1, this mapping is represented by the shape of the green triangle connecting the control interface and the system's operation. The goal of the metaphor is to make users expect a mapping as close as possible to the actual one. However, the expectations that each user creates from transferring their

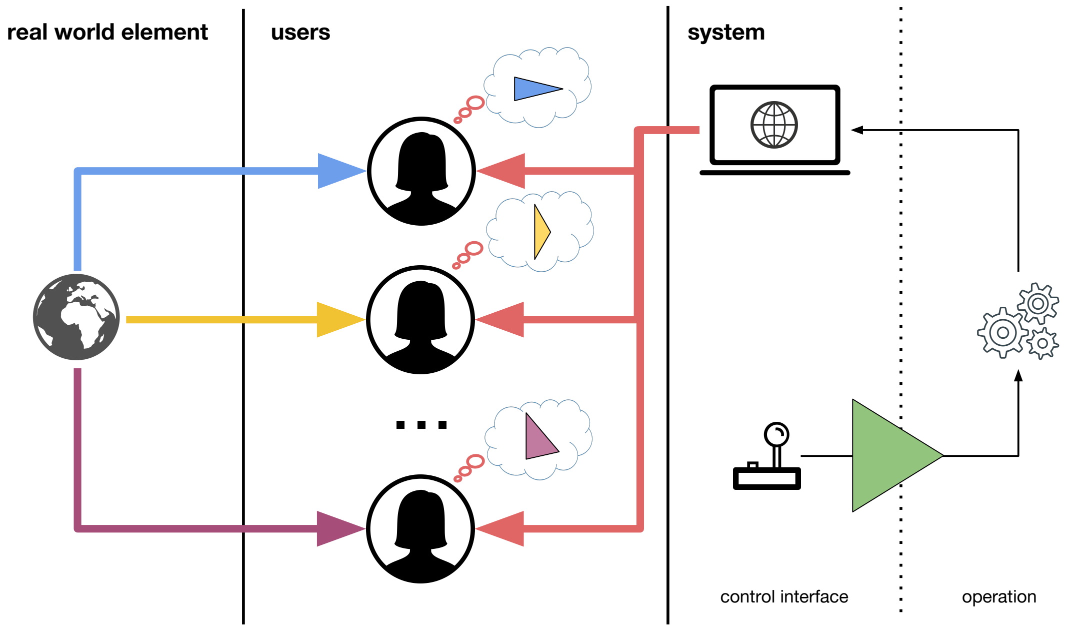


Figure 1.3: User-specific expectations from an interface metaphor.

knowledge of the real-world element represented in the metaphor may differ. In the figure, this is represented by the differently-shaped triangles for each user.

In this work, we deal with this problem in two different scenarios of DMIs using the conductor metaphor. In both cases, detailed below, we follow two steps. First, we perform observational studies to identify whether these specific user expectations exist and, if so, how they are concretely reflected in the captured data. Second, we develop concrete strategies to allow the system to exploit these differences, and carry out user tests to evaluate its performance.

Controlling tempo and dynamics: parameter-tuning from spontaneous movements

The first case we investigate is a DMI where the user can control tempo and dynamics by means of conducting movements. Following the method and conclusions of professional conductor analysis, we conduct a study where we present different musical fragments to a group of participants, and ask them to perform the conducting movements that they would perform to make the orchestra sound as in those fragments.

The results of the study, as we will see in detail, suggest that what is observed in these

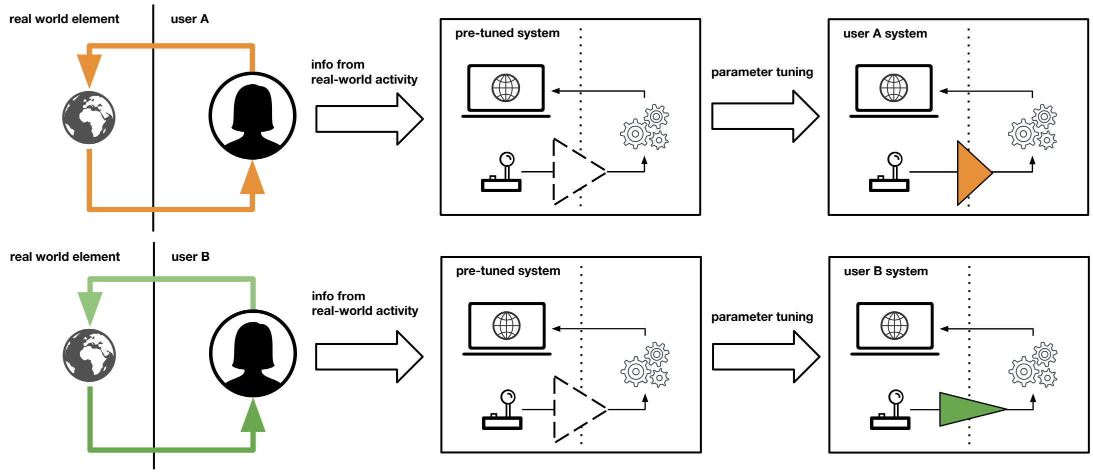


Figure 1.4: User-specific parameter tuning from observation of the real-world activity mimicked by the interface metaphor.

spontaneous (without any instructions) movements can be incorporated into the system to adapt the mapping to user-specific tendencies. Accordingly, the strategy we follow is that illustrated by Figure 1.4. The system is designed with a mapping in which some parameters can be adjusted in a specific way for each user. For this, in the first place, the user performs the activity that the metaphor replicates. In our case, the user makes spontaneous conducting movements on top of fixed music. The information of this activity is used to tune the system parameters, which results in a specific mapping for each user.

Controlling articulation: explicitly learning user expectations

The next case we investigate is a DMI that allows to control musical articulation through conducting movements. In a similar way to the previous case, we start with an observational study where we ask participants to perform conducting movements on top of fragments of the same melody interpreted with different articulation.

Following the conclusions of the study, we propose a different strategy. Instead of inferring the concrete parameters for each user from information obtained by observing spontaneous movements, we allow the user to explicitly define her own mapping by providing movement examples to the system. Figure 1.5 illustrates this strategy. In this case, the system has no predefined mapping, and needs to be trained. It is the user who, according to her expectations, explicitly defines the desired mapping.

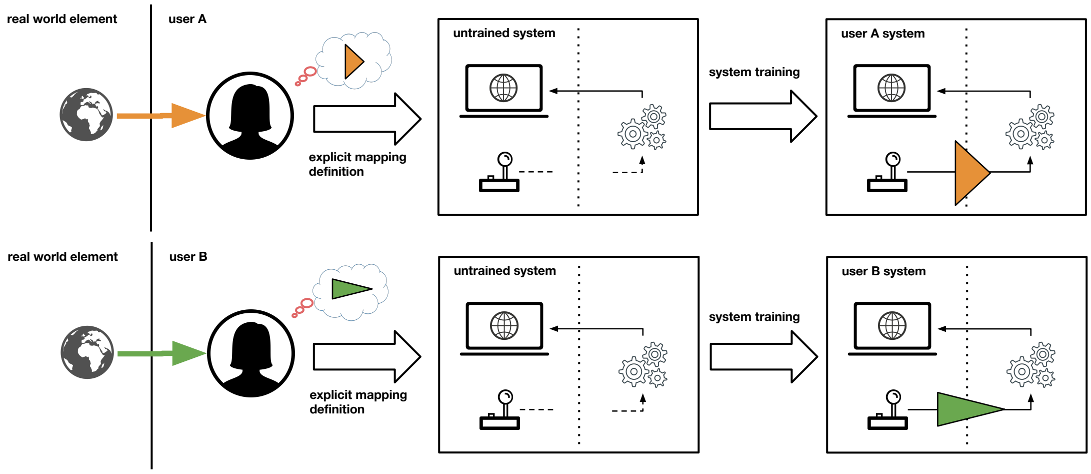


Figure 1.5: User-specific system training.

Learning about music-related movement

Investigating how users interact with DMIs based on the conductor metaphor also helps to understand certain aspects of music-related movement. We are interested in finding out how different musical parameters (beat, dynamics, articulation) are reflected in conducting movements performed by people with different musical expertise. The proposed methodology allows to study this from spontaneous movements performed *following* fixed music as well as in an interactive context.

1.2.3 Attracting new audiences to classical music

As we stated when introducing this project within the framework of PHENICX, one of the objectives of this thesis is to create experiences that can attract new audiences to classical music. In a way, this objective is transversal throughout the work presented in this dissertation, since the strategies discussed above are directly applicable in contexts such as museum installations.

However, we also pursue this goal in a more explicit way by investigating the potential of the conductor metaphor in a gaming context. For this, we developed *Becoming the Maestro*, a game specifically designed to create an attractive experience for these new audiences to which the project was intended, using the result of technologies developed during the project. According to this idea, the game is evaluated precisely in relation to its ability to provide the user with a fun experience and to raise interest for the music contained in the game.

1.3 Outline of the dissertation

In this section, we detail the structure of the rest of the dissertation in relation to the presented objectives.

In Chapter 2, *Background*, we offer a review of relevant literature for the different aspects covered in the thesis. First, we review feature extraction techniques from motion capture (MoCap) data and different strategies for motion-sound mapping. We also briefly discuss the operation of the capture devices used in this work (Kinect V1 and V2). Then, we focus on the case of the conductor. For this, we first offer a brief historical introduction to the figure of the conductor in Western classical music and we review relevant works that have computationally studied the effect of conducting movements on the performance and its perception by the audience. Finally, we make a comprehensive review of systems that have made use of the conductor metaphor.

Chapter 3, *Learning from real performances*, deals with the first area of the thesis: learning from real conductors. In this chapter, therefore, we present in detail the results of the interview with conductors and the analysis performed on the recorded performance. Also, we discuss some technical issues encountered when making multimodal recordings with Kinect, and we explain the strategy we followed to solve them.

The second part, adapting to user-specific expectations, is treated in the following two chapters, corresponding to each of the scenarios previously introduced.

Chapter 4, *Adapting to user-specific tendencies for beat and dynamics control*, focuses on the case of the system that allows to control tempo and dynamics. The first half of the chapter presents the observational study and its results. In the second half, the proposed system, whose mapping is adapted to each user, is explained together with the results of an experiment that compared its usability with a baseline with no user-specific mapping adaptation.

Chapter 5, *Learning user-specific gesture variations for articulation control*, deals with the use of conducting movements to control articulation. This chapter follows a structure similar to the previous one: the first half of the chapter presents the observational study and its results; in the second one, the proposed system and the user study performed to evaluate it are explained in detail.

Becoming the Maestro, the game developed to attract new audiences to classical music, is presented in Chapter 6, *Becoming the Maestro: a conducting game to enhance curiosity for classical music*. More specifically, the chapter frames the game within the PHENICX project, explains its mechanics and presents the results of the user evaluation carried out with its first prototype.

The document ends with chapter 7, *Conclusions*, where we summarize the work carried out in the thesis, its main contributions, and the possible directions for future research in these areas.

Chapter 2

Background

Throughout this chapter we review works relevant to the topics covered in the thesis. We begin by focusing on feature extraction techniques for Motion Capture (MoCap) data. Next, we see different techniques used for motion-sound mapping. We also assess some relevant technical aspects of the capture devices used in this work (Kinect). After that, we focus on the case of the conductor. For this, we first make a short historical review of the conductor figure within Western classical music and review some works that have analyzed the effect of conducting movements in performance and their perception from a computational point of view. Finally, we offer a comprehensive review of DMIs that have used the conductor metaphor.

2.1 MoCap feature extraction

Many of the works reviewed throughout this chapter use motion capture (MoCap) devices. In some of these works, they are used to record the movements of the conductor for its later analysis. In others, as input devices for DMIs. Feature extraction from this raw data is useful in both cases, as these features/descriptors[1] provide useful information. In the case of analysis, features help to interpret the results; in the case of interaction with DMIs, they offer greater possibilities for the mapping between movements performed by the user and the outcome of the system.

An important source for inspiration in the definition of body movement descriptors is Laban Movement Analysis (LMA). LMA is a theoretical system for the observation, description and interpretation of human movement developed by the dance artist, choreographer and theorist Rudolf Laban (Newlove and Dalby, 2004). LMA is based on four components:

[1] In this dissertation, we use “feature” and “descriptor” indistinctly.

- The *body* component describes the spacial characteristics of the movement in terms of the body parts that are moving.
- The *space* component describes how the body moves in relation with the environment.
- The *shape* component describes how the shape of the body changes over time.
- The *effort* component describes the dynamics and energy of the movement. Effort is comprised of four subcategories in a bipolar scale: *weight* (strong-light), *time* (sudden-sustained), *space* (direct-indirect) and *flow* (fluid-bound).

In this section we review some of the most commonly used MoCap features. Even in cases where these features have not been defined explicitly based on LMA, it is possible to situate them in some of its categories. This will help us to understand the kind of information provided by each descriptor. In the following, we first define the representation used in this section and throughout the thesis for raw MoCap data. Next, we list some descriptors that can be obtained from the position of a single point or body joint and some others that can be obtained by combining information from several joints. We refer to them as **joint descriptors** and **body descriptors**, respectively. At the end of the section, we discuss some issues about real-time computation of MoCap descriptors and we briefly review some existing frameworks for MoCap data analysis, also introducing our own framework developed during this work.

2.1.1 MoCap data representation

MoCap devices provide the position of K body joints at regular time intervals. The device sampling rate, f_s, establishes the number of data frames contained in 1 second. For example, Kinect devices work at $f_s = 30$ Hz or frames per second (fps). The time corresponding to a frame i is denoted t_i. Accordingly, two consecutive frames appear at times t_{i-1} and t_i and are separated by the sampling period $\Delta t = {}^{1}/_{f_s} = t_i - t_{i-1}$.

Body movement is represented by a sequence of joint positions[2] of the skeleton over a period of time. The position of the body at a time t_i (commonly referred to as *pose*) is defined by the position of a joints set $\mathbb{K}$ of size K:

$$\mathbf{x}(t_i) = \{\mathbf{p}^1, \mathbf{p}^2, ..., \mathbf{p}^K\}(t_i) \tag{2.1}$$

[2] Some MoCap devices also provide the orientation of joints. Since this is not the case in most reviewed works, and to make our work generalizable to a greater number of cases, we focus on descriptors computed from positional data. See Larboulette and Gibet (2015) for a comprehensive review of MoCap descriptors including some based on joint orientation.

where $\mathbf{p}^k(t_i)$ corresponds to the position of the k_{th} joint at time t_i, composed by 3 components for each dimension, x, y, z:

$$\mathbf{p}^k(t_i) = \{x^k, y^k, z^k\}(t_i) \tag{2.2}$$

The pose of the body at a time t_i, $\mathbf{x}(\mathbf{t_i})$, is thus represented by a $3 \times K$ dimensional vector. Consequently, a motion is represented by a sequence of poses during n frames by a vector of $3 \times K \times n$ dimensions:

$$\mathbf{X} = \{\mathbf{x}(t_1), \mathbf{x}(t_2), ..., \mathbf{x}(t_n)\} \tag{2.3}$$

In some representations, a particular body joint is defined as the *root* joint. Its position is represented in absolute values, while the rest of the joints are represented in relative positions from it. See Müller (2007, p. 195) for a formal definition of this kind of representation.

The amount of time, in seconds, that n frames represent, depends on the sampling rate of the capture device. For example, 1 second captured by the Kinect V1 or V2, which operate at 30 fps, contains 30 frames.

In the following, we use superscript to indicate the joint to which the descriptor corresponds. Bold symbols are used for vectors and non-bold for scalar values.

2.1.2 Joint descriptors

Amongst the most usually computed descriptors we find the instantaneous velocity, acceleration, and jerk, and their magnitudes:

- **Velocity:** $\mathbf{v}^k(t_i) = \{v_x^k, v_y^k, v_z^k\}(t_i)$
- **Velocity magnitude / speed:** $v^k(t_i) = \sqrt{v_x^k(t_i)^2 + v_y^k(t_i)^2 + v_z^k(t_i)^2}$
- **Acceleration**: $\mathbf{a}^k(t_i) = \{a_x^k, a_y^k, a_z^k\}(t_i)$
- **Acceleration magnitude:** $a^k(t_i) = \sqrt{a_x^k(t_i)^2 + a_y^k(t_i)^2 + a_z^k(t_i)^2}$
- **Jerk**: $\mathbf{j}^k(t_i) = \{j_x^k, j_y^k, j_z^k\}(t_i)$
- **Jerk magnitude:** $j^k(t_i) = \sqrt{j_x^k(t_i)^2 + j_y^k(t_i)^2 + j_z^k(t_i)^2}$

These descriptors correspond to the first three time derivatives of the position and describe the dynamics of the movement. In terms of LMA, velocity, acceleration and jerk can be associated to the *weight effort, time effort* and *flow effort* categories, respectively

(Kapadia et al., 2013). Faster movements are stronger than slow ones; sustained movements show low acceleration values; fluid movements are reflected in low jerk values. The magnitudes describe these dynamics ignoring the direction of the movement.

Time derivatives can be computed in a variety of ways, the most straightforward consisting on taking the difference between consecutive values and dividing by the sampling period, Δt. For example, the velocity of the k_{th} joint at time t_i can be computed as

$$\mathbf{v}^k(t_i) = \frac{\mathbf{p}^{\mathbf{k}}(t_i) - \mathbf{p}^{\mathbf{k}}(t_{i-1})}{\Delta t} \tag{2.4}$$

Acceleration and jerk can be computed with similar calculations:

$$\mathbf{a}^k(t_i) = \frac{\mathbf{p}^k(t_{i+1}) - 2 \cdot \mathbf{p}^{\mathbf{k}}(t_i) + \mathbf{p}^{\mathbf{k}}(t_{i-1})}{\Delta t^2} \tag{2.5}$$

$$\mathbf{j}^k(t_i) = \frac{\mathbf{p}^k(t_{i+2}) - 2 \cdot \mathbf{p}^k(t_{i+1}) + 2 \cdot \mathbf{p}^{\mathbf{k}}(t_{i-1}) - \mathbf{p}^k(t_{i-2})}{2 \cdot \Delta t^3} \tag{2.6}$$

These differentiations however amplify noise present in the signal (Skogstad et al., 2012). This noise can be reduced considering samples in a wider time frame. For example, in the case of velocity, by considering the previous and next samples:

$$\mathbf{v}^k(t_i) = \frac{\mathbf{p}(t_{i+1}) - \mathbf{p}(t_{i-1})}{2 \cdot \Delta t} \tag{2.7}$$

However, this computation introduces one more sample delay, which is problematic for real-time applications. For higher order derivatives, reducing noise implicates adding even more delay. In subsection 2.1.4 we discuss these issues (handling noise and delay) for real-time computation of these descriptors with more detail.

For offline computation, delay is not an issue and noise can be dealt with in different ways. One common strategy in this scenario is to perform polynomial approximation to the data points centered around the point of interest, then analytically computing the derivative of the obtained polynomial (Luck and Toiviainen, 2006). Polynomial approximation consists on finding the coefficients $w_m, w_{m-1}, ..., w_0$ of a polynomial of order m, $p(x) = w_m x^m + w_{m-1} x^{m-1} + ... + w_1 x + w_0$ that for a combination of n points $(x_j, y_j), 0 < j < n$ minimize $\sum_{j=0}^{n} |p(x_j) - y_j|^2$.

These descriptors can also be combined to compute others. For example, the **acceleration along the movement trajectory** a_t can be obtained by projecting the acceleration vector on the direction of the velocity vector (Luck and Sloboda, 2008):

$$a_t(t_i) = \frac{\mathbf{a}(t_i) \cdot \mathbf{v}(t_i)}{v(t_i)^2}\mathbf{v}(t_i) \tag{2.8}$$

As we will see throughout this literature review, these descriptors are also commonly used to detect events in conducting movements. For example, using changes of sign in vertical velocity or maxima and minima in vertical acceleration to detect changes from downward to upward movement or vice versa.

The **curvature** c measures the rate at which the curve defined by the trajectory changes its direction. It can be computed from the magnitude of the cross product of velocity and acceleration and the velocity magnitude, as proposed by Camurri et al. 2004:

$$c^k(t_i) = \frac{|\mathbf{a}^k(t_i) \times \mathbf{v}^k(t_i)|}{v^k(t_i)^3} \tag{2.9}$$

This descriptor is commonly associated with the *smoothness* of the movement (Camurri et al., 2004; Larboulette and Gibet, 2015): sharp movements are reflected in high curvature values, while movements with trajectories approaching a straight line approach zero curvature. In this sense, this descriptor would also fall in the *flow effort* LMA category.

Statistics of all these descriptors can also provide meaningful information and be considered as descriptors themselves. For example, the mean and standard deviations of the velocity magnitude can be computed as:

$$v^k_{mean}(t_i, N) = \frac{1}{N}\sum_{j=0}^{N-1} v^k(t_{i-j}) \tag{2.10}$$

$$v^k_{std}(t_i, N) = \sqrt{\frac{1}{N}\sum_{j=0}^{N-1} (v^k(t_{i-j}) - v^k_{mean}(t_i, N))^2} \tag{2.11}$$

where N is the number of frames over which the mean and standard deviation values are computed. In the case of the velocity magnitude, for example, the mean value gives an idea of how fast the movement is in average during those frames (i.e. it is related to *weight effort*), while the standard deviation is related to the stability of the movement, with stable movements showing low standard deviation (i.e. it is related to *time effort*). Note that the formulas above compute the statistics at time t_i considering the $N - 1$ previous frames. For real-time computation, this would be the only option. For offline computation, it is preferable to use values before and after t_i, so that the computation reflects tendencies centered at the time of interest.

Some descriptors can also be computed describing the movement of a joint during a period of time. For example, the length or **size** of the movement trajectory from time t_S to t_E is related to the *shape* category, and it can be computed as the cumulative distance travelled by the joint during that time:

$$size^k(t_S, t_E) = \sum_{i=S}^{i=E-1} \sqrt{(x^k(t_{i+1}) - x^k(t_i))^2 + (y^k(t_{i+1}) - y^k(t_i))^2 + (z^k(t_{i+1}) - z^k(t_i))^2} \tag{2.12}$$

The **directness** of a trajectory between times t_S and t_E, $D(t_S, t_E)$, is related to the *space effort* category in LMA, which describes movements in terms of being direct or indirect. It can be computed as the ratio between the euclidean distance between starting and end points, $\mathbf{p}(t_S)$ and $\mathbf{p}(t_E)$ and the *size* of the trajectory:

$$D^k(t_S, t_E) = \frac{\sqrt{(x^k(t_E) - x^k(t_S))^2 + (y^k(t_E) - y^k(t_S))^2 + (z^k(t_E) - z^k(t_S))^2}}{size^k(t_S, t_E)} \tag{2.13}$$

2.1.3 Body descriptors

There is an unlimited number of descriptors that can be calculated by combining information from various joints. A very simple descriptor could be the distance from one joint to another (e.g. the distance between both hands or the distance between one hand and the torso). We denote this as d_k^l, for the distance between joints k and l. While simple, these descriptors can be suitable for particular applications. Here we mention some of the most commonly found descriptors in the works reviewed in this chapter, and thus most relevant for our own work.

The **Quantity of Motion** (QoM) of a joints set $\mathbb{K}$ of size K can be computed by averaging the velocity magnitude of all joints:

$$QoM(t_i) = \frac{1}{K} \sum_{k \in \mathbb{K}} v^k(t_i)$$

This descriptor is related to the *weight effort* category, describing the overall energy of the set. It is usually calculated on all tracked joints, and it can also be computed as a weighted average, where different joints have different relative importance.

In the same way, the acceleration and jerk information are combined to compute descriptors related to the categories of *time effort* and *flow effort* for the whole body (Hachimura et al., 2005; Kapadia et al., 2013):

$$\textit{Time Effort}(t_i) = \frac{1}{K} \sum_{k \in \mathbb{K}} a^k(t_i)$$

$$\textit{Flow Effort}(t_i) = \frac{1}{K} \sum_{k \in \mathbb{K}} j^k(t_i)$$

Different bounding shapes are commonly used for descriptors related to the LMA *shape* component. For example, the **bounding box** corresponds to the smallest rectangular parallelepiped containing all joints. Other shapes are used in an analogous way, as in the case of the **bounding sphere** and **bounding ellipsoid** (Larboulette and Gibet, 2015).

Camurri et al. (2000) define the **Contraction Index** (CI), that uses the **bounding box** and applies normalization to make values range from 0 (when all joints are kept tightly together) to 1 (when body limbs are fully stretched).

In fact, both **Quantity of Motion** and **Contraction Index** are descriptors which are also used to describe body movement from video recordings instead of MoCap data (Camurri et al., 2000; Jensenius, 2007). In the case of video, these can be computed by tracking the body silhouette. For QoM, by looking at the number of moving pixels in the silhouette; for CI, by comparing the total number of pixels inside the bounding region of the silhouette with the number of pixels covered by the silhouette itself (Camurri et al., 2004).

A common descriptor related to the LMA *body* component is the **Center of Mass** (**CoM**), which describes the unique point around which the body mass is equally distributed in all directions. It is estimated computing a weighted average of all joints positions, with the weights of each joint k, w_k, assigned according to an anthropomorphic model (Kapadia et al., 2013):

$$\mathbf{CoM}(t_i) = \frac{\sum_{k \in \mathbb{K}} w_k \mathbf{p}^k(t_i)}{\sum_{k \in \mathbb{K}} w_k}$$

Descriptors related to the *space* component inform about the movement in relation with the surrounding space. This is relevant for dancing movements (or other more general movements such as walking, running...) but not so much for conducting movement analysis. Aristidou and Chrysanthou (2014), for example, review a set of space descriptors related to the distance and area covered by the body.

Müller (2007) defines a set of binary descriptors[3] that are particularly suitable for retrieving movement templates stored in a database. These descriptors are based on relations between joints and work independently of the global position and orientation of the body, which is desirable for the mentioned retrieval application. The descriptors have a generic form, which is concreted by choosing specific joints and parameters. For example, one of the descriptors proposed by Müller, $F^{(j1,j2)}_{\theta,touch}$, determines whether the distance between two joints $j1$ and $j2$ is less than a value θ. For instance, $F^{(LH,RH)}_{0.01,touch}$ takes value 1 when the distance between the two hands (denoted LH and RH, respectively) is less than 0.01 m, and 0 otherwise. This descriptor would fall in the *shape* LMA category, as it describes the relative position between two joints. Another example feature is $F^{(j1,j2;j3)}_{\theta,move}$, which looks at the velocity of joint $j3$ relative to joint $j1$ and takes value 1 if the component of this velocity in the direction determined by $(j1, j2)$ is above θ. Müller mentions the example of $F^{(belly,chest;RH)}_{\theta,move}$, which tests whether the right hand is moving upwards or not. This descriptor falls in the *body* LMA category, since it describes which parts of the body are moving. Müller proposes a concrete combination of 39 descriptors of this kind to build a feature matrix to identify MoCap data recordings in a database.

Table 2.1 contains a summary of some of the mentioned descriptors and their correspondences to LMA categories.

Throughout this literature review we see works that make use of some of these descriptors. In our work, we use mainly those that allow to detect events in the movement of the hands to study conductor-orchestra synchronization and for event triggering in interaction, but we also explore in general descriptors that can provide information related to other aspects relating body movement and musical parameters.

2.1.4 Real-time computation of MoCap features

As we have seen, descriptors computed from joints position time derivatives are very relevant, since they are the basis of other descriptors related to dynamic aspects of the movement. However, real-time computation of these descriptors has some issues, since differentiation can introduce delay and amplify noise.

Differentiation acts as a high pass filter (Skogstad et al., 2012), which is a problem due to the fact that MoCap signal is concentrated in the lower part of the spectrum.

[3] Binary descriptors can take only two values: 0 or 1.

Table 2.1: Examples of reviewed MoCap descriptors for different LMA categories.

	Joint descriptors		Body descriptors	
LMA cat.	Name	Symbol	Name	Symbol
Body			*Joint moving in given direction*	$F_{\theta,move}^{(j1,j2;j3)}$
			Center of Mass	**CoM**
Shape	Movement size	*size*	*Joints touching*	$F_{\theta,touch}^{(j1,j2)}$
			Distance between joints k and l	d_k^l
			Contraction Index	CI
Weight effort	Velocity	$\mathbf{v}^k$	Quantity of Motion	QoM
	Velocity magnitude	v^k		
Time effort	Acceleration	$\mathbf{a}^k$	Time Effort	*Time Effort*
	Acceleration magnitude	a^k		
	Acceleration along trajectory	${a_t}^k$		
Space effort	Directness	D^k		
Flow effort	Jerk	$\mathbf{j}^k$	Flow effort	*Flow Effort*
	Jerk magnitude	j^k		
	Curvature	c^k		

Skogstad et al. asked a group of participants to perform 2 types of movements with their hands: first, moving as fast as they could (to determine the highest frequency reached in extreme cases); second, simulating they were controlling some application "with more *articulated* and *controlled* motion" (to determine the frequency band of interest in the general case). From these recordings, they identified that most of the content for normal movements is below 10 Hz, and below 15 Hz for the take with fast movements.

Raw MoCap data contains noise whose frequency content depends mainly on the device. For most cases, though, it is evenly distributed in all frequencies as white noise. In this sense, it is desirable to get rid of the noise above the frequencies containing actual MoCap data. Low pass filters can be applied to the data for this purpose. Differentiation from this filtered data would result in a less noisy signal. However, it is actually more optimal in terms of delay to use low pass differentiators (digital filters that perform differentiation and attenuate high frequency content) than to first perform filtering and then differentiating. Throughout this thesis, we use the filters proposed by Skogstad et al. for real-time computation of time derivatives. These filters are specifically designed for filtering MoCap data for musical applications, minimizing delay in the computation.

2.1.5 Existing MoCap feature extraction frameworks

The *MoCap Toolbox* is a set of functions for Matlab[4] developed at the University of Jyväskylä (Burger and Toiviainen, 2013). It was developed for research on music-related movement but is suitable for other areas as well. The toolbox includes a set of MoCap feature extraction routines including some of the aforementioned and others. Interestingly, it includes algorithms for computing the *eigenmovements* of a full body motion capture segment using principal component analysis (PCA). PCA is a data dimensionality reduction method that consists on projecting the original data onto orthogonal components that explain as much of the variance in the data as possible. PCA analysis has proved to be useful both for MoCap data analysis (Toiviainen et al., 2010; Tits et al., 2015) and musical performance (Bevilacqua et al., 2002).

EyesWeb, by Camurri et al. (2004), is a software that supports a wide number of input data streams including MoCap, video, audio or other analog inputs (e.g. physiological signals). The software includes modules that perform feature extraction from these different streams, including *QoM* and *CI* (available from video, 2D and 3D MoCap data). It includes modules that specifically support some devices such as the Kinect V1 and supports a variety of standards including OpenSound Control (OSC) and MIDI.

Jensenius (2007) developed the *Musical Gestures Toolbox* including some of the features in *EyesWeb* as a set of Max[5] modules for the *Jamoma* framework (Place and Lossius, 2006). This toolbox, which includes several modules for visualizing motion, is however limited to work with video data.

Two more platforms were released during the realization of this thesis. *Mova* (released as a prototype), by Alemi et al. (2014), is an interactive movement analytics framework for feature extraction, feature visualization, and analysis of human movement data. Interaction with the platform mainly consists in choosing the features to visualize. It is in this sense a platform more oriented to online movement visualization and exploration.

More recently, Tilmanne and D'Alessandro (2015) released *MotionMachine*, a C++ software toolkit for rapid prototyping of MoCap features particularly targeted for motion-based interaction design. It includes a layer for visualization based on the OpenFrameworks[6] environment. It allows to provide input data in a flexible format, working both in offline and real-time scenarios.

[4] https://www.mathworks.com/products/matlab.html
[5] https://cycling74.com/products/max/
[6] http://openframeworks.cc/

MoDe: a real-time MoCap feature extractor for creative applications

As we have seen, there are different libraries to compute features from MoCap data. Also, we have seen that real-time differentiation involves some issues which are particularly relevant in this kind of data. Filters proposed by Skogstad et al. (2012) offer an optimal solution both in terms of noise reduction and introduced delay. In this sense, we consider it is desirable for our work to have a software that implements these filters and allows an easy integration for musical applications. Based on these needs, we implemented MoDe, an open-source C++ library for real-time MoCap feature extraction.

There are other reasons that motivated the implementation of this library, which became the requirements for its design:

- Easy integration with creative toolkits. The library can be compiled as an OpenFrameworks addon. OpenFrameworks is a cross-compatible, open source C++ creative toolkit widely used by the creative community. Applications developed throughout this thesis are in fact built with this toolkit.
- Compatibility with any MoCap device. The library Application Programming Interface (API) allows to provide positional data using standard C++ containers, so it does not require to use any specific device or library.
- Easy handling of temporal events. The library allows to *subscribe* to events detected in the descriptors. For example: it is possible to get notifications when a given descriptor reaches a local maxima or changes its sign. This can be useful for triggering actions based on such events.

The library architecture and its API are explained with more detail in Annex C.

2.2 Motion-Sound Mapping

In Chapter 1 we already introduced the concept of *mapping* referring to how actions performed by the user and control parameters are connected in a DMI. In motion-controlled DMIs, such as those of interest in this thesis, we speak of *motion-sound mapping* (Schacher and Stoecklin, 2011; Jensenius, 2012; Françoise, 2015). In this section, we review different strategies and approaches for motion-sound mapping.

2.2.1 Explicit mapping and event triggering

Hunt and Kirk (2000) introduced a terminology to classify mapping approaches into different categories. They distinguish between *explicit* mapping, for strategies where movement parameters are directly *wired* to sound control parameters; and *implicit* mapping, for strategies that use some model to encode more complex behaviors at the interface between motion and sound parameters.

For explicit mappings, they identify different approaches depending on the way in which input control dimensions are connected to output control parameters. The simplest case is *one-to-one* mapping, where one input control dimension is wired to one output control parameter. The problem of this kind of setting is that, usually, it is desirable to control more dimensions than those provided by the input device. In this sense, different authors such as Rovan et al. (1997) and Wanderley et al. (1998) point out that this kind of mapping makes more sense from the performer perspective when the controlled parameter is associated to some perceptual quality. For example, we could have a DMI whose synthesis engine allows to control the relative amplitude of different harmonics in the sound. While accurately controlling the amplitude of each harmonic would require as many input dimensions as harmonics, a single dimension at the input could be connected to different combinations of these amplitudes, which in effect would be controlling the timbre or brightness of the sound.

Alternative options to one-to-one mapping are *one-to-many* and *many-to-one* mappings. Hunt and Kirk (2000) illustrate both ideas with the example of a violin. In this instrument, the bow controls a variety of sonic parameters, such as the volume, timbre or articulation. In this sense, it is an example of one-to-many mapping. The control of volume in the violin is an example of many-to-one mapping; it is not influenced by a single control, but by a "combination of inputs such as the bow speed, bow pressure, choice of string and even finger position". Rovan et al. (1997) refer to one-to-many and many-to-one mappings as *divergent* and *convergent* mappings, respectively. Following the same idea, *many-to-many* mappings involve associations between multiple input dimensions and control parameters.

An important aspect in the interaction with DMIs is the control of temporal events. Accordingly, it is common for the input device to allow triggering events, and in most applications it is desirable to do it with high temporal precision. Systems based on the conductor metaphor are an example where this is particularly relevant. As we see in section 2.5, where we make a comprehensive review of systems based on this metaphor, there are different strategies for this. These can be as simple (and effective) as detecting changes in hand movement direction, or may involve more elaborate strategies such as the detection of specific gestures.

2.2.2 Mapping gestures: from recognition to temporal and dynamical modeling

With the term *gesture* we refer to movements that extend for a certain time (have a beginning and an end) and express a meaning[7]. Gesture recognition and characterization have been extensively used for interaction with DMIs.

As "movements with a meaning", a common paradigm for musical interaction consists on triggering musical events when a particular gesture is recognized. For example, Sawada and Hashimoto (1997) use features computed from accelerometers data describing the projection of the acceleration in different planes to recognize gestures such as swings and rotations in different directions, which are associated to actions such as starting and stopping the performance or adjusting the volume. Bevilacqua et al. (2002) used dimensionality reduction based on PCA to detect dancing gestures such as jumps in different directions, triggering different sample sounds. There are also examples that we explore later in Section 2.5.2 for the specific case of systems based on the conductor metaphor. For example, Kolesnik (2004) uses gesture recognition based on Hidden Markov Models (HMM) to detect expressive conducting gestures, which are associated with specific indications for the virtual orchestra (*crescendo*, *diminuendo*, *fermata*...).

Artificial Neural Networks (ANN) have also been used to classify input gestures, mapping this classification for sound control. ANN connect input and output variables through hidden layers. In the case of motion-sound mapping, input variables are different gestures captured by the input device, and output variables are the classification. ANN are powerful to learn non-linear relationships between both. For example, Modler (2000) and Mitchell and Heap (2011) used ANN to classify hand postures performed with sensor gloves, mapping the classification to control parameters.

Gestures however are not always performed in the same way. The same gesture can be executed, for instance, with different speeds or sizes. Incorporating this information about not only *what* gesture is executed but *how* it is executed opens new possibilities for interaction. The *Gesture Follower* by Bevilacqua et al. (2010) allows precisely to perform continuous gesture recognition and following. It is built upon a template-based implementation of HMM and it can learn a gesture from a single example. What makes the system interesting in regards to interaction is that it is able to classify the gesture while being executed and estimate its time progression with respect to the corresponding template.

[7] Readers interested in the use of the term *gesture* in NIME might refer to the analysis that Jensenius (2014) performed on such use through the NIME conference proceedings.

Caramiaux et al. (2014b) use particle filtering to extend this approach with an adaptive system. The *Gesture Variation Follower* (GVF) is a template-based method that allows to track several movement features from a gesture in real time, during execution. These features include the time progression (as the Gesture Follower) but also other variations such as the offset position, the gesture size, and the orientation. Caramiaux et al. present an example application where the gesture recognition is used to select different sounds and the estimated variations are used for sound "manipulation": the gesture speed execution controlling the playback speed, and the gesture size modifying the volume.

2.2.3 Mapping through examples

Machine Learning (ML) techniques allow to analyze motion-sound mapping or to explicitly design mappings through examples. Different algorithms have specific advantages and disadvantages, so they are generally selected depending on the particular application.

Canonical Correlation Analysis (CCA) has for example been used by Caramiaux et al. (2010) and Nymoen et al. (2011) to deduce intrinsic motion-sound mappings from gestures performed while listening to some stimuli. CCA allows to extract features from each modality that are the most correlated over time. As Caramiaux and Tanaka (2013) point out, the main drawback of CCA is that it assumes that the temporal evolution of movement and sound are synchronous and that the relationship between gestural and sonic features remains linear over time.

Bevilacqua et al. (2011) introduce the *temporal mapping* paradigm taking advantage of the capabilities of the *Gesture Follower* (Bevilacqua et al., 2010). Temporal mapping focuses on the temporal dimension of the sound. As explained by the authors, temporal mapping can be understood as a synchronization procedure between "input gesture parameters and sound process parameters", for which the progression index provided by the *Gesture Follower* is used. The system can be trained performing a gesture while listening to an audio file, setting a direct correspondence between the gesture and audio time progressions. In the cited paper, the authors already identify the potential of temporal mapping for a "conducting scenario", with a conducting gesture (or, in fact, any gesture) controlling the playback speed of a recording.

Beyond classification and temporal modeling of gestures, ML allows modeling of cross-modal relationships between movements and sound parameters through regression. The

Wekinator[8], by Fiebrink (2011), is a toolkit based on *Weka*[9] that implements a variety of supervised learning methods including ANN, K-nearest neighbors, decision trees, Adaboost and Support Vector Machines (SVM). The *Wekinator* allows to explore different regression analysis methods to design mappings between performer's movements and sound synthesis parameters (or, in general, control parameters for other domains). The user can choose an algorithm and a different combinations of control parameters, for which she can provide examples of the gestures she wants to associate to those parameters.

Caramiaux et al. (2014a) introduce a design principle they call *Mapping through Listening*, which considers embodied associations between gestures and sounds as the essential component of mapping design. They enumerate three categories of mapping strategies: *instantaneous*, *temporal*, and *metaphoric*. *Instantaneous* refers to the translation of magnitudes between instantaneous gesture and sound features or control parameters. *Temporal* refers to the same idea introduced by Bevilacqua et al. (2011), translating and adapting temporal structures between gesture and sound data streams. *Metaphorical* refers to relationships determined by metaphors or semantic aspects. In all cases, the interaction of the user with the system starts in its training stage, in its mapping design.

Similar to this idea is *play-along mapping* as introduced by Fiebrink and Cook (2009). In this paradigm, the user pretends to play along with a musical score in real-time. As the user "performs", the underlying machine learning system builds a training dataset looking at the user's gestures and the concurrent audio synthesis parameters. From this training data, the algorithm learns a mapping from user inputs to synthesis parameters. For this, Fiebrink and Cook implemented a specific functionality in the *Wekinator* to allow the user to perform this "play-along" training, controlling the parameters and behavior of the play-along mapping process.

This idea of integrating the user into the mapping design is in line with the HCI subdomain of Interactive Machine Learning (IML). IML, as introduced by Fails and Olsen (2003), investigates ways to make ML more usable by end users by putting them in the training loop, providing training data for training and iteratively evaluating ML models. This human-centered approach to ML has also been explored in DMI design. Fiebrink and Caramiaux (2016) report how learning algorithms can indeed be used in a creative way by allowing users to convey concepts and intentions to the machine through examples, the approach being of particular interest in DMI building.

Françoise (2015) introduces a framework that he calls *Mapping by Demonstration*. Fol-

[8] http://www.wekinator.org/

[9] Weka is a widely used suit of ML algorithms for data mining. http://www.cs.waikato.ac.nz/ml/weka/

lowing the principles of *Mapping through Listening*, this framework considers listening as the starting point for the design of the mapping. The mapping is learnt from a set of *demonstrations* that explicitly show the relationship between motion and sound as an acted interaction. Françoise uses joint recordings of gestures and sounds to learn a mapping model using statistical modeling. In this sense, following the principles of IML, the user is involved in the design loop, creating training examples and (iteratively) evaluating the designed motion-sound relationship. As a result of this research, Françoise et al. (2014) released XMM[10], a cross-platform library that implements Gaussian Mixture Models and Hidden Markov Models for recognition and regression. The library is specifically targeted at movement interaction in creative applications and implements an IML workflow with fast training and continuous, real-time inference.

2.2.4 Conclusion

As we have seen, there are multiple motion-sound mapping strategies, ranging from the simplest connections between input and output dimensions to frameworks and paradigms that involve the end user in the design of the mapping itself. We believe that these later trends that place the user at the center of the mapping design are particularly relevant to our case, where we want to explicitly exploit the possible different expectations that users may have when using a DMI based on the conductor metaphor.

2.3 The Kinect devices

2.3.1 Kinect V1

The hardware of the Kinect V1 was licensed by Israeli company PrimeSense. It is depicted in the left-hand side of Figure 2.1, and consists on a color camera, an infrared (IR) projector, an IR camera, and a four-microphone array. The color camera delivers video at 30 fps and 640×480 pixels. The IR projector and camera together form the depth sensor, which works with structured light technology. The IR projector casts an IR speckle dot pattern, invisible to the eye and the color camera, that is captured by the IR camera. The dot pattern itself is comprised of local unique patterns which are perceived differently depending on the depth (distance to the plane of the camera lens) of the surface where they are projected. This allows to infer this depth, via software, from the IR image. The IR camera delivers video also at 30 fps at 320×240 pixels.

[10] https://github.com/Ircam-RnD/xmm

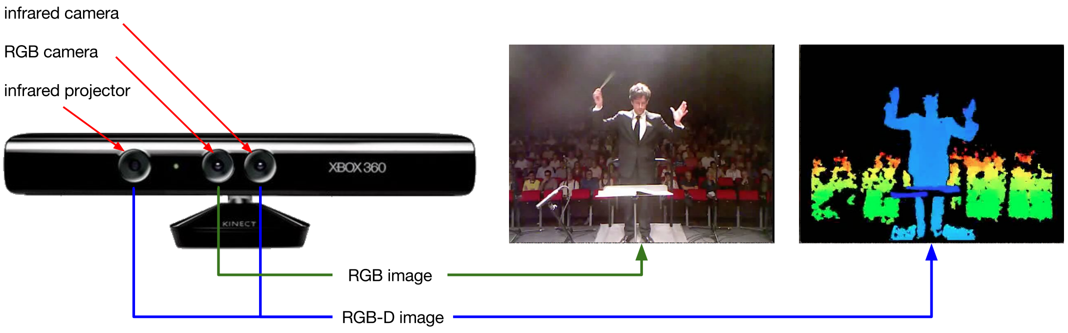


Figure 2.1: Kinect V1 sensor hardware and sample RGB and RGB-D images.

There are different options for software that allow to get the depth image (usually called RGB-D) from the IR video, and that also provide skeletal tracking from it. The most commonly used ones have been Microsoft's official Kinect Software Development Kit (SDK)[11] and Primesense's OpenNI (NI standing for Natural Interaction). OpenNI must be used in compliance with a middleware called NITE in order to perform skeletal tracking (while OpenNI is open source, NITE is not). Both options are comparable in performance (Jungong Han et al., 2013), although the Microsoft SDK is only available for Windows operating system and the OpenNI+NITE option is available in other platforms and works with other devices similar to the Kinect. Examples of these sensors are Asus Xtion[12] and Primesense Carmine[13]. Even though OpenNI ceased to be officially developed when PrimeSense was acquired by Apple in 2014 and the official sites of the library were closed, it is still possible to find the library in sites that maintain it available for the community[14].

A disadvantage that derives from the method that the Kinect V1 uses to perform skeleton tracking is that it introduces a high latency, which is particularly inconvenient in musical applications. Livingston et al. (2012) report minimum latencies of 106 ms that are increased with the number of tracked users.

2.3.2 Kinect V2

The Kinect v2 sensor, officially released as Kinect for Xbox One, offers some improvements over the capabilities of the first version. Details on its capture methods can be

[11] https://www.microsoft.com/en-us/download/details.aspx?id=40278
[12] https://www.asus.com/3D-Sensor/Xtion_PRO/
[13] http://xtionprolive.com/primesense-carmine-1.09
[14] https://structure.io/openni

found in Sell and O'Connor (2014). The depth-sensing mechanism is based on the time-of-flight (ToF) measurement principle. The IR emitter illuminates the scene and the light is reflected by obstacles. The IR camera captures the reflected light and wave modulation and phase detection are used to estimate the distance to obstacles. The Kinect V2 offers higher resolution images than its predecessor. Color image has 1920×1080 pixels, and the depth image has 512×424 with better depth resolution (Gonzalez-Jorge et al., 2015). In any case, the most important improvement of this version with respect to its predecessor for musical applications is its highly reduced latency. The Kinect V2 performs skeleton tracking with a minimum latency of 20 ms (Sell and O'Connor, 2014).

Regarding the software, the only reliable option to use this sensor for skeletal tracking during the realization of this work was the official Microsoft's Kinect for Windows SDK[15]. Open source drivers for the device such as libfreenect2[16] have been developed, but do not provide skeletal tracking.

2.4 Music Conducting

After reviewing MoCap feature extraction techniques and motion-sound mapping strategies, we can move on to focus on the case of the conductor. In this section, we review works that computationally analyze the role of the conductor during performance. Before that, we provide a short historical introduction to the figure of the conductor in Western classical music.

2.4.1 A short historical introduction

Conductors play a special, vital role in orchestral and choral music. During performance, conductors *lead* the ensemble by coordinating all musicians, evaluating the performance and providing additional instructions on how it must be carried out. However, performance is only a very tiny fraction of the work that conductors do nowadays, which also includes analyzing and interpreting the (full) score, communicating this interpretation to the orchestra and practicing the performance. In some orchestras, conductors also carry managing tasks such as defining the repertoire, scheduling rehearsals or even casting new orchestra members.

The purpose of this Section is not to provide a complete historical overview of the figure of the conductor, and we stick to its most immediate predecessors within western classical

[15] https://developer.microsoft.com/en-us/windows/kinect

[16] https://github.com/OpenKinect/libfreenect2

music. Readers interested in a complete overview are referred to Galkin (1988). Shorter, yet very complete, overviews of music conducting can also be found in Platte (2016); Gambetta (2005).

The reason why we say *nowadays* when we talk about these tasks *beyond* performance is because the role of the conductor has greatly evolved during time. The figure of the conductor as we currently understand it was not stablished until the end the 19th century (Galkin, 1988). Up until then, the conducting tasks in instrumental music were usually done by the *Kapellmeister*, who reinforced the harmony and rhythm playing an instrument, usually the cembalo -specially for *basso continuo* in the first half of the 18th century- or the violin, which ended up being more used as it can be played while standing. It was actually a common practice for composers themselves to lead performances of their pieces while playing. For choral music, there was usually a dedicated time beater, who used rolls of paper or wooden sticks to indicate the time visually or audibly (by beating the ground).

However, the higher complexity of music in the romantic period, with increasing orchestra sizes, required new strategies for maintaining precision and synchrony. This is when it became common to see composers (e.g. Mendelsohn, Berlioz, Brahms, Wagner) leading the performances of their pieces without an instrument, using gestures, and when the conducting technique started to become established. Berlioz, in *Le chef d'orchestre: théorie de son art* (1856), gave detailed technical instructions for beating patterns using diagrams. Wagner, in *Über das Dirigieren* (1870), concentrated on more aesthetic aspects about the execution. Here, we confirm the increasing relevance of the conductor as a *performer*: he is not just supposed to lead the performance or to maintain temporal consistency, but also to embody the expressivity of the musical piece and to augment the perception of this expressivity by the audience. Section 2.4.2 discusses how this aspect has been studied in existing literature.

The last step in the evolution of the conductor towards what we know today was forced by the taste of the public, which had an increasing interest on music by deceased composers. The conductor was not necessarily the composer of the piece, but a musician that was able to interpret a score, communicate this interpretation to the orchestra and develop the right technique to make this communication with gestures.

Although the art of conducting allows each conductor to develop his own style and technique, conducting is highly codified and taught in many institutions. Traditional schools of orchestral conducting have developed a well-defined and structured grammar of conducting gestures. There is a vast literature in the field of musical conducting, but probably the most relevant and commonly cited reference is *The Grammar of Conducting* by Rudolf (1980). The book argues that most conducting information can and *should* be communicated with gestures of the right hand with or without a baton. Accordingly, even though in some cases we find that the tasks of both hands are divided (the right hand for time beating, the left hand for expressive communication) (Kolesnik, 2004), it is more common to consider that the same hand can indicate other expressive aspects that are translated into modifications with respect to the dynamics or articulation of the performance. Rudolf describes gestures with illustrations of two-dimensional shapes covering different musical expression as well as time-beating.

In the rest of this dissertation (and as usual in related works) we focus on the most visible and recognizable task of conductors: silently leading performances with gestures. However, it is important to keep in mind that this vision of the conductor as a musician who can just stand in front of an orchestra and start performing is clearly reductionist.

2.4.2 The effect of conducting movements in performance

The way in which the movements of the conductor influence the performance has been studied from two main points of view. One set of works try to establish causal relationships between the movement execution and specific aspects of the resulting musical outcome (mainly timing), either by recording and analyzing the performance itself or by studying how performers perceive specific aspects of the conducting movements, which are assumed to affect how they would perform (e.g. perception of beat in the movement). Other works assess, usually through questionnaires, whether the conductor's movements affect the perception of subjective performance qualities, such as its interest or expressivity.

Objective assessment of the effect of conducting movements

Existing literature dealing with the objective analysis of conducting movements effects has focused on timing, usually trying to establish the concrete moment of the (hand) gesture in which musicians perceive the ictus or beat. Clayton (1986) examined the contribution of different factors in string instrument players timing during performance of orchestral music: the rest of the ensemble players, the conductor, the score, and the player's own sense of rhythm. He found that the most important source of timing information was the sound of the rest of the ensemble players, followed by the conductor. From the results, he suggested that the role of the conductor was to provide general timing information to the ensemble by, for example, setting the initial tempo or monitoring

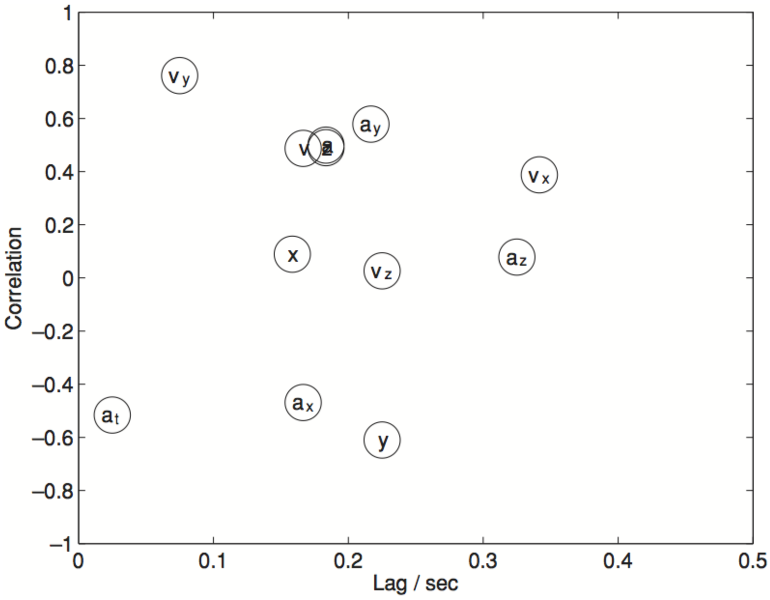


Figure 2.2: Correlation vs lag plot for one of the “high-clarity” excerpts studied by Luck and Toiviainen (2006). The position of each variable corresponds to the lag at which its correlation with the pulse was maximum and the value of this correlation. Image from Luck and Toiviainen (2006), p. 194.

the tempo during the performance and keeping the whole ensemble together. He also investigated which part of the conductor’s gesture communicated the ictus by taking recordings of string instrument players playing and tapping their foot in synchrony with two-beat conducting gestures. He found that the lowest part of the trajectory in the vertical axis corresponded to the perceived beat. However, Luck (2000) questioned the accuracy of this definition of the beat after asking participants with different musical expertise to tap the beat in synchrony with recordings of a conductor performing different gestures. He found that musical expertise was negatively correlated with synchronization accuracy following that definition and suggested that the beat in conducting gestures may need to be defined in more complex terms.

In order to explore the phenomenon of conductor-ensemble synchronization in a more ecologically-valid way, Luck and Toiviainen (2006) recorded the movements of an expert conductor with a high-precision MoCap optical system while directing an ensemble of expert musicians, together with the performance audio. They examined the features of the conductors’ gestures with which the ensemble performance was synchronized. More concretely, they extracted twelve MoCap features from the position of a marker attached to the tip of the baton and cross-correlated them with the pulse of the performance, for which they used the spectral flux computed from audio of the orchesta. They manually annotated short (close to 10 seconds) excerpts where the conductor communicated the beat with high and low clarity. For each of the descriptors, they found the lag that produced the highest correlation with spectral flux. This way they could establish, for each descriptor, the delay between the feature occurring and the ensemble playing. For the high clarity excerpts, they found that the ensemble performance tended to be highly synchronized with periods of maximal deceleration along the trajectory. This is suggested by the descriptor a_t (acceleration along the trajectory, computed by projecting the acceleration in the direction of the velocity vector) having a high negative correlation at a short lag. Periods of high vertical velocity (v_y) showed higher correlation with a longer delay. Figure 2.2 shows, for one of the "high-clarity" excerpts considered in this study, the correlation and lag values for each of the twelve descriptors under consideration[17].

More recently, Platte (2016) proved a direct correlation between the movement qualities of conductor's gestures and the reactions of musicians to those with respect to sound qualities. For one of the experiments she performed, a professional conductor was recorded performing the three different conducting patterns shown in Figure 2.3 (concave, convex and mixed, a combination of both) which were expected to produce different reactions. The patterns were recorded in different sizes and tempi. Then, she asked participants in the study to perform a set of beat-tapping tasks by pressing a touch sensor while watching the recordings. The output of the touch sensor was associated, during the analysis, to different musical qualities. The applied pressure was associated to volume, while the length of the touch was associated to articulation. In addition to the output of the touch sensor, she also recorded physiological data to measure stress levels. The requested tasks involved *playing* the touch sensor following the presented video: first, without specific instructions, in order to observe differences between intuitive reactions to different patterns and sizes; then, by giving coherent and incoherent tasks to the participants (e.g. asking to play loud with a big and a small gesture, respectively). The experiment showed, for example, that loudness was mainly influenced by the size of the gesture, while articulation was influenced by different shapes, being the convex gesture the one evoking significantly longer pressure duration over the concave one. It was also shown that convex gestures showed the highest predictability in terms of beat communication.

Karipidou (2015) performed MoCap recordings of a conductor directing a piece for string quartet with different expressive intentions. From these recordings, she explored dimensionality reduction techniques based on Gaussian Process Latent Variable Model

[17] We use the same naming convention for these descriptors in Subsection 3.2.3.

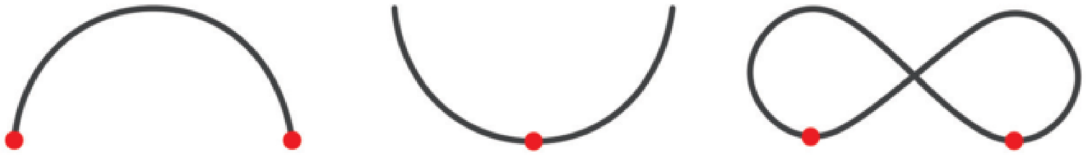


Figure 2.3: Concave (left), convex (middle) and mixed (right) conducting patterns used by Platte in her experiment. Red circles represent the beat points. Image from Platte (2016), p. 36.

(GP-LVM) to represent these expressive variations, being able to classify conducting movements based on the expressive intent. Following this work, Ahnlund (2016) looked for correlations between these representations of conducting expressivity and expressive features computed from audio. While some correlations are found, the conclusion of her work is that the utilized representations of conducting gesture do not capture musical expression to a sufficient extent.

Influence of conducting gestures in performance perception

During performance, conductors embody expressive elements of the music being played. Even if this can be considered as something they need to do in order to correctly communicate with the orchestra and achieve the best sonic result, their movements also influence the overall perception of the performance by the audience.

Previous research has widely shown that there is a link between visual and auditory perception of music performance. Davidson (1993) showed that visual information plays an important role in conveying expressivity. She presented observers with excerpts of recordings of deadpan, projected and exaggerated performances in sound only, sound and vision, and vision only modes. She found coherence between performer intention and audience perception in all conditions, with the vision-only recording providing more information about expressive intention. Thompson et al. (2005) asked listeners to rate the perceived dissonance and emotional valence of two recordings by Judy Garland and B. B. King (known for being particularly expressive in their facial gestures) and found significant differences when those recordings were presented with and without the video. Dahl and Friberg (2004) investigated wether emotional intentions could be conveyed through musicians' movement. They recorded a marimba player playing the same piece with the intentions Happy, Sad, Angry and Fearful and found that the first three were correctly identified by observers who watched the recordings without sound. Platz and Kopiez (2012) performed a meta-analysis of fifteen studies on audiovisual music perception, concluding that the visual component is an important factor in the communication of meaning.

A number of works have studied the perception of conducting gestures and their influence in the overall perception of the performance. Bender and Hancock (2010) performed a study to analyze the influence of the conductor's intensity and ensemble performance in the rating of the conductor, for which they combined high and low intensity gestural conducting with high and low performance qualities. As expected, the high intensity conductor was rated higher, but the quality of the performance also influenced the conductor assessment. Price and Mann (2011) found out that the conductor also had an effect on the rating of the performance. They asked participants (undergraduate music students and music education majors) to rate the recordings of seven performances with different conductors for which the audio was actually the same. Not only conductors were rated differently, but performances as well. More recent studies started to address this effect with more detail. For instance, Morrison et al. (2014) found that the perception of articulation and dynamics in gesture was strongly correlated with evaluations of overall ensemble expressivity.

Kumar and Morrison (2016) recently analyzed whether the effect of the conductor in the performance perception could be due to the gesture as they delineate and amplify some specific aspects in the music. They recorded two conductors focusing on different musical elements for two musical excerpts, one consisting on an *ostinato* paired with a lyric melody, and another one with long chord tones paired with rhythmic interjections. They asked participants to rate the performances according to a number of musical elements (articulation, rhythm, style and phrasing). For the first excerpt, they found out that listeners were sensitive to how the conductors delineated musical lines as an indication of overall articulation and style. According to the authors, the same effect was not observed in the second excerpt probably due to listeners' preconceptions on the importance of melody over rhythm or some instruments versus others.

Following a different strategy, Luck et al. (2010) investigated the relationships of conductors' gestures kinematics and ratings of perceived expression. They recorded two professional conductors with a MoCap system and presented participants with point-light representations of those recordings, asking them to provide continuous ratings of perceived valence, activity, power and overall expression. They performed regression from movement features, extracted from the MoCap data, to the provided ratings. They found out that higher levels of expressivity seemed to be conveyed by gestures with high amplitude, variance and speed of movement.

2.4.3 Conclusion

In this section we have seen several works that analyze conductor-orchestra interaction from a computational point of view. In most cases, these studies are performed in controlled settings, which is necessary when trying to observe and analyze a particular phenomenon. There is however a lack of studies dealing with actual performances and, in our work, we want to approach this analysis trying to see whether it can inform the design of DMIs based on the conductor metaphor.

We have also seen studies that analyze how the conductor movements affect the audience's perception of a performance. We have included these works since, as we have been saying, the purpose of an interface metaphor is allowing the user to transfer the knowledge she has from the activity replicated by the metaphor to the interaction with the system. In this sense, it is relevant to know how the conducting activity is perceived by those who see it. Something we do not find in the literature, however, is studies that investigate how people understand the role of the conductor by turning this approach. That is, instead of analyzing how conducting movements are perceived, asking the person to perform these conducting movements. We believe that, with methods similar to those used to analyze the movement by real conductors, there is an interesting unexplored opportunity to analyze how people with different musical backgrounds perform conducting movements.

2.5 The Conductor Metaphor in Digital Musical Instruments

The conductor metaphor has been widely used in DMIs. For this review, we consider that a DMI belongs to this category when:

- The control consists on the **modification of an existing musical piece** (which can be stored either as a score to be synthesized or as audio to be modified). This modification can consist on controlling the performance tempo, the overall volume or balance between different sections, the articulation, etc.
- The **interaction** occurs **in real-time**, with the user being able to hear the effects of her gestures on the outcome.

As we will see immediately below, this type of interaction motivated the creation of control devices that allowed to use **movements that resemble those of a real conductor** during performance. For this review, we focus on works that also include this characteristic.

2.5.1 Immediate predecessors and first works

Even though we could trace the predecessors of conductor-based musical interfaces to many works in the Computer Music field, here we just mention the most immediate ones.

The first of them is GROOVE, a system developed by Mathews and Moore (1970) which allowed to compose, store and edit functions of time interactively. They defined a file system to store functions of time generated by human actions and, very importantly, they included feedback on the performed actions during the interaction as a key element of their system, as depicted in Figure 2.4. In fact, the authors already talk about the conductor metaphor when they discuss the first application for which they had used the system (which was designed to be used for a wide range of applications), an electronic music synthesizer:

> *The desired relation between the performer and the computer is not that between the player and his instrument, but rather that between the conductor and the orchestra. The conductor does not personally play every note in the score; instead he influences (hopefully controls) the way in which the instrumentalists play the notes. The computer performer should not attempt to define the entire sound in real-time. Instead, the computer should have a score and the performer should influence the way in which the score is played.* (Mathews and Moore, 1970, p. 716)

Mathews (1976) used GROOVE to develop the *Conductor Program*. The *Conductor Program* was motivated by Mathews' conversations with composer and conductor Pierre Boulez. As Mathews said in an interview with Richard Boulanger:

> *The Conductor concept was actually the result of a request from Pierre Boulez. In 1975 he asked me for a flexible way of playing back a tape so that the tempo of the music would be controlled. [...] Boulez wanted to be able to change the timing of the playback without changing the pitches or timbres. [...] I began asking myself, "What do conductors do during a performance?" Clearly, they control the tempo.* (Boulanger et al., 1990), p. 34.

The *Conductor Program* allows to perform music with a synthesizer and a computer that has a score stored in a file. In its first version, the *Conductor Program* allowed to control the performance in real time using the same input devices that GROOVE used to record gestures (i.e. typewriter, switches and knobs).

Inspired by the *Conductor Program,* Buxton et al. (1980) developed the *conduct* system in the late '70s. It consisted on a digital synthesizer controlled with a graphics tablet,

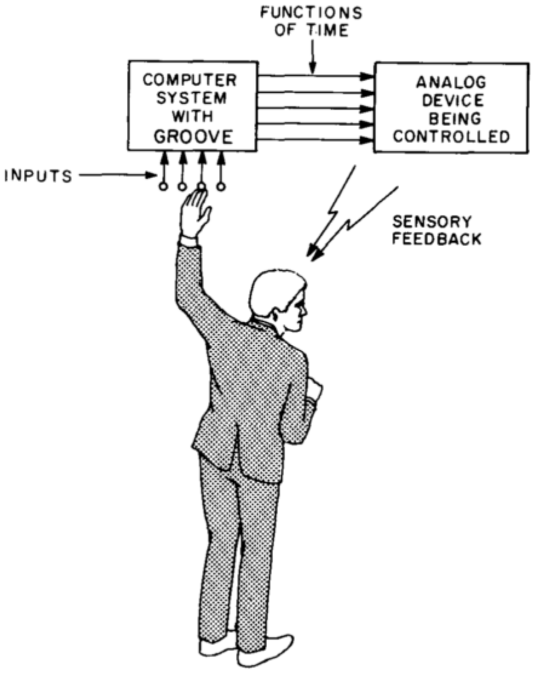


Figure 2.4: Feedback loop for composing functions of time in GROOVE. Image from Mathews and Moore (1970), p. 715.

switches and sliders. It had prerecorded scores for which pitch, tempo, articulation, amplitude and *richness* (timbre) could be controlled in real time. This was done using the graphics tablet to select the desired parameter to change on the screen and changing its value directly, either by inputting it through the keyboard or moving it up and down using sliders.

These works, as we see, are explicitly based on the conductor metaphor. However, the interaction was still done with input devices that were not meant to be used with gestures that resembled those of a real conductor.

Mathews kept developing the *Conductor Program* looking for more natural ways to interact with it. As a result, he developed a mechanical baton called *Daton* (Mathews and Barr, 1988), which consisted on a plate mounted on gauges that generate electrical pulses when struck with a drumstick. The *Daton* was used to control the performance of the stored score by triggering beats and controlling other qualities such as loudness and balance of different voices (in these cases, in a less natural way, by varying the position of the hit). It was also possible to control these other qualities using a joystick and knobs, keeping the *Daton* for tempo control. Figure 2.5 shows the hardware configuration of the *Conductor Program* when the *Daton* was included.

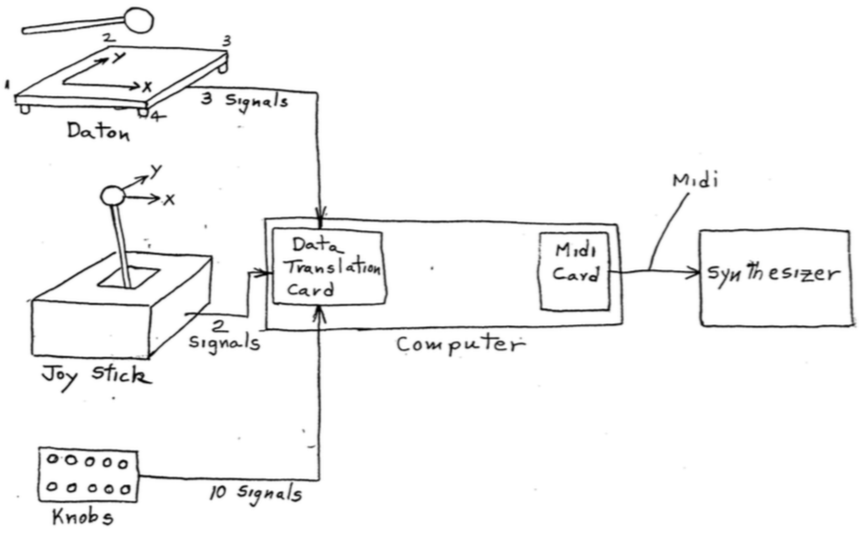


Figure 2.5: Diagram of the *Conductor Program* hardware in the version using the *Daton*. Image from Mathews and Barr (1988), p. 12.

Mathews, together with Bob Boie, later developed the famous *Radio Baton* (Boie et al., 1989; Mathews, 1991) to allow a more natural control of the *Conductor Program*. The *Radio Baton* consists in a low-frequency radio transmitter in the end of a baton, whose 3D position over a plate is measured by an array of receiving antennas. Using two *Radio Batons*, the beat is triggered when the distance of one of them (usually the one held with the right hand) to the plate is lower than a certain threshold. The position of the other baton is used to control dynamics, as it was done with the joystick in the previous version. This makes the interaction with the *Conductor Program* more natural and similar to real conducting. Figure 2.6 shows Mathews using the *Radio Baton*.

These works can be (and very commonly are) considered the first conductor-based musical interfaces that fulfill all the aforementioned criteria. They have a stored score whose performance can be controlled in real-time using conductor-like movements as, in this case, beating with a baton.

2.5.2 Relevant DMIs using the Conductor Metaphor

The works included in this subsection fulfill the criteria stablished at the beginning of this section. They are considered relevant for their novel contributions with respect to different aspects such as the motion capture procedure (the first works usually make contributions in novel input devices) or the motion-sound mapping strategy. However,

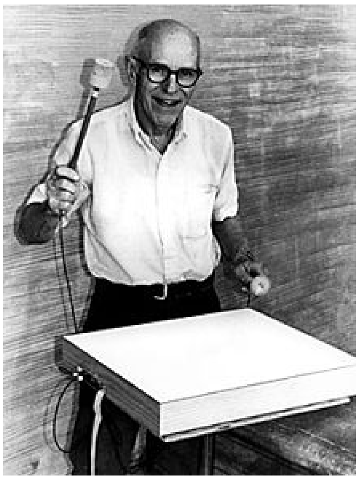


Figure 2.6: Max Mathews using the *Radio Baton* to control the *Conductor Program.* Image from Marrin (2000), p. 44.

the contributions of many works are multiple, which makes difficult to establish a classification based on the kind of contribution. For this reason, we review these works in a chronological order, grouping those that were created by the same person or group.

Haflich and Burnds (1983) developed the *Conductor Follower* with a particular focus on the input device. They used ultrasonic rangefinders developed by Polaroid for their automatic cameras. Two sonar devices on the floor with their beams positioned upward toward the conductor's arm tracked its position in two dimensions to an accuracy of about one inch. It was the first system that tracked the arm position in a completely unobtrusive way. Following Mathews' advice (according to Marrin (2000)), they built a baton consisting on a passive device with a reflector attached to its tip that was better traced by the rangefinders.

Keane and Gross (1989) designed the *MIDI Baton* to enable a conductor to coordinate, in real time, live performers and a computer or sequencer. In this sense, it is only meant to control timing. It consists on a metal tube with a metal ball inside, separated by a spring. When accelerated with sufficient strength, they make contact and generate an electrical signal. Some post-processing is done to discard false positives and detected beats are sent to a sequencer. Refinements to the system were done in the following years, with the basic concept for interaction and the device being the same (Keane and Wood, 1990, 1991).

The first computer vision based interface using the conductor metaphor was the *Computer Music System that Follows a Human Conductor* by Morita et al. (1989). It used a

CCD camera and specific hardware to track the 2D position of the baton or a white glove on the conductor's hand. Tempo and volume were computed from the turning points of the tracked trajectory and used to control a MIDI sequencer. In later versions (Morita et al., 1991), the baton incorporated an infrared light on its tip and a VPL Research DataGlove (Zimmerman et al., 1987) for the left hand, which allowed to include new actions such as instrument selection. In addition, the system included a database mapping gestures to musical expression information which could be altered by the conductor by telling the system how good the gestures had been interpreted. In this sense, this system is also the first to incorporate some sort of supervised learning stage to gain information from how users interacted with it.

The *Conductor Follower*[18] by Brecht and Garnett (1995), which builds on the *Adaptive Conductor Follower* by Lee et al. (1992), used a Mattel Power Glove and a Buchla Lightning baton as input devices. The 2D position of the baton was processed in the Max/MSP environment with different methods for the beat analysis. The first method used the time between the last two detected beats to predict the time of the following one. The second looked for six characteristic points in a beat curve, using zero crossings in velocity and acceleration of the baton. The third method used an ANN trained from the relationship between these six characteristic points and time, estimating the probability of the next beat happening. This ANN was previously trained by a conductor conducting along with a metronome at different tempo. The synthesis was done by controlling MIDI output.

Guy Garnett (coauthor in the aforementioned works) also coauthored the *Virtual Conducting Practice Environment* (Garnett et al., 1999), which in this case was not meant for performance but to be used by student conductors to get proper audiovisual feedback on the goodness of their gestures. Similarly to the previous systems, it used a Buchla Lighting baton. It was able to provide graphic representations of specific aspects like the position of recognized beats or the articulation (from *legato* to *staccato*). However, the authors stated that it could not be used as a successful substitute of a real teacher or practice with musicians.

The *Interactive Virtual Ensemble* (Garnett et al., 2001) is an evolution of the previous *Adaptive Conductor Follower* and *Conductor Follower* systems that replaces the Buchla Lightning baton by a MotionStar sensor that is able to recognize not only the position of the baton, but also the orientation of the hand holding it. The system used one computer to process sensor input and another one for synthesis getting the required output from

[18] Not to be confused with the system with the same name by Haflich and Burns (1983) explained previously in this Subsection.

the first computer. Also, the synthesis used an analysis/resynthesis paradigm instead of MIDI, with a dedicated algorithm to deal with the artifacts caused by transients when slowing down the resynthesized music.

Teresa Marrin's contributions in the creation of musical interfaces is highly influenced by her experience as a conductor. The *Digital Baton* (Marrin and Paradiso, 1997) included three acceleration sensors to sense the movement, five pressure sensors to sense the pressure of each finger of the hand, and a LED at its tip to track its position with a camera. It could track conducting gestures (Marrin, 1996) but it also ended up being used as an input device for other sort of interactions such as triggering samples and placing them in the stereo panorama.

A very different an innovative approach is the *Conductor's Jacket* (Marrin and Picard, 1998) which, for the first time, uses physiological sensing to track conducting gestures, as opposed to just tracking the position of the hands or the baton. The jacket incorporated four electromyogram (EMG) sensors for muscles in the arms, a respiration monitor, a heart rate monitor, a temperature sensor and a Galvanic skin response sensor. In its second version (Marrin, 2000), it only included the EMG and respiration sensors. The processing software detected beats based on maxima in muscle tension from the biceps, as well as cutoffs or pauses, allowing the conductor to control tempo. The system also allowed to perform other actions on the synthesis beyond those commonly associated to conductors, such as changing the pitch or panning of specific voices.

The first system that used Hidden Markov Models (HMM) to follow right hand conducting patterns was the *Multi-Modal Conducting Simulator* by Usa and Mochida (1998b). It used acceleration sensors to track the movement of the baton in two dimensions. HMMs in their case were used in combination with fuzzy logic based on the current tempo to estimate the probabilities of a detected beat to correspond to the different beats within a measure. In addition, the system also allowed to control dynamics and articulation based on the movement trace between beats (e.g. mapping the movement size to the loudness and its smoothness to the articulation). The gaze and breath of the user was also detected with an eye camera and a breath sensor, which allowed to couple previously marked parts in the score with the breathing of the user.

The *Virtual Orchestra* by the Digital Interactive Virtual Acoustics (DIVA) consisted on 3D models of musicians in a band whose movements were rendered from the notes in a MIDI score. The system included a conductor following system by Ilmonen and Takala (1999) to control the interpretation of these virtual musicians using MotionStar sensors as input device, mounted on both hands and the conductor's neck (as a reference). The sensors provided three-dimensional positions of the hands relative to the neck. These positions were inputted to ANNs to predict beats similarly to Brecht and Garnett (1995).

The *Personal Orchestra* series is a set of works by the RWTH Aachen University's Media Computing Group (Borchers et al., 2002; Lee et al., 2004, 2006; Borchers et al., 2006). They made a number of installations which are based on providing control over ad-hoc recordings of orchestras.

The *Virtual Conductor* (Borchers et al., 2002) is the main attraction of the House of Music in Vienna. It consists on an audiovisual recording of the Vienna Philharmonic, with manual annotations of the beat positions. An infrared baton is used as the input device. Its position is tracked in 2D by a receiver under the screen that the user faces, as shown in Figure 2.7. The downward-turning points are detected as beats and used to control the tempo of the playback, applying low-pass filtering to avoid sudden unnatural tempo changes. As opposed to the case of the *Conductor Program*, this system does not provide control over a synthesizer, but over a recording (which, in a sense, is closer to Boulez's original idea about controlling the tape). This requires to perform time-stretching on the recordings, in a way that changes the playback speed without affecting the pitch or timbre of the original sound. At the time of development of the first version, state-of-the-art algorithms did not allow to do this in an efficient way, so it was solved by cross-fading between tracks at different speeds and changing the playback speed of the video. The installation was upgraded in 2009 with new recordings, an electronic music stand, high-quality time stretching and some improvements in the robustness of the interaction (Borchers et al., 2006).

A time-stretching system based on the phase-locked phase vocoder was first used in a version for children called *You're the Conductor* (Lee et al., 2004), which was installed in the Children's Museum of Boston. Regarding the interaction, the mapping was made intentionally simple for children, with the speed of the movement controlling the tempo (avoiding the necessity to follow beats) and the size controlling the dynamics.

Maestro! (Lee et al., 2005) and *iSymphony* (Lee et al., 2006), installed at the Betty Brinn Children's Museum in Milwaukee, incorporated a gesture analysis framework (*conga*, Conducting Gesture Analysis framework (Grull, 2005)) that allows the system to detect if the user is performing one of two possible gestures or none, thus being able to switch its behavior depending on what the user is doing.

The *Conducting Gesture Recognition, Analysis and Performance System* by Kolesnik (2004) used two USB cameras as input devices. A computer processed the input using the EyesWeb software (Camurri et al., 2004) to track the two-dimensional position of the

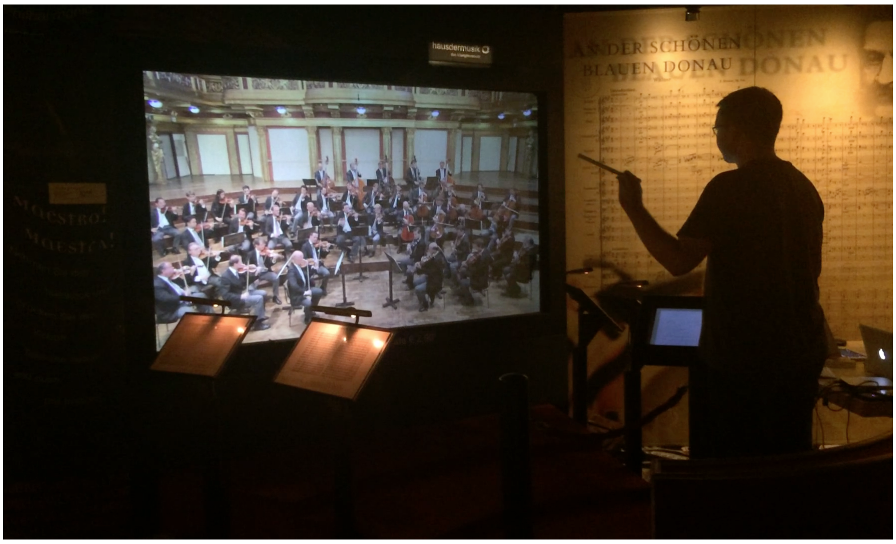


Figure 2.7: *The Virtual Conductor* installation at the House Music in Vienna.

hands. A second computer received this tracked position to compute, using a dedicated set of HMM tools in Max/MSP, control information from both hands (right hand for tempo control, left hand for expressive control). A pair of HMM (one for each camera) was trained for each of the gestures to be recognized by the system. This allowed to recognize the most likely gesture for the input at each moment, controlling the playback of an audio file. Phase vocoder was used to control the tempo, for which the controlled sound needed to have beat annotations.

Another set of works use the KTH[19] rule system for music performance (Friberg et al., 2006). This system models principles of interpretation used by musicians in performance. It takes a score as input and outputs a expressive rendering of it by applying contextual modifications on the low-level performance parameters (tempo, note duration, dynamics...). For example, a rule called *Overall articulation* is used to change the articulation of all notes in the score by changing the ratio between the note duration and the inter-onset-interval. In *Home Conducting*, Friberg (2005) used the KTH rule system as a means to provide indirect control of expressive musical details on the note level. It uses a webcam as input device, but instead of directly extracting the position of the hands, it computes parameters such as the quantity of motion, position of the overall motion, and the size and velocity of horizontal and vertical gestures. These features are mapped to values of the KTH rules to guide the performance. Interestingly, Friberg envisions three

[19] KTH is the acronym for the *Kungliga Tekniska Högskolan* (Royal Institute of Technology) University in Stockholm.

possible levels of interaction depending on the complexity of the mapping. A first level (*listener level*) would allow to map semantic descriptors such as emotions (e.g. *happy* movements result in *happy-sounding* music); a second one (*simple conductor level*) would use kinematic descriptors to directly control the music expression (e.g. a *fast* movement results in *faster* music); and a third one (*advanced conductor level*) would combine the previous ones with explicit control of the beat.

Fabiani (2011) focused on the *listener level* (or *naïve level*, in his publication) in *PerMORFer* and *MoodifierLive*. *PerMORFer* is similar to *Home Conducting*, except in that it is based on the manipulation of audio recordings instead of using MIDI, which requires a complex audio analysis and transformation stage in order to apply note-level modifications to the original audio. *MoodifierLive* is a version using mobile phones as input devices (concretely, their accelerometers) and, again, using MIDI to generate the resulting audio. In both cases, the movements do not need to resemble those of a real conductor. Instead, the input is processed to obtain high level descriptors which are mapped to performance parameters, as illustrated in the block diagram of *PerMORFer* shown in Figure 2.8[20].

In *VirtualPhilharmony,* Baba et al. (2010) deal with the specific problem of previous interfaces not satisfying users with conducting experience. It is, thus, meant for professional conductors. They use three sensors to track conducting movements: a glove with an accelerometer, a Wii Remote held as a baton and a capacitance sensor (a MIDI theremin). It introduces heuristics of conducing an orchestra based on expert knowledge. They make some interesting considerations, such as considering that musicians in the orchestra do not always obey the directions of the conductor and also have their own musical intentions. They introduced the "Concertmaster Function" imitating the role of a concertmaster (usually the first violin), who conveys instructions from the conductor to the orchestra, and communicates the orchestra members' will to the conductor. Basically, the "Concertmaster Function" adapts he behavior of the system in terms of beat prediction to the musical style or the tempo. They made an interview with an expert conductor and got some feedback from a few public demonstrations which seems to indicate that the system feels like having "musical persuasiveness". However, it is difficult to tell wether the heuristics have any particular effect on this, provided that the evaluation is solely based on informal feedback provided by users who used the system with this heuristics activated.

The *Conductor Follower*[21] by Bergen (2012) uses a Kinect V1 as input device. It is

[20] Note that the block diagram for *MoodifierLive* would be the same, excluding the audio analysis part.

[21] Not to be confused with the systems with the same name by Haflich and Burnds (1983) and Brecht

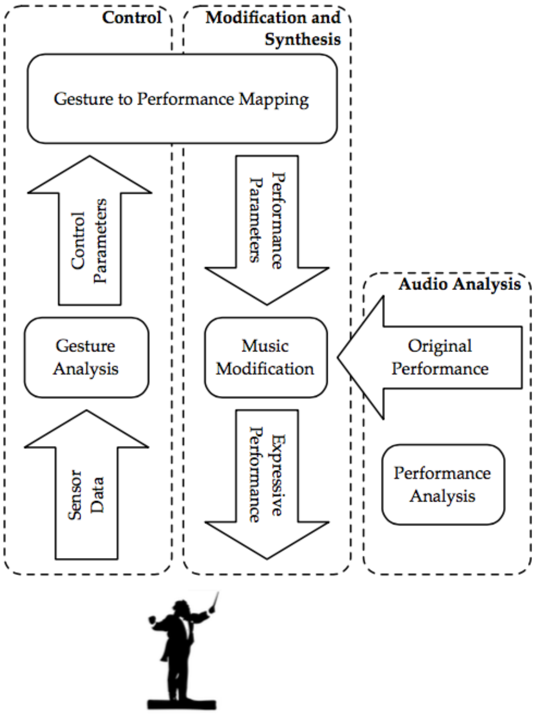


Figure 2.8: Block diagram of *PerMORFer.* Image from Fabiani (2011), p. 11.

developed as a VST plugin that takes a MIDI score as input and it allows to control the beat. The beat detection algorithm uses only the vertical position of the hand and some heuristics to determine whether a change from downward to upward movement is actually a beat. It also includes some heuristics to deal with very sudden changes in tempo or the conducting pattern (e.g. switching from marking all 4 beats in a 4/4 bar to marking just the first and third beats).

The *Interactive Conducting System Using Kinect* by Toh et al. (2013) allows to control tempo, adjust volume and emphasize instrument sections of a virtual orchestra. It uses a Kinect V1 as input sensor, and the interaction is based on a priori knowledge gained from interviews with conductors. To detect beats, the system looks at the trajectory of the right hand movement and finds down-turning points between two consecutive up-turning points. Volume is controlled by raising or lowering the left hand at a certain distance from the rest of the body. Instrument emphasis is achieved by looking at the position of the head and the depth information (distance to the camera) of the trajectory (e.g. if the head moves more than a certain threshold to the left and the conducting trajectory moves towards the camera - towards the orchestra - the instrument section

and Garnett (1995), explained previously in this Subsection.

placed on upper left position is emphasized).

Similarly to the previous example, the *Virtual Ensemble* by Rosa-Pujazon and Barbancho (2013) allows to control the beat and the volume of different instrumental sections using a Kinect V1. In their case, beats are detected from direction changes in horizontal hand movement[22], and the volume can be adjusted with the left hand by first pointing at the instrumental section of interest and then raising or lowering the hand. Visual feedback helps to understand the interaction by, for example, drawing a red arrow on top of the instrumental section being pointed at each time, if any.

With a very different approach, Diakopoulos et al. (2015) used a Kinect V2 to allow a professional conductor to control a prepared grand piano. In this case, the conductor can make “grabbing” gestures with the right hand at different positions in space to activate different modes of control, waving the other hand to modify different parameters. In this case, the gestures controlled the behavior of the actuators in the piano.

Diesbach et al. (2013) propose an original installation using a Kinect V1 where the orchestra to be controlled is actually formed by up to 24 laptops arranged in space similarly to a symphonic orchestra. Each laptop is associated to a different sample, and the performer can just move her hands through the space to make different laptops sound. Holding a hand in one area makes the sound of the corresponding computer increasing in volume. Tsui et al. (2014) followed the same idea using mobile phones, but in this case using Leap Motion[23] as input device.

More recently, Bacot and Féron (2016) report the creative process on the creation of a piece where a conductor used two Kinect sensors to interact with *Gestrument*[24], an application initially conceived for the iPad. Here, two Kinects are used to provide the performer with two “immaterial” touch screens, one for each hand, to control the application.

2.6 Conclusions

In this chapter, we have reviewed works relevant to this thesis in several areas. First, we have seen common techniques for feature extraction from MoCap data. Next, we have reviewed different mapping paradigms that have been used for DMI design. Then, we

[22] According to the authors, the decision of using horizontal movement is based on previous explorative tests asking non-conductors to perform conducting gestures. This is something we have not found elsewhere in the literature, nor something we have observed in our own observation studies of the sort (see Chapter 4).

[23] https://www.leapmotion.com/

[24] http://www.gestrument.com/

have reviewed studies that computationally analyze conducting movements. Finally, we made a comprehensive review of DMIs based on the conductor metaphor.

Having done this review, we can identify some opportunities for research with DMIs based on the conductor metaphor.

Regarding the conclusions from existing studies analyzing conducting movements, we identify two problems with respect to their applicability in the design of DMIs. First, most of these studies are performed under conditions other than those found in a performance. This is desirable in some cases if what is desired is to have a controlled environment where a particular phenomenon is observed. However, when designing a DMI based on the conductor metaphor, we must keep in mind that the idea that the average user has about what a conductor does (and, therefore, what she will tend to imitate) is associated with what she can see in a performance. For this reason, it is desirable to observe conductors in actual performances, although this implies that the observed effects should be treated with greater caution. Second, most of these studies have been performed for areas other than the design of DMIs. We also think it is adequate to perform this analysis with such use case in mind. For this, we must consider what parameters can be controlled in a DMI based on this metaphor and observe the performance looking for causal relations between the movements of the director and these parameters.

Regarding the design of DMIs based on the conductor metaphor, we believe that the progress in motion-sound mapping paradigms offers new opportunities for exploration. We have seen how the vast majority of the revised DMIs offer important improvements in many aspects (more precise control of the tempo, extension of the dimensions under control, etc.). However, in a scenario where users may have different expectations about how to interact with the system or where they may even wish to develop their own style, this aspect has not been explored in depth. Recent motion-sound mapping paradigms encourage to involve the user in the design of the DMI. Following this idea, we believe that the case of the conductor metaphor offers a good use case for investigating the adaptation to user-specific styles or specificities.

Finally, it is difficult to find cases where, even when the instrument has been designed with the goal to attract new audiences to classical music, the extent to which this objective is achieved is discussed. Accordingly, in a context such as that of the PHENICX project in which this thesis is developed, it is important that this is not left aside. For this reason, we believe that it is relevant to design experiences where not only this goal is key but also where, accordingly, its success in this matter is thoroughly evaluated.

Chapter 3

Learning from real performances

Music conducting is a highly complex musical art. It involves many different aspects. Among the many tasks for which conductors are responsible, leading the ensemble using gestures during musical performance is probably the most characteristic and prominent one. In this context, conductors use gestures to establish the tempo, indicate entries to different sections in the orchestra (or voices in a choir) or convey expressive intentions that get reflected in variations in, for example, dynamics and articulation. Although the goal of this work is not to build a realistic model for virtual conducting targeted to professional conductors or students, a better understanding of the causal relationships between conducting gestures execution and the resulting performance informs the design of interactive systems based on the conductor metaphor. In this context, we made an interview with professional conductors and students to identify which specific aspects of a performance could be observed and automatically analyzed. Motivated by the conclusions of this interview, we recorded and analyzed specific parts of a conductor performance focusing on two aspects: the musicians' synchronization with conductor's gestures and the relationship between some body movement descriptors and the overall loudness of the performance. This analysis provides the basis for the subsequent parts of this thesis in terms of the methods to estimate movement-music synchronization and estimate correlations between body movement and loudness.

3.1 Introduction

There are roughly two strategies to follow when designing interactive applications based on the imitation of an already established activity as orchestral music conducting.

The first one is to build a system that replicates as closely as possible the real scenario. This is useful in cases where the application is meant to be used as a teaching or educational tool. In the case of conducting, we could think of applications for different contexts. One could replicate the performance scenario and would consist on a virtual orchestra that interprets and responds to the gestures of the conductor in the same way a real orchestra does. Another application could replicate a teacher during a lesson. In this case, the system should be able to properly identify the correctness of performed gestures and to provide coherent instructions accordingly. These applications could be particularly useful for conductors (professional or students). Obviously, designing this kind of applications requires an extensive knowledge of the replicated scenario and thus can be significantly improved by studying the scenario to be replicated *in situ*.

The second strategy consists on taking the real activity just as a *metaphorical* inspiration. This is, the interaction is designed taking elements from the original activity but in a way that does not try to maximize realism. This is is the case of our work, where we look for strategies to make virtual conducting applications more intuitive and appealing to non-conductors and classical music outsiders. Here, there is total flexibility regarding how close or far we want to be from the actual conducting scenario. For example, a virtual conducting application can consist on a DMI where the user performs just one of the tasks that a real conductor does (e.g. indicating tempo). In this sense, it is possible to decide whether to use knowledge from the replicated activity or not. For instance, in the case of indicating tempo, there are different options. One would be to observe and analyze how real conductors communicate tempo and how orchestras react and to replicate this accurately in the system. Another one would be to predefine a simple rule to trigger beats. And yet another one would be to allow the user to define how she wants to indicate tempo.

As we advanced in the Introduction, this work has been developed, within the PHENICX project, at ESMUC, the Catalan Higher School of Music. Thanks to this, we had the chance to work with professional conductors in different scenarios. We attended conducting lessons, rehearsals and concerts. The availability of commercial contactless unobtrusive motion capture (MoCap) devices such as the Microsoft Kinect allowed us to take recordings of conductors without affecting their performance with an easy setup. An analytical approach of the motion-sound relationships taking place in real performances can help to identify the body movement descriptors that are more useful in the design of DMIs based on the conductor metaphor. Considering that the device used for the recordings is suitable to be used in interactive applications, we ensure that what we learn about these descriptors is applicable.

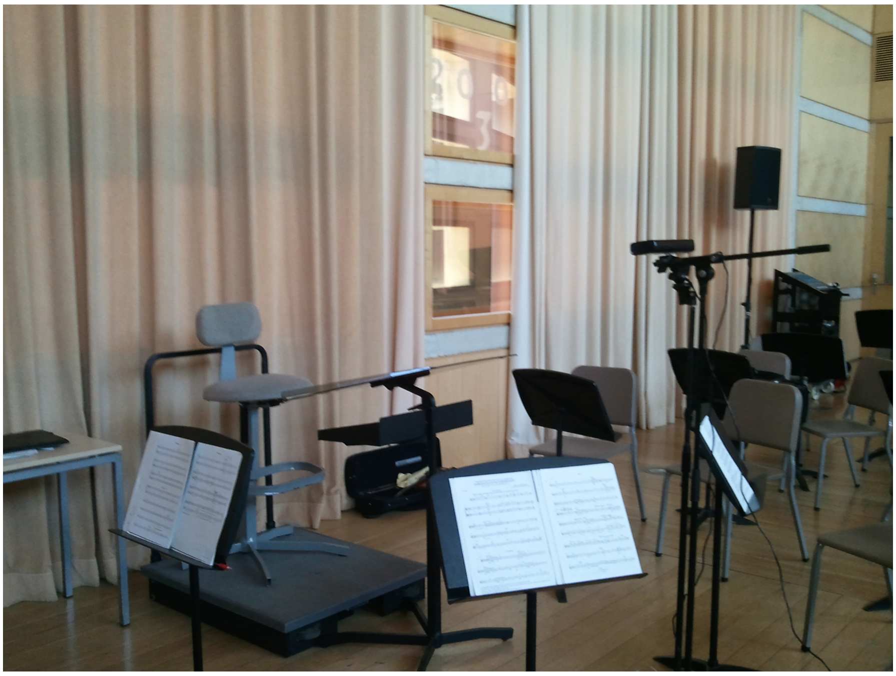


Figure 3.1: Kinect setup for recording in a rehearsal of ESMUC students orchestra.

3.2 Analyzing conducting movements during performance

In this Section, we explain the strategy we followed to computationally analyze the movements of a conductor during an actual performance and present the results of this analysis. First, we briefly explain the very first tests we carried out to address technical issues. Then, we report an interview we had with professional conductors and students to get their feedback about how to face the analysis. Finally, we explain how we built the publicly available multimodal recording of the *Orquestra Simfònica del Vallès* and the conclusions we got from its analysis.

3.2.1 Initial tests

As a first step, we performed some tests in order to identify technical issues to be solved for posterior recordings.

One of the goals of the PHENICX project was to perform and share multimodal recordings including multichannel audio, video and MoCap from the conductor (also enriched with metadata such as the aligned score). The challenge in this kind of recordings is to have all streams aligned in time. In order to test the concrete issues we would face using a Kinect V1 as a MoCap device, we attended two rehearsals of the student orchestra in ESMUC and made recordings including audio from a ZOOM H4n hand recorder and video, audio and MoCap from the Kinect.

For the Kinect data, we developed *KinectVizz*[1], an OpenFrameworks[2] application that uses OpenNI for skeleton tracking. The application allows to record up to four data streams: audio from the Kinect microphone array or laptop built-in microphone, RGB video, RGB-D video and MoCap. The MoCap information is stored in a Tab-Separated-Values (TSV) file where each line contains the information of a frame, including its ID (which increases by 1 at each frame), a timestamp and the x, y and z positions of the fifteen joints provided by OpenNI+NITE skeleton tracking (i.e. forty five position values per frame in total). The application includes functionality to export the recorded data to *Repovizz*[3], an integrated online system developed by Mayor et al. (2011) capable of structural formatting and remote storage, browsing, exchange, annotation, and visualization of synchronous multi-modal, time-aligned data. *Repovizz* in this sense has been not only a platform that we used for sharing data recorded during this work, but also as a visualization tool for all recorded streams.

To avoid the need to have a person close to the conductor running the application, we used RealVNC[4] for remote control through a local network. This way, we can have a laptop capturing the Kinect data close to the conductor's podium and control it from outside the orchestra. Figure 3.2 shows a laptop controlling another one running *KinectVizz* during the recording of a rehearsal by *Orquestra Simfònica del Vallès*. These recordings are explained below, in Section 3.2.3.

Provided that all the streams captured by *KinectVizz* are aligned, our assumption was that a simple manual alignment of the audio stream to the audio captured by the hand recorder would allow to have all streams aligned. These rehearsals were useful to test the application and to make sure that the setup was easy and unobtrusive for all musicians, including the conductor. The location of the Kinect camera during one of these rehearsals is shown in Figure 3.1.

The posterior analysis of these recordings showed that, for long recordings, some frames were dropped in the recorded Kinect streams[5]. This shortens the duration of the resulting recorded data and dramatically affects the alignment to other streams recorded

[1] https://github.com/asarasua/KinectVizz

[2] http://openframeworks.cc/

[3] https://repovizz.upf.edu/

[4] https://www.realvnc.com/

[5] We tested this with different computers and operative systems and the effect persisted. The laptop we used for all reported recordings has a 2,9 GHz Intel Core i7 processor and 8GB RAM, which is quite beyond the Kinect requirements.

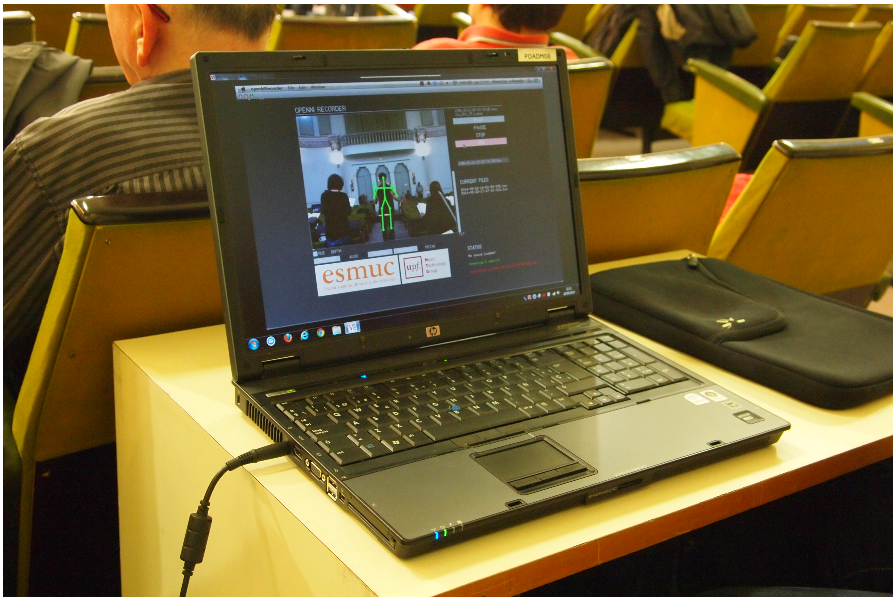
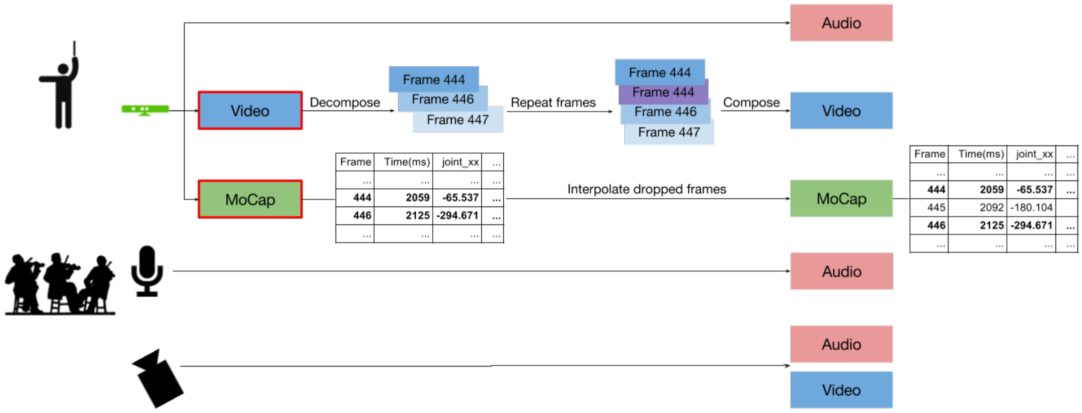


Figure 3.2: Laptop remotely controlling another laptop on stage, running *KinectVizz* for recording aligned multimodal data during a rehearsal by *Orchestra Simfònica del Vallès*.

externally, such as the audio from the hand recorder in this case. The way we overcame this issue was using the information stored in the MoCap TSV file. If no frames are dropped, the ID of consecutive frames are consecutive numbers. When the ID of two consecutive frames stored in the TSV file are not consecutive we can tell how many frames were dropped between them by inspecting the difference between their IDs. Using this information, we generate corrected video and MoCap streams with the process illustrated in Figure 3.3:

- For video, we use *ffmpeg*[6] to decompose the original video into frames. The corrected video is generated, also using *ffmpeg*, by locating the times were frames were dropped and repeating the frame previous to the dropping as many times as frames were dropped. For example, if the TSV file has two consecutive rows with IDs 100 and 106, frame 100 is repeated 5 times in the resulting video. This video freezes for a moment in times were frames had been dropped, but it can be perfectly aligned with other streams and we can automatically annotate where the freezing occurs.
- For MoCap, we follow a similar procedure but perform linear interpolation for the position of joints where frames were dropped. This way, the resulting MoCap file

[6] https://www.ffmpeg.org/



Frame	Time(ms)	joint_xx	...
...	...	...	...
444	**2059**	**-65.537**	**...**
446	**2125**	**-294.671**	**...**
...	...	...	...

Frame	Time(ms)	joint_xx	...
...	...	...	...
444	**2059**	**-65.537**	**...**
445	2092	-180.104	
446	**2125**	**-294.671**	**...**
...	...	...	...

Figure 3.3: Recording setup scheme. Video and MoCap streams recorded from Kinect cannot be directly aligned to other streams due to dropped frames. Information about dropped frames is used to generate corrected video and MoCap data.

does not freeze as the video. However, the interpolation might differ from the actual performed motions during recording, specially if these movements were fast (and increasingly with the number of dropped frames).

3.2.2 Interview with conductors

The interaction between the conductor and the orchestra is highly complex. When we approach the analysis of real conductors during performance we do not want to model all its complexity, but to learn specific insights to later be used in interactive conducting applications. In this context, we carried out an interview with professional conductors and their students, in an educational context, to get their feedback about how to get the most from analyzing a real performance. We made participants discuss about the following questions:

- What actions of the conductors have a causal and measurable effect on the resulting performance by the orchestra?
- Are there useful insights to learn from actual performances to apply in DMIs based on the conductor metaphor for non-conductors?

Seven participants (all male) attended the interview, which took place in ESMUC. Three of them were professional conductors who teach the Bachelor of Music Degree in the Orchestra Conducting speciality, as well as the Master Degree in Advanced Conducting. The other four were students of this Master program.

The interview was organized by asking these two questions to the participants followed by open discussion. We started the meeting by showing the test recordings we already had from rehearsals to explain the kind of recordings we were planning to make and the sort of descriptors that could be extracted from motion. The session was recorded for later analysis of verbal responses. Below we detail the conclusions of such analysis.

Causality and measurability of conductors actions during performance

Conclusion 1 Causal relationships and correlations between conductors gestures and resulting performance are not always present. Participants identified a number of reasons for this:

1. Conductors sometimes decide not to make any actions at specific moments if not required, specially with experienced orchestras.
   - "*You can get great results performing very few gestures. It is sometimes better to leave the orchestra play if it is a good one and they are performing the way you want.*" [Teacher 1]
   - "*I saw Bernstein live once. He made a gesture for the orchestra to start and nothing else. When we feel the orchestra gets to a certain feel, we stop conducting and the orchestra sounds better. Not to conduct is also to conduct*". [Teacher 2]
2. As suggested by one participant, some gestures contrast to the music in order to express a desired change in the performance. This means that the observed gesture could be seen as contradictory with the concurrent outcome.
   - "*You might find that an orchestra is playing an exaggerated staccato and the conductor performs a legato gesture only to correct the performance. The gesture does not always have to be directly related to the sound, he can just be making corrections*". [Teacher 2]
3. The indication of some intention might just appear at the beginning of a phrase and disappear for the rest of it if the orchestra plays as intended.
   - "*You cannot expect, for example, to see that the conductor makes forte gestures all the time that the music is forte. If you want to indicate forte you just indicate it at the beginning of the phrase, there is no need to be constantly communicating it.*". [Teacher 3]

Conclusion 2 The personal style of the conductor and the experience of the orchestra affects the way in which the interaction occurs.

- "*This is all very relative. Even with professionals, every conductor has his own style.*" [Teacher 1]
- "*It is not the same to conduct a young orchestra or a professional one, which is able to immediately react to your gestures.*" [Teacher 2]

Insights to be learned for virtual conducting applications

Conclusion 3 Whatever a user will do with an interactive application is not conducting but *something else*. In this sense, the analysis of the conductor movements during performance has to focus on the specific set of tasks that the user will have in the application (e.g. controlling tempo). One participant (Student 4), who had used an existing interactive conducting application[7], suggested that it could actually benefit from a better knowledge of real conducting gestures in terms of becoming more realistic.

- "*Whatever a non-conductor will do in front of an application will not ever be conducting: it is a game. Here we have students who spend six years studying and even so they do not get enough information.*" [Teacher 2]
- "*Conducting is just something too complicated to be replicated in an application. As we mentioned, it really depends on the orchestra.*" [Student 2]
- "*Conducting is not just communicating the pulse. This is something we realize now as students. It is so complex that I do not think you can automatically measure all its complexity.*" [Student 1]
- "*There is this application in Vienna where you can conduct an orchestra. The funny thing is that if you use actual conducting gestures it does not work properly. The gestures you have to use are very hieratic. I guess that could be improved if you actually look at what conductors do*". [Student 4]

Conclusion 4 There are mainly three aspects of conducting movements to observe on an automatic or semiautomatic analysis: synchronization (beat induction), dynamics and articulation.

- "*Generally speaking, it is true that if you do a gesture like this [waves his hands strongly] it is forte and this other way [waves his hands softly] it is piano.*" [Teacher 1]
- "*I think there are a number of things you could see. For example, legato and staccato gestures are clearly different and you most likely can see that reflected*

[7] The application this participant referred to is the *Virtual Conductor* (Borchers et al., 2002), to which we referred in Section 2.5.2.

on automatic descriptors. However, you must be careful when relating that to the resulting audio." [Teacher 3]

- "*There are orchestras that follow exactly your gesture and others that take a while to react. You might be able to measure this.*" [Teacher 2]

Summary

Even though at the beginning of the interview participants seemed quite skeptic about any kind of computational analysis of conductors movements during performance, they ended up identifying some elements that could be subject to this kind of study. However, always having in mind the considerations that derive from the aforementioned conclusions:

- We can encounter parts where there is no direct correlation between the conductor's movement and the resulting performance.
- The conclusions of the analysis of a conductor with an orchestra might not be applicable to other conductors or orchestras.
- Any analysis trying to get insights for the design of virtual conducting applications has to be focused on the specific tasks that users will have. Good candidates in this sense are synchronization (beat induction), dynamics and articulation.

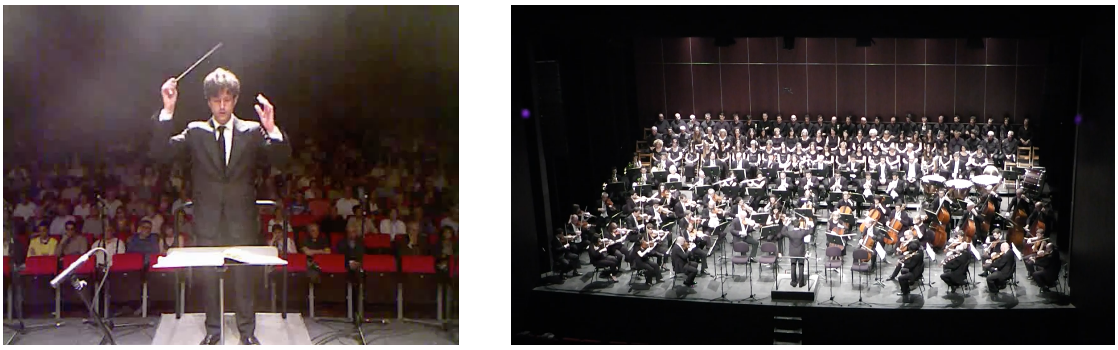
3.2.3 Performance analysis

Having in mind the previous conclusions from the interview, we recorded a performance of Beethoven's 9th Symphony played by the *Orquestra Simfònica del Vallès* and conducted by Rubén Gimeno[8]. The concert took place on May 25th 2014 at the Kursaal theatre in Manresa, Spain. This recording was made in the frame of the PHENICX project and includes data for other tasks such as automatic score alignment and source separation[9]. It is publicly available at http://mtg.upf.edu/download/datasets/phenicx-conduct and includes the following data:

- From the Kinect V1 facing the conductor (all at 30 fps):
  - MoCap with 3D position of nine joints: head, neck, torso, shoulders, elbows and hands. Because of the position at which the conductor stood with respect to the camera (see Figure 3.4a), the six joints below the torso (hips, knees and feet) were occluded by the lectern and could not be correctly tracked.

[8] http://www.rubengimeno.es/

[9] Readers interested in other analyses resulting from this recording are referred to the project academic website http://phenicx.upf.edu.



(a) RGB stream from the Kinect camera.

(b) Video from the orchestra.

Figure 3.4: Snapshots of RGB Kinect Stream (a) and orchestra video (b) of the OSV recording analyzed in this Section.

  - RGB video (640×480 pixels).
  - RGBD video (320×240 pixels).

- 32 channels audio, with microphones placed close to each of the instrument sections.
- Video and audio from a videocamera capturing the whole orchestra.

Figure 3.4 shows snapshots of the RGB Kinect video and orchestra video streams.

The beats in the 4th movement of the Symphony were manually annotated by a musicologist following the score and thus the dataset also provides aligned score information for this movement.

Following the conclusions from the interview, we analyze the performance with two specific goals:

- First, to examine which MoCap descriptors are best synchronized with the available beat annotations and with which delay, and whether this varies depending on the excerpt.
- Second, to examine whether there is a correlation between certain MoCap descriptors and loudness, and whether this correlation varies depending on the excerpt.

Articulation, which was also mentioned by the participants in the interview as a potential aspect to be observed, was left out of this analysis. The reason for this is that we could not identify excerpts in the performance where we could clearly establish differences in articulation for the whole orchestra.

We also recorded, for internal support and testing, the main rehearsal that took place three days before in the orchestra's regular rehearsal space in Sabadell, Spain. In this

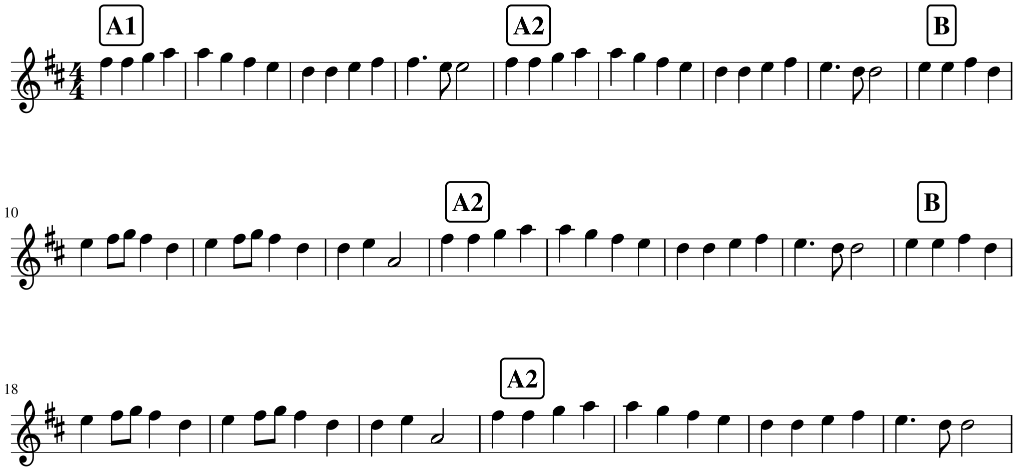


Figure 3.5: *Ode to Joy* melody simplified score.

case, a Zoom H1 digital recorder was used to record stereo audio (instead of the 32 channels available in the concert). This recording is not analyzed here, but is also available online[10].

Excerpts under study

We decided to restrict the analysis to the eight excerpts of the 4th movement where the usually known as "Ode to Joy" theme appears. Figure 3.5 shows a simplified score of the main melody of this theme. All eight excerpts under study are 24-bars long and have the same A1-A2-B-A2-B-A2 structure shown in the score. Other parts in the piece where the theme appears partially or with different structures were left out of the analysis. Having a number of excerpts that share their length (in bars) and structure allows us to look for differences that depend on other elements such as the instrumentation. All excerpts sum 294.44 seconds in total and their complete scores, in PDF and electronic format can be found in the online repository. Table 3.1 contains information about the position of these excerpts in the score and their respective durations along with overall information about the musical content.

[10] http://mtg.upf.edu/download/datasets/phenicx-conduct

Table 3.1: Information about analyzed excerpts from Beethoven's 4th movement. SB: starting bar. D(s): duration of the excerpt (in seconds)

ID	SB	D (s)	Musical information
I1	93	37.00	Cellos and doublebass playing the melody.
I2	117	37.11	Violas play the melody, accompanied by doublebass and bassoons.
I3	141	37.31	Violins play the melody, accompanied by doublebass, viola and bassoons.
I4	165	36.67	Melody played by winds, accompanied by strings and percussion.
V1	242	33.60	A1-A2-B-A2 sung by bass with strings, oboes and clarinets; second B-A2 louder, sung by choir with strings, oboes, clarinets, horns, trumpets and percussion.
V2	270	35.37	A1-A2-B-A2 sung by four soloists with cellos, horns, flutes and bassoons; second B-A2 louder, sung by choir with strings, oboes, clarinets, horns, trumpets, basoons and percussion.
C	376	41.35	A1-A2-B-A2 melody played offbeat by winds with tenor singing a different variation; choir and string join in B-A2 as music gets louder.
T	544	41.14	Constantly loud with choir singing the melody accompanied by strings, horns, flutes, bassoons, trumpets and percussion.

Beat analysis

The goal of this part of the analysis is to examine how musicians synchronized with MoCap descriptors extracted from the conductor. This is very similar to the analysis by Luck and Toiviainen (2006) we referred to in section 2.4.2. However, there are some specific details that are different in our case:

- Luck and Toiviainen used a *Qualisys ProReflex* optical MoCap device, which is able to record at 120 fps with high precision. We were using a Microsoft Kinect, which is not designed to be used as a high precision recording device, but for real-time full-body interaction in video games. While the Kinect is in this sense not ideal in terms of tracking precision and time resolution, we were actually interested in using the same device we would use in an interactive scenario. In addition, it is interesting to test how reliable this sort of device is to perform an analysis of this kind, provided that it is much less intrusive and thus more prone to be used in real performances.
- The recording under study corresponds to an actual public performance in front of an audience. This makes the recording more ecologically valid, but above all

it is particularly necessary in our case, as we want to analyze the movements conductors do in the situation that users of DMIs based on the conductor metaphor will imitate.

- Instead of focusing on some short excerpts selected by the clarity of the communication of the beat by the conductor, we analyze all the occurrences of a specific theme in a movement and look for differences among them.
- Our ground truth is not automatically computed from the audio; it consists on manual annotations of beat positions.

Ground truth The manual annotation of the beat positions in time were used as ground truth. These annotations were manually created by a musicologist following the score with quarter-note precision (i.e. 4 annotations per bar for the 4/4 parts under study).

Motion capture descriptors For the beat analysis, we only considered the position of the right hand, which the conductor used to hold the baton and indicate the beat. The 3D-position of the hand, as well as the rest of the joints, is tracked by the Kinect camera at 30 fps. From this position data, we extracted the same descriptors used in the study by Luck and Toiviainen using a similar procedure. Velocity and acceleration along the three dimensions were extracted from the raw position data using numerical differentiation. More concretely, for each point, we fitted a second-order polynomial to the 7 consecutive points centered at it and computed the derivative of the obtained polynomial. For this, we used Python's `polyfit`[11] function from scientific tools package SciPy (Jones et al.). We also computed the instantaneous speed and acceleration as the length of the vector for both velocity and acceleration and the acceleration along the trajectory by projecting the acceleration vector on the direction of the velocity vector. This results on a total of 12 descriptors. The naming convention for these descriptors, introduced in Section 2.1, is an agreement with Luck and Toiviainen's. In summary, the following 12 variables were used: x, y, z (position; with x = left-right, y = up-down, and z = forward-backward); v_{x}, v_y, v_{z} (velocity components); a_{x}, a_{y}, a_{z} (acceleration components); v (speed); a (magnitude of acceleration); and a_{t} (magnitude of acceleration along the movement trajectory).

Analysis procedure Provided that our ground truth consists on manual annotations of beat positions, we based the analysis on, for each descriptor, (1) evaluating its ability

[11] https://docs.scipy.org/doc/numpy/reference/generated/numpy.polyfit.html

```
input  : $\mathbf{b}^a$, Ground-truth Annotated beats; $\mathbf{b}^p$, Predicted beats
output: $\mathsf{e}$, Error distribution
$\mathsf{IBI} \leftarrow$ mean(diff($\mathbf{b}^a$));
$\mathbf{e} \leftarrow$ Empty array;
$\mathbf{e}^- \leftarrow$ Empty array;
$\mathbf{e}^+ \leftarrow$ Empty array;
for $\mathbf{b}^a_n$ in $\mathbf{b}^a$ do
    $d \leftarrow$ Closest($\mathbf{b}^a_n$, $\mathbf{b}^p$) - $\mathbf{b}^a_n$;
    $d^- \leftarrow$ ClosestPrevious($\mathbf{b}^a_n$, $\mathbf{b}^p$) - $\mathbf{b}^a_n$;
    $d^+ \leftarrow$ ClosestPosterior($\mathbf{b}^a_n$, $\mathbf{b}^p$) - $\mathbf{b}^a_n$;

    Append($\mathbf{e}$, $d$);
    Append($\mathbf{e}^-$, $d^-$);
    Append($\mathbf{e}^+$, $d^+$);
end
if abs(mean($\mathbf{e}$)) $> 0.2 \times \mathsf{IBI}$ then
    if abs(mean($\mathbf{e}^-$)) $<$ abs(mean($\mathbf{e}^+$)) then
        $\mathbf{e} \leftarrow \mathbf{e}^-$
    end
    else
        $\mathbf{e} \leftarrow \mathbf{e}^+$
    end
end
```

Algorithm 1: Procedure to build the error distribution. $\mathbf{b}^a$ and $\mathbf{b}^p$ consist on arrays with the annotated and predicted positions of beats in seconds, respectively.

to correctly estimate these beat positions and (2) estimating its delay with respect to them. For every excerpt, we followed the same procedure for each descriptor:

1. Extract positions of local maxima and minima as predicted positions of beats, respectively $\mathbf{b}^{p_M}$ and $\mathbf{b}^{p_m}$.
2. For both $\mathbf{b}^{p_M}$ and $\mathbf{b}^{p_m}$, compute the error distributions $\mathbf{e}^M$ and $\mathbf{e}^m$ with respect to the annotated beats $\mathbf{b}^a$. To build the error distribution, an error value ϵ_n is stored for every annotated beat $\mathbf{b}^a_n$ corresponding to the difference to the closest value in the predicted beats vector ($\mathbf{b}^{p_M}$ or $\mathbf{b}^{p_m}$, respectively). In cases where the lag of a descriptor with respect to the annotated beat positions is close to half the Inter-Beat Interval (IBI), this process could end up building a bimodal distribution, as a little difference can change the beat to which the predicted position is closest. To avoid this, we consider two cases. (1) If the mean absolute value of $\mathbf{e}^M$ ($\mathbf{e}^m$) is smaller than 0.2 times the mean IBI for the excerpt, we do not apply any corrections. (2) If the mean absolute value of $\mathbf{e}^M$ ($\mathbf{e}^m$) is greater than 0.2 times

the mean IBI for the excerpt, we compute two candidate distributions, $\mathbf{e}^{-}$ and $\mathbf{e}^{+}$, using only beats appearing before and after the annotations, respectively. Then, the distribution for which the absolute value of the mean is lower is chosen. This process to build the error distribution is illustrated in detail in Algorithm 1. The mean of the chosen distribution is stored as an **estimation of the lag** between the descriptor and the annotated beats. For each descriptor, then, we have lag^M and lag^m for $\mathbf{b}^{p_M}$ and $\mathbf{b}^{p_m}$, respectively.

3. For both $\mathbf{b}^{p_M}$ and $\mathbf{b}^{p_m}$, compute the F-measure to determine the **quality of the beat estimation.** The F-measure is computed as the harmonic mean of the *precision* and *recall* values: $F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$. *precision* refers to the proportion of estimated beats that are correct; *recall* corresponds to the proportion of annotated beats that are correctly estimated. In our case, we consider that an annotated beat $\mathbf{b}_n^a$ has been correctly detected if the closest predicted beat is closer than 66 ms (equivalent to 2 frames at 30fps). This F-measure is inspired on the evaluation for audio beat trackers proposed by Dixon (2001), who uses 70 ms for tolerance. Note that we need to take the estimated *lag* into consideration. Otherwise, a descriptor for which we detected beats consistently at time positions 70 ms from the annotated beats would get 0 *precision* and *recall.* For this reason, we correct $\mathbf{b}^{p_M}$ and $\mathbf{b}^{p_m}$, by adding lag^M and lag^m respectively before computing the F-measure.

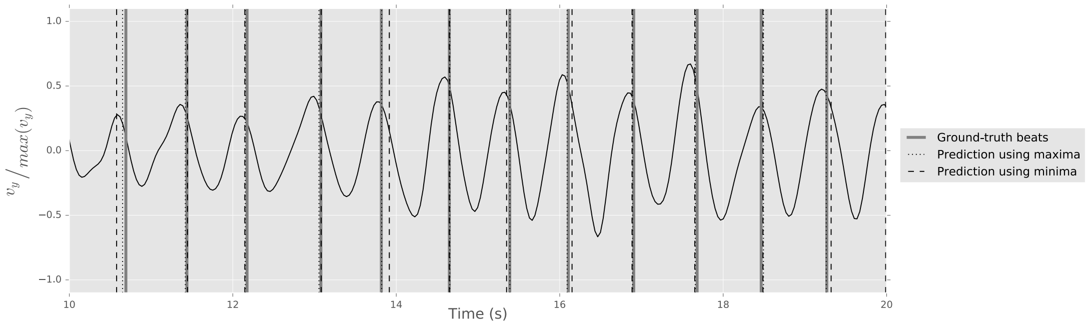
Results Table 3.2 shows a summary of the results across all excerpts under study. More concretely, it shows the estimated lags and computed F-measures of every descriptor and for every excerpt. The results highlighted in bold in Table 3.2 correspond to cases where the obtained F-measure is higher than 0.5. Analogous tables with the results for each of the excerpts can be found in Appendix B.

As an example, if we look at the results for the v_y descriptor in the I2 excerpt, the F-measure of the estimation using its local maxima is 0.84 with an estimated lag of -75 ms. The next two rows show the equivalent results when using local minima to predict beat positions. In this case, the estimated lag is 299 ms and the computed F-measure is 0.73. This means that local maxima of v_y occur closer to actual beats (75 ms before) than local minima do (209 ms after). Figure 3.6a shows how indeed, for the I2 excerpt, local maxima of v_y consistently appear very close before the annotated beats, while local minima appear a while after.

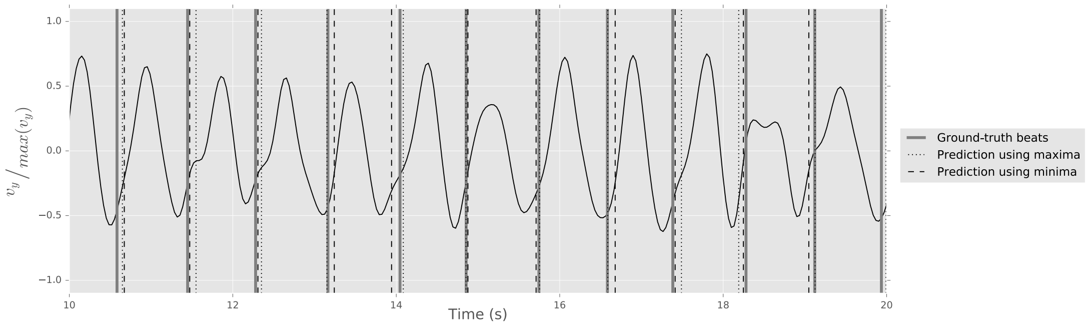
Looking at the results across all excerpts, we observe that the information in the y axis is clearly the most relevant: y, v_y and a_y are the descriptors that consistently get the best results for beat estimation. However, the *lag* of descriptors with respect to the beat

Table 3.2: Lags and F-measures for each of the 12 MoCap descriptors for each of the eight excerpts. “max” and “min” indicate the use of local maxima and minima as beat position candidates, respectively. Results in bold correspond to cases where the F measure is higher than 0.5.

ID			x	y	z	v_x	v_y	v_z	a_x	a_y	a_z	v	a	a_t
I1	max	lag (s)	-.477	**.162**	-.179	.342	**-.126**	**.330**	.046	**-.355**	.087	-.476	.384	.274
		F	.08	**.72**	.44	.34	**.69**	**.51**	.23	**.64**	.28	.31	.21	.24
	min	lag (s)	.239	**-.266**	.318	-.312	**.214**	-.094	-.272	**.050**	-.223	.279	-.021	.258
		F	.27	**.62**	.23	.09	**.63**	.48	.47	**.65**	.44	.17	.33	.19
I2	max	lag (s)	-.396	**.220**	-.155	-.375	**-.075**	-.381	**.215**	**-.277**	.230	.349	.308	.311
		F	.14	**.83**	.36	.44	**.84**	.27	**.50**	**.72**	.41	.12	.19	.22
	min	lag (s)	.227	**-.177**	.379	**-.047**	**.299**	**.010**	-.219	**.120**	-.272	.256	.223	.286
		F	.36	**.77**	.42	**.51**	**.73**	**.56**	.41	**.74**	.30	.21	.22	.19
I3	max	lag (s)	.785	**.194**	-.458	.356	**-.088**	.259	.220	**-.319**	.177	.373	.401	.282
		F	.17	**.80**	.17	.40	**.82**	.33	.39	**.64**	.37	.27	.19	.31
	min	lag (s)	.440	**-.200**	.781	.034	**.267**	-.331	-.242	**.094**	.334	.226	.008	.276
		F	.10	**.72**	.19	.30	**.71**	.15	.37	**.79**	.37	.39	.22	.17
I4	max	lag (s)	-.389	.335	-.172	-.347	.031	.356	.203	-.239	.262	.433	.356	.282
		F	.11	.33	.23	.31	.43	.28	.42	.44	.32	.27	.09	.34
	min	lag (s)	.365	-.069	.478	.031	**.345**	.020	-.278	**.196**	-.248	.292	-.252	-.326
		F	.26	.44	.10	.41	**.55**	.28	.24	**.58**	.36	.24	.17	.20
V1	max	lag (s)	-.317	**.210**	-.566	.308	**-.055**	-.318	.178	**-.230**	.230	.229	.210	-.251
		F	.24	**.91**	.15	.40	**.96**	.45	.31	**.78**	.32	.42	.34	.24
	min	lag (s)	.202	**-.127**	.377	-.022	**.277**	-.010	-.228	**.109**	-.180	.208	.233	.296
		F	.37	**.90**	.24	.43	**.84**	.42	.37	**.87**	.37	.38	.23	.24
V2	max	lag (s)	-.480	**.175**	-.525	.273	**-.094**	.340	.073	**-.263**	.161	.341	.242	.207
		F	.13	**.91**	.10	.39	**1.00**	.33	.45	**.83**	.51	.23	.19	.28
	min	lag (s)	.225	**-.180**	-.205	-.179	**.251**	-.289	-.274	**.070**	-.295	.209	-.296	.304
		F	.22	**.93**	.21	.26	**.79**	.04	.28	**.83**	.34	.33	.16	.22
C	max	lag (s)	.303	**-.264**	.267	-.098	**.295**	-.082	-.243	.084	-.273	-.094	-.294	-.266
		F	.15	**.73**	.29	.29	**.61**	.44	.21	.32	.33	.35	.29	.27
	min	lag (s)	-.434	**.207**	-.290	.318	**-.162**	.333	.067	**-.372**	.058	.231	-.038	.307
		F	.27	**.55**	.20	.36	**.57**	.34	.36	**.57**	.37	.27	.26	.24
T	max	lag (s)	.327	-.211	.488	-.045	**.421**	.051	-.210	.238	-.308	-.502	-.402	.032
		F	.27	.42	.17	.21	**.54**	.24	.29	.40	.28	.11	.10	.27
	min	lag (s)	-.461	**.328**	-.337	-.375	-.012	-.340	.331	**-.234**	.259	-.266	.209	-.324
		F	.08	**.52**	.19	.29	.48	.29	.21	**.50**	.23	.23	.31	.16



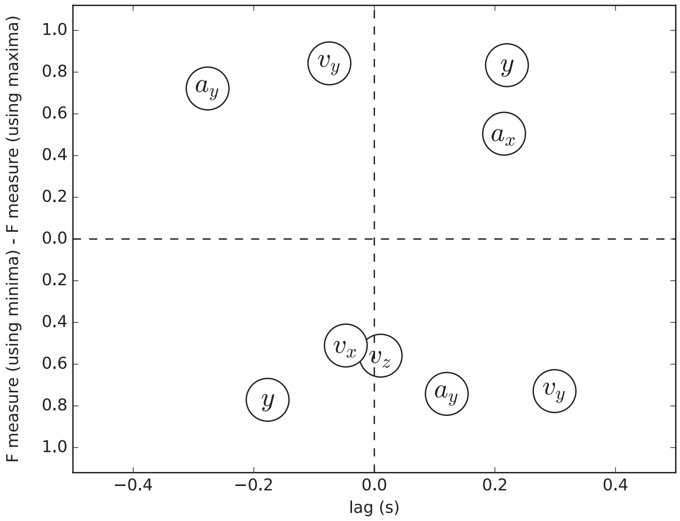
(a) Excerpt I2.



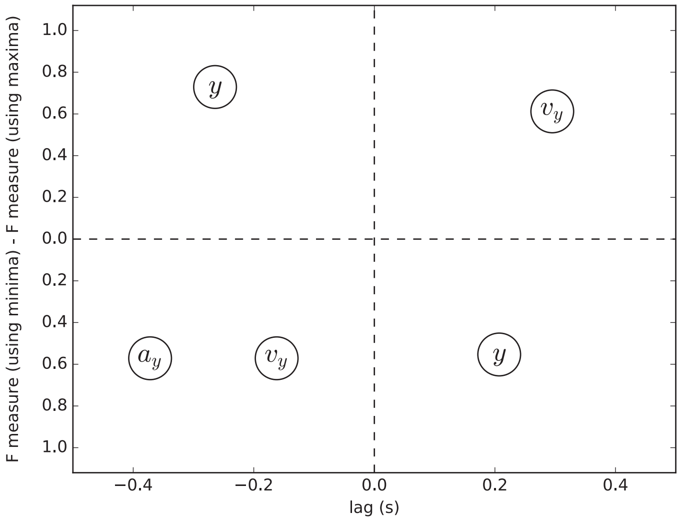
(b) Excerpt C.

Figure 3.6: Fragments of velocity component in the vertical axis (v_y) for the I2 (a) and C (b) excerpts. Solid vertical lines are annotated beat positions (ground truth); dotted vertical lines are estimated beat positions using v_y maxima; dashed lines are estimated beat positions using v_y minima.

is not consistent. If we take again v_y as an example, we can observe that while the lag of the beats estimated taking its maxima is relatively small for most excerpts (e.g. -126 ms for I1, -75 ms for I2 and -88 ms for I3), it is larger for others (e.g. 295 ms for C, 421 ms for T). Looking at Figure 3.6 again, we can compare v_y for excerpts I2 (Figure 3.6a) and C (Figure 3.6b). In the case of I2, local maxima appear very close to the annotated beats (solid vertical lines), while in the case of C they appear later. The Figure also shows the estimated positions of beats using maxima (dotted vertical lines) and minima (dashed vertical lines). Note that the estimated position of beats takes the estimated *lag* into consideration.



(a) Excerpt I2.



(b) Excerpt C.

Figure 3.7: F-measure vs lag plots for excerpts I2 (a) and C (b). Only descriptors for which F-measure > 0.5 are shown.

For the sake of comparison with Figure 2.2 by Luck and Toiviainen, Figures 3.7a and 3.7b show equivalent representations for excerpts I2 and C, respectively, with descriptors for which the computed F-mesaure was higher than 0.5. Cases using maxima and minima are mirrored with respect to the y axis (F-measure) for better comparison with Figure 2.2, where the y axis corresponds to the correlation (a high negative correlation corresponds to a high F-measure using minima, in our case). In these figures, the farther points are from 0 in the y axis, the better the beat estimation is; the closer they are from 0 in the x axis, the lower the lag. For the descriptors that appear in both Figures 3.7a and 3.7b such as y, v_y and a_y, it is clearly noticeable the shift in the x axis between both graphs. This shift reflects the aforementioned difference in the lag between both excerpts.

It is interesting to notice that, in our case, a_t was not a good descriptor to predict the beats in any of the excerpts under analysis, as opposed to the case of the study we are comparing with. Further inspection of the complete data (available in Appendix B) reveals that the low F-measure values are not the result of low *recall* only (a combination of high *precision* and low *recall* values could correspond to a descriptor that, for example, correctly predicts half of the beats). It is however the result of a_t being quite noisy. In this sense, our results for this descriptor are not necessarily in contradiction with those by Luck and Toiviainen. Instead, this shows that the correct computation of a_t, which results from projecting the acceleration vector in the direction of the velocity vector, is not reliable using a device like the Microsoft Kinect, which operates at 30 fps.

Dynamics analysis

In this part, we examine which MoCap descriptors are best correlated with the loudness of the performance.

Ground truth Loudness computed directly from the audio is used as ground-truth. We used a stereo mix of the whole orchestra as input audio and computed its loudness using Essentia's (Bogdanov et al., 2013) Loudness[12] algorithm. This algorithm computes the loudness of the audio signal using power law by Stevens (1975), as the energy of the signal raised to the power of 0.67.

Motion capture descriptors For this part of the analysis, we computed three features describing the general characteristics of the body movement: Quantity of Motion (QoM), Contraction Index (CI) and highest hand position (Y_m). To compute QoM, we averaged the mean velocity of all nine available joints. For CI, we looked at maximum and minimum values along every axes and empirically derived an equation to make its value approximately 1 when arms are completely stretched out and 0 for a very contracted pose:

[12] http://essentia.upf.edu/documentation/reference/std_Loudness.html

$$CI(t_i) = \frac{-4 + \frac{|x_{max}(t_i) - x_{min}(t_i)| + |y_{max}(t_i) - y_{min}(t_i)| + |z_{max}(t_i) - z_{min}(t_i)|}{h(t_i)}}{6} \tag{3.1}$$

Where $h = (x^{head}(t_i) - x^{torso}(t_i))^2 + (y^{head}(t_i) - y^{torso}(t_i))^2 + (z^{head}(t_i) - z^{torso}(t_i))^2$ denotes the squared distance between the head and torso joints, which is proportional to the participant's height and $i_{max}(t_i)$ and $i_{min}(t_i)$ denote the maximum and minimum values along i axis across all joints at time t_i.

Y_{max} is a simple descriptor that takes at each time the highest y position of both hands:

$$Y_{max}(t_i) = max(y^{LH}(t_i), y^{RH}(t_i)) \tag{3.2}$$

Analysis procedure In this analysis we are interested in studying the overall relationship between MoCap descriptors and loudness. In order to emphasize this overall relations as opposed to doing a frame-by-frame analysis, we followed this procedure for every excerpt:

1. Compute the average IBI for the excerpt.

2. Low-pass filter the three descriptors and the computed loudness at $(4 \cdot IBI)^{-1}$ Hz. $4 \cdot IBI$ corresponds to the average length of a bar in the excerpt, so by low-pass filtering the descriptors at the frequency corresponding to this time interval, the signals are softened to represent variations at the bar time scale.

3. Compute a linear regression model through least squares regression using MoCap features as predictors and loudness as the independent variable. From the computed model, we look at the Pearson correlation values between each feature and loudness, and at the adjusted coefficient of determination, R^2_{adj}, which measures the proportion of the variance in the dependent variable (loudness) that the independent variables (the MoCap descriptors) account for.

In addition, we computed another linear regression model taking the data from all excerpts together with the same procedure, in order to test how generalizable the MoCap descriptors - loudness relationship was across excerpts.

Variables	P corr.	p value
Intercept		<0.001
QoM	0.400	<0.001
CI	**0.505**	<0.001
Y_{max}	0.408	<0.001
		R^2_{adj}: 0.358

(a) I1

Variables	P corr.	p value
Intercept		<0.001
QoM	0.432	<0.001
CI	0.212	>0.1
Y_{max}	0.097	<0.05
		R^2_{adj}: 0.191

(b) I2

Variables	P corr.	p value
Intercept		<0.001
QoM	**0.544**	<0.001
CI	0.093	<0.001
Y_{max}	0.373	<0.001
		R^2_{adj}: 0.399

(c) I3

Variables	P corr.	p value
Intercept		<0.001
QoM	0.281	<0.001
CI	0.167	<0.001
Y_{max}	0.334	<0.001
		R^2_{adj}: 0.164

(d) I4

Variables	P corr.	p value
Intercept		<0.01
QoM	**0.693**	<0.001
CI	0.352	<0.001
Y_{max}	**0.803**	<0.001
		R^2_{adj}: 0.816

(e) V1

Variables	P corr.	p value
Intercept		<0.001
QoM	**0.795**	<0.001
CI	0.412	>0.1
Y_{max}	**0.524**	<0.001
		R^2_{adj}: 0.789

(f) V2

Variables	P corr.	p value
Intercept		<0.001
QoM	**0.553**	<0.001
CI	0.416	<0.001
Y_{max}	**0.551**	<0.001
		R^2_{adj}: 0.404

(g) C

Variables	P corr.	p value
Intercept		<0.001
QoM	0.340	<0.001
CI	-0.073	<0.05
Y_{max}	-0.157	<0.001
		R^2_{adj}: 0.158

(h) T

Table 3.3: Statistics from computed linear regression models for each excerpt. Pearson correlation (P. corr) values with absolute value greater than 0.5 are highlighted in bold.

Results Tables 3.3a to 3.3h show statistics from the computed regression models for the eight excerpts. Taking the coefficient of determination, R^2_{adj}, of each of the models as a measure of their quality for loudness estimation, we observe that the excerpts where the best models are computed are V1 and V2 ($R^2_{adj} = 0.816$ and $R^2_{adj} = 0.789$, respectively), followed by C ($R^2_{adj} = 0.403$). The first two are precisely the excerpts where the clearest

Table 3.4: Statistics from computed linear regression model using all excerpts. Pearson correlation (P. corr) values with absolute value greater than 0.5 are highlighted in bold.

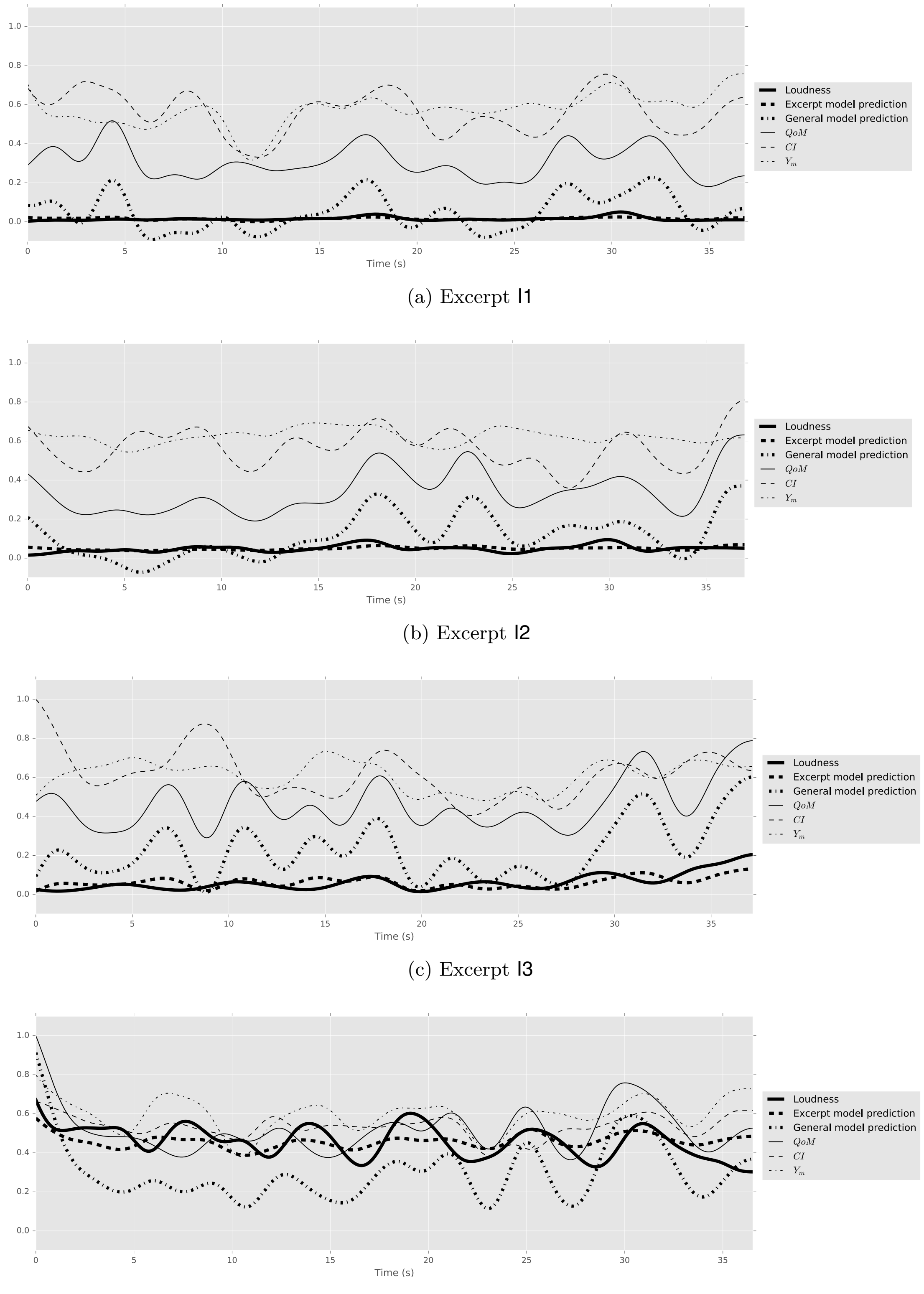
Variables	P corr.	p value
Intercept		<0.001
QoM	**0.593**	<0.001
CI	0.194	<0.001
Y_{max}	0.342	<0.001
		R^2_{adj}: 0.390

variations in loudness occur, as already pointed out in Table 3.1. In these excerpts, the second repetition of B-A2 in the theme is much louder than the rest, with the whole choir singing the melody and most instrument sections playing. In excerpt C, the choir also joins in the second repetition of B-A2 but loudness increases less and more gradually than in V1 and V2. The rest of the excerpts do not contain clear variations in loudness.

In V1, V2 and C, QoM and Y_{max} show high (>0.5) positive correlation with loudness. This suggests that the conductor performs more *energetic* movements with the hands in a higher position in loud parts of these excerpts, and less *energetic* movements with hands in a lower position in soft parts. In the case of the position of the hands, this correlation is very likely due to the fact that the choir (standing at the farthest position from the conductor) sings in loud parts, so these loud parts coincide with parts where the conductor gives instructions to the choir raising his hands.

The statistics from the regression model that results when using all excerpts together are shown in Table 3.4. In this case, only QoM shows a high (>0.5) positive correlation with loudness.

To better illustrate the results, Figures 3.8 (excerpts I1, I2, I3 and I4) and 3.9 (excerpts V1, V2, C and T) show the time series of MoCap descriptors and loudness together with the predicted loudness by the models of each excerpt and the general one. In excerpts I1, I2 and I3 (Figures 3.8a, 3.8b and 3.8c), where loudness remains mostly constant, the general model fails at correctly predicting the loudness for excerpts. In the case excerpts I4 and T (Figures 3.8d and 3.9d), which contain some more fluctuations in loudness, the general model fails at predicting the correct values, but succeeds in replicating these fluctuations. The best performance of the general model occurs in excerpts V1, V2 and C (Figures 3.9a, 3.9b and 3.9c), although it overestimates the influence of QoM in the latter.



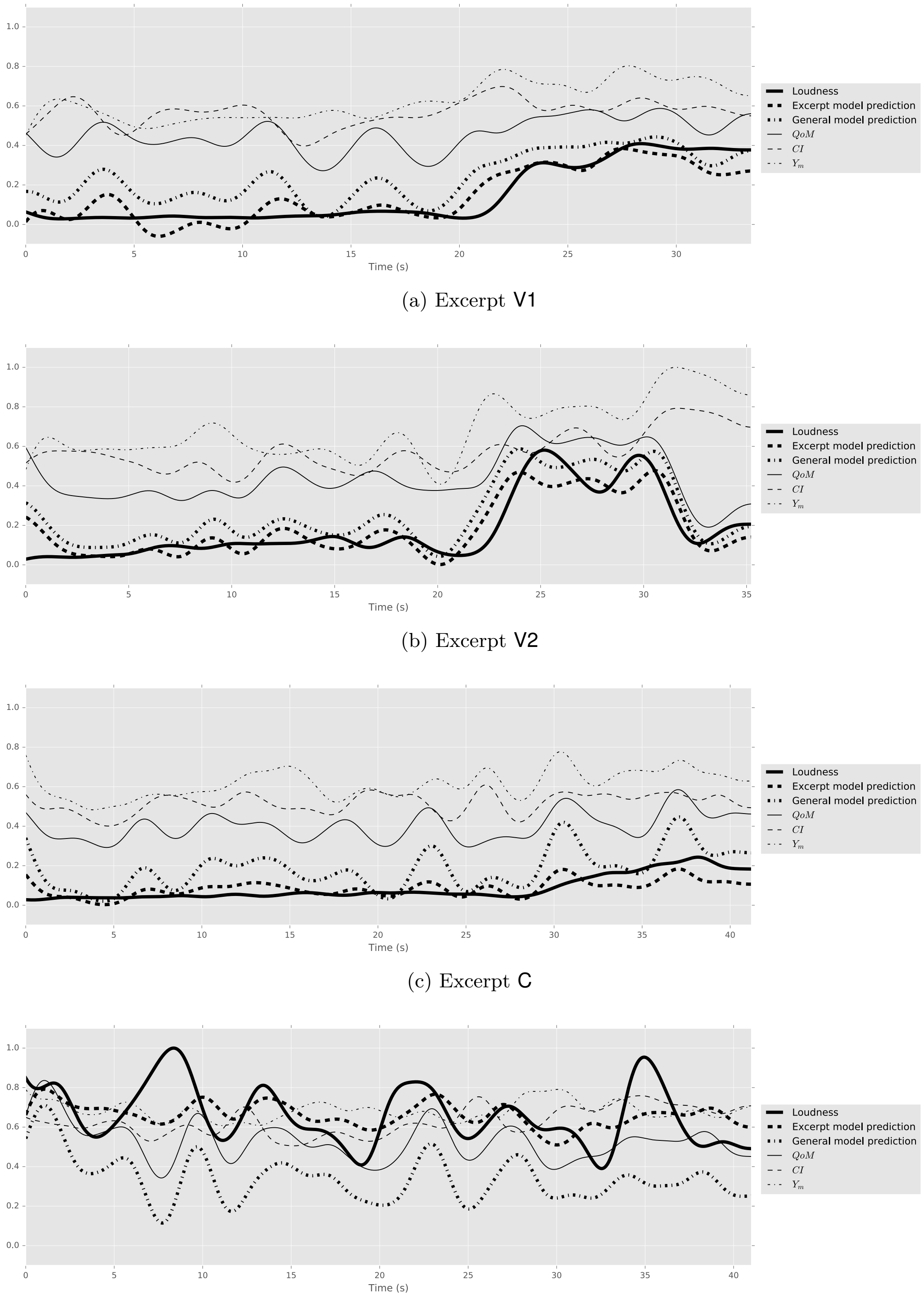
(a) Excerpt I1

(b) Excerpt I2

(c) Excerpt I3

(d) Excerpt I4

Figure 3.8: MoCap descriptors, loudness and loudness estimations for excerpts I1, I2, I3 and I4. All time series are normalized with 0 and 1 corresponding to the minimum and maximum values taken by each variable.



(a) Excerpt V1

(b) Excerpt V2

(c) Excerpt C

(d) Excerpt T

Figure 3.9: MoCap descriptors, loudness and loudness estimations for excerpts V1, V2, C and T. All time series are normalized with 0 and 1 corresponding to the minimum and maximum values taken by each variable.

3.3 Conclusions

We analyzed a performance with the same kind of device we planned to use to build DMIs based on the conductor metaphor; in this case, a Kinect V1. We did so to identify which MoCap descriptors are best to describe the relationships between the movement of the conductor and specific aspects of the performance potentially controllable in an interactive scenario.

In this context, we proposed a technical solution to make long recordings with this device and align them with other data streams. This tool, which allows to generate a *Repovizz* datapack that can be uploaded and visualized in this platform, is publicly available online[13].

Before actually doing the recording and its analysis, we carried out an interview with professional conductors and their students, in an educational context, to get their feedback about how to plan and what to expect from the analysis. The interview led to the conclusion that the analysis of a live performance in our context needs to be done focusing on the specific tasks that users of the application will have, and that good candidates for this are the synchronization, the dynamics and articulation. Also, participants in the interview warned that causal relationships between the conductor's movement and the resulting performance are not always present.

We recorded and made public[14] a live performance of Beethoven's 9th Symphony by the *Orquestra Simfònica del Vallès.* For this analysis, we selected the excerpts in the 4th movement where the usually known as *Ode to Joy* theme appears. We examined which MoCap descriptors were best synchronized with the available beat annotations and which are best correlated with the loudness of the performance. Regarding the beat, we saw that descriptors related to the movement of the hand holding the baton in the y (up-down) axis were best to correctly estimate the beat in the performance. We also observed that the lag between these descriptors and the beat was not the same across all excerpts. The analysis of the relationship between MoCap descriptors and loudness revealed that there are excerpts in which this relationship can be very clear, while in others it is hard to establish, in accordance with the warning raised in the interview. For the excerpts where the relationship existed, the quantity of motion, which averages the velocity measuring the general movement intensity, was the best correlated with loudness. The height at which the conductor raised his hands was also correlated with loudness, although in this case this could be an effect of the specific excerpts under

[13] https://github.com/asarasua/KinectVizz
[14] http://mtg.upf.edu/download/datasets/phenicx-conduct

study, where the choir sang in loudest parts. The scores of the analyzed fragments are also available in the online repository, and the beat analysis summary tables for each excerpt are available in Annex B.

Beyond the concrete conclusions of this analysis, the most useful outcome of this part is the methodology we used. It allowed us to determine the relationships between the movement of the conductor and the beat and loudness of the performance in specific excerpts. In subsequent chapters, we use the same approach to study how people with different musical expertise, potential users of a virtual conducting application, embody the beat and loudness when performing conducting movements. We also apply it to articulation, which was left out of this analysis due to the lack of excerpts in the performance where we could clearly establish differences in articulation.

Chapter 4

Adapting to user-specific tendencies for beat and dynamics control

In this Chapter, we move to the exploration of strategies to adapt the mapping of a DMI based on the conductor metaphor by explicitly exploiting the value of this interface metaphor. As we advanced in the Introduction, interface metaphors are used so that the user can transfer her knowledge from the activity replicated by the metaphor to the interaction with the computer or DMI. Following this idea, in this Chapter we begin by observing what different users do when asked to "conduct" spontaneously, i.e. without specific instructions. With the methods used in Chapter 3 to analyze a conductor during performance, we analyze the different tendencies of users in terms of beat anticipation and loudness communication. Next, we apply this in an interactive context, with a DMI that allows to control beat and loudness with a predefined mapping, but where some parameters are adjusted specifically for each user according to the analysis of their spontaneous movements. We refer to this strategy as *Mapping by Observation.*

4.1 Introduction

As we saw in Section 2.5.1, the commonly considered first DMI using the conductor metaphor is the *Conductor Program* by Mathews (1976). In the version using two *Radio Batons* (Boie et al., 1989; Mathews, 1991), one of them (usually the one held in the right hand) is used to trigger beats when its vertical position is below a certain threshold. The position of the other baton can continuously control other parameters, most commonly loudness, or balance of different instruments. The same strategy of using information directly derived from the hand or hand-held device position to trigger beats is commonly found in systems that came after Mathews' (Haflich and Burnds, 1983; Keane and Gross, 1989; Morita et al., 1989; Borchers et al., 2002; Lee et al., 2004; Bergen, 2012; Toh et al., 2013; Rosa-Pujazon and Barbancho, 2013). Other systems exploit machine learning techniques such has Hidden Markov Models (HMMs)(Usa and Mochida, 1998a; Kolesnik, 2004) or Artificial Neural Networks (ANNs) (Brecht and Garnett, 1995; Ilmonen and Takala, 1999) to deal with temporal information from the gesture. Regarding loudness control, there are some cases where it is performed through specific gestures like raising and lowering one hand (Toh et al., 2013; Rosa-Pujazon and Barbancho, 2013). However, it is more common to find cases where descriptors computed from the movement (most commonly its size) are directly *wired* to loudness control (Morita et al., 1989; Usa and Mochida, 1998a; Lee et al., 2004; Toh et al., 2013).

An important aspect in systems where tempo is controlled by triggering beats is to provide accurate control on the exact time when the orchestra plays following the performed movement. In most cited works, it is commonly assumed that the “beat induction” instant in the movement (the ictus) corresponds to the change from downward to upward hand motion, so beats are triggered when this change is detected. However Lee et al. (2005) identified some usability breakdowns when qualitatively analyzing how people performed with their systems in public spaces (Borchers et al., 2002; Lee et al., 2004), and decided to analyze with more detail the temporal relationship between users’ conducting gestures and the beat on a musical piece. In order to do so, they asked conductors and non-conductors to “conduct” a fixed musical clip from the *Radetzky March* using up-down movements making them aware that they were not affecting the resulting sound in any way. They found that conductors tended to lead the music beat by an average of 150 ms, while it was 50 ms for non conductors, who also showed larger variance in the placement of the gesture beat with respect to the music beat. Lee et al.’s hypothesis, following the conclusions from their study, was that incorporating this knowledge to conducting systems could improve their usability.

However, in a context where the target application is a public installation, as in their case, we believe it is potentially better to perform user-specific rather than profile-specific interface adaptations. The problem in this case is that this user-specific tailoring must be done just before or during the interaction. With this in mind, we performed an observation study, presented in Section 4.2, where we analyzed the movements of different participants when asked to “conduct” on top of a musical excerpt. In our case, we did not ask to perform any specific gesture like up-down movement or to focus on any specific aspect of the performance like the beat. Instead, as introduced above, we are interested in analyzing spontaneous (i.e without instructions) movements. Later, in Section 4.3, we explore whether, as we hypothesize, this kind of analysis is useful in an interactive context. We present a DMI with a predefined mapping to control beat and loudness with some parameters that are adjusted specifically for each user analyzing spontaneous movements. We refer to this strategy as *mapping by observation*, as the mapping adaptation is done by *observing* the user perform the activity that inspires the interface metaphor.

4.2 Observation study

4.2.1 Objectives

In this study, we analyze how different people perform spontaneous (i.e. without instructions) conducting movements on top of fixed musical excerpts. More concretely, the specific goals are:

- To investigate if participants move following the beat of the music, and whether the anticipation to the beat is different across participants.
- To identify if the loudness of the music is reflected in the participants' body movement, and whether it does in similar or different ways across participants.

For this analysis, we use a similar strategy to the one followed in Chapter 3 with a real conductor. For beat analysis, we examine whether beats extracted from the same descriptor are aligned with the musical beat differently for each participant. For loudness analysis, we investigate whether different descriptors extracted from body movement can predict loudness, and whether there are differences across participants.

4.2.2 Materials and methods

Materials

We used *KinectVizz*[1] for the recordings. For this, we incorporated a new functionality to the application which allows to select an audio file and play it while recording video and motion capture from the Kinect aligned with this audio. For the analysis, we only used information from the nine upper-body joints: head, neck, torso, shoulders, elbows and hands.

We selected excerpts from a performance of Beethoven's 3$^{\text{rd}}$ Symphony (*Eroica*) 1st Movement performed by the Royal Concertgebouw Orchestra[2] for which multimodal data (including high quality audio for every section, multi-perspective video and aligned

[1] https://github.com/asarasua/KinectVizz

[2] http://www.concertgebouworkest.nl/

score) were made available within the PHENICX project[3]. This movement, *Allegro con brio*, is in 3/4 time.

We selected 35 seconds fragments so we have enough data while allowing users to memorize them in a short time period. Fragments were chosen to have some dynamics and tempo variations. All files were converted to mono so participants did not have to pay attention to spatialization. Beat annotations are available in the dataset and were used as ground truth beats location. Loudness values were computed from audio using Essentia's (Bogdanov et al., 2013) Loudness[4] algorithm. This algorithm computes the loudness of the audio signal using power law by Stevens (1975), as the energy of the signal raised to the power of 0.67. Computed values were resampled to 30 Hz (the rate of the MoCap data) in order to make a frame-by-frame comparative analysis with respect to MoCap descriptors.

During the study, participants used over-ear headphones and stood approximately two meters from the Kinect sensor, placed on a flat speaker stand, approximately 1.4 m from the floor. The experimenter read instructions to participants and controlled the application from a laptop to which the Kinect sensor and headphones were connected.

The recorded data and the scores of the fragments used in the study are available online[5].

Methods

Motion Capture descriptors Here we detail all MoCap descriptors that were extracted from raw position data (x, y, z) of the nine upper-body joints. They are classified into *joint descriptors*, computed for every joint, and *body descriptors*, describing general characteristics of the whole body movement.

- **Joint descriptors**:
  - (v_x, v_y, v_z), (a_x, a_y, a_z): Velocity and acceleration components, computed by fitting a second-order polynomial to 7 subsequent points centered at each frame and taking the derivative of the polynomial. We used Python's `polyfit`[6] function from scientific tools package SciPy (Jones et al.) for this.
  - v, a: Velocity and acceleration magnitudes.
  - v_{mean}, v_{std}: Velocity magnitude mean and standard deviation, computed from 31 (1.03 seconds) values around each frame. They are expected to account

[3] https://repovizz.upf.edu/phenicx/datasets/
[4] http://essentia.upf.edu/documentation/reference/std_Loudness.html
[5] http://mtg.upf.edu/download/datasets/phenicx-conduct
[6] https://docs.scipy.org/doc/numpy/reference/generated/numpy.polyfit.html

for the “quantity” and “regularity” of the joint movement, respectively.

  - $(x_{tor}, y_{tor}, z_{tor})$: Relative position with respect to the torso components. These are computed from the position of the joint and the position of the torso. For the x axis, points to the left (from the subject perspective) of the torso are positive and points to the right are negative. In the y axis, points over the torso are positive and points below are negative. In the z axis, points in front of the torso are positive and points behind are negative. Appropriate weights were empirically estimated to make the values approximately 1 for the case of hands completely extended in the corresponding axis.
  - d_{tor}: Distance to torso, computed from the position of the torso and joint of interest.

These last two sets of descriptors, $(x_{tor}, y_{tor}, z_{tor})$ and d_{tor}, are related to the *shape* component in LMA (see Section 2.1). Even though they are computed using information from two different joints, we list them as joint descriptors as they inform about how stretched a joint is with respect to the torso.

- **Body descriptors:**
  - QoM: Quantity of Motion. It is computed as the average magnitude velocity of all tracked joints.
  - CI: Contraction Index. Is is computed by looking at maximum and minimum values along every axis. We used equation 3.1, empirically derived to make its value approximately 1 when arms are completely stretched out and 0 for a very contracted pose.
  - Y_{max}: maximum hand height. This is a simple descriptor that takes at each time the highest y position of both hands.

Loudness analysis In order to study the relationship between the participants movement and the loudness of the fragments, we performed least squares linear regression, using movement descriptors as predictors and the computed loudness as the independent variable. As opposed to the case of the real conductor in the performance studied in Section 3.2.3, where we preselected three descriptors based on expert knowledge (QoM, CI and Y_m), we now follow a *blind* approach with a larger set of descriptors and allowing the model to identify the relevant ones. We created different linear models for different levels of specificity (general to subject-specific). In all cases, we started from maximal models including all descriptors and kept simplifying by removing non-significant explanatory variables until the resulting model only contained descriptors with a significant effect on loudness, to get the minimal adequate model.

Beat analysis We took maxima in vertical acceleration of the hand, a_y, as beat position estimates. For each participant, we automatically selected her hand by looking at which of the two of them showed more activity, estimated as the total energy of a_y for each hand during the n frames under analysis: $E = \sum_n a_y(t_n)^2$. We then used the manual annotations of beat positions as ground truth to build an error distribution[7] $\mathbf{e}$, following the procedure detailed in Algorithm 1. The mean of the distribution is used as an estimation of the *lag* between the detected and the annotated beats.

We compute the F-measure as the harmonic mean of the *precision* and *recall* values: $F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$. *precision* refers to the proportion of estimated beats that are correct; *recall* corresponds to the proportion of annotated beats that are correctly estimated. In our case, we consider that an annotated beat a_n has been correctly detected if its closest predicted beat is closer than 66 ms (equivalent to 2 frames at 30 fps). The estimated *lag* is used to correct the position of detected beats before computing F. An F-measure of 1 indicates that the position of annotated music beats can be perfectly estimated from hand movement acceleration with a ±66 ms precision.

Participants

Participants were recruited via convenience sampling through department members and their students. They subsequently signed an informed consent form informing about the type of data being recorded and the intention of making it publicly available, filled out a brief pre-questionnaire, performed the tasks of the study, and filled out a post-questionnaire. The study involved approximately 10 minutes per participant.

Procedure

For each of the 35 seconds fragments, participants were allowed to listen to them twice (so they could focus on learning the music). Then, they were asked to “conduct” the fragment three times (so they could keep learning the music while already practicing their conducting). For each of the fragments, the analysis was only performed in the last of the three takes, where participants are assumed to be able to better anticipate changes (i.e. this way being closer to “conducting” and not just “reacting” to changes).

[7] Although we speak of "error distribution", a beat estimated from hand movement appearing far from an annotated beat does not imply that there is anything *wrong*, participants did not have to accomplish any task.

Also, to get rid of the effects of initial synchronization, the analysis was done on the 30 seconds from second 4 to second 34. This makes a total a total of 90 seconds (30 seconds per fragment) of conducting data for each participant.

At the end of the recording, participants filled another questionnaire with Yes/No questions about how they had faced the study and the problems they had encountered. Concretely, the questions we asked were the following:

- *Were you able to recognize the time signature (3/4) in the excerpts?*
- *In general, did you use rhythm information to guide your conducting movements?*
- *In general, did you use loudness information to guide your conducting movements?*
- *Were you able to anticipate changes in the last take of each excerpt?*

4.2.3 Results

Twenty five participants participated in the study, of which six were female. Ages were distributed as 21-25 (2), 26-30 (9), 31-35 (8), 36-40 (5) and $>$ 40 (1). Their musical background was distributed as "no musical training" (4), "some non formal training" (4), "less than 5 years of formal musical training" (4) "and more than 5 years of musical training" (13). Of these, one was an expert conductor and 5 had some basic notions of conducting technique.

Loudness analysis

In the post-recording questionnaire, participants where asked about whether they had consciously conducted according to loudness variations. Eight participants indicated that they had actually not used this information. In preliminary observations, participant by participant, we looked for descriptors showing a high ($>$ 0.5) correlation with loudness, finding none for the aforementioned participants. For this reason, we left these participants out of the analysis. The resulting set of participants ($n = 17$) is composed by 2 non musicians, 3 non-formal musician, 3 trained musicians and 9 expert musicians.

Grouping participants by tendencies In this preliminary observations where we looked for features correlated to loudness by more than 0.5, we observed a tendency that is also clearly noticeable when playing back the 3D models of the MoCap recordings: while for most participants QoM appears as highly correlated with loudness, for some others we observe a strong correlation between Y_{max} and loudness.

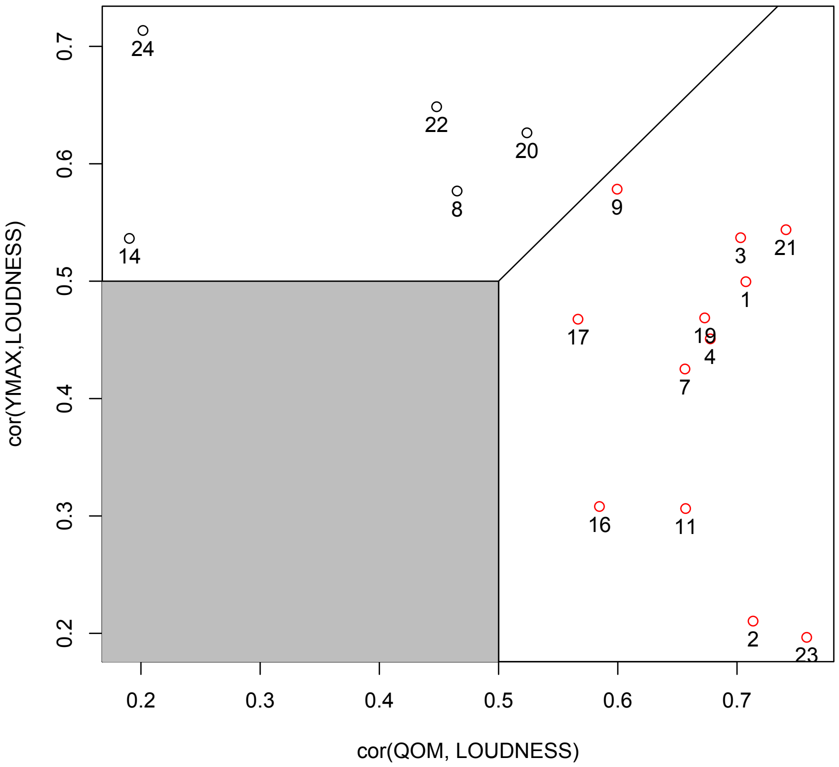


Figure 4.1: Participants in the space formed by $cor(QoM, loudness)$ and $cor(Y_{max}, loudness)$. The shaded area corresponds to points where neither $cor(QoM, loudness)$ nor $cor(Y_{max}, loudness)$ are > 0.5. Participants are divided into the two groups corresponding to areas at both sides of the line: in red, $cor(QoM, loudness) > cor(Y_{max}, loudness)$; in black, $cor(Y_{max}, loudness > cor(QoM, loudness)$.

Following this observation, we split participants into two separate clusters according to the correlation of these two descriptors with loudness: "QoM cluster" ($n = 12$) and "Y_{max} cluster" ($n = 5$), respectively, depending on which of the two correlations is higher. Participants falling into each of the two groups are illustrated in Figure 4.1. Then, we created linear regression models for each of these two clusters.

The statistics from the models that resulted for each of the clusters, illustrated in Table 4.1, are coherent with the rationale applied when splitting participants. For the case of the "QoM cluster", none of the variables relating to the position in the y axis of the hands appeared as significant. QoM does not show in any of the two models, but this

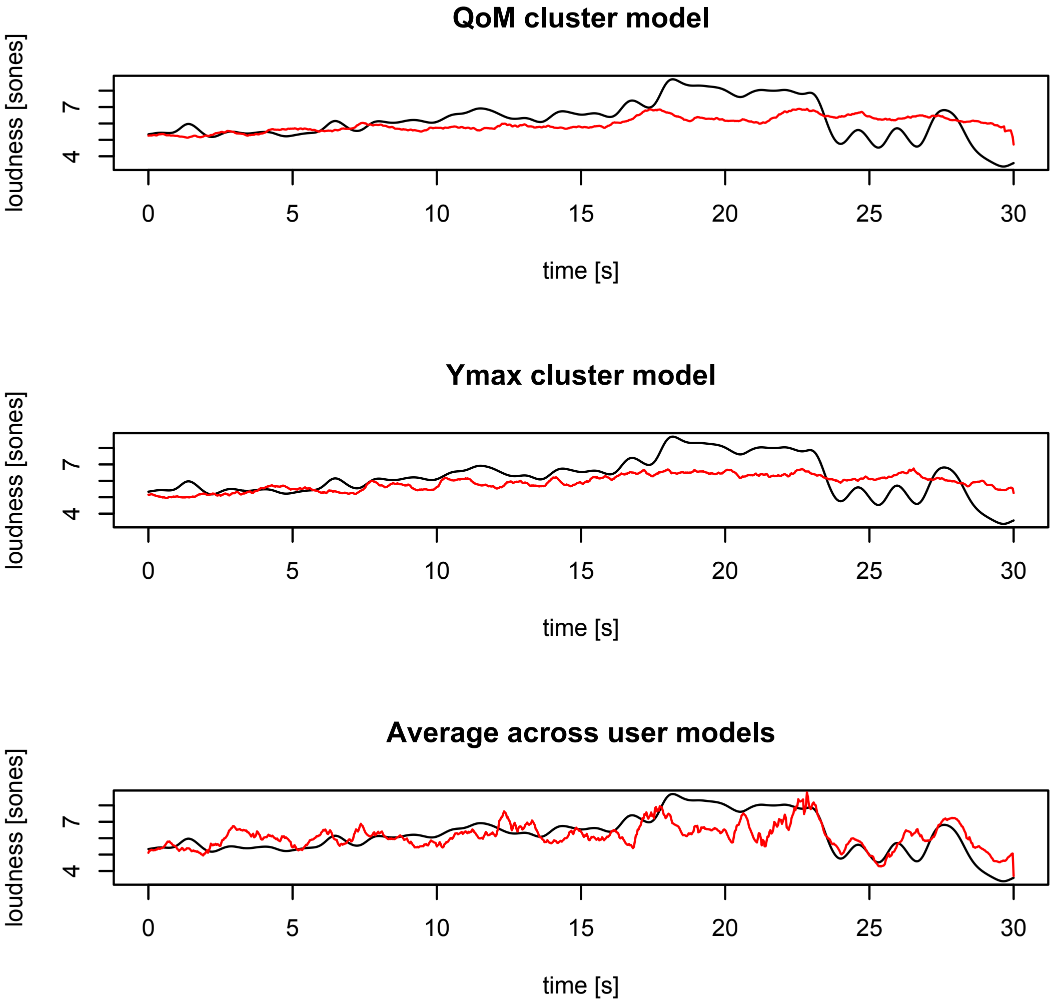
Table 4.1: Statistic from linear regression Models for QoM and Y_{max} clusters. (LH = left hand, RH = right hand). "-" indicates that the descriptor did not appear as significant for the correspondent model. Pearson correlation (P corr) values with absolute value greater than 0.5 are highlighted in bold.

	QoM cluster		Y_{max} cluster	
Variables	P corr	$p$value	P corr	$p$value
(Intercept)		<0.001		<0.001
a^{LH}	0.388	<0.05	0.296	<0.001
a^{RH}	0.431	<0.001	0.293	<0.001
d_{tor}^{LH}	0.142	<0.001	0.293	<0.001
d_{tor}^{RH}	0.183	<0.01	0.348	<0.001
y_{tor}^{LH}	-	-	0.480	<0.001
y_{tor}^{RH}	-	-	0.482	<0.001
v_{mean}^{LH}	**0.519**	< 0.001	0.389	<0.001
v_{mean}^{RH}	**0.585**	< 0.001	0.393	<0.001
v_{dev}^{RH}	**0.552**	< 0.001	-	-
Y_{max}	-	-	**0.593**	<0.001

is not contradictory looking at other descriptors that do appear: v_{mean} values for both hands are correlated to QoM by definition (the latter is calculated as the mean v_{mean} for all joints). In this sense, the fact that we observe no significant effect of QoM on loudness when v_{mean}^{LH} and v_{mean}^{RH} are included in the model, means that QoM does not account for more variability than what is already explained by v_{mean}^{LH} and v_{mean}^{RH}. In terms of LMA, as we explained in Section 2.1, they are all related to the *weight effort* category. The same applies to CI, which is correlated to d_{tor}, being both related to the *shape* LMA component.

The R^2_{adj} statistic computed from the models gives an indication of how much loudness variability is explained by the descriptors. In both cases, R^2_{adj} is greater than 0.4, while it is 0.350 when trying to build a model for all participants. This increment in R^2_{adj} was not due only by the reduced number of participants in each cluster. We performed random splits of participants into reduced groups of 5 participants (the size of the "Y_{max} cluster") and checked that the improvement was smaller when these groups contained participants from both "QoM cluster" and "Y_{max} cluster"; the improvement was maximized with this particular way of splitting participants.

Beyond this, we observed how subjects in the "QoM cluster" did not have the same *dynamic range*, meaning that while all of them performed movements with different amplitudes for soft and loud parts, the amplitude of these movements was not the same


QoM cluster model

Ymax cluster model

Average across user models

Figure 4.2: Different model predictions for the first fragment. Black: loudness extracted from audio, Red: average predicted values.

for all of them. In order to balance this effect, we normalized the values of QoM and v_{mean} compressing or expanding them so all participants had the same dynamic range. This supposes a clear improvement in the "QoM cluster" ($R^2_{adj} = 0.455$ vs $R^2_{adj} = 0.413$ without normalization) and some improvement in the "Y_{max} cluster" ($R^2_{adj} = 0.470$ vs $R^2_{adj} = 0.459$ without normalization).

Creating user-specific models We also created user-specific models for each of the participants. As expected, these models are capable of better predicting the loudness from the movement, with an average $R^2_{adj} = 0.620$ ($\sigma = 0.08$). Nevertheless, although this suggests that the descriptors are able to learn better for specific subjects, the clear improvement in the statistical score of the models is also influenced by the fewer observations from which these models are created. In any case, in the same way that the descriptors we are using can identify different tendencies among participants, it was expected that when a model is created from a single participant it is able to predict the loudness from her movements more accurately.

Figure 4.2 illustrates the average predicted values for the different regression models we have presented in this Section. While the models learned for the different clusters are useful in the sense of observing which descriptors appeared as relevant, the quality of the fit clearly shows that they are not adequate as good predictors. The average behavior of user-specific models is far better in terms of being able to approximate the original loudness curve.

We performed a one-way ANOVA on the quality of the fitted models (R^2_{adj}) to analyze the effect of musical expertise. No significant effects were found. This indicates that for all participants who claimed to have reflected the loudness variations in their movements, the prediction of the loudness from MoCap descriptors was equally good. However, this result must be taken cautiously, since the number of participants for each of the levels for musical expertise is unbalanced.

Table 4.2: Analysis results for all participants. Participants unable to recognize the time signature are marked with †; participants unable to anticipate changes in the last recording are marked with ‡; participants who claimed not to have used rhythmic information in their movement are marked with. $F > F^*$ are highlighted in bold.

Participant	*lag*(s)	σ(s)	F	p	r	F^*
1	-0.016	0.21	**0.54**	0.58	0.51	0.50
2	-0.002	0.27	0.26	0.32	0.22	0.26
3	-0.032	0.16	**0.55**	0.58	0.53	0.40
4	-0.058	0.09	**0.85**	0.86	0.84	0.64
$5^{\ddagger\star}$	-0.057	0.67	**0.43**	0.55	0.38	0.37
$6^{\ddagger\star}$	-0.027	0.18	**0.43**	0.46	0.40	0.42
7	-0.041	0.21	**0.61**	0.68	0.57	0.44
8	-0.051	0.21	**0.58**	0.62	0.56	0.53
9	-0.070	0.14	**0.71**	0.73	0.70	0.39
$10^{\dagger}$	-0.014	0.20	0.45	0.49	0.42	0.47
$11^{\ddagger}$	0.004	0.28	0.40	0.47	0.34	0.39
12	-0.044	0.14	**0.71**	0.74	0.69	0.53
13	0.003	0.14	0.74	0.74	0.73	0.74
14	-0.012	0.19	**0.45**	0.48	0.42	0.39
$15^{\dagger}$	0.015	0.19	0.34	0.38	0.31	0.40
16	-0.016	0.12	**0.63**	0.63	0.62	0.58
17	-0.084	0.20	**0.60**	0.63	0.57	0.37
$18^{\ddagger}$	-0.094	0.43	**0.37**	0.42	0.34	0.36
$19^{\dagger}$	-0.054	0.21	**0.45**	0.47	0.44	0.36
20	-0.041	0.13	**0.61**	0.62	0.59	0.51
21	0.016	0.13	**0.67**	0.68	0.66	0.61
$22^{\dagger}$	-0.020	0.14	**0.57**	0.59	0.56	0.57
$23^{\dagger}$	-0.042	0.09	**0.78**	0.78	0.77	0.64
$24^{\dagger}$	-0.017	0.27	**0.49**	0.54	0.45	0.45
$25^{\dagger\ddagger}$	-0.034	0.19	0.34	0.35	0.33	0.38

Beat analysis

Table 4.2 contains the analysis results for all participants, averaged across the three excerpts. More concretely, the Table shows, for each participant, the estimated *lag* values (and corresponding standard deviation σ) and computed F values. *precision* and *recall*, as well as the value of the F-measure when the correction of the estimated *lag* is not applied (F^*) are also presented for completeness. Problems indicated by participants in the post-recording questionnaire are also highlighted in the table: participants unable

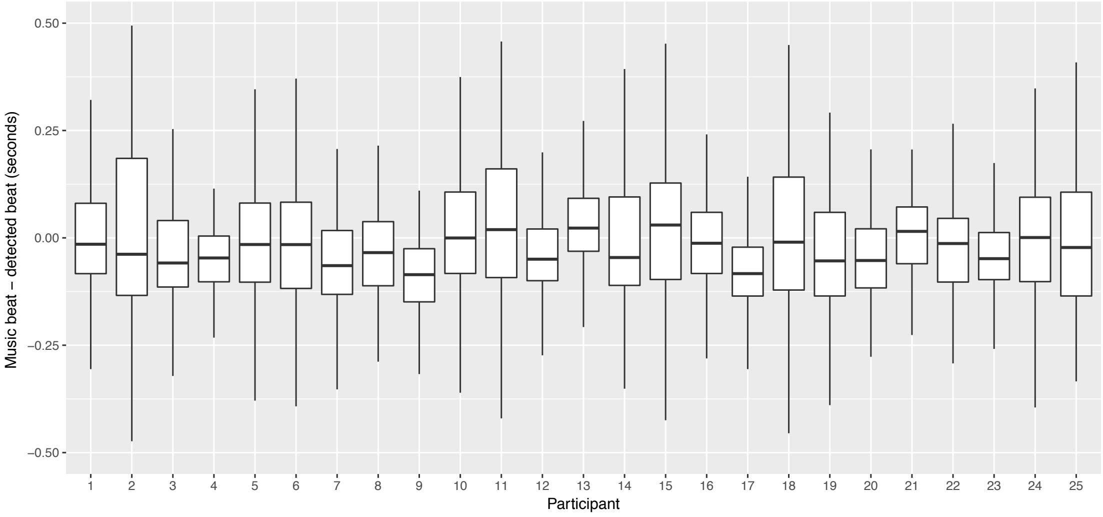


Figure 4.3: Distribution of time differences between beat annotations and beat estimations for each participant.

to recognize the time signature are marked with †; participants unable to anticipate changes in the last recording are marked with ‡; participants who claimed not to have used rhythmic information in their movement are marked with ⋆. The error distribution of each participant is also illustrated in Figure 4.3.

Results show that good beat positions estimations are achieved from hand movement for some participants (e.g. $F = 0.85$ for participant 4, $F = 0.71$ for participant 9 or $F = 0.78$ for participant 23), while in other cases the estimation is quite poor (e.g. $F = 0.26$ for participant 2, $F = 0.34$ for participant 15 or $F = 0.34$ for participant 25). In general, worst cases correspond to participants who indicated some issue in the post-recording questionnaire, with the exception of participants 2 and 14. The average F-measure for participants who did not raise any of these issues was $F = 0.61$ ($\sigma = 13.54$), which suggests that, in general, beat information was indeed reflected in their movement.

Beyond the quality of beat estimation from body movement, which informs us about this idea of beat being reflected in hand movement, we are more interested in different anticipation tendencies of participants reflected in the estimated *lag* values. In most cases, estimated beats tend to appear before annotated beats, as reflected in negative *lag* values. However, we observe big differences across participants. As an illustrative example, *lag* estimated for participant 13 is almost 0 (3 ms), while it is -58 ms for participant 4. High F-measure values of 0.74 and 0.85, respectively, indicate that this estimated *lag* values are indeed indicative of a consistent tendency. In cases like participant 18, with a

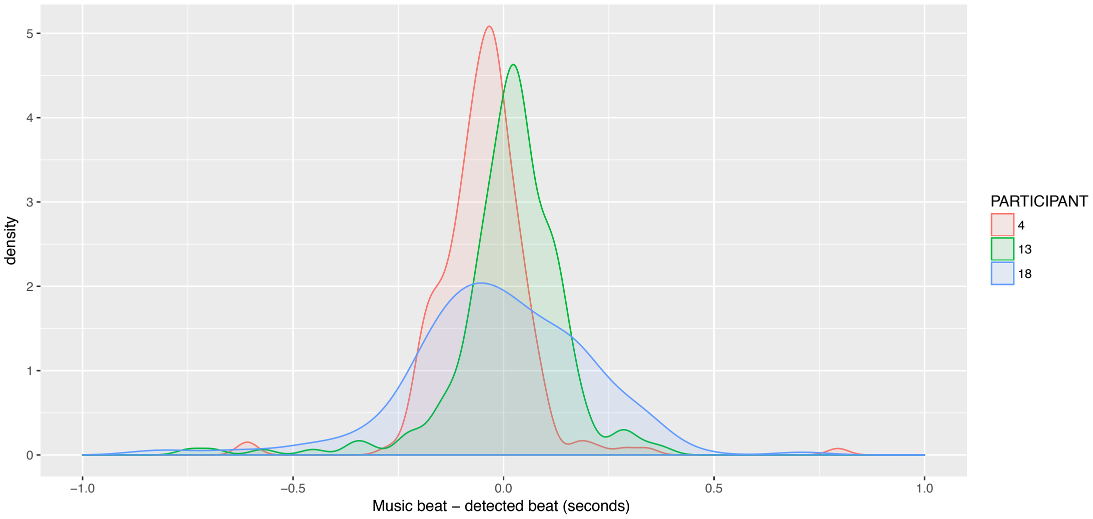


Figure 4.4: Distribution of time differences between beat annotations and beat estimations for participants 4 (red), 13 (green) and 18 (blue).

low $F = 0.37$, the estimated *lag* of -94 ms is less likely to reflect an actual tendency. The error distributions of these three participants is shown in Figure 4.4 as an illustrative example of these different tendencies: participants 4 (red curve) and 13 (green) show narrow distributions centered at different points representing different *lag* values, while participant 18 (blue curve) shows a wide distribution whose mean is less likely to be reflecting an actual tendency.

We performed a one-way ANOVA on the quality of the beat estimation (F) to analyze the effect of musical expertise. No significant effects were found in this case either. Again, this result must be taken cautiously, since the number of participants for each of the levels for musical expertise is unbalanced.

4.2.4 Additional considerations

The study shows some limitations that need to be considered. First, the group of participants was unbalanced in terms of musical training. In this sense, the effect of musical expertise on the observed differences could not be thoroughly analyzed. Also, we took different excerpts from the same piece. Extending the analysis to more pieces with more variations in tempo, dynamics, articulations, instrumentation, etc. might help to better analyze these tendencies in depth.

In terms of beat anticipation, our results are in agreement with those by Lee et al. (2005) in the sense of finding different tendencies. However, in our case we found that these are user-specific, while in their case they found general trends for conductors and non-conductors. However, their study was also different in the sense that they did not look at spontaneous movements, but they asked participants to perform a specific up-down gesture following the beat.

In any case, our purpose in this study was to check whether participants moved differently when asked to "conduct" on top of classical music fragments. We intentionally decided to make this analysis asking participants to move on top of fixed pieces of music being aware that their movements were not changing the performance in any way. While this implicates that the observed conclusions are not directly applicable in an interactive scenario, we were indeed interested in observing intuitive, spontaneous conducting movements performed without instructions. In the broader context of this thesis, the relevant issue following the conclusions of this study is whether the analysis we perform of spontaneous conducting movements can be used to improve the usability of a DMI based on the conductor metaphor that learns its mapping from these movements. This is explored in detail in the following Section.

4.3 Building a user-tailored DMI from spontaneous movements: mapping by observation

Following the conclusions of the study, we propose a system that explicitly exploits the knowledge that users have from the metaphor that inspires the interface (in our case, music conducting). As we saw in Chapter 2, *Mapping through Listening* (Caramiaux et al., 2014a) considers listening as the starting point for mapping design. The mapping is learned from a set of demonstrations where the user explicitly shows the relationship between motion and sound as an acted interaction. In our case, taking advantage of the fact that the instrument is based on a metaphor, we propose to learn the mapping by observing each user making spontaneous conducting movements such as those in the observation study presented above. This is preferable in public installations than to allow each user to explicitly define her own mapping, as far as the learning is simple and fast. In this sense, our approach is also similar to *play-along mapping* as introduced by Fiebrink and Cook (2009), since the user performs gestures while listening to the music which are used to train the system. Again, the difference in our case is that we focus on movements performed without this training *purpose*. For this reason, we refer to this approach as *mapping by observation*.

We explore the idea anticipated in the Introduction and illustrated in Figure 1.4. We propose to have a system where the mapping is roughly predefined, but where some parameters can be adjusted in a specific way for each user. For this parameter tuning, we analyze how the user performs the activity that the metaphor replicates. In our case, the user makes spontaneous conducting movements on top of fixed music. The *predefined* mapping in the proposed system consists on controlling tempo by triggering beats in changes from downward to upward hand movement and controlling loudness with the gesture size. The *adaptation* learned from spontaneous movements consists on incorporating the tendency of the user to anticipate or fall behind the beat, compensating the effect, and analyzing which descriptors, apart from the gesture size, are correlated with loudness.

We evaluate the usability and intuitiveness of the proposed system in a setup where the user does not receive instructions on how the system works and instead just learns by experimenting. For comparison, we consider the system with the *predefined* mapping, which does not adapt to user-specific tendencies, as a baseline. The experiment includes a series of tasks to compare both systems using both subjective feedback and objective measures about the participants' ability to control loudness and the exact time of beats in the resulting music. In addition, we recruited both musicians and non musicians to study possible differences caused by musical expertise.

4.3.1 Proposed system

Here we explain in detail the functioning of the proposed system. As we said, it has a predefined mapping which is tuned specifically for each user. Accordingly, we first explain the system without adaptation, to which we refer as BASELINE (since we compare the proposed approach with it). Then, we continue with the proposed system, to which we refer as TRAINED, highlighting the aspects in which they differ.

Both allow to control loudness and tempo on a musical piece using body movements captured by a MoCap device, in this case a Kinect v2.

BASELINE

Inspired by previous approaches (Haflich and Burnds, 1983; Keane and Gross, 1989; Morita et al., 1989; Borchers et al., 2002; Lee et al., 2004; Bergen, 2012; Toh et al., 2013; Rosa-Pujazon and Barbancho, 2013), the system allows to control **tempo** by triggering beats in changes from downward to upward hand movement. For this, we use the vertical velocity (v_y) of both hands, computed with low-pass differentiators of degree one proposed by Skogstad et al. (2013), as implemented in MoDe[8]. The ictus is detected whenever a change from negative to positive sign in v_y occurs (change from downward to upward movement), as illustrated by the red circles in Figure 4.5. Notes falling between beats are played according to the tempo estimated from the time interval between the last two beats. Two extra rules are applied to avoid false positives in beat detection:

- If the last local minimum before the current change of sign of v_y is not below a threshold v_{th}, the beat is not triggered. This avoids detecting beats from almost-still movement.
- Two consecutive beats must be detected separated by at least a certain number of frames n_{th} from each other. This is done to avoid detecting beats closer in time than musically meaningful, and is particularly necessary to avoid triggering two beats from simultaneous movements from both hands.

Loudness is controlled by means of the gesture *size*, similarly to Morita et al. (1989); Usa and Mochida (1998a); Lee et al. (2004); Toh et al. (2013). When a beat is detected at time t_B, the *size* is computed as the cumulative squared distance traveled by the hand where the beat has been detected since the detection of the previous beat, t_{PB}[9]:

$$size(t_{PB}, t_B) = \sum_{i=PB}^{i=B} (x^k(t_{i+1}) - x^k(t_i))^2 + (y^k(t_{i+1}) - y^k(t_i))^2 + (z^k(t_{i+1}) - z^k(t_i))^2 \quad (4.1)$$

The mapping from *size* to MIDI velocity values is set in preliminary user tests, in order to cover the whole MIDI velocity range. We used MIDI velocity values provided that, as we explain below, we are considering a MIDI sound engine. MIDI velocity values can range from 0 to 127. In the following, we refer to MIDI velocity units (mvu) for loudness values represented in this scale.

TRAINED

The proposed system adapts its mapping individually to each user by performing a previous analysis of spontaneous conducting movements. By “spontaneous” we refer to conducting movements that the user performs on top of a musical excerpt without having received any specific instructions. In this sense, the system needs the user to “conduct” on top of a musical piece for which there is available information on the loudness and location of beats, just as in the case of previous observation studies by Lee et al. (2005)

[8] https://github.com/asarasua/MoDe

[9] We use the squared distance instead of the distance as it requires less computation.

or the one previously presented in this Chapter. More concretely, this system takes into consideration how the user tends to anticipate or fall behind the beat, and which body movement descriptors are best correlated with loudness. For this, we need to store the time position of beats in the music and beats detected from hand movement (using the same method we detailed for the BASELINE system), as well as the value of different body movement descriptors together with the corresponding loudness values at different instants.

The mean difference in seconds between beats in the music and beats detected in hand movement, lag, provides an estimation of the tendency of the user to anticipate or fall behind the beat. Negative values indicate that beats detected in hand movement tend to appear before the beat in the music, while positive values indicate that beats detected in hand movement tend to appear after the music beat. From lag, we compute n_{ant} as the number of frames at the device sampling rate, f_s, (in the case of the Kinect V2, 30 fps) that corresponds to the time closest to lag:

$$n_{ant} = round(lag \cdot f_s) \tag{4.2}$$

Tempo in the TRAINED system is controlled exactly the same way as in the BASELINE system, but including this additional parameter n_{ant}. If $n_{ant} = 0$, there is no difference with respect to the BASELINE. If $n_{ant} < 0$, the beat is triggered $-n_{ant}$ frames after the change of sign in v_y. Figure 4.5 illustrates the method for for $n_{ant} = -2$ (green circles). If $n_{ant} > 0$, beats are no longer detected looking at changes of sign in v_y. Instead, beats are triggered when two consecutive values of v_y are, respectively, smaller and greater than a new threshold $v_{trigger} \neq 0$. $v_{trigger}$ corresponds to the value that v_y takes n_{ant} frames after the last change from positive to negative sign (upward to downward movement). In Figure 4.5, blue circles illustrate the samples where the beat would be triggered in the case of $n_{ant} = 2$, while blue crosses show the samples that determine the different values of $v_{trigger}$.

Loudness is controlled through a linear combination of three different MoCap descriptors:

$$\text{loudness} = \omega_s \cdot size + \omega_Q \cdot QoM + \omega_Y \cdot Y_{max} + \beta \tag{4.3}$$

- Gesture $size$, as defined in Equation (4.1) for BASELINE.
- Quantity of Motion QoM, computed by averaging the mean speed values of all

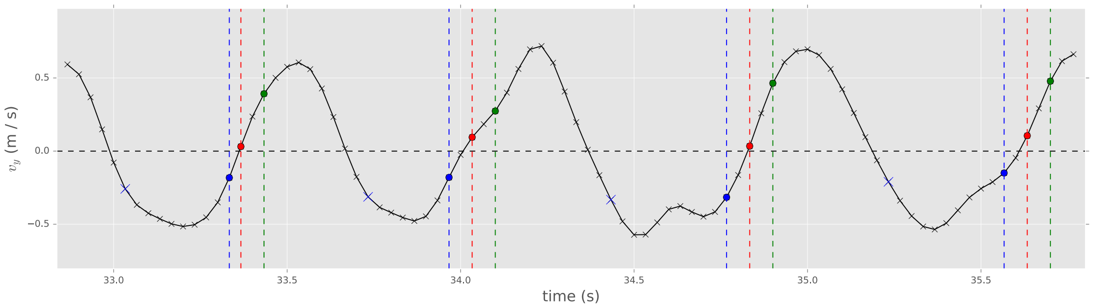


Figure 4.5: Beat triggering from v_y with BASELINE (red circles) and TRAINED systems (green circles, $n_{ant} = -2$; blue circles, $n_{ant} = 2$). Samples highlighted as blue crosses set the $v_{trigger}$ values for $n_{ant} = 2$ in the TRAINED system.

tracked joints $\mathbb{J}$ during N frames as

$$QoM(t_n) = \frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{J}\sum_{j\in\mathbb{J}}\sqrt{v_x^j(t_i)^2 + v_y^j(t_i)^2 + v_z^j(t_i)^2}, \tag{4.4}$$

$v_x^j(t_i)$, $v_y^j(t_i)$, $v_z^j(t_i)$ are the x, y and z components of the velocity of joint j, i frames before n, and J is the number of tracked joints. We use $N = 30$ (1 second at 30 fps).

- Highest hand position Y_{max}, a simple descriptor that in every frame looks at the vertical position y of both hands and takes the maximum value.

The descriptors and loudness values recorded during the execution of spontaneous movements are later used to compute the weights assigned to each descriptor (ω_S for *size*, ω_Q for QoM, ω_Y for Y_{max} and β for the intercept) using least squares linear regression.

4.3.2 Experiment

Materials

We built a dedicated Windows application with OpenFrameworks[10] to be used with a Kinect v2. It uses ofxKinectForWindows2[11] (an OpenFrameworks wrapper for Kinect for Windows SDK) to track skeleton data and MoDe for real-time feature extraction and event triggering.

[10] http://openframeworks.cc/

[11] https://github.com/elliotwoods/ofxKinectForWindows2

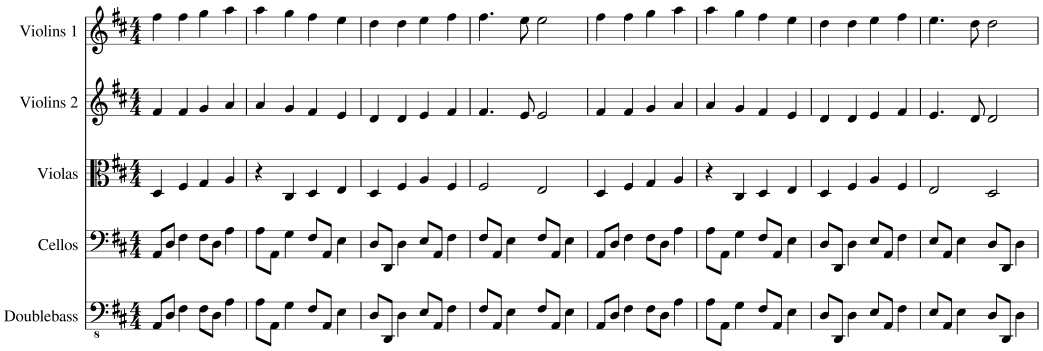


Figure 4.6: Excerpt from Beethoven's 9th symphony used in the experiment.

The application allows to control the experiment procedure using a set of keyboard commands and records all necessary data (training results, tasks results and MoCap data) into text files. It implements the previously presented conducting systems, BASELINE and TRAINED, to conduct a musical piece using movements captured by the Kinect v2. For synthesis, it reads a music score in MusicXML or MIDI format and outputs MIDI events which can be rendered by any external software. For the experiment, we used Native Instrument's Kontakt with Session Strings library and a simplified 8-bar long score for strings from the *Ode to Joy* theme from the fourth movement in Beethoven's 9th Symphony, shown in Figure 4.6. We chose this excerpt for two reasons: first, it is a very popular melody that all participants in the study knew in advance (as they later confirmed); second, the selected melody mostly contains quarter notes. This makes the beat and rhythm of the melody equivalent and avoids possible confusions with participants tending to conduct to the onsets of the predominant melody instead of the beat (this effect was observed by Lee et al. (2005)). The application also provides visual feedback consisting on the mirrored image captured by the Kinect v2 and specific visualizations for each of the phases in the experiment. The content and design of these visualizations is explained with more detail below.

During the experiment, participants used over-ear headphones and stood approximately two meters from a 46-inch TV screen showing the visual feedback from the application. The Kinect v2 sensor was placed below the screen, using a flat speaker stand, approximately 1.4 m from the floor. The experimenter read instructions to the participants and controlled the application from a laptop to which the screen, Kinect v2 sensor and headphones were connected. Another laptop was placed close to the participants for them to provide some demographic information and feedback after each task on a Google Form.

Hypothesis and Experiment Design

As previously indicated, we hypothesize that observed user-specific tendencies in spontaneous conducting movements can be used to build user-specific mappings in a DMI based on the conductor metaphor, improving its usability. Provided that we deal with the concrete case of loudness and beat control, this main hypothesis can be separated in two:

- **H1**: Computational analysis of spontaneous conducting movements can be used to design user-specific mappings between motion and loudness in a DMI based on the conductor metaphor, the resulting system having better usability and being more intuitive than one where the mapping is fixed.
- **H2**: Computational analysis of spontaneous conducting movements can be used to build a DMI based on the conductor metaphor where tempo control considers user-specific tendencies to anticipate or fall behind the beat, the resulting system providing more precise control over beat than a system not considering these tendencies.

To test these hypotheses, we designed an experiment to compare the TRAINED and BASELINE conducting systems. The concrete procedure of the experiment is explained with detail below, but we first enumerate the factors we controlled.

In the experiment, participants use both systems to perform a series of tasks in which we retrieve objective measures of the performance, as well as subjective feedback provided by participants. The most relevant factor we investigate in the experiment is thus the SYSTEM (TRAINED / BASELINE) being used for each of the tasks. All participants use both systems, so the order in which they use them is counterbalanced to compensate the possible effect of learning. Because of this, we also consider the SYSTEM INDEX (first / second) factor.

We retrieve objective measures and subjective feedback related to the control over loudness and beat separately. In addition, we created tasks that challenge participants to only control loudness, beat, and both at the same time. More specifically, participants are presented with the following tasks:

- **Loudness tasks**. The participant is asked to make the orchestra play following a pattern of loudness variation (e.g. “first play *loud*, then *soft*, then *loud*...”).
  - Objective measure: Loudness error, ε_L. At each beat, we define ε_L as the

  difference between the target and achieved loudness levels, both represented in MIDI velocity units (mvu).
  - Subjective feedback: Loudness control rating, r_L. At the end of the task, the participant rates her ability to control loudness in a 5-point scale ranging from 1 = "Could not control loudness at all" to 5 = "Could perfectly control loudness".

- **Metronome tasks**. The participant listens to a metronome at a fixed tempo and has to make the orchestra play in synchrony with it.
  - Objective measure: Beat error, ε_B. Every time a beat is triggered, ε_B corresponds to the difference in seconds with respect to the closest metronome beat.
  - Subjective feedback: Beat control rating, r_B. At the end of the task, the participant rates her ability to control the exact moment in which the instruments sound in a 5-point scale ranging from 1 = "Instruments played much sooner than I intended" to 5 = "Instruments played much later than I intended", with 3 = "Instruments played exactly when I intended".
- **Combined tasks**. A combination of the previous tasks (i.e. the participant listens to a metronome and has to make the orchestra play in synchrony with it while following a loudness variation pattern).

In order to test our hypotheses, we investigate the following effects:

- For **H1**, we expect significantly lower values of $|\varepsilon_L|$ and significantly higher values of r_L for the TRAINED system with respect to the BASELINE system.
- For **H2**, we expect significantly lower values of $|\varepsilon_B|$ and values of r_B (beat control rating) significantly closer to 3 (which corresponds to "Instruments played exactly when I wanted") for the TRAINED system with respect to the BASELINE system. In this case, however, we only expect to observe this effect when the number of frames for anticipation n_{ant} estimated for the user is different to 0. Recall that the BASELINE and TRAINED systems are equivalent for beat control when $n_{ant} = 0$. We explore this with an additional factor ANTICIPATION that codes, for each participant, whether $n_{ant} = 0$ or $n_{ant} \neq 0$.

Tasks with a different TARGET are presented. In the case of loudness tasks, the TARGET corresponds to different loudness levels, coded by the corresponding MIDI velocity. In the case of metronome tasks, the TARGET corresponds to different tempi. We also investigate the influence of the TASK TYPE (simple or combined). Finally, we investigate whether the musical EXPERTISE of participants influences the results.

Participants

Participants were recruited via convenience sampling through department members and their students. They signed an informed consent which contained the approximate duration of the experiment (30 minutes), as well as the type of data being recorded.

Procedure

After signing the consent form, participants were informed about the general setup for the experiment.

Once the participant agrees to start, she fills a form with information about her age and musical expertise. Then, a procedure consisting on three phases is repeated twice, once for each SYSTEM (BASELINE and LEARNED), counterbalancing the order across participants. These phases are (1) *Warm up*: the parameters for the LEARNED system are adjusted and the participant familiarizes with the set up; (2) *Adaptation*: the participant is allowed to explore how the SYSTEM works; (3) *Tasks*: the experimenter asks the participant to perform the tasks introduced above. The concrete procedure was the following:

- **Warm up phase**. In this phase we learn the parameters for the TRAINED system. We only use the information from the first time this phase appears (regardless of the order in which the systems are presented to the participant). We do this because we are interested in learning from "spontaneous movements", and these only occur at the beginning of the experiment. In this sense, the information from the second Warm up phase is not considered, but we still make it to provide the same set up for both systems. There are two steps:
    1. The experimenter informs the participant that she will listen to the musical excerpt used throughout the experiment, preceded by four metronome counts. In this phase, the excerpt (8 bars, 32 beats) is played once at a fixed tempo (90 Beats Per Minute or BPM) and with loudness changing on every bar, following the pattern MID-LOUD-MID-LOUD-MID-SOFT-MID-SOFT. The MIDI velocities corresponding to each of the loudness levels is 60 mvu for "MID", 127 mvu for "LOUD" and 30 mvu for "SOFT". The visualization of the pattern consists on a set of red parallel lines separated proportionally to the loudness. The space between the lines is filled with red color as the music advances. This visualization, for which a snapshot is shown in Figure 4.7, is designed to be self-explanatory and to allow participants to memorize and

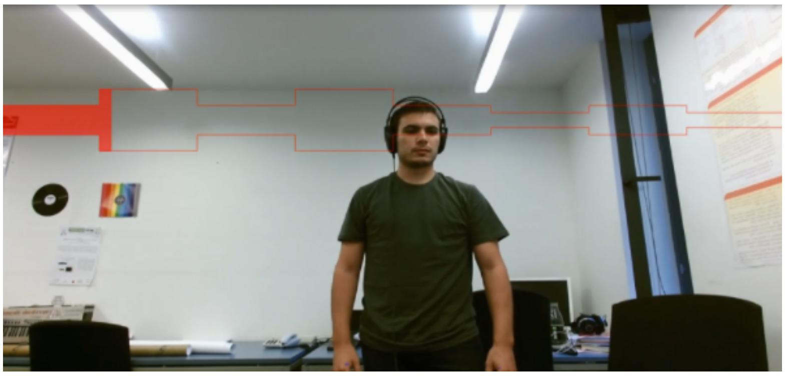


Figure 4.7: Visualization shown during warm up phase.

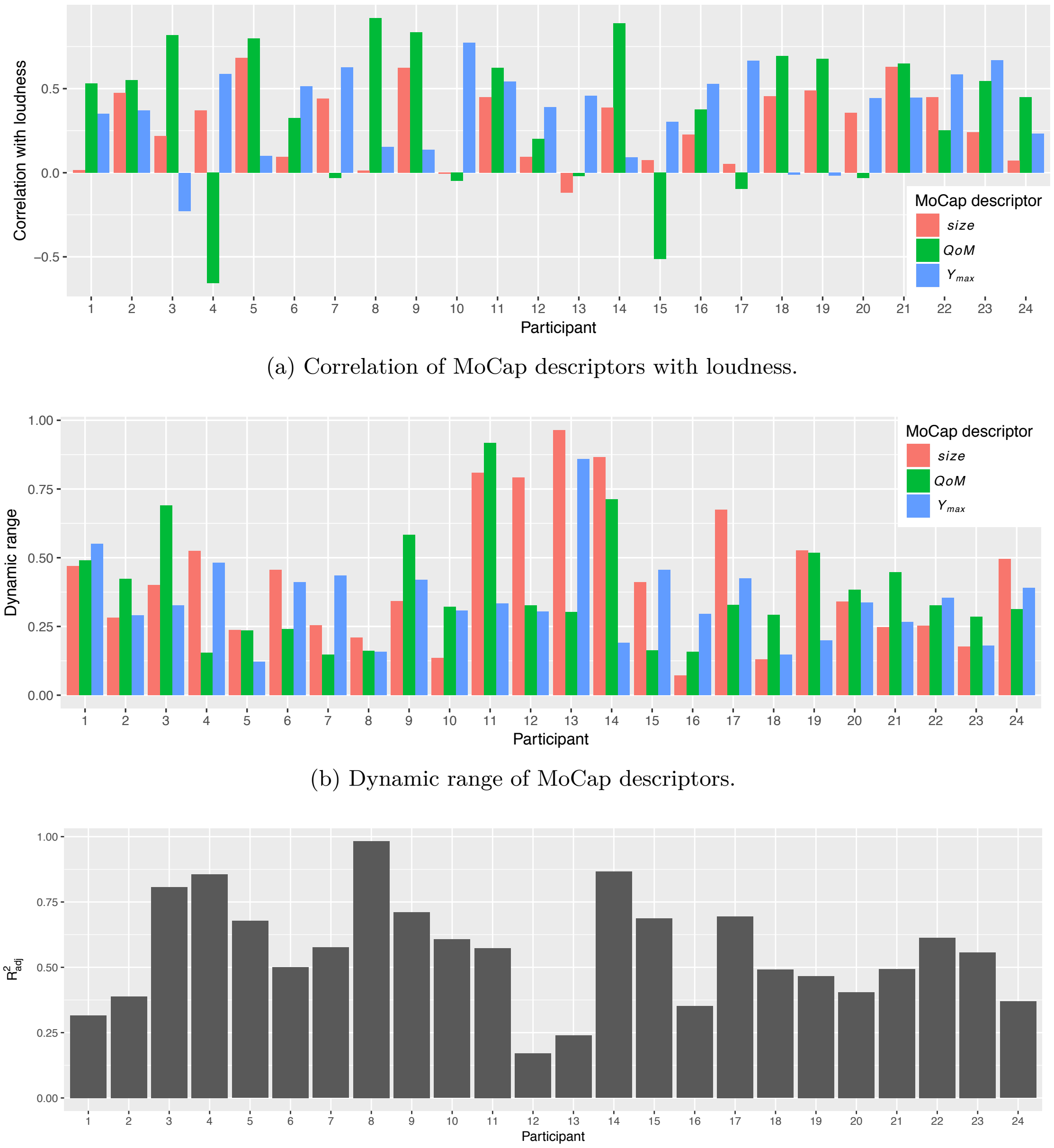
anticipate loudness changes. The excerpt is played as many times as necessary until the participant correctly understands the visualization.

2. The experimenter asks the participant to imagine she has to conduct this excerpt exactly as it sounded, and to perform those conducting movements while listening again to the same excerpt. The fact that no actual conducting is occurring during this phase and the excerpt plays exactly the same way it did before is remarked to avoid confusion. After allowing the participant to rehearse her movements as many times as needed to feel comfortable, the experimenter asks her to perform it again. Here, the application computes the necessary information to compute the parameters for the LEARNED system. More specifically, it stores the exact time at which beats occur in the played excerpt and, for each beat detected in the participant's movement, the exact time at which it is detected, the MIDI velocity at which the music plays, and the MoCap descriptors ($size$, QoM and Y_{max}) values at that time. This information is used to determine the parameters of the TRAINED system as explained in Section 4.3.1.

- **Adaptation Phase**. During this phase, the participant is allowed to experiment with the conducting system. The experimenter does not give any information about possible motion-sound mappings; he only indicates that the system should allow to control tempo and loudness using conducting gestures and that these are not necessarily related to what the participant did in the warm up phase (and, in the case of it being the second tested system, also not necessarily similar to the previous one). A maximum of three trials (each of them consisting of two repetitions of the excerpt) is given to the participant to optimize her control of the performance.

- **Tasks Phase**. Here, the participant performs the tasks introduced above. For all tasks, the participant must conduct the excerpt twice in a row (16 bars, 64 beats). The order in which the loudness and metronome tasks are presented is counterbalanced across participants; the combined tasks always come last. After every task, the participant rates the perceived sense of control over loudness and/or beat. The specifics of the presented tasks are the following:
    - There is one single **Loudness task** where the participant must make the orchestra play with the same loudness variations from the Warm up phase (represented in Figure 4.7) at any tempo. The application shows an equivalent visualization during the task. The red parallel lines now illustrate the target loudness on every bar, and the color fill between the red parallel lines is green and corresponds to the loudness at which the participant is actually making the orchestra sound. Note that in a single loudness task there are three different TARGET levels (LOUD, MID and SOFT). For every loudness task, we have 64 values of ε_L and one r_L rating.
    - There are two **Metronome tasks** at 80 and 100 BPM. In this case, the only visualization is a red progress bar. For each task, we have 64 values of ε_B and one r_B rating.
    - There are also two **Combined tasks**, at 80 and 100 BPM, and with the same pattern of loudness variations and visualization from Loudness tasks. For each task, we have 64 values of ε_L, 64 values of ε_B, one r_L rating and one r_B rating.

After completing these three phases with both systems, the participant is allowed to freely perform with each system. Then, she provides feedback about her preferred one ("first" or "second", as the participant does not know about the difference between both) by answering three questions: "Did you feel any difference between both systems?", "Which one did you prefer in terms of loudness control?" and "Which one did you prefer in terms of your ability to make instruments sound exactly when intended?".



(a) Correlation of MoCap descriptors with loudness.

(b) Dynamic range of MoCap descriptors.

(c) Coefficient of determination (R^2_{adj}) of linear regression models.

Figure 4.8: Correlation with loudness, dynamic range of MoCap descriptors, and coefficient of determination for each participant computed from first Warm up phase.

Results

Twenty four people (6 female) participated in the experiment. Their average age was 27.79 years ($\sigma = 5.84$), with ages ranging from 19 to 41. Half of them were musicians (considering musicians participants with any musical training) and the other half were non-musicians. The experiment was carried out during four different days, taking approximately 35 minutes for each participant.

We first analyze the results from the first Warm up phase, where the parameters of the TRAINED system are learned. In this phase, participants performed spontaneous conducting movements on top of fixed music.

First, we focus on the results that determine the loudness control. Figure 4.8a shows, for each participant, the correlations found between each of the three MoCap descriptors ($size$, QoM and Y_{max}) and loudness (MIDI velocity). In most cases, as expected, MoCap descriptors show a positive correlation with loudness. There are a few exceptions where negative correlations appear, with only two cases where the absolute value of these correlations are greater than 0.5 (QoM for participants 4 and 15). In most cases (70%), QoM is the most correlated descriptor, with an average absolute correlation of 0.48, followed by Y_{max} (0.38) and $size$ (0.29). Correlation is not the only factor influencing the computed linear models. Figure 4.8b shows, for each participant, the *dynamic range* of the three MoCap descriptors. For consistent visualization across descriptors, the dynamic range for a participant and descriptor is computed by dividing the difference between the maximum and minimum descriptor values for *that* participant by the difference between the maximum and minimum descriptor values across *all* participants. As an illustrative example, participants 11 and 21 show a similar positive correlation between QoM and loudness, but the former has a larger dynamic range. This positive correlation indicates that the mapping for loudness control with the TRAINED system would assign louder output for more energetic movements for both participants. The different dynamic ranges indicate that the difference in QoM of movements resulting in soft and loud output would be larger for participant 11 than for participant 21. From the computed linear regression models, we compute the adjusted coefficient of determination R^2_{adj} as indicative of how much loudness variability is explained by the MoCap descriptors. Computed values for each participant are depicted in Figure 4.8c. We use these values below to check whether results during the tasks are affected by the quality of the learned models.

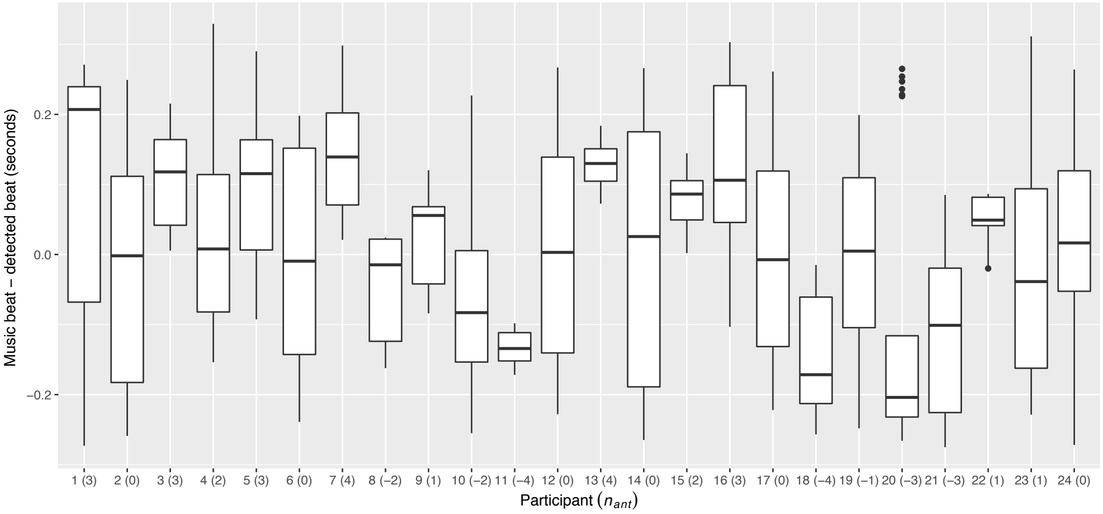


Figure 4.9: Distribution of differences between beats in music and beats detected in hand movement during the first Warm up phase for each participant. The resulting estimated number of frames for anticipation (n_{ant}) for each participant is indicated between parenthesis.

Regarding beat control, Figure 4.9 shows, for each participant, the distribution of the differences in seconds between beats in the music and beats detected from hand movement. In the figure, we also indicate the number of frames for anticipation n_{ant} estimated from the mean of this distribution for each participant. There were 6 participants (2, 6, 12, 14, 17 and 24) for whom $n_{ant} = 0$, i.e. BASELINE and TRAINED systems were equivalent in terms of beat control. n_{ant} values range -4 to 4. For participants with $n_{ant} = 4$, beats are triggered 9 frames (300 ms) before than for participants with $n_{ant} = -4$. From these distributions, we also computed F-measure values for each participant following the same method from preceding parts of this thesis. In this case, this measure is an indication of how consistent is the anticipation effect that the TRAINED system uses for user-specific adaptation. We use these F-measure values below to check whether this affects the results.

Both for loudness and beat control, the results indicate that the TRAINED system was quite different across participants. In the following, we analyze the results from the Tasks Phase.

Loudness control Regarding the objective measures taken from the tasks, Figure 4.10 shows the distributions of absolute loudness error ($|\varepsilon_L|$) for each participant across all loudness and combined tasks, with 64 values of ε_L per task (one per beat). In most

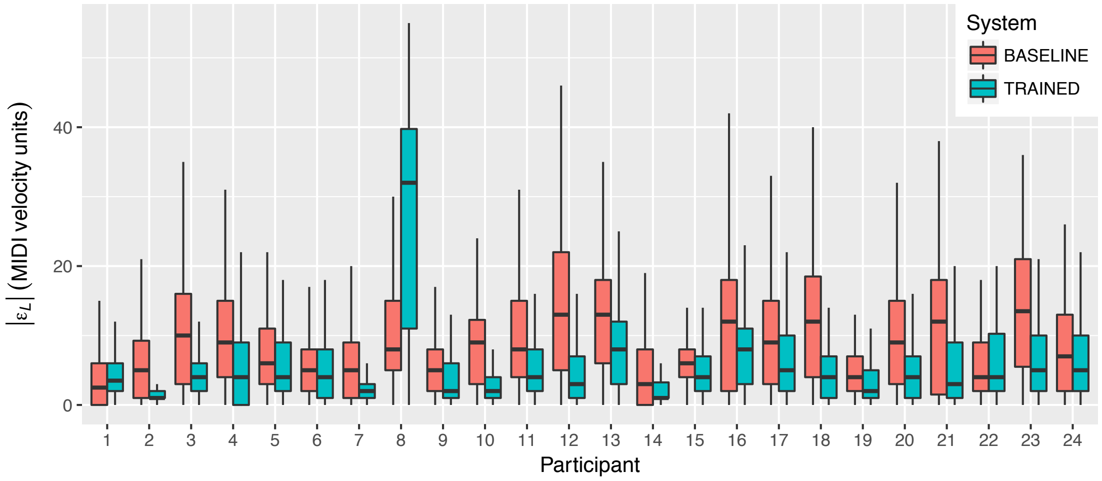


Figure 4.10: Absolute loudness error ($|\varepsilon_L|$) for both systems, averaged across tasks, for each participant.

cases, we observe the expected tendency of lower ε_L values with the TRAINED system. Participant 8, however, shows clearly worse results with the TRAINED system than in any other case. Coming back to the results from training (Figure 4.8), we see that actually this participant showed a very strong correlation between QoM and loudness. Also, the R^2_{adj} metric of the fitted regression model is 0.98, which is very close to ideal in terms of the loudness variability explained by the MoCap descriptors. The low dynamic range suggests that what may have happened is that the observed correlation is spurious; the participant performed with very little variations in QoM that just happened to be very correlated with loudness, resulting in a model whose functioning the participant was not able to learn. Given that this is an outlier case, we removed this participant for the overall statistical analysis presented below.

We fitted a linear model to System, System Index, Expertise, Task Type, Target and their two-factor interactions, and ran an ANOVA to study their effect on the absolute value of ε_L.

A strong effect was observed for System, $F_{(1,8784)} = 644.11, p < 0.001$. As expected, the absolute value of the loudness error was significantly lower using the TRAINED system than using the BASELINE, the average error being of 5.70 mvu for the former and 10.14 mvu for the latter. System Index does not cause any main effect, nor does its interaction with System, indicating that the observed effect of System does not depend on the order in which the systems were presented to the participants.

Results reveal that the performance varies depending on the Target, $F_{(2,8784)} = 563.8, p <$

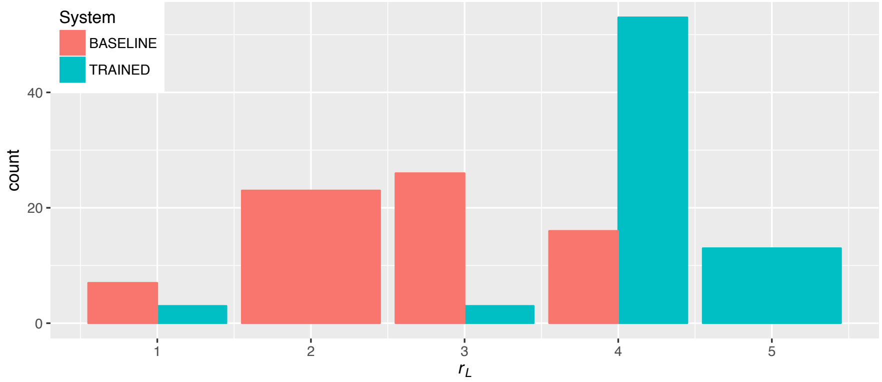


Figure 4.11: Ratings for loudness control (r_L) provided by participants at the end of loudness and combined tasks.

0.001. Absolute error is higher for parts where the target was to play "LOUD". However, this effect is mostly caused by tasks performed using the BASELINE system. The interaction between System and Target also has a significant effect on the absolute error, $F_{(2,8784)} = 394.80, p < 0.001$. The errors were similar in the case of the TRAINED system (4.98 mvu for "SOFT", 5.74 mvu for "MID" and 6.37 mvu for "LOUD"), but participants had more difficulties to achieve louder levels using the BASELINE system, with 3.49 mvu for "SOFT", 9.27 mvu for "MID" and 18.53 mvu for "LOUD". This suggests that the better performance of the TRAINED system is due to its ability to provide accurate control over the whole range of loudness levels. The BASELINE system, where the gesture *size* is mapped to loudness, was problematic for loudest levels.

The effect of Expertise also shows that musicians achieve significantly better control over loudness than non-musicians, $F_{(1,8784)} = 394.54, p < 0.001$. This difference is however significantly reduced when using the TRAINED system. The difference between musicians and non musicians using the BASELINE was 4.30 mvu, while it was 2.64 mvu using the TRAINED system. This suggests that even though both groups achieved better performance with the TRAINED system, musicians were more able to learn the functioning of the BASELINE and adapt in order to complete the tasks.

Finally, no effect is observed for the Task Type, but its interaction with Expertise indicates that musicians performed slightly better in combined tasks, while the opposite happened for non musicians, $F_{(1,8784)} = 12.08, p < 0.001$. Recall that combined tasks always come after simple ones. In this sense, the improvement in combined tasks for musicians can be due to learning. In the case of non musicians, the effect might be explained by the higher complexity of combined tasks.

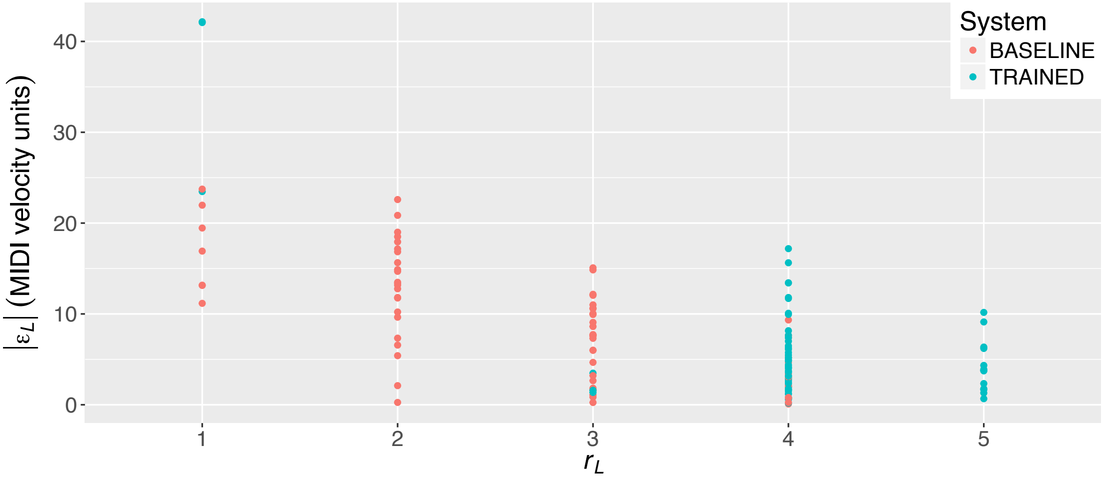


Figure 4.12: Control over loudness rating (r_L) and absolute loudness error ($|\varepsilon_L|$) of all loudness and combined tasks.

Regarding the subjective feedback provided by participants at the end of each task, Figure 4.11 shows the distribution of ratings in a 5-point scale ranging from 1 = "Could not control loudness at all" to 5 = "Could control loudness perfectly". With the TRAINED system, participants rated their ability to control loudness with 4 in most cases, followed by 5. With the BASELINE, ratings were in most cases evenly distributed between 2 and 4. This suggests that participants felt they had better control over loudness when using the TRAINED system.

Again, we fitted a linear model to SYSTEM, SYSTEM INDEX, EXPERTISE, TASK TYPE and their two-factor interactions, this time running an ANOVA to study their effect on r_L. Note that here we do not investigate TARGET, provided that the three targets appear in all tasks and we obtained one rating per task.

Results confirm that the reported sense of control over loudness is better using the TRAINED system, with an average rating of 4.14, than using the BASELINE, with 2.74, $F_{(1,128)} = 91.39, p < 0.001$. The analysis revealed no other significant effects.

We also examine the correlation between the subjective feedback provided by participants and the objective measures reflected in the values of ε_L. We expect a negative correlation (lower error for higher ratings). In Figure 4.12, every point corresponds to the average absolute value of ε_L and the rating provided by the participant for a task, with the color indicating the SYSTEM being used. The correlation between r_L and $|\varepsilon_L|$ is -0.66. This indicates that, as expected, participants were able to achieve a better performance in the tasks when they had a better sense of control over loudness. One-way ANOVA

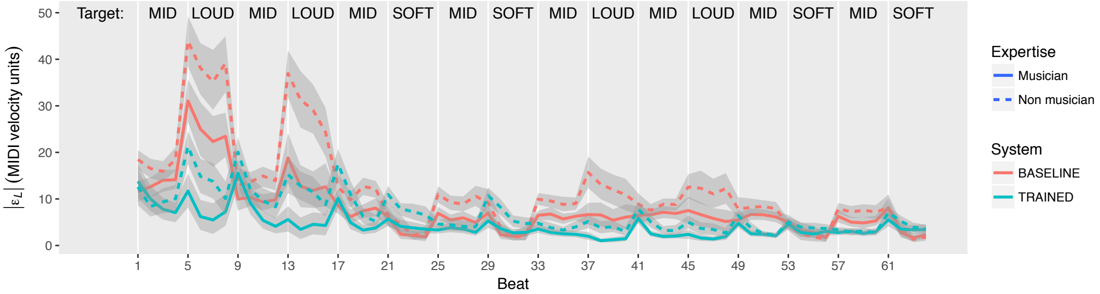


Figure 4.13: Average evolution of absolute loudness error ($|\varepsilon_L|$) for different combinations of System and Expertise.

shows that the difference of absolute values of ε_L for different ratings is significant, $F_{(4,139)} = 37.32, p < 0.001$.

We also investigate whether the *quality* of the linear models computed to adjust the mapping of the TRAINED system for each participant influences the results. For this, we take the R^2_{adj} statistic of each participant's model, which gives a measure of how much loudness variability is explained by the MoCap descriptors. We then compute Δ_{ε_L} for each participant as the difference between average $|\varepsilon_L|$ values for the BASELINE and TRAINED systems. Accordingly, Δ_{ε_L} measures how much improvement the TRAINED system introduces in comparison with the BASELINE. We have thus one R^2_{adj} and Δ_{ε_L} value for each participant. The correlation between both variables across participants is 0.49. This positive correlation indicates that, as expected, better models result in higher improvement introduced by the proposed system.

Another interesting aspect we investigated is the learning effect that occurs during the realization of each task. Figure 4.13 shows the evolution of the absolute loudness error along the 64 beats each task lasted, averaged across all participants. A different curve is shown for each combination of System and Expertise. One of the visible effects in the graph is that the error is in general higher for every first beat with a new target. In the curves, this is reflected by the peaks appearing every 4 beats.

It is also clearly visible that the afore-mentioned effect of the Target using the BASELINE system is particularly higher in the first two appearances of the "LOUD" target (beats 5-8 and 13-16). This is most likely caused by the fact that these are the first loudness changes that participants had to perform. Having observed this effect, we repeat the ANOVA by only using the information from the second half of every task (i.e. from beat 33), to check that the observed effects are consistent along the task. Indeed, the largest effect is the one caused by the System used in the task, $F_{(1,4404)} = 588.22, p < 0.001$.

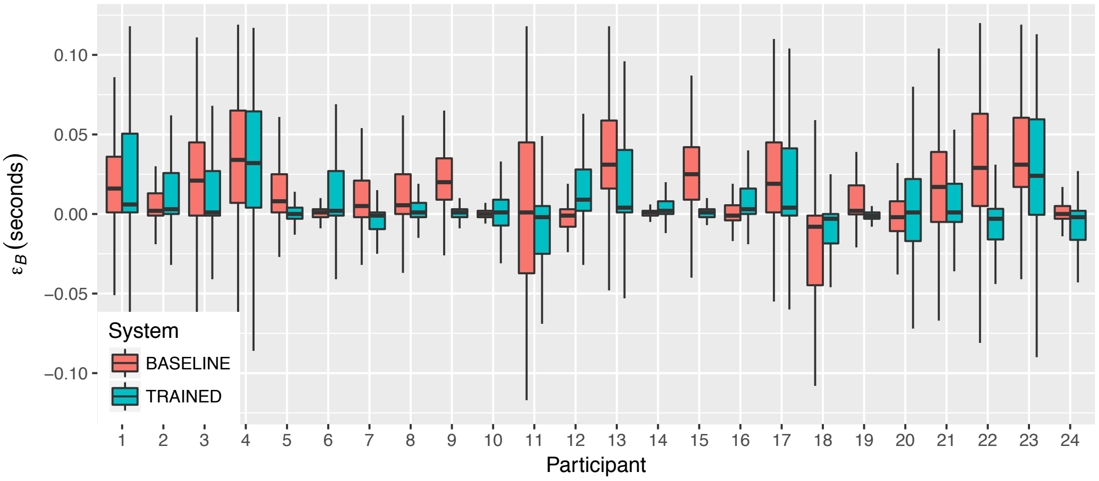


Figure 4.14: Beat error for both systems, averaged across tasks, for each participant.

The absolute loudness error is still significantly lower with the TRAINED system (3.43 mvu) than with the BASELINE (6.84 mvu). The effect of musicians performing better than non musicians is also preserved, $F_{(1,4404)} = 185.98, p < 0.001$. The effect of the TARGET and its interaction with the SYSTEM also appears when looking at the second half of the tasks, but much more mitigated than when considering the whole task duration.

Beat control We now focus on beat control, by analyzing metronome and combined tasks. Regarding the objective performance measures in these tasks, Figure 4.14 shows the distributions of beat errors (distance in time between metronome and performed beats) for each participant across all metronome and combined tasks, with 64 values of ε_B per task (one per beat). As for the case of loudness, we observe that the general tendency is to find these distributions closer to 0 when the TRAINED system is used.

In the case of beat control, both systems work equivalently if the estimated number of frames for anticipation $n_{ant} = 0$. For this reason, the analysis has one more factor than in the case of loudness control: ANTICIPATION. This factor has just two levels ($n_{ant} = 0$ -no difference expected between systems- and $n_{ant} \neq 0$). We fitted a linear model to SYSTEM, SYSTEM INDEX, EXPERTISE, TASK TYPE, TARGET, ANTICIPATION and their two-factor interactions, and ran an ANOVA to study their effect on the absolute value of ε_B.

A strong effect is in fact caused by ANTICIPATION, $F_{(1,12232)} = 100.06, p < 0.001$. The absolute beat error for participants with $n_{ant} \neq 0$ (n=18) is 0.009 seconds higher than

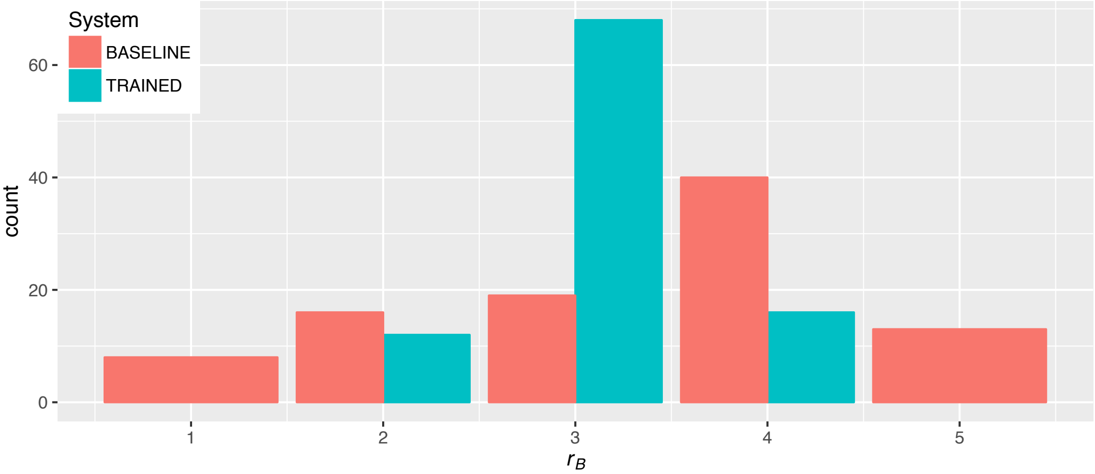


Figure 4.15: Ratings for beat control provided by participants at the end of metronome and combined tasks.

for participants with $n_{ant} = 0$ (n=6). The underlying effect is better explained by the interaction between ANTICIPATION and SYSTEM ($F_{(1,12232)} = 50.84, p < 0.001$). In the case of participants with $n_{ant} \neq 0$, the absolute beat error is 0.013 seconds smaller using the TRAINED system. For the 6 participants for whom $n_{ant} = 0$, the error is slightly smaller (0.003 seconds) using the BASELINE system. These results indicate that the compensation introduced by the TRAINED system is indeed useful to improve the performance of participants who tended to anticipate or fall behind the beat during the Warm up phase ($n_{ant} \neq 0$), i.e. when they performed spontaneous conducting movements.

The results also indicate that the error differed depending on the musical EXPERTISE. Musicians show 0.009 seconds less absolute error than non musicians, $F_{(1,12232)} = 160.09, p < 0.001$. Interestingly, the TARGET also affects the absolute beat error, $F_{(1,12232)} = 96.69, p < 0.001$. However, this only occurs for participants with $n_{ant} \neq 0$ using the BASELINE system. This indicates that the correction that the TRAINED system applies is particularly necessary for slower tempi. Indeed, focusing on the 18 participants with $n_{ant} \neq 0$, the TRAINED system outperforms the BASELINE by reducing the absolute beat error in 0.007 seconds for 100 BPM tasks and 0.019 seconds in 80 BPM tasks.

Figure 4.15 shows the results of subjective ratings of beat control, where participants rated in a 5-point scale ranging with 1 = “Instruments played much sooner than I intended”, 3 = “Instruments played exactly when I intended”, 5 = “Instruments played much later than I intended". In this case, the best rating is thus 3 ("exactly when intended"). The Figure shows the tendency of participants to give a better rating when using the TRAINED system.

In order to statistically analyze the effect of the different factors on the ratings, we perform the following analysis. We define $r^*_B = 3 - |r_B - 3|$, which ranges from 1 to 3, being 1 = "Instruments played much sooner/later than I intended", 2 = "Instrument played a bit sooner/later than I intended" and 3 = "Instruments played exactly when I intended". r^*_B, then, gives a measure of how good or bad the participant felt the system was in providing accurate control of beats, independently of whether a possible bad behavior was caused by instruments playing sooner or later than intended.

Again, we fitted a linear model on the factors of the analysis and performed an ANOVA to study their effect on r^*_B. Participants rated their ability to make instruments play when intended with an average 2.71 for the TRAINED system and 1.97 for the BASELINE, being this difference significant, $F_{(1,136)} = 743.5, p < 0.001$. As expected, the perceived difference was bigger for participants with $n_{ant} \neq 0$. They rated the BASELINE with an average 1.72 and the TRAINED system with 2.72. The 6 participants for whom both systems were equivalent gave slightly better rating to the BASELINE (2.75 vs 2.67 for TRAINED). The interaction between TARGET and SYSTEM ($F_{(1,136)} = 11.84, p < 0.001$) shows that the reported sense of control was significantly worse for 80 BPM tasks using the BASELINE. As we saw before, this is the case where the highest values for absolute beat error appeared. This suggests that the ability to correctly perform the task (to make the orchestra play in synchrony with the metronome) influenced the perceived ability to make instruments play when intended.

As in the case of loudness, we examined the correlation between the subjective and objective measures. In this case, we expect a positive correlation, with negative values of ε_B for low ratings, positive values of ε_B for high ratings, and ε_B values close to 0 for $r_B = 3$. Every point in Figure 4.16 corresponds to the average value of ε_B and the rating provided by the participant for a task, with the color indicating the SYSTEM being used. The correlation in this case is weaker (0.48), but still in the expected direction. This indicates that those participants who felt that instruments came too early with respect to their gesture tended to make the orchestra play in anticipation to the metronome, while those who felt that instruments came too late tended to make the orchestra beats fall behind the metronome. One-way ANOVA shows that the difference of values of ε_B for different ratings is significant, $F_{(4,187)} = 9.912, p < 0.001$.

As we indicated earlier, we computed F-measure values from the training data as indica-

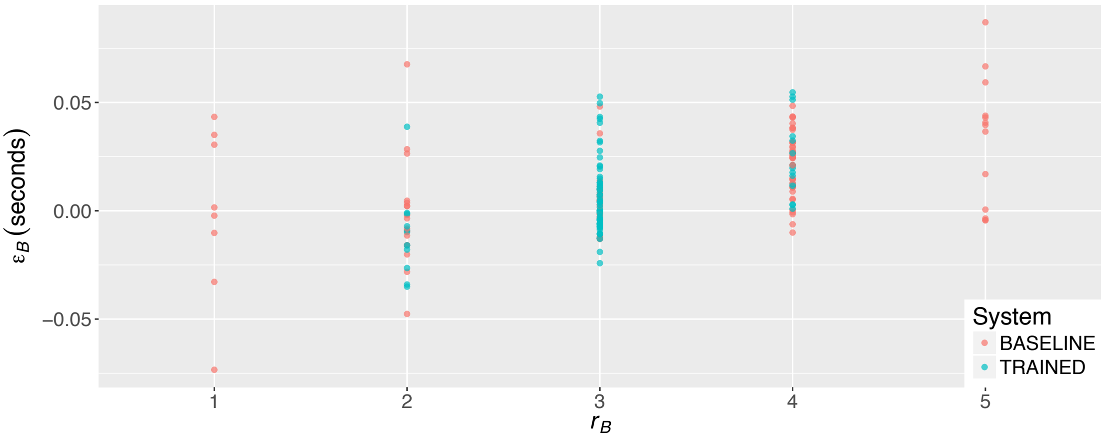


Figure 4.16: Control over beat rating (r_B) and beat error (ε_B) of all metronome and combined tasks.

tive of the consistency of participants to anticipate or fall behind the beat during the Warm up phase. In order to test whether this had an effect on the results, we compute Δ_{ε_B} for each participant as the difference between average $|\varepsilon_B|$ values for the BASELINE and TRAINED systems, i.e. Δ_{ε_B} measures how much improvement there is using the TRAINED system in comparison with the BASELINE. Then, we compute the correlation between Δ_{ε_B} and F-measure values across participants, obtaining a high value of 0.81. This indicates that after the Warm up phase, just by looking at the data used for adapting the TRAINED system, we can guess whether the adaptation will introduce an improvement or not. To put it another way: if time differences between beats in the music and beats detected from hand movement are not consistent in the warm up phase, then the adaptation introduced by the TRAINED system does not guarantee an improvement.

Finally, we explore the possible learning and adaptation effects during tasks. Figure 4.17 shows the evolution of the absolute beat error along the 64 beats each task lasted, averaged across all participants. A different curve is shown for each combination of System and Expertise. We observe a more stable tendency than in the case of loudness control. The error is higher during the first bars, where participants seem to adapt to make the orchestra synchronize with the metronome. The error looks much more stable in the second half (from beat 33), so we also ran the ANOVA again to check if the observed effects also appear in the moment where participants seem to have adapted.

The results indicate that there is still a difference of 0.003 seconds between musicians and non musicians, $F_{(1,6140)} = 35.98$. This difference is however smaller than when con-

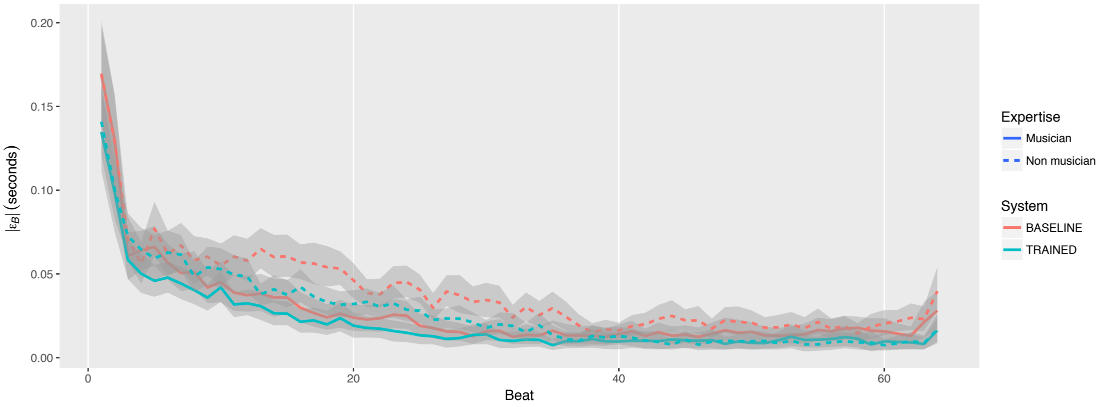


Figure 4.17: Average evolution of absolute beat error ($|\varepsilon_B|$) for different combinations of SYSTEM and EXPERTISE.

sidering the full task (0.009 seconds), which indicates that part of the better performance of musicians is due to their ability to adapt faster. A greater difference is still observed for the SYSTEM: the performance is still notably better (0.008 seconds improvement) with the TRAINED system than with the BASELINE, $F_{(1,6140)} = 203.42, p < 0.001$.

Overall evaluation As we indicated, participants were able to freely perform with both systems again at the end of the experiment, after which they were asked whether they have found differences between both systems and whether they preferred any of them in terms of loudness and beat control.

All participants indicated that they had indeed noticed differences between both systems. Regarding loudness control, all participants preferred the TRAINED system, except for participants 22 and 8 (the outlier), who preferred the BASELINE. Regarding beat control, three participants (2, 6 and 14) indicated that they did not have any preference between both systems, and one (12) showed preference for the BASELINE system. All these four participants were amongst those with $n_{ant} = 0$ (i.e. both systems were equivalent in terms of beat control). The rest of the participants showed preference for the TRAINED system.

4.3.3 Discussion

In this section, we proposed a DMI based on the conductor metaphor that allows to control tempo and dynamics and adapts its mapping specifically for each user by observing spontaneous conducting movements. We refer to this as *mapping by observation* given that, even though the system is trained specifically for each user, this training is not done explicitly and consciously by the user. More specifically, the system adapts its mapping based on the tendency of the user to anticipate or fall behind the beat and observing the MoCap descriptors that best correlate to loudness during spontaneous conducting.

For evaluation, we compared the proposed approach with a baseline that does not perform this user-specific adaptations. We compared both systems in a context where the user does not receive instructions and, instead, is allowed to discover by playing. We consider this an interesting use case, particularly for public installations. We designed tasks where participants had to make the orchestra play at different loudness levels or in synchrony with a metronome in order to objectively evaluate the usability of both systems. We also asked participants to report their sensations using both systems and to compare them.

Results suggest that both hypothesis are confirmed: the usability of the proposed system, which adapts its mapping using the analysis of spontaneous conducting movements, is better both in terms of providing a more intuitive control over loudness (**H1**) and a more precise control over beat timing (**H2**). In both cases, results of objective evaluation and subjective feedback provided by participants are coherent.

We believe that the proof of these hypotheses is particularly relevant considering that parameters were learned from spontaneous movements, i.e.: participants were not making a conscious training of their personalized systems when the parameters for control were learned. This is important for public installations where, if the interaction designers want to take advantage of user customization, it is preferable to make it in a way that is transparent to the user. Beyond the concrete scope of systems for music conducting, this is relevant for other interaction design scenarios using metaphors: the knowledge of the user from the original activity can be explicitly exploited in the system. For example, user-specific tendencies are also observed in “air instrument” performance (Godøy et al., 2006) and sound-tracing (Glette et al., 2010), which suggests that DMIs based on these activities could benefit from adapting their mapping for each user, analyzing how the original activity is performed.

Precisely because the focus of this work was to test whether the information observed in spontaneous movements is useful to be applied during interaction, the parameters under control and the systems under comparison were kept simple. The learned parameters are applied, in the end, to modify the rules of the system used as baseline (by using appropriate descriptors and weights to control loudness and by compensating for the observed anticipation for beat). However, as we pointed out in the introduction, previous conducting systems have used more sophisticated techniques to deal with temporal information from the gesture (Usa and Mochida, 1998a; Kolesnik, 2004; Brecht and Garnett, 1995; Ilmonen and Takala, 1999). We believe that the conclusions from this experiment are not restricted to the case of simple rule-based systems, nor to just the control of beat and loudness. Particularly suitable for more sophisticated and complex gesture-sound mappings to be learned from few observations would be real-time gesture recognition and following systems such as the *Gesture Follower* by Bevilacqua et al. (2010), dynamical models that adapt dynamically to variations such as de Gesture Variation Follower by Caramiaux et al. (2014b) or probabilistic models that learn spatio-temporal variations from gesture (Françoise et al., 2014).

In the field of NIME, it is often hard to establish a criterion for evaluating the quality or usability of musical interfaces. In the concrete case of systems using the conductor metaphor, evaluations, when provided, are most of the times based on subjective feedback provided by participants (Lee et al., 2004; Bergen, 2012; Rosa-Pujazon and Barbancho, 2013) or are focused on evaluating technical aspects specific to the method being used (Brecht and Garnett, 1995; Toh et al., 2013). The warm up and learning phases of the procedure we followed in our experiment are specific to the scenario where the user receives no instructions and observation from her spontaneous movements is required. However, we believe that the kind of tasks we used are suitable to other cases where it is necessary to objectively assess the suitability of a musical interface to control some specific parameters.

In our experiment we also were interested in the effect of musical expertise in the interaction. We observed that, in general, musicians achieved better performance than non musicians. However, focusing on loudness control, this difference was reduced with the proposed system. This suggests that musicians were better at learning how to conduct with the BASELINE system, while non musicians probably tried to stick to their intuitions and were less able to learn by playing. Provided that this effect (greater improvement for non musicians) was not observed for beat control, this might also indicate that non musicians, when using the BASELINE system, tended to focus more on beat control and "forget" about loudness. In accordance with this idea, non musicians got worse results in combined tasks than in simple ones for loudness control, while the opposite happened for beat control. Results also seem to reveal that this difficulty of non musicians to control loudness with the BASELINE system was particularly noticeable in louder parts. This might indicate that they were unable to discover that loudness was controlled with the size of the gesture or that they were probably unable to perform big enough gestures at a given tempo.

In the analysis of loudness control, we removed participant 8, whose results were causing spurious effects for a number of factors and interactions. However, the case of this participant must be carefully considered, as it shows the problems that can be encountered when applying knowledge extracted from analyzing spontaneous movements on top of fixed music. Even though the results from the warm up phase, where the parameters for the TRAINED system are learned, indicated that the learned model could be expected to be good, it was clearly not intuitive for this participant to control loudness. In most cases, mappings learned by the TRAINED system were more intuitive, but the possibility of learning wrong clues is present and should be considered.

Latencies around 20-30 ms are commonly considered acceptable for most musical applications (Lago and Kon, 2004). The Kinect v2 has a ∼20 ms latency (Sell and O'Connor, 2014), and the computation of velocity from raw positional data using low-pass differentiators introduces two samples of delay (Skogstad et al., 2013). This means that this latency is implicit in observed differences in anticipation to the beat. In this sense, the observed improvement introduced by compensating for different tendencies to anticipate or fall behind the beat is also compensating for the device and computation latencies.

Having this consideration in mind, we can further explore the results for beat control. BASELINE and TRAINED systems were equivalent for beat control for participants for whom the estimated anticipation was $n_{ant} = 0$. Results show how a strong difference in the performance for beat control between both systems was just observed in participants with $n_{ant} \neq 0$. Strictly speaking, however, there is a difference between both systems when $n_{ant} = 0$: the mapping for loudness control is different. This could have caused a better performance for beat control of the TRAINED system, specially in combined tasks, but this effect was not observed. Interestingly, the results also show how participants with $n_{ant} \neq 0$ had special difficulties with the slowest tempo (80 BPM) task that were mitigated when the estimated anticipation was compensated (i.e. when they used the TRAINED system). This is unlikely to be caused by the time granularity limitations of the input device, which in fact would penalize the faster task. At 80 BPM (750 ms) there are 22.5 Kinect frames between two consecutive beats, while there are 18 at 100 BPM (600 ms)[12]. In this sense, the results suggest that observed differences in terms of anticipation of the beat are particularly relevant for slower tempos.

[12] We can assume that the input device limitations would start to harm the performance for faster tempo, even though it was not observed for the selected tempos in our tasks.

4.4 Conclusions

Throughout this chapter we have proposed a strategy to exploit, in a DMI based on the conductor metaphor, the expectations that the user has when using the system. To this end, we have carried out a study that has allowed us to observe how, indeed, there are specific tendencies in the way people make spontaneous conducting movements. Specifically, we have used an analysis with descriptors extracted from body movement to observe two effects: the tendency of each user to anticipate or fall behind the beat and the way in which loudness changes are reflected in this movement. The recordings of this observation study are available online[13].

In this regard, the findings are mainly two. First, the position of the musical beat with respect to the beats that can be extracted from hand conducting movements tends to vary among users, with some tending to anticipate and others to fall behind or be in synchrony. Second, regarding loudness variations, there are certain general tendencies. For example, there are some participants who tend to move more energetically in louder parts (as suggested by a strong correlation between loudness and Quantity of Motion) while others tend to raise their hands higher (as suggested by a strong correlation between the maximum hand height and loudness). However, each person shows particularities that are reflected not only in the descriptors most correlated with loudness but in the "dynamic range" of these descriptors. Also, there are participants for whom no correlations between MoCap descriptors and loudness are found.

The question that follows these observations is whether they are applicable in an interactive context. For example, will a user tending to anticipate moving over fixed music continue to do it when his gesture controls the music? And, accordingly, will she be able to better control a system that compensates for that effect?

Following this idea, we have proposed a DMI based on the conductor metaphor that allows to control beat and loudness in a way similar to those most commonly found in previous systems, but adapting to these user-specific tendencies. The mapping of the system is predefined, but some of its parameters are adjusted from what is observed in spontaneous movements. That is, the user does not consciously and explicitly train her own mapping. We refer to this strategy as *Mapping by Observation*. To verify that what the system learns from these spontaneous movements is useful, we have performed an experiment where we have compared it with a baseline that does not adapt to each user. The experiment has been carefully designed and we have studied the effect of different factors to verify, in a reliable way, that the knowledge incorporated by the proposed

[13] http://mtg.upf.edu/download/datasets/phenicx-conduct

system is indeed useful.

We have not dealt with the underlying mechanisms that may cause differences between participants. We did not analyze whether the different tendencies to anticipate or fall behind the beat are intentional, caused by different sensorimotor synchronization to the beat (Aschersleben, 2002) or by different hand gestures. In the experiment, we could even expect different results if the music material or chosen sound engine had been different. Observation studies of sound-accompanying movements by Jensenius (2007) show that these movements are influenced, among other things, by *action-sound types* (impulsed, sustain, iterative) that depend on the instrument and articulation with which it is played. In any case, the way in which the proposed system compensates different tendencies for anticipation is by compensating an observed effect, regardless of the mechanisms causing it. Also, we selected a musical excerpt where the main melody mostly contains quarter notes, avoiding possible problems with participants conducting to the rhythm instead of the beat, as observed by Lee et al. (2005). This is something to take into consideration, particularly when the goal is to create a system that users can learn to use by themselves.

While this learning from spontaneous movements may be particularly useful in public installations or similar settings, it would obviously be possible to involve the user explicitly in the mapping design. In the next chapter we explore this idea in a context where the user can control the musical articulation.

Chapter 5

Learning user-specific gesture variations for articulation control

As we have seen in Chapter 4, the usability of a musical interface based on the conductor metaphor can improve by making user-specific adaptations built upon knowledge extracted from the analysis of spontaneous conducting movements. So far, we focused on the control of most commonly found parameters of tempo and loudness, but other aspects of the performance can be communicated through conducing gestures. In this Chapter, we introduce articulation, understood as the performance quality that defines the transition and continuity between consecutive notes. For example, *legato* articulation refers to notes played with smooth, connected transitions, while *staccato* refers to the case where notes are played with short duration and detached from each other.

5.1 Introduction

Some existing DMIs based on the conductor metaphor provide articulation control. For example, Garnett et al. (1999) estimate the degree of *legato* or *staccato* based on the ratio of the maximum acceleration to the overall acceleration during a beat period: *staccato* beats are reflected in high acceleration peaks around the beat ictus, while *legato* beats tend to show a more uniform velocity. Usa and Mochida (1998b) also derive articulation from conducting gestures based on two parameters: the gesture smoothness (computed from the sharpness of acceleration peaks) and the "degree of halt", which measures the proportion of time that the baton is stopped between consecutive beats. Curved and smooth gestures without halts are considered *legato*, while straight gestures with clear halts are considered *staccato*. In both cases, the approach is based on expert knowledge that is in accordance with conducting technique theory. Figure 5.1 shows two standard beat patterns at $^4/_4$ with *legato* and *staccato* articulations from Rudolf (1980). For

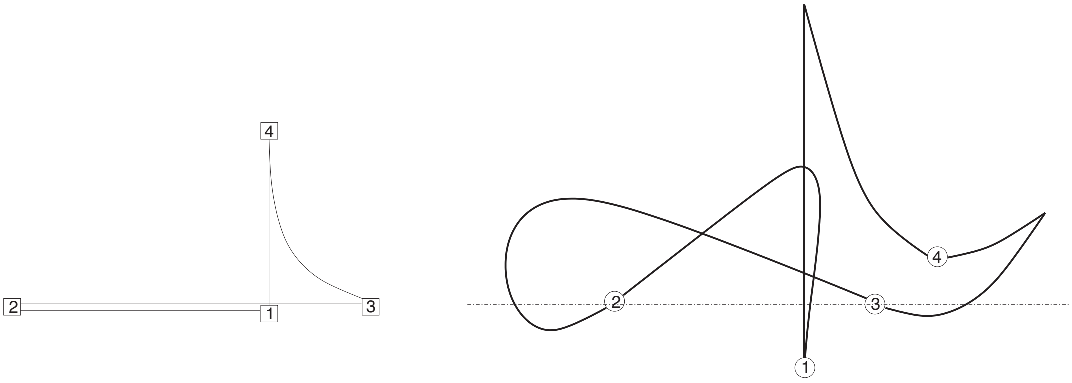


Figure 5.1: 4/4 conducting patterns with *staccato* (left) and *legato* (right) articulation according to Rudolf (1980).

staccato, the shape of the gesture is straight and pauses occur at every beat. For *legato*, the gesture is curved and continuous.

This musical theory is in agreement with the results from the experiment by Platte (2016) (see Section 2.4.2). In the experiment, participants were asked to perform beat tapping on a touch sensor while watching gestures with different shapes and sizes. The length of the touch during the task was associated to articulation (short touch, *staccato*; long touch, *legato*). Significantly shorter touch lengths for concave gestures (Figure 2.3, left) than for convex ones (Figure 2.3, middle) prove that subjects perceived different articulations for different gesture shapes. The fact that the difference was clearer and more coherent for musicians than for non musicians suggests that, at least in part, this shape-articulation association is learnt.

In this Chapter, we also study how different subjects intuitively associate gesture variation and musical articulation. However, we do it in the opposite direction. Instead of presenting them with different gestures and studying which articulation these gestures convey, we present them with sound stimuli played with different articulation and ask them to perform the same gestures with the variations they feel better correspond to the perceived sonic differences. As a starting point, in Section 5.2, we perform an observation study where we investigate how articulation affects conducting gestures in terms of timing (i.e. whether beats detected in hand movement are lagged differently with respect to the musical beat depending on the articulation) and movement dynamics (i.e. whether the articulation is reflected in dynamic descriptors computed from hand movement). Then, in Section 5.3, we propose a model based on the conclusions of this study where the user can control musical articulation through dynamic variations of the same gesture. Our approach follows the *Mapping through Listening* scheme proposed by Caramiaux et al. (2014a). Listening is conceived as the first step in the design of the motion–sound relationship. In the proposed approach, the user first listens to variations of a melody played with different articulations and then teaches the system how she embodies these articulations by performing the same gesture with expressive dynamic variations. As opposed to the case of the previous chapter, where we *mapped by observation* (i.e. the mapping was designed analyzing spontaneous movements without the user being aware of the training), here the user explicitly defines the relationship between dynamic gesture variation and articulation. In this sense, the approach we follow in this case is closer to *Mapping by Demonstration* (Françoise, 2015).

5.2 Observation study

In this observation study, we investigate how articulation is reflected in conducting gestures performed by different participants in terms of timing (i.e. whether beats detected in hand movement are lagged differently with respect to the musical beat depending on the articulation) and dynamics (i.e. whether the articulation is reflected in dynamic descriptors computed from hand movement). Specifically, we compare two articulations: *legato*, with smooth and connected notes, and *staccato*, characterized by short and detached notes.

5.2.1 Objectives

We are interested in analyzing the differences in the execution of conducting gestures due to articulation from two points of view:

- First, we analyze whether gesture execution is unconsciously influenced by performing conducting gestures on top of melodies played with different articulations (*legato, staccato*).
- Second, we analyze how gesture execution changes when trying to convey different articulations (*legato, staccato*).

In both cases, we analyze the gesture variations from two perspectives:

- **Timing**, i.e. how beats detected from the gesture are lagged with respect to beats in music. In this case, we concretely focus on analyzing whether beats detected from hand movement appear at a different distance from the musical beat depending on the articulation.

- **Dynamics**, i.e. variations in movement speed and acceleration.

In summary, the general and concrete hypotheses we test in the study are the following ones:

- **H1**. Conducting gestures performed on top of melodies synthesized with different articulations (*legato*, *staccato*) show variations that can be automatically identified from motion capture data, even if the gestures are not consciously performed differently.
  - **H1a**. Conducting gestures performed on top of melodies synthesized with different articulations differ in their timing.
  - **H1b.** Conducting gestures performed on top of melodies synthesized with different articulations differ in their dynamics.
- **H2**. Conducting gestures performed trying to convey different articulations (*legato*, *staccato*) show variations that can be automatically identified from motion capture data.
  - **H2a**. Conducting gestures performed trying to convey different articulations differ in their timing.
  - **H2b.** Conducting gestures performed trying to convey different articulations differ in their dynamics.
  - **H2c.** Differences are more noticeable if performed on top of melodies played with the articulation being conveyed.

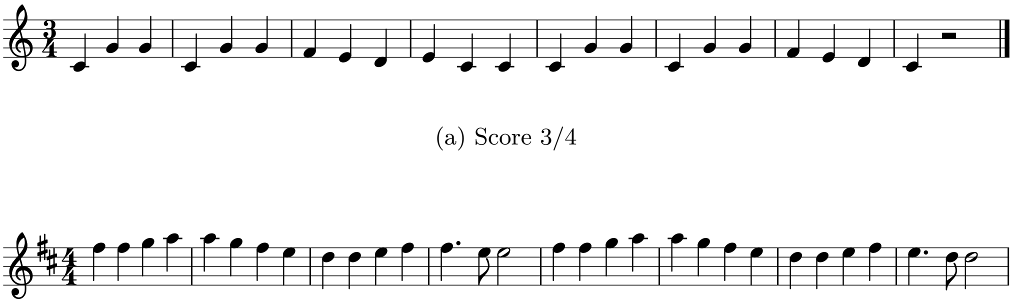
5.2.2 Materials and methods

Materials

We used *KinectVizz*[1] for the recordings. The application allows the experimenter to control the procedure of the study using a GUI and keyboard commands. This control consists on selecting an audio file and playing it while recording aligned video and MoCap from Kinect. In addition, images with the appropriate conducting patterns (Figure 5.3) are shown on screen. Details on the moments of appearance of each of the patterns are given below, when we detail the procedure of the study.

For the study, we use two simple melodies with two different time signatures: 3/4 and 4/4. The scores for each time signature are shown in Figures 5.2a and 5.2b, respectively. We synthesized two versions of each melody, with violin sounds played with *legato* and

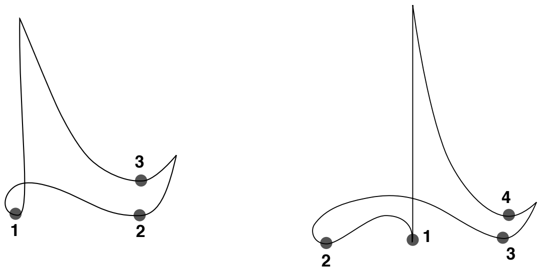
[1] https://github.com/asarasua/KinectVizz



(a) Score 3/4

(b) Score 4/4

Figure 5.2: Scores of melodies synthesized for observation study.



(a) Gesture 3/4

(b) Gesture 4/4

Figure 5.3: Gesture patterns shown to participants in the observation study.

staccato articulations. We used Native Instrument's Kontakt with Session Strings library for the synthesis.

During the study, participants used over-ear headphones and stood approximately two meters from a 46-inch TV screen showing appropriate conducting patterns at each moment. The Kinect sensor was placed below the screen, using a flat speaker stand, approximately 1.4 m from the floor. The experimenter read instructions to participants and controlled the application from a laptop to which the screen, Kinect sensor and headphones were connected.

The recordings of this observation study are available online[2].

[2] http://mtg.upf.edu/download/datasets/phenicx-conduct

Methods

Timing We focus on analyzing how far beats detected in hand movement tend to be from actual beat positions in the music. For this, we take a_y maxima as estimates for beat positions and build the error distribution $\mathbf{e}$ following the same procedure of preceding chapters and detailed in Algorithm 1 on page 68. Since in this case we focus on analyzing how *far* detected beats are from beats in the music, we take the mean of the absolute value distribution $|\mathbf{e}|$ as an estimation of the lag between the beats detected from MoCap data and the actual instants when beats appear in the played music. Accordingly, lag here only takes positive values.

Dynamics We compute two-dimensional velocity and acceleration values in the frontal plane (where participants *drew* the gesture) from the position of the hand used by the participant. We use numerical differentiation from aligned x and y raw position data. At each frame, we fit a second-order polynomial to the 7 consecutive points centered at it and compute the derivative of the obtained polynomial. Instantaneous speed $v(t_i)$ and acceleration $a(t_i)$ are computed as the length of the two-dimensional vectors for both velocity and acceleration at each frame i. For the analysis, we use averaged values of $v(t_i)$ and $a(t_i)$ over 30 frames (1 second) in consecutive, non-overlapped windows. In previous Chapters, we have denoted this averaged values $v_{mean}(t_i, 30)$ and $a_{mean}(t_i, 30)$, with 30 indicating the number of frames used for averaging. In the following, for simplification, we denote these variables v and a, respectively. These are the variables we use for the gesture dynamics representation space.

Separability In addition to studying the effect of different factors in the aforementioned variables, we are interested in evaluating the concrete potential of the representation space formed by dynamic descriptors we used in the study (v and a) to discriminate between articulations. For this, we compute the separability S between both classes (articulations) in this representation space. A low separability indicates that articulations are ambiguous in the representation space, while a high value suggests that it is possible to discriminate between classes. The separability is a common criterion in machine learning, and it is implemented in well-know classification techniques such as Fisher Linear Discriminant Analysis (LDA) (Zhao et al., 2012). The separability measure S is defined as the distance ratio between the data belonging to different classes (articulations) to the variance of data within each class. More concretely, it is computed as the ratio between the norms of the "between classes scatter matrix" S_B and the "within classes scatter matrix" S_W:

$$S = \frac{\|S_B\|}{\|S_W\|}, \tag{5.1}$$

with both matrices defined as

$$S_B = \sum_c N_c(\mu_c - \bar{x})(\mu_c - \bar{x})^T \tag{5.2}$$

$$S_W = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T \tag{5.3}$$

where,

$$\mu_c = \frac{1}{N_c} \sum_{i \in c} x_i \tag{5.4}$$

$$\overline{x} = \frac{1}{N} \sum_i x_i \tag{5.5}$$

and N is the total number of cases, and N_c the number of cases in class c.

Participants

Participants were recruited via convenience sampling through department members and their students. They signed an informed consent which contained the approximate duration of the study (around 15 minutes per participant) and its general objectives, as well as the type of data being recorded and the intention of making it publicly available.

Procedure

The study was divided in two parts. In the first one, all participants follow the same procedure. In the second one, participants are randomly split in two groups, depending on the sound stimuli used as background.

Part 1 The experimenter asks the participant to perform the gesture shown on screen (Figures 5.3a and 5.3b for 3/4 and 4/4, respectively) following the tempo of the played melody. Before actually recording, the participant is first allowed to listen to the melody as many times as necessary to memorize it. Finally, the melody is played twice preceded by two meters of metronome and the application records aligned video and MoCap data from the Kinect.

This process is repeated four times, once for each of the two melodies and articulations (4/4 *legato*, 4/4 *staccato*, 3/4 *legato*, 3/4 *staccato*), counterbalancing the order of appearance across participants. The difference between both versions (articulations) of each melody is not mentioned, and no instructions on how to execute the gesture are given beyond the pattern image shown on screen.

Part 2 The experimenter explains to the participant that the difference between both versions of each melody in Part 1 is due to the articulation with which they have been synthesized. This time, the participant is asked to repeat the process but trying to convey those articulations with the variations in gesture execution that she feels best match the variations in articulation. Half of the participants perform the gestures on top of the melodies (as in Part 1); the other half perform the gestures on top of a metronome, receiving instructions on the gesture to execute and articulation to convey for each case. Again, the process is repeated four times (4/4 *legato*, 4/4 *staccato*, 3/4 *legato*, 3/4 *staccato*).

5.2.3 Results

Twenty four people (5 female) ranging in age from 21 to 50 years old (μ=31.1, σ=5.9) volunteered to participate in the study. Participants had different musical expertise levels: 5 had no musical training, 7 had some non-formal training, 5 had less than 5 years of musical training, and 7 were expert musicians with more than 5 years of training. No conductors were recruited for the study.

All results presented in this section consider data from the takes with both time signatures (3/4 and 4/4).

Tables 5.1 and 5.2 show a summary of the results for Parts 1 and 2 of the study, respectively. In both cases, each row shows mean and standard deviation values of the absolute value of the error distribution $|\mathbf{e}|$, v and a with each articulation (*legato, staccato*) for a participant. Values in every column are computed from data with both time signatures, 4/4 and 3/4. Recall that the mean of the absolute value of the error distribution is our estimation of the *lag* between beats detected from hand movement and beats in the music. Accordingly, the mean of $|\mathbf{e}|$ is denoted as *lag* in the tables.

Figure 5.4 shows the absolute value of the error distributions for different conditions and articulations.

Table 5.1: Summary of results in Part 1.

	Legato						*Staccato*					
	$\lvert\mathbf{e}\rvert$ (s)		speed ($^{\mathrm{m}}/_{\mathrm{s}}$)		acc. ($^{\mathrm{m}}/_{\mathrm{s}^2}$)		$\lvert\mathbf{e}\rvert$ (s)		speed ($^{\mathrm{m}}/_{\mathrm{s}}$)		acc. ($^{\mathrm{m}}/_{\mathrm{s}^2}$)	
	lag	σ	μ	σ	μ	σ	*lag*	σ	μ	σ	μ	σ
1	.077	.06	1.37	.11	11.07	.94	.076	.05	1.32	.11	10.74	1.25
2	.237	.13	1.11	.11	8.88	1.07	.159	.11	1.14	.07	9.64	.58
3	.114	.06	1.10	.19	9.14	1.64	.053	.06	1.09	.15	9.54	1.27
4	.100	.08	.94	.07	7.78	1.01	.052	.04	.93	.04	7.79	.48
5	.090	.05	.96	.07	7.46	.56	.045	.03	.80	.11	6.66	.86
6	.170	.08	.86	.06	6.42	.77	.068	.05	.79	.05	6.55	.54
7	.246	.12	.85	.15	5.86	1.61	.139	.09	.79	.15	5.71	1.46
8	.078	.07	.54	.07	3.74	.50	.061	.07	.53	.07	3.91	.55
9	.092	.08	.87	.09	7.01	.96	.169	.09	.83	.05	6.60	.62
10	.169	.07	1.09	.11	8.85	1.12	.066	.04	1.13	.19	9.66	1.68
11	.130	.08	.92	.05	7.36	.86	.053	.05	.88	.09	7.11	1.06
12	.268	.18	1.20	.19	9.01	1.30	.204	.19	1.24	.33	9.94	2.47
13	.177	.09	1.09	.18	8.43	1.70	.065	.04	1.05	.18	8.22	1.58
14	.065	.06	.93	.11	7.02	.77	.097	.06	.81	.08	6.34	.56
15	.129	.13	.83	.08	6.08	1.04	.262	.19	.79	.06	5.57	.51
16	.093	.08	1.09	.09	8.41	.83	.087	.09	1.02	.15	8.16	1.30
17	.072	.06	.73	.13	5.06	2.16	.068	.06	.68	.07	5.25	.93
18	.138	.10	1.35	.17	12.10	1.78	.155	.14	1.18	.18	10.56	2.12
19	.108	.08	1.42	.34	11.28	3.26	.084	.08	1.43	.24	11.87	2.15
20	.145	.10	.74	.05	5.35	.48	.111	.11	.77	.05	5.71	.50
21	.104	.11	.91	.08	6.70	.72	.073	.06	.73	.08	5.32	.62
22	.232	.11	1.05	.13	7.41	1.25	.167	.10	.94	.14	6.79	1.34
23	.187	.07	1.30	.22	10.98	2.37	.097	.06	1.23	.13	10.67	1.76
24	.253	.15	.71	.09	5.07	1.02	.221	.13	.79	.16	5.89	1.68

Part 1

The average values across all participants for *lag* are 0.142 s ($\sigma = 0.11$) and 0.108 s ($\sigma = 0.11$) for *legato* and *staccato* articulations, respectively. An independent-samples t-test confirmed these differences due to Articulation are significantly different; $t(5079.2) = 11.076, p < 0.001$. These results suggest that, in general, beats detected from hand movement vertical acceleration fall closer to the actual beat positions when participants perform on top of melodies played with *staccato* articulation. An F test comparing the variances of both distributions indicated no significant differences, which suggests that the tendencies are equally consistent.

Significant differences are also observed for v, t (3405.9) = 4.97, p < 0.001. Average

Table 5.2: Summary of results in Part 2.

	Legato						*Staccato*					
	$\lvert\mathbf{e}\rvert$ (s)		speed ($^{\mathrm{m}}/_{\mathrm{s}}$)		acc. ($^{\mathrm{m}}/_{\mathrm{s}^2}$)		$\lvert\mathbf{e}\rvert$ (s)		speed ($^{\mathrm{m}}/_{\mathrm{s}}$)		acc. ($^{\mathrm{m}}/_{\mathrm{s}^2}$)	
	lag	σ	μ	σ	μ	σ	*lag*	σ	μ	σ	μ	σ
1	.091	.06	1.48	.20	11.76	1.84	.050	.07	1.55	.13	13.73	1.48
2	.074	.08	1.22	.13	10.46	1.10	.069	.04	1.45	.11	13.33	1.24
3	.087	.06	1.31	.16	10.64	1.26	.060	.06	.99	.22	8.80	2.00
4	.047	.04	.96	.10	7.38	1.12	.073	.08	1.06	.13	9.59	1.39
5	.084	.08	1.19	.14	9.19	1.26	.038	.03	1.00	.07	8.40	.66
6	.079	.08	.81	.05	6.13	.71	.042	.04	.75	.06	6.56	.53
7	.202	.08	.94	.15	6.78	1.63	.079	.06	1.07	.16	8.55	1.63
8	.080	.06	.65	.04	4.31	.37	.045	.07	.53	.05	4.49	.46
9	.099	.08	1.04	.14	7.98	1.37	.074	.07	.86	.16	7.71	1.27
10	.104	.05	1.35	.13	11.24	1.09	.083	.06	1.07	.09	8.85	.86
11	.137	.12	.93	.06	7.42	.69	.054	.07	.73	.11	6.15	.98
12	.227	.20	.93	.07	6.05	.69	.180	.16	1.12	.09	9.76	1.05
13	.184	.07	1.11	.11	8.57	1.13	.059	.10	1.04	.07	8.77	.65
14	.078	.05	.98	.12	7.33	1.35	.085	.04	.79	.06	6.32	.64
15	.068	.05	1.10	.25	8.27	1.85	.112	.12	.76	.28	5.97	1.90
16	.119	.08	1.14	.10	8.10	.84	.052	.08	1.18	.12	9.83	.90
17	.220	.21	.64	.06	3.75	.52	.072	.10	.66	.06	5.10	.67
18	.220	.07	1.48	.11	12.95	1.04	.302	.06	1.29	.15	12.17	1.36
19	.106	.06	1.73	.11	14.22	1.19	.057	.07	1.69	.11	14.39	1.02
20	.179	.11	.93	.07	6.15	.61	.090	.10	.81	.11	6.13	.84
21	.067	.05	1.03	.06	7.38	.72	.067	.07	.78	.07	6.24	.69
22	.207	.08	1.09	.07	7.44	.87	.094	.06	1.04	.10	8.46	1.21
23	.085	.06	1.69	.20	14.78	1.99	.050	.04	1.59	.17	14.76	1.94
24	.258	.12	.86	.08	6.28	.89	.082	.05	.89	.07	7.20	.67

values are $v = 1.00$ $^{\mathrm{m}}/_{\mathrm{s}}$ ($\sigma = 0.26$) for *legato* and $v = 0.95$ $^{\mathrm{m}}/_{\mathrm{s}}$ ($\sigma = 0.26$) for *staccato*. No significant differences are observed in the case of a.

Musical Expertise, introduced as a factor with 4 levels corresponding to the 4 groups previously identified, did not yield any significant effects on the dependent variables. No significant effect was observed for the time Signature (3/4 or 4/4) either. Recall that a a different melody and gesture were used for each time signature.

Part 2

In this part, participants were split into two different groups depending on whether they performed the gestures on top of a metronome, or again on top of the synthesized

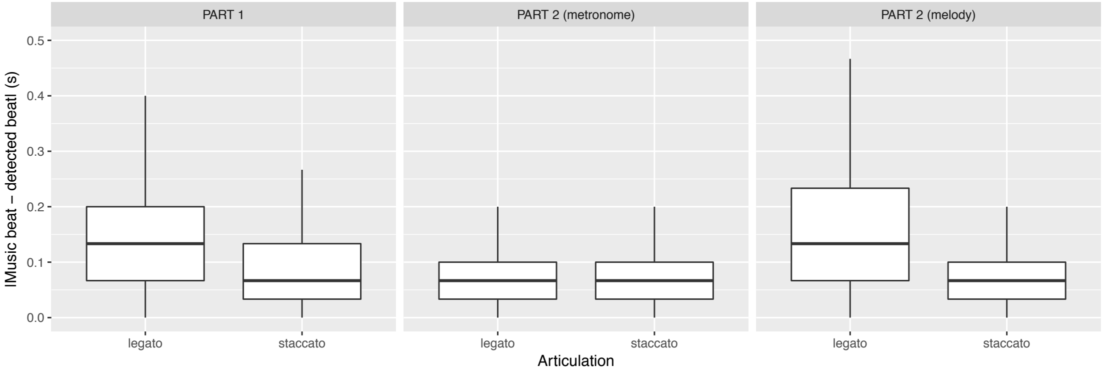


Figure 5.4: Beat error distributions for different conditions and articulations.

melodies. We perform 2x2 ANOVA to analyze the effect of Articulation (*legato* or *staccato*) and the Background (metronome or synthesized melody) on the values of *lag*, v and a.

For *lag*, the strongest effect is caused by the Articulation, with average values of 0.131 s ($\sigma = 0.11$) for *legato* and 0.082 s ($\sigma = 0.09$) for *staccato,* $F_{(1,4469)} = 267.76, p < 0.001$. Significant differences were also observed for Background, with average *lag* values of 0.088 s ($\sigma = 0.10$) when the gestures were performed on top of a metronome, and 0.120 s ($\sigma = 0.12$) performing on top of the synthesized melodies, $F_{(1,4469)} = 114.37, p < 0.001$. The interaction between both factors also had a significant effect, $F_{(1,4469)} = 59.89, p < 0.001$. The differences between average *lag* values for different articulations were bigger with gestures performed on top of the melodies (0.154 s ($\sigma = 0.10$) for *legato* and 0.086 s ($\sigma = 0.11$) for *staccato*) than on top of the metronome (0.099 s ($\sigma = 0.11$) for *legato* and 0.077 s ($\sigma = 0.09$) for *staccato*). These results indicate that the melody on top of which the gestures are performed has a stronger effect on the timing than the intention to convey one or other articulation.

We perform an F test comparing the variances of both distributions in the cases of metronome and melodies for background. Only in the case of participants performing the gestures on top of melodies we observe a significant difference between the standard deviations, F (2212) = 1.36, p < 0.001. This indicates that, in this case, the observed tendency is more consistent for *staccato* than for *legato*, with a wider distribution.

Only Articulation has an effect on v values, $F_{(1,2948)} = 50.54, p < 0.001$. Average values are $v = 1.05$ m/s ($\sigma = 0.29$) for *legato* and $v = 0.99$ m/s ($\sigma = 0.29$) for *staccato.* The same is observed for a, with average values of $a = 8.54$ m/s² ($\sigma = 2.96$) for *legato* and $a = 8.81$ m/s² ($\sigma = 3.04$) for *staccato*, $F_{(1,2948)} = 5.98, p < 0.05$.

As in Part 1, musical Expertise does not cause any significant effects on the dependent variables. No significant effects are observed for the time Signature (3/4 or 4/4) either.

Idiosyncrasy of gesture variations

The results presented so far evaluate how values of lag, v and a vary across all participants throughout the different conditions of the study. However, participants were not given any instructions on how to perform depending on the articulation. In this context, it is interesting to study whether these variations are similar across participants, or if they are idiosyncratic.

For this reason, we included the Participant as a new factor in the analysis, and evaluated the effect of its interaction with Articulation on lag, v and a performing 2x24 ANOVA for each of these variables.

In Part 1, the interaction between Articulation and Participant yielded significant effects on lag ($F_{(23,5045)} = 20.75, p < 0.001$), v ($F_{(23,2904)} = 8.45, p < 0.001$) and $a(F_{(1,2904)} = 8.19, p < 0.001)$.

Significant and stronger effects of this interaction appear in Part 2: lag ($F_{(23,4425)} = 25.88, p < 0.001$), v ($F_{(23,2904)} = 45.32, p < 0.001$) and a ($F_{(1,2904)} = 52.51, p < 0.001$).

The results in Tables 5.1 and 5.2 help to understand where the effect of this interaction comes from. As an example, we observe different variations of lag in Part 1 (Table 5.1). For instance, some participants show similar lag for both articulations, such as participants 1 ($lag = 0.077$ s for *legato* and $lag = 0.076$ s for *staccato*) and 17 ($lag = 0.072$ s for *legato* and $lag = 0.068$ s for *staccato*). Others, however, show very different values depending on the articulation, such as participants 3 ($lag = 0.114$ s for *legato* and $lag = 0.053$ s for *staccato*) and 6 ($lag = 0.170$ s for *legato* and $lag = 0.139$ s for *staccato*).

Discriminating articulations from gesture variations

Figures 5.5 and 5.6 illustrate all v and a values for each participant in Parts 1 and 2, respectively, and the computed separability values. These figures show how clusters corresponding to each articulation (red = *legato* and blue = *staccato*) are more separated in the case of Part 2 (where participants where indeed trying to convey different articulations through gesture variations). An independent-samples t-test confirmed that the separability values of Part 1 ($S = 0.32$, $\sigma = 0.59$) and 2 ($S = 1.28$, $\sigma = 1.36$) are significantly different ; $t(31.264) = -3.14, p < 0.01$.

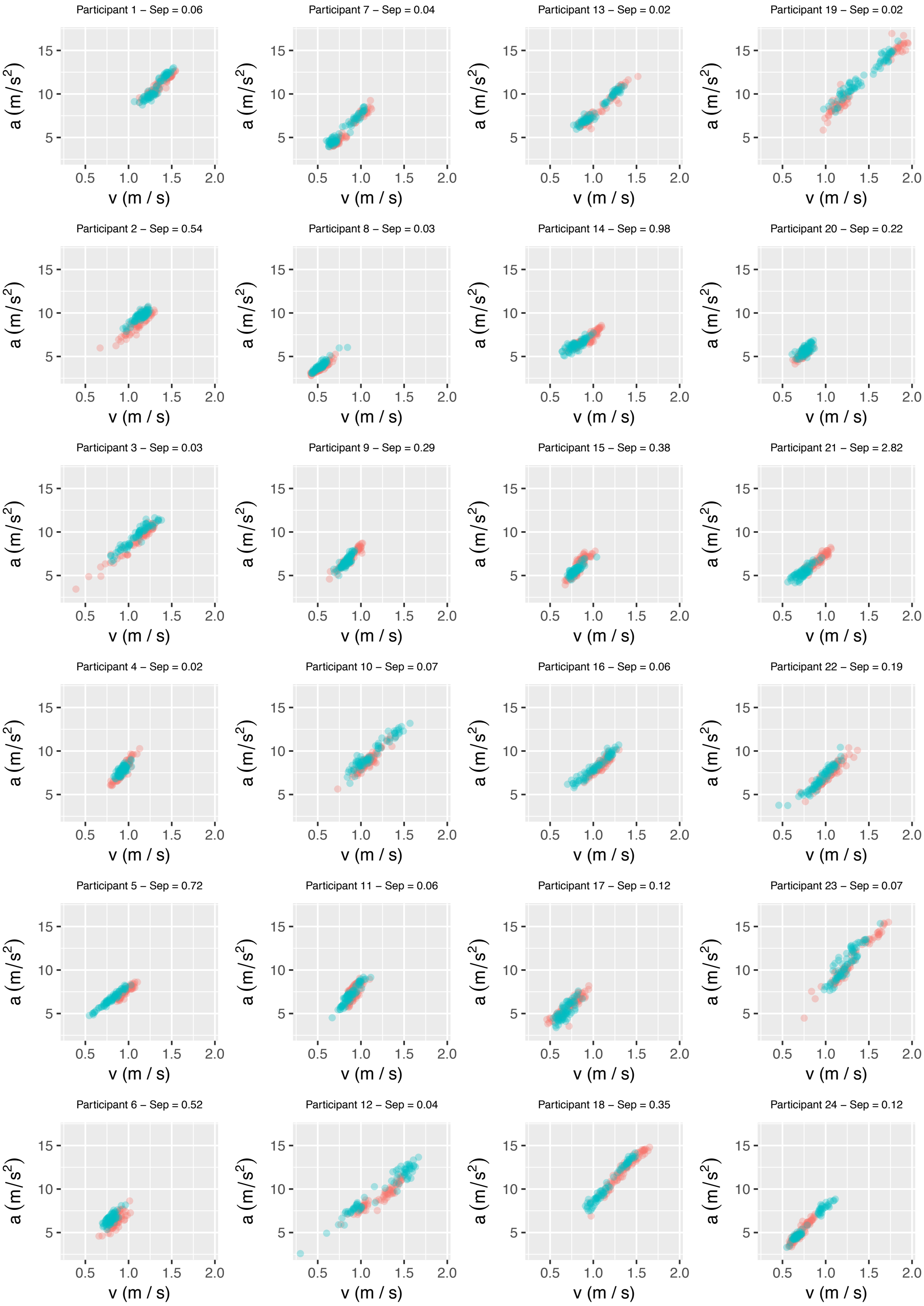


Figure 5.5: v vs a values for each participant in Part 1 of the study. red = *legato*, blue = *staccato*. Computed Separability between articulations (S) values are indicated for each participant.

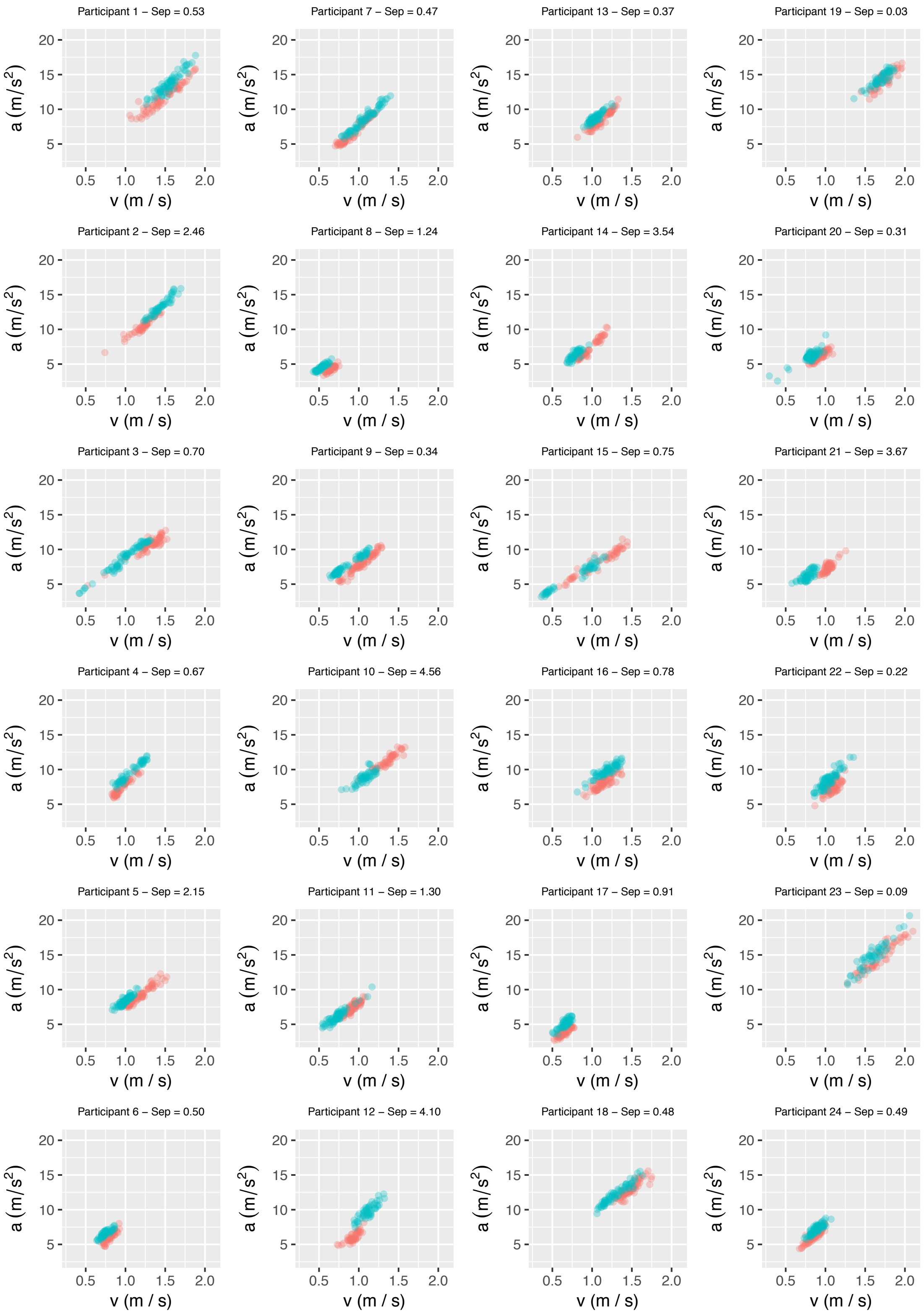


Figure 5.6: v and a values for each participant in Part 2 of the study. red = *legato*, blue = *staccato*. Computed Separability between articulations (S) values are indicated for each participant.

Discussion

In Part 1 in this study, we investigated whether conducting gestures performed on top of melodies synthesized with different articulations (*legato*, *staccato*) show variations that can be automatically identified from MoCap data. Our hypothesis **(H1)** was that these variations would appear and would be reflected in different timing (**H1a**) and dynamics (**H1b**) for each articulation. **H1a** is confirmed by the results, with a tendency to detect beats in hand motion closer to beats in the music for *staccato* articulation. Dynamics between both articulations reflected in values of v and a tend to differ, particularly when considering separately cases for each participant. However, these differences seem to be weak, specially compared with the case of the second part of the study, where participants tried to actually convey different articulations through gesture variations. For this reason, we consider that **H1b** is not completely backed up by the results.

Differences between both articulations were much clearer in Part 2 of the experiment, where participants were asked to convey articulations through gesture variations. The hypothesis **H2** tested in this second part is that the aforementioned differences also appear in this case. In this sense, **H2** was confirmed both in terms of different timings (**H2a**) and dynamics (**H2b**). Interestingly, participants who performed gestures on top of a metronome showed smaller differences in timing. This suggests that observed differences between both articulations might me mainly due to different sensory-motor synchronization to melodies played with different articulations. In any case, **H2c** (differences are bigger for gestures performed in top of the melodies than on top of a metronome) was only confirmed in terms of timing.

The confirmation of **H2b** is particularly relevant in terms of its potential to be applied in real-time control of articulation. Simple descriptors extracted from hand movement such as v and a, which can be computed in real-time, can be used to discriminate between different articulations conveyed by the user. It is also relevant to point out that, in this study, participants were not given instructions on how to convey these articulations, so the observed gesture variations are the result of their intuitive motion-sound mappings having listened to examples with different characteristics associated to articulation (note length, connection between consecutive notes). As the results suggest, the variations introduced by each participant were idiosyncratic and, accordingly, a model that provides control over articulation through these gesture variations should be able to adapt to these idiosyncrasy.

5.3 Controlling articulation with idiosyncratic gesture variations

Motivated by the conclusions of the observation study, we developed a system where users train their own models to control music articulation with conducting gestures. As opposed to the case explored in Chapter 4, where we learned the mapping by *observing* spontaneous conducting movements, here the user explicitly designs her own mapping through gesture variation examples. This follows the idea we introduced in Chapter 1 and illustrated in Figure 1.5. The approach presented in this Section is more aligned with *Mapping by Demonstration* as defined by Françoise (2015). Listening is the first step in the mapping design, and the mapping is explicitly defined by the user through gesture variation examples performed while listening to sound stimuli.

5.3.1 Proposed system

The system is depicted in Figure 5.7. First, the user teaches the system how she embodies music articulations by performing the same gesture with expressive dynamic variations. For this, the user first listens to a melody synthesized with different articulations, and then performs the example gestures while listening to the different stimuli. Phrase articulation can range from totally *legato* to totally *staccato*. The system is however not constrained to these particular articulations, nor limited to two or three articulations. In this sense, the system is generic.

During performance, the user starts to execute a new gesture with a given dynamic variation. The trained model takes this gesture as input and infers which articulation the user is doing. The inferred articulation may be one of the learned articulations (from the training dataset), but it may also be a combination of learned articulations. In other words, the articulation space is not discrete, but continuous. The inferred articulation then controls the way the synthesized melody is rendered. For the aforementioned example, the sound engine creates long notes with long attack and release for *legato* and short notes with short attack and release for *staccato*.

Learning a model of articulations

The computational design can be formulated as a supervised learning problem: the user provides a set of data input, each one representing an articulation of the same gesture, paired with outputs encoding the articulations. This is *classification*. In the

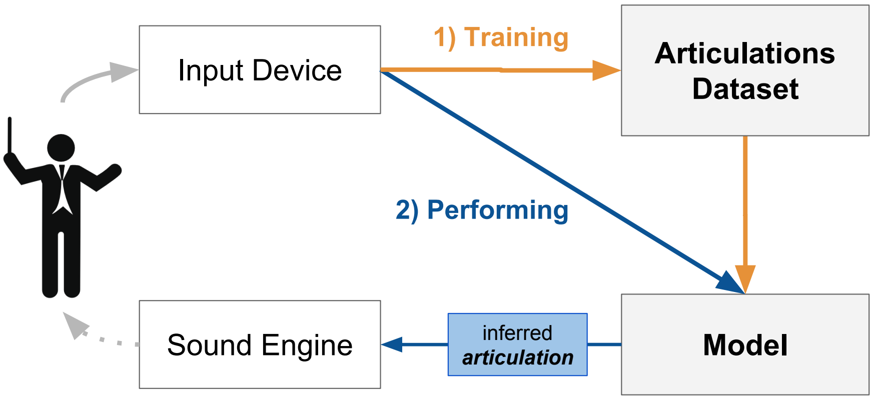


Figure 5.7: System diagram. During training, the user teaches the system how she embodies different articulations. During following, the model estimates the *inferred articulation* from gesture dynamic variations, driving the sound synthesis.

interaction scenario described above, we are specifically interested in having continuous output informing the proportionate level of each articulation within a given gesture. One solution is to interpolate between classes, achieving a form of *soft classification*.

The learning procedure is represented in Figure 5.8 with actual data from the user study presented in Subsection 5.3.2. From the gesture (a shape drawn by the user), we extract dynamic features (velocity and acceleration). These features feed a probabilistic model based on a Gaussian Mixture Model (GMM). We chose GMM for different reasons. Firstly, as a generative model, it can be used either for soft classification or as regression model. Secondly, GMM is a probabilistic model that can handle (gaussian) noise efficiently. We use GMM in a supervised mode, by providing the algorithm with the training dataset and a code for each articulation. The articulation code is an integer index, incremented for each new articulation added to the training dataset. In this sense, for best performance, training instances should be provided in a meaningful order from which the system can infer a continuum. We initialize the model by providing the means of each class (see Figure 5.8, bottom-right). An Expectation-Maximization (EM) (Dempster et al., 1977) algorithm iteratively adapts the covariance matrices, creating a model of each articulation. At test time, incoming gestures are analyzed online. The model will then assign a continuous value to it, representing the relative distance between each articulation.

More concretely, in the GMM model, the dataset is represented by a mixture of C (number of trained articulations) Gaussian components defined by the probability density function

$$p(x_i) = \sum_{j=1}^{C} w_j p(x_i|A_j) \tag{5.6}$$

where $x_i \in \mathbb{R}^D$ is a D-dimensional datapoint, w_j is the prior probability (or weight) of the jth component ($w_j \geq 0, \sum_{j=1}^{C} w_j = 1$) and $p(x_i|A_j)$ is the conditional probability function:

$$p(x_i|A_j) = \mathcal{N}(x_i; \mu_j, \Sigma_j) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_j|}} e^{-\frac{1}{2}\left((x_i-\mu_j)^T \Sigma_j^{-1}(x_i-\mu_j)\right)} \tag{5.7}$$

with A_j denoting the jth articulation, and $\mu_j \in \mathbb{R}^D$ and Σ_j the mean and $D \times D$ covariance of the jth gaussian component, respectively.

For training, each articulation A_c is represented by a set of N inputs $\mathbf{x}_i^c, i = 1..N$ (note that the number of inputs per articulation can vary). For each articulation, the mean of the N inputs is computed and denoted μ_c. The training process then finds the value of the weights $w_j, j = 1..C$ and the covariances $\Sigma_j, j = 1..C$, using EM. The EM algorithm estimates the values of these parameters that maximize the likelihood of the training data. See Dempster et al. (1977) for a complete derivation of the algorithm.

The posterior likelihood is then given by

$$p(A_c|x) = \frac{w_k \mathcal{N}(x; \mu_c, \Sigma_c)}{\sum_{j=1}^{C} w_j \mathcal{N}(x; \mu_j, \Sigma_j)} \tag{5.8}$$

Having coded each articulation in the dataset by an integer index incremented for every new articulation, we can use the posterior likelihoods for an input x to derive a value for articulation $A(x)$ in a continuum via *soft classification* as previously introduced:

$$A(x) = p(A_1|x) + 2 \cdot p(A_2|x) + ... + C \cdot p(A_C|x) \tag{5.9}$$

5.3.2 User study

We carried out a user study to evaluate if users' gesture articulations can be learned with the proposed model and if users can subsequently use this model to control music articulation by varying gesture. We also inspect the effect of musical expertise and, in addition, we use two input devices to capture gestures: a Kinect v2 and a mouse.

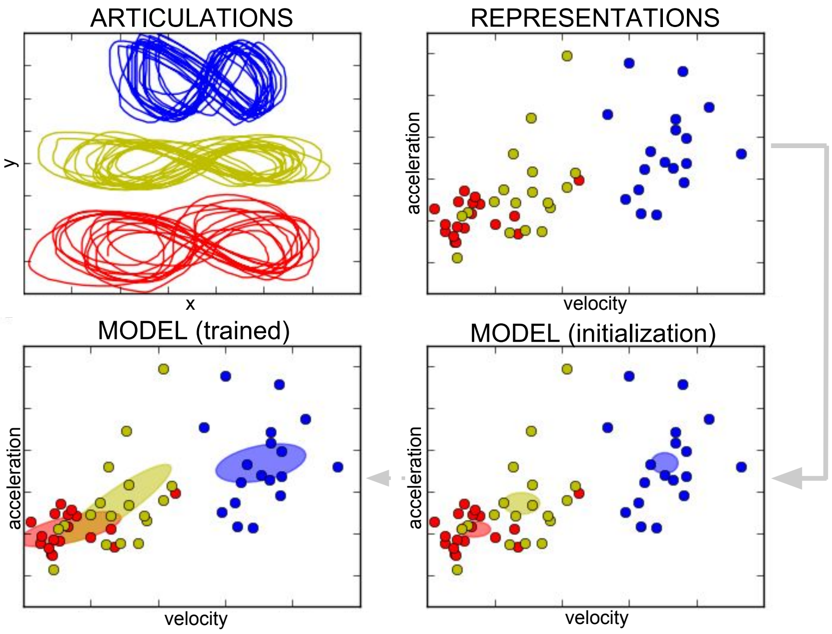


Figure 5.8: Learning procedure for participant 5 with mouse as input device. Input examples are represented in the velocity-acceleration feature space and associated to an articulation label. The representation feeds a GMM initialized with the means of each class and adapted using Expectation-Maximization.

We do this to test possible differences in the ability to control articulation through expressive gesture variation using a MoCap device such as the Kinect, where movements are performed freely, or a mouse, which restricts movements to a certain plane and space.

Materials

We built a Windows application to be used with a Microsoft Kinect v2 or mouse with OpenFrameworks. It uses ofxKinectForWindows2[3] (a wrapper for Microsoft's official SDK) to track skeleton data and Maximilian[4] (Grierson and Kiefer, 2011) for sound synthesis. Maximilian is an MIT-licensed C++ library for audio synthesis and signal processing, particularly suitable for applications built with OpenFrameworks. The application allows the experimenter to control the procedure of the study using keyboard commands. During training, it plays back the melody with articulations *legato*, *normal*, and *staccato* (respectively coded 1, 2 and 3) and records v and a values computed in real

[3] https://github.com/elliotwoods/ofxKinectForWindows2
[4] https://github.com/micknoise/Maximilian

time to form the training set. *Normal* refers to a melody synthesized with parameters in between the ones used for *legato* and *staccato.* Two-dimensional position is also recorded for posterior visualization. During performance, the application determines the articulation value (from 1 to 3) from v *and* a values computed in real time, controlling the way the synthesized melody is rendered. The hand position and the estimated articulations by the model are recorded frame by frame for analysis.

For the study, we use the simplified score of Beethoven's *Ode to Joy* also used in the observation study and depicted in Figure 5.2b. Participants used over-ear headphones. When using the Kinect v2, they stood approximately two meters from a 46-inch TV screen showing the trace of the tracked hand (left or right depending on participant's preference); when using the mouse, they sat in front of the laptop showing the trace of the mouse position. In both cases, during performance, a slider shows the fixed target articulation value together with the one being inferred in real time. Figure 5.9 shows a snapshot of the application during performance, with target articulation *normal*[5]. The Kinect v2 sensor was placed below the screen, using a flat speaker stand, approximately 1.4 m from the floor.

Regarding the sound synthesis, we use simple sine waves and ADSR modulation. The concrete parameters being modified depending on the articulation were the attack and release times (also affecting the duration) of each note. For articulation $A = 1$ (*legato*), values were set as 600 ms for attack and 3000 ms; for articulation $A = 3$ (*staccato*), they were 50 ms for attack and 50 ms for release. Intermediate values for attack and release were linearly interpolated according to the value of A.

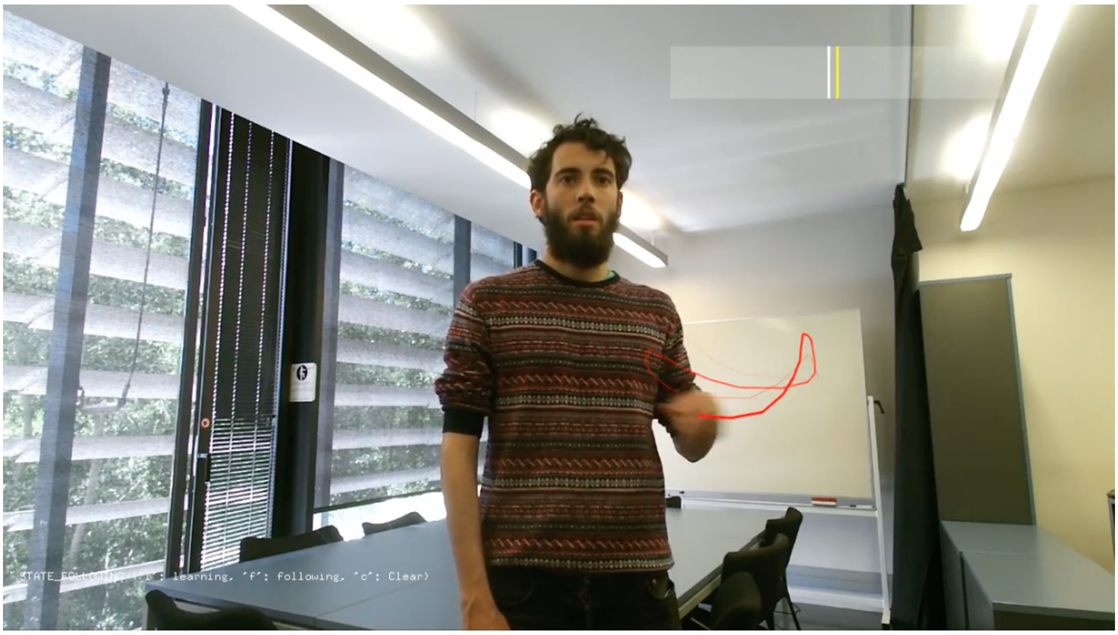
The experimenter read instructions to participants and controlled the application from a laptop to which the screen, Kinect v2 sensor, mouse and headphones were connected.

Methods

The method for training the model for articulations and computing the resulting articulation value during performance is the one explained in Subsection 5.3.1.

We compute two-dimensional velocity and acceleration values. In the case of the Kinect, we use values in the frontal plane (where participants *draw* the gesture) from the position of the hand used by the participant. We use Low Pass Differentiators (LPD) proposed by Skogstad et al. (2012) to compute instantaneous velocity ($v(t_i)$) and acceleration ($a(t_i)$). For the representation space, we compute a running average to smooth the

[5] The snapshot was not taken during the actual study: video from the Kinect was not recorded to minimize performance issues.



```
STATE_FOLLOWING (´l´: learning, ´f´: following, ´c´: Clear)
```

Figure 5.9: Snapshot of the application during performance using a Kinect v2 as input device. Here, target articulation $A_T = 2$ (*normal*) is indicated by a yellow line on the slider; the white line indicates the inferred articulation at the moment.

values, considering the last second (i.e. 30 frames for the Kinect V2, 60 frames for the mouse). As in the case of the observation study, we denote these variables v and a for velocity and acceleration, respectively. Every datapoint is thus composed by two values $x_i = \{v_i, a_i\}$.

Participants

Participants were recruited via convenience sampling through department members and their students. They signed an informed consent which contained the approximate duration of the study (around 25 minutes per participant) and its general objectives, as well as the type of data being recorded.

Procedure

The study procedure is repeated twice, once for each input device, counterbalancing the order across participants. The participant is briefed that she will control the articulation of a melody (a excerpt from Beethoven's *Ode to Joy* from the 9$^{\text{th}}$ Symphony) using figure-eight gestures.

In the Training Phase, the experimenter plays the stimulus melody with articulations *legato*, *normal*, and *staccato* (respectively coded 1, 2 and 3). *Normal* refers to a melody synthesized with parameters in between the ones used for *legato* and *staccato*. The participant is encouraged to perform these gestures with the variations she feels best match the articulations. She can rehearse until she feels confident and then records the training examples (one gesture variation for each articulation).

In the Task Phase, the participant is presented with one of the melody versions used for training, the articulation of the version being the target articulation. After listening to it, she is asked to start drawing a figure eight in order to control the melody articulation such as to reach the articulation target, *as close to the example as possible* until the melody ends. Two bars with a metronome are played before the melody starts. This process is repeated twice for each of the 3 target articulations appearing in random order. As visual feedback, a screen shows the trace of the gesture. During performance, a slider shows the fixed target articulation value together with the inferred one, as shown in Figure 5.9. The hand position and the estimated articulations by the model are recorded frame by frame for analysis.

At the end of the study, participants are asked to rate the following aspects of the task on a scale from 1 (total disagreement) to 7 (total agreement):

- *Globally, do you think you managed to fulfill the tasks asked during the study?*
- *When you were controlling the music, was the audiovisual feedback of your movement variations what you were expecting?*

In addition, they are asked whether, during control, they focused on audio and visual feedback, only audio feedback, or only visual feedback.

Evaluation metrics

The accuracy of the estimated articulation compared to the intended one is assessed by computing ϵ, the mean error between the running articulation estimation (along time) and the given target, during the Task Phase. Figure 5.10 reports an example of estimated articulation for participant 19, target 2, using Kinect v2. ϵ is the mean difference between

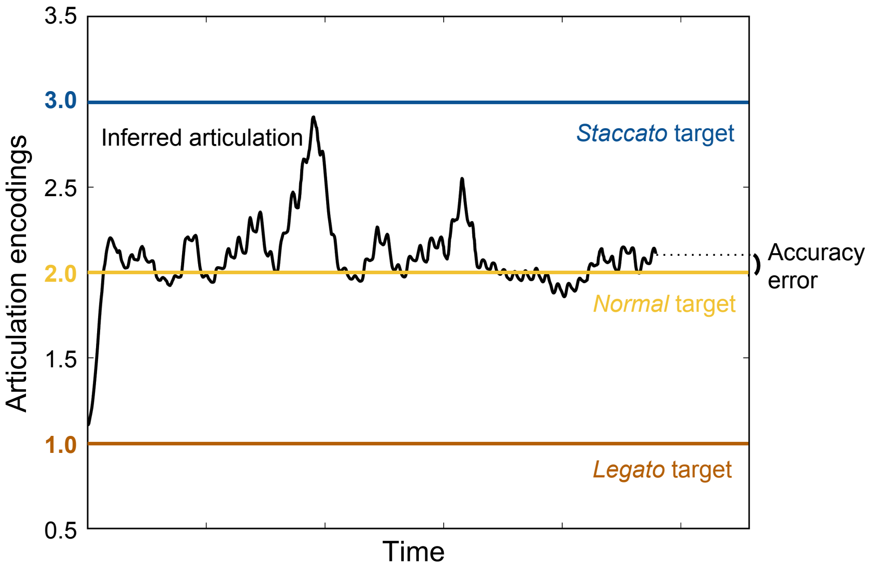


Figure 5.10: Inferred articulation (black curve) for participant 19, target $A_T = 2$ (*normal*), using Kinect v2. Colored straight lines represent the three possible target articulations in the study. In this case, the target articulation is represented by the yellow line ($A_T = 2$; *normal*).

the black curve and the yellow straight line. For a sequence of n frames it is computed as:

$$\epsilon = \frac{1}{n} \sum_{i=0}^{n-1} |A_T - A(t_i)|, \tag{5.10}$$

where A_T is the target articulation (1 for *legato*, 2 for *normal* and 3 for *staccato*) and $A(t_i)$ the inferred articulation in frame i.

Results

Twenty participants (7 female, 13 male) aged between 22 and 38 ($\mu = 29.7$, $\sigma = 4.2$) volunteered to participate. Half of them were musicians (considering musicians participants with any musical training) and the other half were not.

Analysis of articulation performance The questionnaire revealed that participants, in general, felt that they had fulfilled the task according to their answer to “Do you think you managed to fulfill the tasks asked during the study?” (μ=5.3; σ=0.9 in a scale from 1 to 7). They also replied positively to “When you were controlling the music, was the audiovisual feedback of your movement variations what you were expecting?” (μ=5.2; $\sigma = 0.8$). There was no significant difference between musicians and non-musicians in response to these questions.

The questionnaire also asked participants about which element(s) of the audiovisual feedback they focused on during the Task Phase. 9 out of the 10 musicians reported to have used both audio and visual feedback; the other one used audio feedback only. In the case of non musicians, 6 out of 10 reported to have focused on visual feedback only, while the other 4 used both audio and visual feedback.

We averaged the mean error across participants, devices and target articulations. The resulting global error is $\epsilon = 0.31$ ($\sigma = 0.21$). To compare with subjective measures, we computed the correlation coefficient between each participant's rating of their perception of task fulfillment and the mean accuracy values over all of that participant's performances. We found that subjective ratings and objective measure are correlated with a coefficient of 0.6.

We then inspect how the accuracy given by the mean error is affected by three factors: the Target articulation, the participants musical Expertise and the Device used for the task. A repeated-measure analysis of variance (ANOVA) showed that there is a significant effect of Target ($F_{(2,108)} = 3.992, p < 0.05$) and Expertise ($F_{(1,108)} = 7.264, p < 0.01$), while there is no effect of Device. A Tukey's HSD (Honestly Significant Difference) post-hoc analysis shows that the accuracy is higher for Target 2 and 3 compared to Target 1 ($p < 0.05$), while there is no significant difference between Target 2 and 3. For Expertise, the analysis shows that the accuracy is significantly better ($p < 0.01$) for musicians ($\epsilon_m = 0.26$, $\sigma = 0.19$) than for non-musicians ($\epsilon_{nm} = 0.36$, $\sigma = 0.24$).

Analysis of the model training From the Training Phase, we examine the quality of training by computing the separability S between articulations in the representation space (Figure 5.8, top-right), following the same method used in the observation study. ANOVA reveals that the Expertise does not affect separability, while the Device does ($F_{(1,36)} = 5.911, p < 0.05$). Also, we found that articulation separability is not correlated to model accuracy (correlation coefficient is 0.12). Figure 5.11 illustrates, for each participant (using the Kinect), the articulations in the representation space formed by v and a and the computed separability values.

We finally examine an important aspect of the gestures considered in the study: the idiosyncrasy of articulations performed by users. For that, we perform cross-validation on the training data: for each participant i, we train the model with the articulations from that participant and test with training data from the other participants $j = 1..20, j \neq i$. From these tests, we compute the average error between the estimated articulation value given by the model and the expected articulation. We found that the global error is $\epsilon_{idiosyn} = 0.80$. We then computed an "individual error"' by training the model and testing with the training data from the same participant i, for each participant. The global individual error is $\epsilon_{indiv} = 0.46$. A statistical test (t-test) shows that the two errors $\epsilon_{idiosyn}$ and ϵ_{indiv} are significantly different ($p < 0.001$).

5.3.3 Discussion

According to the questionnaire results, the model succeeds at providing sense of control over articulation. Also, objective measures provide acceptable errors for both kind of users (although significantly better for musicians). Importantly, the model managed to learn intended articulations even if the way participants performed the articulations to train the system embedded idiosyncratic elements that were not shared across participants. Indeed, while we imposed a particular base gesture (figure-eight) and tempo, users were not told how to vary their gesture to achieve the different articulations. Instead, variations in execution were free, but asked that they be coherent with the sound stimuli. As a result, a model learned on a user's set of data may not be transferable to another user, but embeds a given user's own expressive gesture qualities.

We saw that musicians performed significantly better than non musicians. Although no significant differences were found for the training quality for participants of different musical expertise, the differences during performance might be due to the musicians' better ability to better understand the task from a musical perspective. We think that their musical ability allows them to concentrate on dynamic variations and to better interpret the synthesized sonic differences representing *staccato* and *legato* articulations. This is supported by the fact that most non-musicians reported that they focused exclusively on visual feedback during performance.

Interestingly, the individual error of the model (ϵ_{indiv}), obtained when training and testing offline on a participant's data from the Training Phase (so considering the same data for the training and the testing) is higher than the average accuracy error of the model obtained from the Task Phase, ϵ (where participants trained the system and performed through it online). We believe that online task execution with audiovisual feedback involves an action-perception loop which helps users adapt their gesture to achieve the task. This could indicate that participants deliberately adapted to the outcome of the system during the task phase with unnatural gesture variations. However, the results from the questionnaire reveal that the audiovisual feedback was consistent with participants' expectations, indicating that this was not the case, and that the audiovisual feedback was a reinforcing confirmation to the user on her actions. Such aspects of sensorimotor learning that may enter into play constitute an important direction for future research.

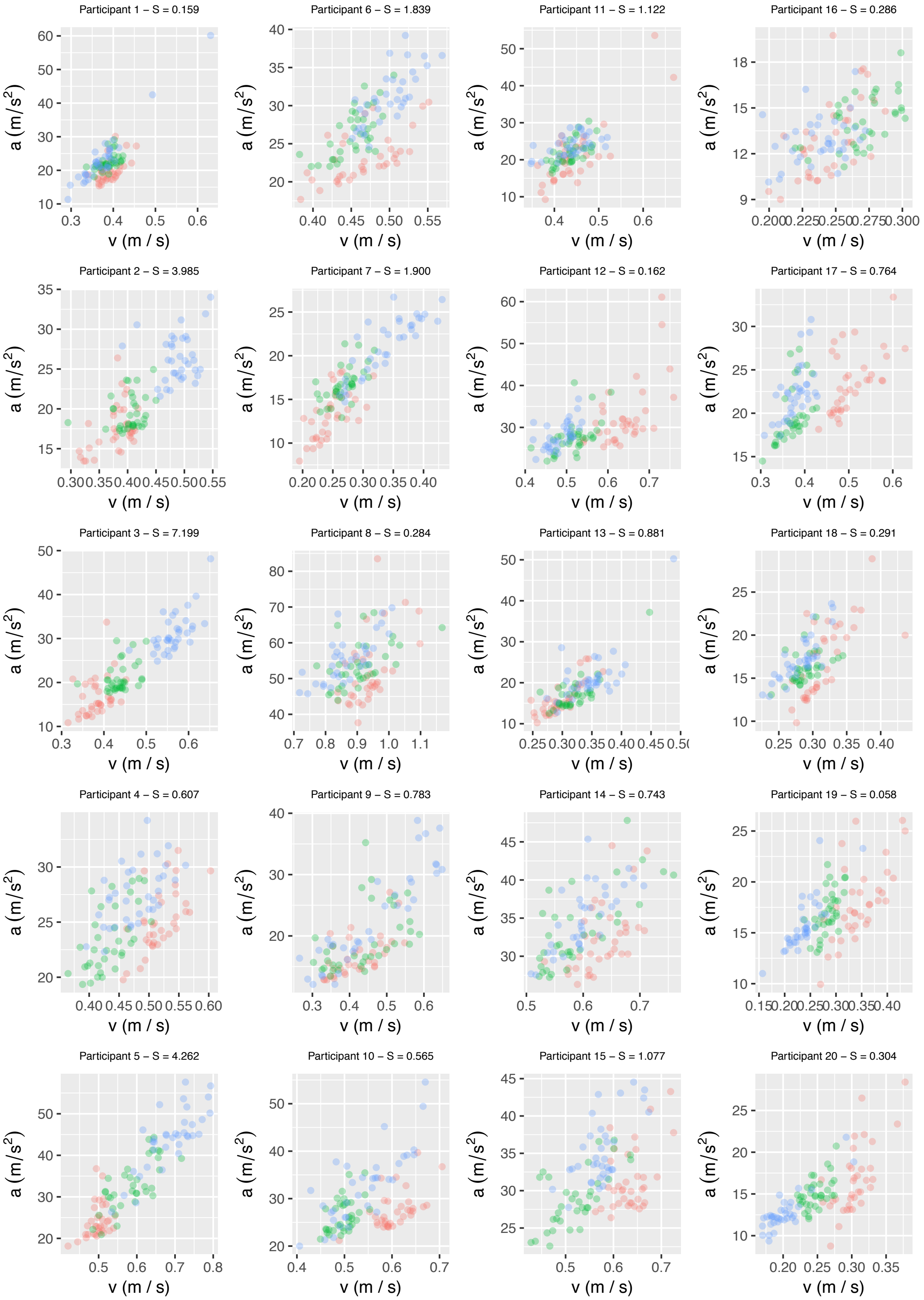

Participant 1 – S = 0.159
Participant 6 – S = 1.839
Participant 11 – S = 1.122
Participant 16 – S = 0.286
Participant 2 – S = 3.985
Participant 7 – S = 1.900
Participant 12 – S = 0.162
Participant 17 – S = 0.764
Participant 3 – S = 7.199
Participant 8 – S = 0.284
Participant 13 – S = 0.881
Participant 18 – S = 0.291
Participant 4 – S = 0.607
Participant 9 – S = 0.783
Participant 14 – S = 0.743
Participant 19 – S = 0.058
Participant 5 – S = 4.262
Participant 10 – S = 0.565
Participant 15 – S = 1.077
Participant 20 – S = 0.304

Figure 5.11: v and a values for each participant during the Training Phase. red = *legato*, green = *normal,* blue = *staccato*. Axes limits are set differently for each participant. Computed separability (S) values for each participant are also indicated.

5.4 Conclusions

We have analyzed the way in which subjects intuitively embody musical articulation when performing conducting gestures. For this, we performed a study with twenty four participants with different musical expertise. The study was divided in two parts: in the first one, participants performed gestures on top of melodies played with different articulations but did not get any indications; in the second one, they were asked to actually convey different articulations with expressive gesture variations. The results show that musical articulation affects gesture timing in both cases, with beats estimated from hand acceleration data falling closer to the beat with *staccato* articulation. Dynamic variations reflected in speed and acceleration computed from the hand movement are particularly noticeable when subjects actually try to convey different musical articulations. Interestingly, the results suggest that the variations introduced by each participant were idiosyncratic. The recordings of this observation study are available online[6].

Based on this, we proposed a model to control musical articulation where the user first listens to music played with different articulation and then explicitly tells the system how she embodies it through examples of gesture variations. We tested the model in a user study with twenty participants of different musical expertise. The results of the study show satisfactory results in terms of the ability of participants to control music articulation. Also, the results are in agreement with those from the observation study indicating that performed variations are idiosyncratic, which reinforces the need for user-specific mappings. Interestingly, we found that the quality of the trained models, computed as the separability of different articulations in the representation space, did not have a significant effect on the results during performance. This, combined with the fact that musicians achieved better performance, suggests that participants were able to adapt their performance in an action-perception loop during the execution of the tasks.

The proposed model has some limitations that are worth discussing. The scheme considers a single gesture at a fixed tempo, and the descriptors we used can be affected by changes in, for example, tempo, that should not affect articulation. The main strategies we foresee to address this issue are two. The first possible direction would be to build a representation space with descriptors that are more robust to these variations (for example, descriptors related to the shape of the gesture). Then, we believe that incorporating a temporal model of gesture could also be useful in this case. For example, combining this model with gesture recognition that tracks variation based on dynamical systems (as proposed by Caramiaux et al. (2014b)) could afford the user the possibility to train

[6] http://mtg.upf.edu/download/datasets/phenicx-conduct

the system to recognize different gestures and a set of potential variations which could then be dynamically explored, in performance, by the user. Likewise, the effects seen in the observational study regarding how gesture timing varies with articulation, combined with the results of the experiment in the previous chapter, suggest that articulation should also be considered as a factor that can interact with the beat detection from hand movement.

Chapter 6

Becoming the Maestro: a conducting game to enhance curiosity for classical music

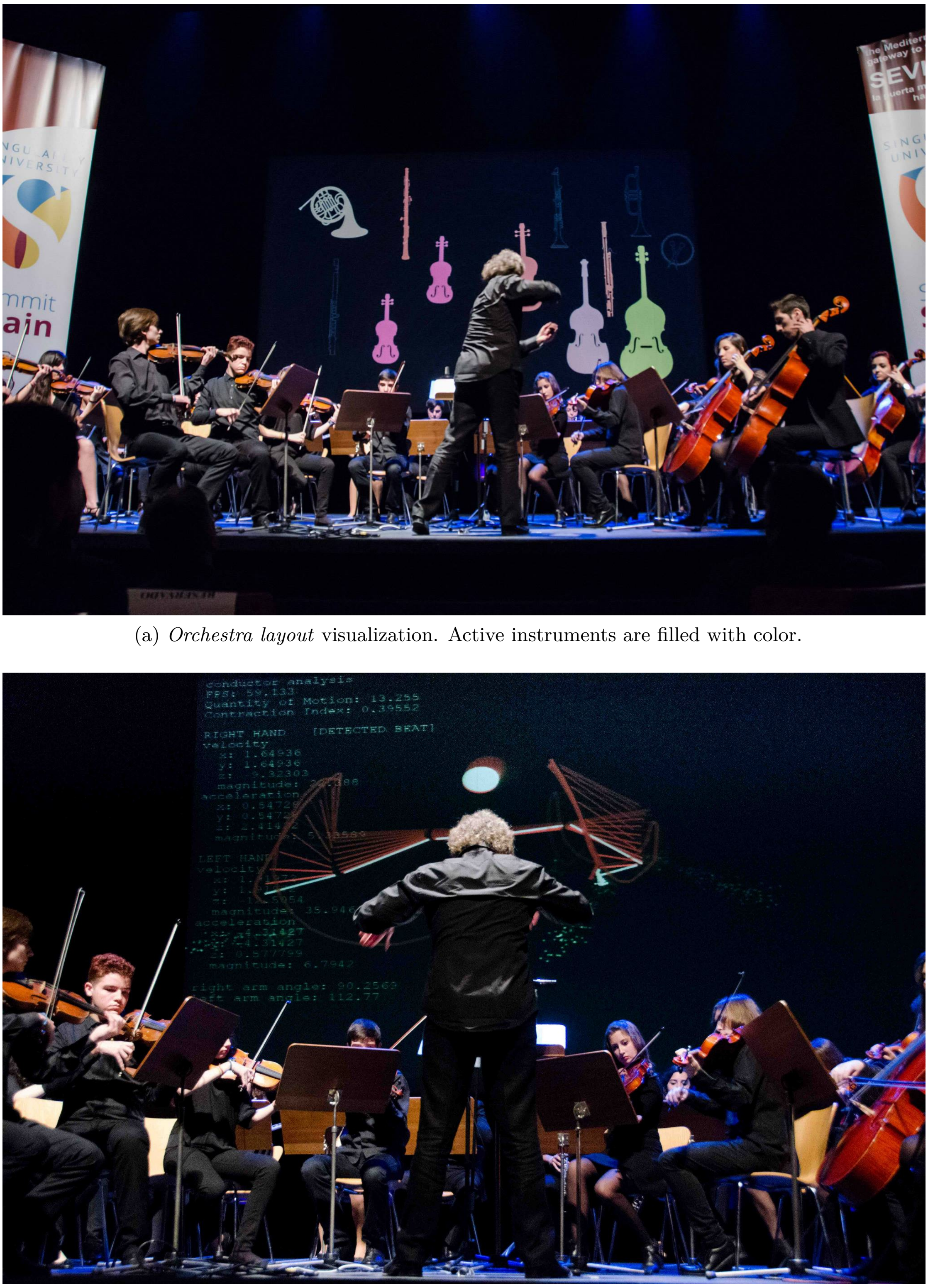
In preceding chapters we explored techniques to adapt or learn the mapping of DMIs based on the conductor metaphor to user-specific tendencies or expectations. Lessons learned in these chapters are useful in the case, for example, of installations in museums or, in general, contexts where it is desirable to provide an intuitive interaction that does not require too much training. Also, adapting to user's expectations is not only useful in terms of reducing the learning curve, but also in terms of allowing a more expressive control of the musical outcome.

These advances are aligned with the transversal goal of this thesis and the PHENICX project of finding means to attract new audiences to classical music through technology. However, we also wanted to pursue this goal in a more explicit way. In this context, we explore the potential of a game using the conductor metaphor to attract these new audiences. In this chapter, we present *Becoming the Maestro*, the game we developed with this objective.

6.1 Introduction

6.1.1 *Becoming the Maestro* in the context of PHENICX

Becoming the Maestro was developed as one of the demonstrators of PHENICX. In this project, state-of-the-art technologies were used to enrich the classical music concert experience *before, during* and *after* the performance. The game falls into the last category in this division: it is designed to be played *after* the concert.



(a) *Orchestra layout* visualization. Active instruments are filled with color.

(b) *Conductor* visualization.

Figure 6.1: *Exponential Prometheus* visualizations, during a performance in *Teatro de la Maestranza*, Seville (Spain) in March 12, 2015.

Also, in agreement with the general objectives of the project, the game tries to appeal classical music outsiders and boost their interest for this kind of music. Observation of and interviews with first-time attendees demonstrate that classical music novices often feel a lack of sense of belonging (Dobson and Pitts, 2011), whereas novices point out the importance of having a certain level of knowledge before they can enjoy the music (Dobson and Pitts, 2011; Kolb, 2000). A user-requirements study for classical music applications made within PHENICX by Melenhorst and Liem (2015) showed that creating experiences that provide opportunities for learning can be a successful strategy to motivate users for classical concerts, and that active, physical engagement with the music increases enjoyment. The game thus tackles the problem of appealing classical music outsiders by providing an experience that involves physical engagement and raises attention about specific aspects of the performance which might be unnoticed when attending one as audience.

Children have been the target audience of many educational games released by orchestras such as the Dallas Symphony Orchestra[1], the New York Philarmonic Orchestra[2] or the San Francisco Symphony[3]. Other systems we have already referred to in previous chapters also introduce game-like elements, particularly when targeted at children (Lee et al., 2004). There are also commercial video games that employ the conductor metaphor. Nintendo's *Wii Music*[4] includes different game modes, including one where the player controls the tempo of a virtual orchestra using the Wii remote controller as a baton. *Fantasia: Music Evolved*[5] by Harmonix for Xbox, uses Kinect and takes inspiration from the *Sorcerer's Apprentice* segment in Disney's 1940 classic *Fantasia*. Even though classical music played a key role in the film (in fact, it features music performed by the Philadelphia Orchestra conducted by Leopold Stokowski), the game focuses on pop music. Here, we did not want to limit ourselves to a game focused on children, and we designed the game following the aesthetics and using the technologies of other PHENICX project demonstrators.

6.1.2 Relation with other PHENICX demonstrators

The game can be played and enjoyed separately, but its design, coherent with other PHENICX demonstrators, is meant to make it more appealing when used as part of the

[1] http://www.dsokids.com
[2] http://www.nyphilkids.org
[3] http://www.sfskids.org
[4] http://wiimusic.com
[5] http://www.harmonixmusic.com/games/disney-fantasia-music-evolved/

whole *before, during* and *after* PHENICX concert experience.

Two demonstrators were developed to be enjoyed *during* the performance. The first one consists on a tablet/smartphone application to be used and controlled by a concert attendee, while the other consists on visualizations projected on stage. Both exploit real-time score following (Arzt et al., 2015) to show synchronized visualizations of different aspects of the piece or the performance, including the score, the structure of the piece (at different levels of detail), the instruments that are currently playing, etc.

Becoming the Maestro was designed to be coherent particularly with two visualizations from the second demonstrator[6] (projections on stage). All visualizations from this demonstrator were pictured by the same designer to have coherent aesthetics. The first of these visualizations used in *Becoming the Maestro* is the *Orchestra layout* (Figure 6.1a), in which different instrument sections in the orchestra are represented as silhouettes with the shapes of the instruments. The silhouettes are arranged on the screen with the same disposition of the orchestra on stage. Then, during performance, the silhouettes of instrumental sections which are playing are highlighted by filling the silhouette with color. The second one is the *Conductor* visualization (Figure 6.1b). It consists on a simplified torso of the conductor (head and arms only) enriched with information extracted in real-time using MoDe from the position of body joints tracked with a Microsoft Kinect. For example, every time a beat is detected in any of the two hands as a change from negative to positive sign in vertical velocity, the animation shows some particles coming from the hand with speed and direction related to those of the hand movement. Also, the skeleton leaves a trace whose intensity is proportional to the quantity of motion computed from all tracked joints. Next to the figure, the values of the involved descriptors (velocity and acceleration of both hands, quantity of motion, etc.) are shown as text strings in the background in order to reinforce the sensation of the visualization being computed in real time.

In this context, even though the game was developed such that it can be enjoyed separately, its design was highly influenced by the afore-mentioned visualizations. The reason for this was to make the game more appealing when played after attending a concert with those enriched visualizations: the user re-enjoys the same visualizations, this time interactively, by *Becoming the Maestro*.

[6] http://phenicx.upf.edu/SUPrometheus

6.2 Game description

In the game, the user employs her body position (mainly the position of her hands and arms in space) to interact with presented challenges using a Microsoft Kinect v2 as input device. A stick figure representing the user, similar to the one in the *Conductor* visualization from the PHENICX live demonstrator (Figure 6.1b), is displayed on the screen, mirrored to match the users' expectations. Its arms are used as cursors, and most interaction consists in making the projection of the arms intersect with circles that appear on screen, or moving hands following a particular pattern.

Two different tasks are presented in the game: **giving entrance to** the different **instrument sections** in the orchestra and **performing** the appropriate **conducting gestures with the right timing**. They appear separately in the first levels and combined when the player progresses to higher levels. Points are awarded (or lost) according to the player performance. Score events and the total score are shown on the screen, giving a performance cue to the user. Textual messages are also temporally displayed in order to give the sense of how well the user is doing, from praising messages ("Maestro!") to complaints ("Can't hear the Oboe!").

6.2.1 First task: giving entrance to instrument sections

This task is inspired in the mechanics of games like Guitar Hero[7], where the user has to perform guitar notes from a popular song, according to the visual annotations progressively appearing on the screen. The correct sound plays when the user plays the right notes in time; incorrect performance results in awkward sounds and silences. In this sense, the game needs to have separated stems from the performance (either from a multitrack recording or achieved through source separation) and annotations for the moments where each of the sections are playing. A snapshot of the game during this task is shown in Figure 6.2.

In *Becoming the Maestro*, the player has to indicate the entrance to the different instrument sections of the orchestra when needed (and displayed on the screen). As in the *Orchestra layout* visualization from the PHENICX live demonstrator (Figure 6.1a), the sections are represented by the instrument silhouettes spatially arranged as they would be in a real concert. Whenever a section has to be quiet, its image disappears. If the section is playing correctly, its shape is displayed *vibrating* at the tempo of the music and filled with a light color; if it should be playing but it is not, the shape is filled with

[7] https://www.guitarhero.com/

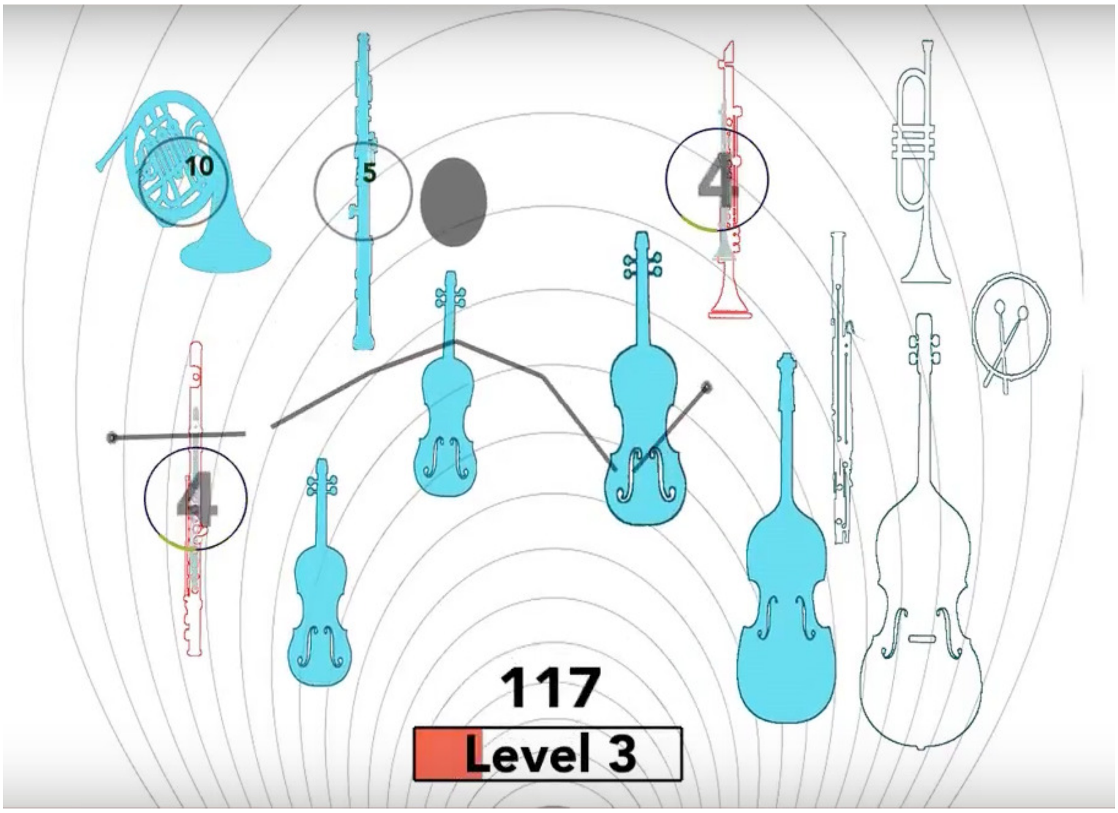


Figure 6.2: *Becoming the Maestro*; first task screenshot.

a dark color and it *vibrates* very fast.

A circle is displayed over the instrument section whenever an action is required from the user. A number of beats (dependent of the difficulty level) before the instrument section is supposed to play, a countdown starts showing the remaining number of beats until the entrance. The user can give the entrance to the section by *touching* this circle (i.e. making the projection of her forearms intersect with them). If she fails to do so in time, the separated audio of this section fails to play, resulting in an incomplete and awkward audio mix. The circle also will remain (this time pulsating) over the section until the user gives the entrance.

The number of instrument sections that the user needs to give entrance to depends on the difficulty level. The best score (20 points) is achieved when the user is able to give the entrance to the section just before it is supposed to play (i.e. 1 beat before). Lower scores are given if the entrance is given some beats before. Then, if the instrument should be playing but is not, the score is decreased by 1 point at each beat. Animations are shown on top of the corresponding silhouette every time a score event occurs.

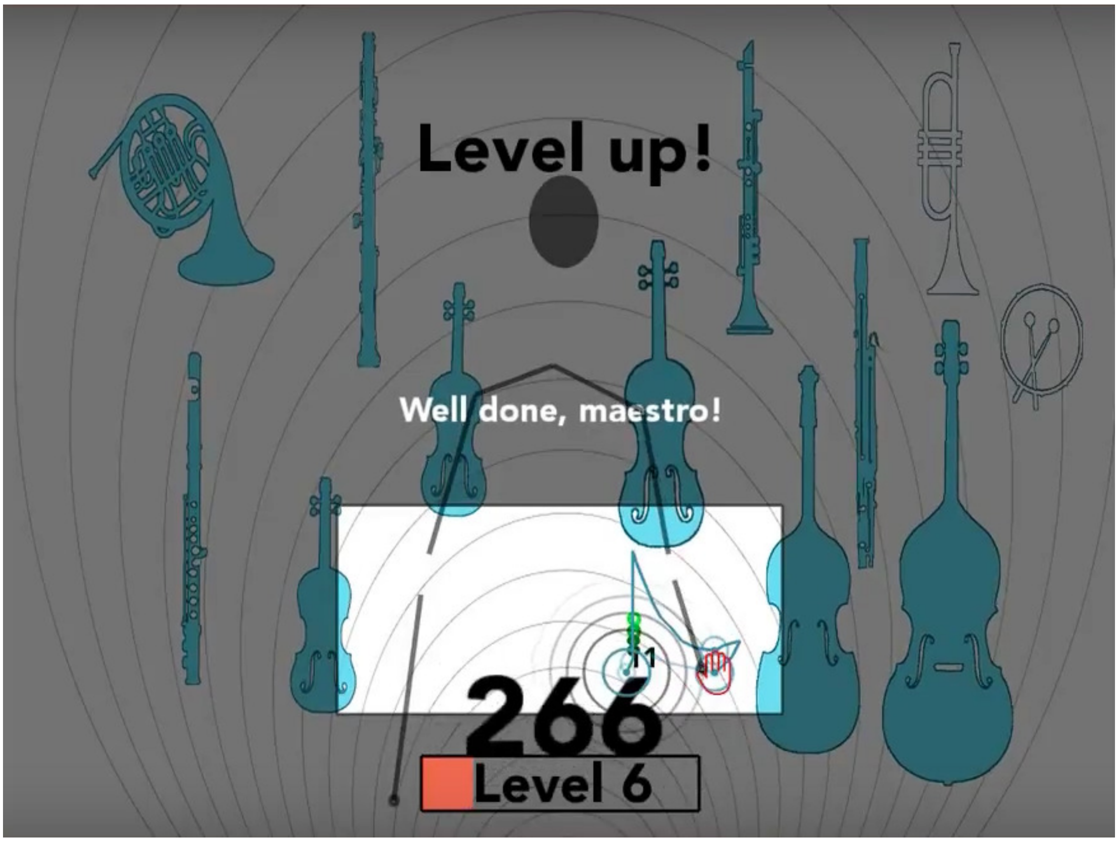


Figure 6.3: *Becoming the Maestro*; second task screenshot.

6.2.2 Second task: performing beat pattern gesture

In order to communicate rhythm information, conductors employ a set of gestures that depend on the meter of the music. For example, a 3/4 gesture appears in the screenshot of this task shown in Figure 6.3. This task challenges the player to perform these gestures with the correct timing. The gesture is introduced as a virtual hand that follows a trace in the screen. A trace is also drawn by the hand of the user, allowing her to check if the gestures match.

The performed gesture is evaluated and rated using the *Gesture Variation Follower*, a template-based real-time gesture recognition method based on particle filtering by Caramiaux et al. (2014b). This system can be trained with a single instance per gesture. Then, during gesture execution, it updates estimated parameters of the gesture. In this case, the procedure to rate gestures using this method is the following:

- First, the system is pre-trained with the gesture templates the player will have to match. For example, in the case of the implemented prototype, we used the same

3/4 gesture pattern we used for the study presented in Section 5.2 and shown in Figure 5.3a. During the game, it is displayed as in Figure 6.3). The 2-d position of 99 points in the line describing the gesture (33 between every two consecutive beats) was sampled and used as the template for training.

- Then, for every new MoCap frame during the game task, the projected 2-d position of the hand is used as input for the recognizer. The recognizer estimates the alignment in the gesture (between 0 and 1). The absolute difference between the estimated alignment and the time progression in the current bar is used to rate the correct timing of the gesture. As an example, if a frame arrives in the first beat of the current bar (0.33 progression), and the recognizer estimates that the current alignment of the performed gesture is 0.40, the resulting error at the frame is 0.07. Once the bar concludes, the cumulative error of all frames is mapped to the final score of the bar. The gesture scores go from -1 (bad gesture) to 20 (perfect gesture timing).

6.3 User evaluation

We evaluated a prototype of the game in order to assess its potential to increase engagement with classical music and get feedback for possible future improvements. Since the use case we have in mind is public installations, we evaluated the game in a setup where participants were not giving any indications on the game mechanics. This was done to investigate possible necessary improvements in the intuitiveness of the game interface. A picture from the user study is shown in Figure 6.4.

In this sense, the evaluation of this game is quite different to the evaluation of other prototypes presented in preceding chapters. We wanted to thoroughly assess the ability of the game to create a joyful experience and to enhance curiosity for classical music. In this context, we used some standard metrics for this purpose.

Venkatesh et al. (2003) formulated a model called Unified Theory of Acceptance and Use of Technology (UTAUT) that has been later extended in Venkatesh et al. (2012). The model provides a framework for systematically assessing user acceptance of new technologies and is commonly used to assess their probability to succeed. We used this framework to measure the fun of use and effort expectancy of *Becoming the Maestro*. Effort expectancy is defined as the degree of ease associated with the use of the system. We also used the *AttrakDiff2* questionnaire proposed by Hassenzahl (2008) to measure hedonic stimulation when playing the game.

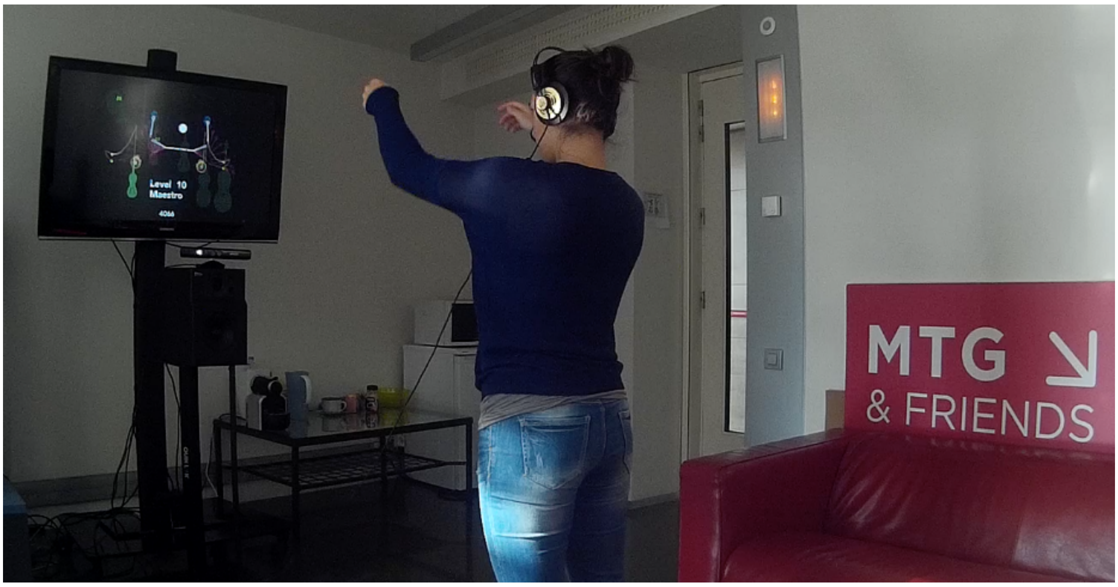


Figure 6.4: *Becoming the Maestro* demonstrator; participant in the user study playing the game.

6.3.1 Materials and methods

The game needs separate audio tracks for each of the instrumental sections and information about whether each section is playing on every bar. Both were available in a performance of Beethoven's 3rd Symphony (*Eroica*) by the Royal Concertgebouw Orchestra[8] that was recorded within the PHENICX project and is publicly available online[9]. Accordingly, the game prototype used in the evaluation was implemented using the 1st movement from this performance. We had individual stems for each of the instrument sections achieved with source separation (Miron et al., 2015), as well as aligned score (Miron et al., 2014). Both techniques (source separation and score alignment) can be used to adapt the game to any performance. In the case of source separation, to avoid needing multichannel recordings; in the case of score alignment, to minimize the manual work for annotation.

Participants were recruited via convenience sampling through department members and their students. They subsequently signed an informed consent form, filled out a brief pre-questionnaire, played the game prototype for nine minutes, and filled out the post-questionnaire. The 30-minute session was concluded with a brief interview.

The pre-questionnaire included the following questions:

[8] http://www.concertgebouworkest.nl/
[9] https://repovizz.upf.edu/phenicx/datasets/

- Attitude towards classical music: 9 items
- Behavioral intention to attend classical music concerts in the next three months: 2 items

In the post-questionnaire, the following questions were asked:

- *AttrakDiff2*, hedonic quality-stimulation (Hassenzahl, 2008): 7 semantic differential items
- Effort expectancy (Venkatesh et al., 2003): 4 UTAUT Likert items
- Fun of use (Venkatesh et al., 2012): 3 UTAUT Likert items
- Engagement with the game: 3 Likert items
- Expected impact on attitude towards classical music and classical music concert attendance: 3 Likert items

In the post-interview, questions were asked about aspects that appealed and did not appeal to the participants, suggestions for improvement, and the impact on the attitude towards classical music. We recorded the scores and levels achieved by each of the participants, as well as the time spent in each level, to test whether the ability to succeed at the game affects the participants attitude towards it.

6.3.2 Results

Twenty participants participated in the test, of which five were female. Ages were distributed as 21-25 (2), 26-30 (4), 31-35 (10), 36-40 (1) and $>$ 40 (3). Following Müllensiefen et al. (2014), classical concert attendance in the last year was measured on a seven-point scale ranging from 0 (coded as 1) to more than 11 times a year (coded as 7). Thus, values could range from 1 to 7. The average classical concert attendance score was 2.9 (s.d. 1.8). Years of musical instrument training was measured in a similar vein, ranging from 0 (coded as 1) to 10 years or more (coded as 7). Average musical training score was 4.0 (s.d. 2.3).

Attitude towards classical music

Attitudes towards classical music was measured with seven seven-point semantic differentials, and two seven-point Likert scale items. Results are shown in Table 6.1.

In general, apart from learnability, participants had a relatively positive attitude towards classical music. Important to note is the potential of classical music for mood regulation, as evidenced from the results on the items about escapism, coming in the right mood, and

Table 6.1: Attitude towards classical music. Seven-point semantic differentials and Likert scale items.

Semantic differentials	μ	σ
Beautiful - Awful	1.9	.9
Stimulating - Boring	2.6	1.2
Tiresome - Relaxing	5.6	1.2
Hard to understand - Easy to understand	3.9	1.3
Easy to enjoy - Hard to enjoy	2.8	1.2
Easy to learn - Hard to learn	5.1	1.4
Something to enjoy with others - Something to enjoy alone	4.8	1.2

Likert items	μ	σ
Classical music helps me to escape from my daily worries	4.2	1.5
Classical music helps me to come in the right mood	4.8	1.1

the tiresome-relaxing continuum. These participant characteristics and their attitudes suggest that the participants can be considered as classical music outsiders, in the sense that they are not regular consumers of this kind of music, but with a general positive attitude towards it. Their answers also suggest that they identify the genre as more prone to be enjoyed individually.

Hedonic quality, fun of use, and effort expectancy

As a first step in the analysis of the dependent variables, reliability analyses were conducted on the scales that were administered. The Cronbach's alpha levels represent the internal consistency of the scales that were used. The alpha levels for hedonic quality (stimulation), fun of use, and effort expectancy demonstrated good internal consistency (alpha $>$ .85).

Scale means and standard deviations for hedonic quality, fun of use and effort expectancy are displayed in Table 6.2.

Table 6.2: Hedonic quality, fun of use, effort expectancy.

	μ	σ
Hedonic quality (stimulation) [1 to 7]	4.3	.6
Fun of use [1 to 5]	4.2	.7
Effort expectancy [1 to 5]	3.4	.9

Hedonic simulation was measured with *AttrakDiff2* (Hassenzahl, 2008) 7 semantic differential items with values ranging from 1 to 7. The average results for each of the elements in the questionnaire is shown in Table 6.3. An average of 4.3 suggests that the game was moderately stimulating.

Table 6.3: Hedonic quality measurement.

	μ	σ
Typical - Original	4.4	.9
Standard - Creative	4.4	.9
Cautious - Courageous	4.1	.7
Conservative - Innovative	4.4	.8
Lame - Exciting	4.3	.9
Harmless - Challenging	4.4	.7
Commonplace - New	4.2	.9

Fun of use was measured following Venkatesh et al. (2012) with three five-point Likert scale items with values ranging from 1 to 5. The average results for each element are shown in Table 6.4. The average of 4.2 shows that participants liked playing *Becoming the Maestro*. The results for fun of use were confirmed by the results for engagement, which are shown in Table 6.5. As can be seen from the table, participants in general would like to continue playing the game at home. Additionally, the results suggest that the game succeeds in appealing to the user's motivation to engage in competition.

Table 6.4: Fun of use measurement.

	μ	σ
Playing *Becoming the Maestro* is fun	4.3	.9
Playing *Becoming the Maestro* is enjoyable	4.2	.8
Playing *Becoming the Maestro* is very entertaining	4.0	.7

Table 6.5: Engagement with the game measurement.

	μ	σ
I would like to continue playing *Becoming the Maestro* at home	3.7	1.2
I expect *Becoming the Maestro* to become boring after a while	3.0	1.0
Becoming the Maestro motivates me to achieve the highest score possible	4.0	1.1

Effort expectancy was measured with four five-point Likert scale items with values ranging from 1 to 5, as proposed by Venkatesh et al. (2003). Table 6.6 shows the average results for each of the items. Even though participants were in most cases not

Table 6.6: Effort expectancy measurement.

	μ	σ
Learning how to play *Becoming the Maestro* is easy for me.	3.5	1.0
My interaction with *Becoming the Maestro* is clear and understandable	3.4	1.2
I find *Becoming the Maestro* easy to use	3.5	1.1
It is easy for me to become skillful at playing *Becoming the Maestro*	3.2	1.2

usual consumer of classical music, the effort expectancy levels are still close to the scale average of 3. Further inspection of the data reveals that the items on the learnability ($\mu = 3.5, \sigma = 1.0$), ease of use ($\mu = 3.5, \sigma = 1.1$), and the interaction design ($\mu = 3.4, \sigma = 1.2$) yielded positive values around 3.5, while the item "It is easy for me to become skillful at playing *Becoming the Maestro*" yielded an average of 3.2 ($\sigma = 1.2$). When we relate this result to the high joy of use measures, we can conclude that the tasks the users had to do were difficult enough to keep them engaged, but not so difficult that it compromised the joy of use they experienced. At the same time, the results also suggest that, at least for some participants, some tasks might be difficult to understand. Some participants did in fact raise this point during the interview. In particular, some took some time to understand that different points were awarded depending on when the entrance to the instrument was given during Task 1.

During the interview, participants usually raised the educational aspect of the game as its most interesting aspect. They appreciated how they could realize the effect of not giving the entrance to a section in the resulting audio, thus helping them to be more conscious about what different sections in the orchestra are playing. More related to the second task, participants also liked the way in which the game taught them how to perform the 3/4 gesture. That part was specially challenging for some participants, who often commented how the fast tempo of the song made it more difficult.

Independent samples t-test revealed no significant effects of gender. A one-way ANOVA test was used to analyze the effect of age. No significant effect of age was found on any of the scales either.

No significant correlations were found between the measured variables and game performance (i.e. final score, level achieved, time spent in levels).

Correlations with musical training and classical concert attendance

Engagement with classical music was expected to impact the perception of the game. To assess this hypothesis, we computed correlations between on the one hand fun of use, hedonic stimulation, effort expectancy, and on the other hand musical training and classical concert attendance. The results revealed however no significant correlations.

Estimated motivational impact on attitude and concert attendance

Participants were asked to report on the effects of playing the game, in terms of their attitude towards classical music, and their intention to attend a concert in the near future. Three five-point scale Likert items were used for this purpose. The results are shown in Table 6.7.

Table 6.7: Estimated motivational impact on attitude and concert attendance.

	μ	σ
Becoming the Maestro makes me curious about the piece (Beethoven's 3$^{\text{rd}}$ Symphony)	3.6	.9
Becoming the Maestro makes me more enthusiastic about classical music in general	3.8	1.0
Becoming the Maestro motivates me to attend a classical music concert in the following 3 months	3.1	1.0

The averages above the neutral point in the scale point to the potential for the game to increase engagement with classical music. The game increases curiosity, and enthusiasm for classical music in general. In contrast, participants responded neutrally on the impact of the game on the intention to attend a classical concert.

No significant correlations were found between the three items on participants' estimated motivational impact and game performance (i.e. final score, level achieved, time spent in levels).

6.3.3 Discussion

The evaluation shows the potential of the game to engage people who are not regular consumers of classical music but show a moderately positive attitude towards the genre. For this group of users, the game succeeds at offering an engaging experience and increasing their curiosity for classical music. Although it did not have an immediate impact on their intention to attend a concert in the near future, the positive attitude of the players towards the game makes us believe that it could have an impact on classical music consumption in the long run, in the same way Guitar Hero increased the sales of records by older artists appearing in the game[10].

During the interview, participants often mentioned how the game made them pay more attention to the role of the different sections in the orchestra, which indicates that the game gave them a new way of looking at orchestral music and a way to appreciate and better understand one of the roles of the conductor. Following this idea, it seems that it would be positive to include more educational aspects in the game.

6.4 Conclusions

In this Chapter we have presented *Becoming the Maestro*, the game developed within the PHENICX project to attract new audiences to classical music. In the game, the player takes the role of a conductor in a concert. The game does not try to realistically replicate the role of the conductor during performance, but to provide the user with an entertaining and joyful experience. As we have seen, the game is greatly inspired in other PHENICX demonstrators and prototypes, as it is coherent with the whole *before, during* and *after* concert experience envisioned in the project. It is, in any case, designed so that it can also be played and enjoyed separately.

In this part of the work we have thus moved from exploring motion-sound mappings in the context of DMIs based on the conductor metaphor to using this metaphor in a gaming scenario. As explained throughout the Chapter, the motivation for this was to explicitly pursue the transversal goal of attracting new audiences to classical music, which in the rest of the chapters was present implicitly. Following this motivation, we have evaluated the game precisely in terms of its ability to create a joyful experience and to enhance curiosity for classical music. The evaluation was done with participants who are not regular consumers of classical music but show a moderately positive attitude towards the genre. For them, the game succeeds at providing this joyful experience, and it also has a positive impact in terms of engaging their curiosity for classical music.

A pending challenge is to evaluate the game with groups of users more reluctant to classical music, as well as in certain age groups such as children. After the evaluation of the prototype presented in this chapter, the game has been presented in public venues, and our feeling is that its acceptance has always been positive. Something that we have observed in these presentations is that children are often those who show greater interest in interacting with the game, and are also those who, generally, obtain higher scores. In

[10] Cultural impact of the Guitar Hero series, in Wikipedia: https://en.wikipedia.org/wiki/Cultural_impact_of_the_Guitar_Hero_series

any case, it is still necessary to evaluate the game with groups of different profiles and to study if it would be appropriate to adapt it depending on the venue.

Participants in the study also indicated that it would be interesting to find levels where they take control of other aspects of the music. According to this idea, one possibility to extend the game is to take advantage of the prototypes for beat, dynamics and articulation control and *gamify* the tasks used for their evaluation turning them into challenges within the game.

Chapter 7

Conclusions

Throughout this thesis we have made an in-depth study of interaction with systems based on the conductor metaphor. The proposed approach for this analysis has been to focus on the very fact that motivates the use of interface metaphors: providing the user with a control interface to which she can transfer her knowledge of a real world activity; in this case, conducting. We hypothesized that part of the knowledge that users have from the domain that the interface metaphor replicates is user-specific and that this can be exploited for DMI mapping design. As we have seen in our literature review, current trends in motion-sound mapping, which place the user at the center of the mapping design, have practically not been used in DMIs based on the conductor metaphor when, in fact, it is a very suitable use case for this.

In this context, we have dealt with the problem from different perspectives. In the first place, we have analyzed the performance of a conductor in order to identify to what extent this analysis can inform the design of instruments based on this metaphor. Then, using the methods developed for this analysis, we have performed observational studies with different participants to observe whether there are indeed user-specific tendencies in terms of how they embody the beat, loudness variations or articulation. Based on the observed differences, we have proposed specific strategies to design the mapping of DMIs that adapt to each user and we have performed experiments and user studies to evaluate their usability. At the same time, there is a transversal objective to the thesis which is to use this knowledge to develop applications that can attract new audiences to classical music. This goal has been explicitly pursued with the development of *Becoming the Maestro*, a game focused on these potential new audiences. We evaluated a prototype of the game precisely in terms of its ability to create a joyful activity that enhances interest in this type of music.

The development of these areas has resulted in a number of concrete contributions. In the following sections, we discuss these contributions, their limitations, and possible directions for future work in this area.

7.1 Contribution 1: Kinect for long, multimodal recordings

Devices such as Kinect are suitable for making recordings in a non-obstructive way. However, there are some drawbacks that may arise when aligning data recorded with this device to other data streams. In our recordings of rehearsals and performances we noted that one of these problems is that some frames (from both video and MoCap) can be dropped, especially in long recordings. To counter this effect, we have developed an application that annotates the frames in which this occurs and generates the missing frames by applying linear interpolation in the case of MoCap and repeating the previous frame to the loss in the video case. This application, *KinectVizz,* records aligned MoCap, audio and video, and allows to directly export the data in a format compatible with the *Repovizz* platform. Following the needs from our own observation studies, the application also allows to load an audio file and play it while recording aligned data from the Kinect. *KinectVizz* is available online at https://github.com/asarasua/KinectVizz.

7.1.1 Limitations and future work

Regenerating lost frames by linear interpolation can result in unrealistic body movement reconstructions. One way to perform better reconstructions would be to consider kinetic models that impose rules on the relative position of joints, or to consider the dynamics of movement before and after the part to be reconstructed to infer the most adequate reconstruction. There are available solutions for MoCap data post-processing that could also be used for this purpose, such as *MotionBuilder*[1].

7.2 Contribution 2: Library for real-time MoCap feature extraction

During this work we implemented and released MoDe, an open-source C++ library for real-time MoCap feature extraction. It computes differentiation from positional data using nearly optimal filters proposed by Skogstad et al. (2013), and has some features that make it suitable for creative applications using MoCap. It can be compiled as an OpenFrameworks[2] addon and is compatible with any MoCap device that provides

[1] http://www.autodesk.com/education/free-software/motionbuilder

[2] http://openframeworks.cc/

positional data. In addition, its API includes functionality to handle temporal events from MoCap features, such as local maxima and minima or zero crossings.

The library is publicly available under a LGPL License online at

https://github.com/asarasua/MoDe

Its architecture and API are briefly explained with more detail in Annex C.

7.2.1 Limitations and future work

The library allows to extract a number of descriptors from positional data. In the future, it is desirable to include features computed from orientation data, as well as to include the ability to compute features from other kinds of devices (e.g. accelerometers, gyroscopes). In addition, we have plans for future updates in the library to allow the creation of new features through the API, as combination of the existing ones.

7.3 Contribution 3: Dataset of conductor movement during performance

Using the aforementioned software, we recorded a live performance of Beethoven's 9th Symphony played by the *Orquestra Simfònica del Vallès*. It includes audio from 32 microphones, 2 video streams (1 of the whole orchestra, 1 of the conductor captured by the Kinect), MoCap data of the conductor captured by the Kinect, and aligned score. This dataset is, as far as we are aware of, the first public one to include MoCap data from the conductor in an actual performance.

This recording is publicly available online, through the *Repovizz* platform, at

http://mtg.upf.edu/download/datasets/phenicx-conduct.

7.3.1 Limitations and future work

Even though the recording is publicly available, we still need to devote specific effort to draw the attention of the academic community to it and propose concrete ways in which the data can be utilized, also providing easier ways to access the data beyond visualizing and downloading it from *Repovizz*.

7.4 Contribution 4: Further understanding of conductor-orchestra interaction

In our literature review, we have seen numerous works that computationally analyze the interaction between the conductor and the orchestra. In all cases, however, this analysis is done in controlled environments and focusing on some particular aspect, such as the synchronization of the orchestra with the movements of the conductor. In our case, since the analysis is motivated by the activity that users will replicate in conducting systems, we wanted to take this analysis to a real performance. This, of course, implies some problems, since there are no variables under control and we must limit ourselves to observing possible relations between conductor's movements and music.

For this reason, we wanted to approach the analysis knowing those specific aspects on which we could expect some correlation between the conductor's movement and the resulting music. From an interview with professional conductors and students, we concluded that the possible aspects to be analyzed were the communication of tempo, dynamics and articulation; being aware that the conductor is not necessarily constantly conveying information of these parameters.

Accordingly, we performed an analysis of the afore-mentioned recording focusing on repetitions of the same motif (the *Ode to Joy* from the 4th movement in Beethoven's 9th symphony) that appears with different variations throughout the piece. The main conclusions of the analysis, presented in Chapter 3, were the following:

- The descriptors that best capture the synchronization between the movements of the conductor's right hand and the music beat are those obtained from vertical movement.
- The lag observed between different descriptors computed from hand movement and musical beat varies across different fragments. This suggests that automatic beat estimation from the conductor's movement must incorporate contextual information.
- As suggested in the interview, it is not always possible to find correlations between conductor's movements and music.
- In terms of loudness, when this correlation exists, the quantity of motion seems to be the feature that best describes it.

The scores of the analyzed excerpt are included in the online repository. The complete results for beat analysis of each excerpt can be found in Annex B.

7.4.1 Limitations and future work

The first limitation of our analysis to keep in mind is that it is focused on a specific performance, with one conductor and one orchestra. In this sense, works that seek to approximate a general model of the director-orchestra interaction, should do so from data that represent greater variability. Focusing on specific aspects of our analysis, there are also some limitations that must be mentioned.

First, although it had appeared in the interview as one of the possible aspects to analyze, we have not dealt with articulation. The reason is that it is difficult to find similar fragments in the symphony where articulation varies. In this respect, one aspect to be taken into account when planning new recordings for this type of analysis is to, as far as possible, select the repertoire according to what is going to be observed.

Second, regarding the observation that the lag between different descriptors computed from hand movement and musical beat varies across different fragments, we have not explored the possible mechanisms that may help to predict this effect. This is, in fact, an area open to future research: is there any way to determine, from the context, how beat must be predicted from motion? The *context* in this sense can be the musical one, i.e. the instruments that are playing at that moment, the current tempo, the possible changes of tempo that are going to occur, etc. But the context can also refer to how the conductor's body is moving, i.e. the speed at which the arms are moving, the amplitude of the gesture, etc. We believe that a more detailed analysis that takes into account these factors can help to better understand the observed phenomenon.

Third, in our analysis relative to loudness we have used audio descriptors as ground truth. The loudness is however a complex perceptual phenomenon, particularly in classical music. For example, in our analysis we have observed parts where the height at which the director placed his hands was correlated with loudness. However, it is possible that this effect is due to the fact that, in those fragments, the conductor was giving indications to the choir, placed in the back part of the orchestra. In this context, we believe that potentially interesting analyses can be made using other information than the loudness extracted from audio. For example, having aligned score, it is possible to know how many musical sections are playing at any given moment. Also, scores usually include dynamic annotations on how they should be played (e.g. *fortissimo*, *pianissimo*).

The shared dataset is a good resource for such analyses.

In any case, a general caution that must be taken when analyzing a performance is that it is very difficult to determine whether the observed correlations involve causality. We are not here discussing that the conductor actually controls the orchestra during the performance. What we are referring to is that, according to our observations and suggested by the conductors in the interview, actions performed by conductors during performance are influenced by a high number of factors difficult to determine. Something that we have observed during this project, although it has not been part of the presented computational analysis, is that the aspects that conductors emphasize during pre-concert rehearsals are also emphasized during the concert. For example, when a conductor asks to repeatedly rehearse a part where there is something he does not like, it is more likely that in the concert he will emphasize the requested correction with his gesture. Also, regarding the loudness analysis, one of the indications we got in the interview with conductors was that sometimes a conductor may make an indication (e.g. play *forte*) at a point, not needing to give that indication again for the time that indication applies (and as long as the orchestra plays according to the conductor intention). We believe in this sense that both ideas suggest that a potentially useful approach for this kind of analysis might focus on these *key points* where concrete significant actions occur.

In accordance with these ideas, and also based on the outcomes and conclusions from other parts of the thesis, we believe that in terms of DMI mapping design it is better to focus on the final user than trying to transfer knowledge from the analysis of conducting performances. It is highly complicated to find causal relationships between conductor movements and music in an uncontrolled environment. Also, what users of a conducting system will do will always be very different from what a conductor does at the concert.

Finally, we would like to point out that conductor-orchestra communication is not confined solely to body gestures. Gaze and facial gestures also play an important role in expressive communication. Future works that deal with this aspect might refer to Poggi (2002), who describes a "lexicon of the conductor's face" including gaze, head movement and facial expression gestures.

7.5 Contribution 5: Analysis of user-specific tendencies in conducting movements

We hypothesize that part of the knowledge that users have from the domain that the conductor interface metaphor replicates is user-specific, and that this is reflected in user-specific tendencies in spontaneous (i.e. without instructions) conducting movements. In order to corroborate this hypothesis, we have performed observational studies where we have asked different participants to perform conducting movements and we have analyzed whether there are indeed differences across participants.

In our first study, presented in the first half of Chapter 4, we have focused on differences in beat and loudness. We asked participants to perform spontaneous conducting movements on top of different audio fragments. Regarding the beat, we analyzed whether there were user-specific tendencies to anticipate or fall behind the beat. For loudness, we analyzed the MoCap descriptors that were most correlated with it for each participant. The main conclusions of the analysis were the following:

- Users show different tendencies in terms of anticipation to the musical beat. Beats estimated from hand movement appear totally synchronized with the beat for some users, while for others they tend to appear sooner or later.
- In the case of some participants, no particular tendency is observed. In our study, this was the case of participants who indicated some issue (not having recognized the measure of the piece, not having been able to memorize the fragment to anticipate the changes...) or of those who stated that they had ignored tempo during the recording.
- Regarding loudness variations, there are certain general tendencies. For example, there are some participants who tend to move more energetically in louder parts (as suggested by a strong correlation between loudness and quantity of motion) while others tend to raise their hands higher (as suggested by a strong correlation between loudness and the maximum hand height).
- Even though there are general trends, there are also user-specific tendencies which are particularly reflected in different "dynamic ranges". This is, even if two users show the same correlation between quantity of motion and loudness, one might show wider differences between soft and loud parts than the other.

This analysis has resulted in two peer-reviewed conference papers (Sarasúa and Guaus, 2014a,b).

In the second study, presented in the first half of Chapter 5, we have dealt with differences related to musical articulation. In this study, we asked participants to perform specific gestures on top of melodies synthesized with different articulation in order to study how articulation is reflected in gesture variations. The study included two parts: in the first one, participants were asked to perform the gestures on top of these melodies with different articulation, but were not given indications on whether they should vary their gesture according to the variations in sound; in the second one, they were specifically asked to convey the perceived sonic differences through dynamic gesture variations (half of the participants on top of the melodies, the other half on top of a metronome). In this case, we studied whether beats detected from hand movement appear at a different distance from the musical beat depending on the articulation, and whether articulation affects gesture dynamics reflected in hand movement velocity and acceleration. The conclusions of the study were the following:

- Different articulations cause different tendencies in terms of distance beats detected from hand movement and musical beat, even if the user does not intend to convey articulation through gesture. More concretely, beats estimated from hand movement acceleration appear closer to the beat with *staccato* articulation. This suggests that the greater clarity of the beat in *staccato* helps to perform the gesture in synchrony with the beat.
- These differences increase when users try to convey articulation through gesture variation. This suggests that dynamic gesture variations also affect beat detection from hand motion. At the same time, the fact that these differences are smaller in the case of the participants who did the gestures on top of metronome instead of the melodies with different articulations indicates that the main mechanism causing the differences is the one mentioned previously (greater beat clarity for *staccato*).
- No dynamic variations are observed in gestures performed on melodies synthesized with different articulations when the user does not try to convey these articulations.
- Gestures performed trying to convey different musical articulation present significant differences in dynamics, reflected in hand velocity and acceleration. In this case, differences were not greater for participants who performed the gestures on top of the melodies and those who performed them on top of the metronome. What this suggests is that, in a non-interactive context where the user knows that her movements are not affecting the resulting sound, a sound consistent with the variation transmitted by the gesture does not necessarily reinforce the user in her actions.
- Dynamical variations observed in gestures trying to convey different articulations are idiosyncratic, i.e. specific to each user.

The data from both studies are available online at

http://mtg.upf.edu/download/datasets/phenicx-conduct

7.5.1 Limitations and future work

The analysis of these studies has been done with the eyes on the applications for which we made them. In this sense, we have focused on observing and identifying differences between users that are potentially useful in an interactive context. Accordingly, we have not analyzed in depth the causes for the observed effects.

For example, regarding observed differences in beat anticipation, we have limited ourselves to verifying and quantifying these differences and we have not studied their possible causes. In our analysis, we have not attempted to relate these differences in beat anticipation to other aspects of movement that can be reflected in different descriptors (e.g. we did not check whether the size of the gesture affects the detected beat location). Also, we have looked for general tendencies for each user. However, the fact that in the second study we have seen that articulation affects beat anticipation, reinforces the idea that this tendency is not constant for each user. This is an interesting direction for future research: studying beat embodiment from a dynamic point of view, not assuming a constant tendency for each user, but evaluating if there are changes and what mechanisms cause them.

We used loudness computed from audio as ground truth for our loudness analysis. However, there are different aspects that affect perceived loudness. In the case of the professional conductor we mentioned that some observed correlations could be due to the arrangement of the orchestra on the stage. This may also be happening here; in this case, with the disposition that each person imagines when listening to the fragment on top of which she conducts. In our study, we left this aside and did not ask participants about it. A possible direction for future research in this area would be to conduct similar studies where participants conduct not only "following" audio but also video. It is very probable that the movements in this case were different, since the user would be able to see the physical layout of the instrumental sections in the orchestra, identifying which of them are playing in each moment.

Regarding dynamic articulation variations, we focused on two gestures (4/4 and 3/4) and a fixed tempo. The descriptors we used for the representation space were valid for this scenario, but are however likely to be affected by gesture variations related to other parameters such as tempo and dynamics. In this sense, other possible more robust representation spaces might use descriptors related to, for example, the shape of the gesture or its temporal evolution.

7.6 Contribution 6: Adaptation to user-specific tendencies by observation

We hypothesized that the conclusions of observational studies with movements performed on top of fixed music have applicability in an interactive context. This, however, is not trivial. As opposed to the case of the studies, where the user is aware that her movements do not affect the result, during interaction there is an action-perception loop through which the user adapts her actions according to the perceived effects that they have. In this context, it is necessary to test, for example, whether a user that tends to anticipate the beat on top of fixed music will prefer a system that compensates for this.

Following this idea, in the second half of Chapter 4 we have proposed a DMI based on the conductor metaphor that allows to control beat and loudness in a way similar to those most commonly found in previous systems, but adapting to user-specific tendencies. We refer to the strategy that the system uses to adapt to user-specific tendencies as *Mapping by Observation*. The basic idea is that the mapping is predefined, but some parameter tuning is done for each user based on analysis of spontaneous movements similar to those from the observational study. That is, the user does not consciously and explicitly train her own mapping. We performed an experiment where we compared the proposed approach with a baseline that does not adapt its parameters for each user, confirming that this strategy serves to create a more intuitive mapping. We believe this is particularly relevant in relation to beat control. The proposed system incorporates a strategy to estimate the beat before there is a change from downward to upward hand movement. This, even without considering user-specific tendencies, is relevant for systems that use input devices with a low frame rate and non-optimal latencies such as Kinect.

There is another relevant conclusion from the experiment. From the analysis of spontaneous movements that we use to adjust the parameters of the mapping, we obtain measures on the reliability of such analysis (the F-measure for the tendency to anticipate the beat and the R^2_{adj} of the regression model for loudness). In the experiment, these measures are correlated with the improvement that the proposed system introduces with respect to the baseline, which suggests that we can estimate in advance if the adjustment introduced by the system will actually improve its usability.

The proposed system and the experiment for its evaluation have lead to a journal paper in preparation at the moment of deposit of this dissertation (Sarasúa et al., 2017).

7.6.1 Limitations and future work

The proposed strategy of adapting mapping to each user observing their spontaneous movements has only been tested in the concrete scenario of conducting. We believe this is a good use case for this type of research. However, it would be interesting to see to what extent the proposed strategy can be extrapolated to other similar cases. For example, Jensenius (2007) performed observation studies looking for "music-movement correspondences" in three different activities: air instrument performance, free dance to music, and sound-tracing. While these studies showed that users with different musical expertise seem to associate similar body movement with features in the musical sound, some user-specific tendencies also appeared. Following this idea, we believe that the *mapping by observation* scheme can be explored in these cases. Just as we have analyzed spontaneous conducting movements to build user-specific mappings, a similar use could be explored in these other activities that can serve as an interface metaphor in a DMI.

In our case, the proposed implementation is focused on our concrete use case. We have some predetermined rules that are adapted, after observation, to each user. However, this makes the implementation not directly applicable to other use cases. We believe that supervised learning techniques are suitable for a more generalizable approach to this *mapping by observation* scheme. In fact, as we stated when introducing this approach, the idea is very similar to *play-along mapping* as proposed by Fiebrink and Cook (2009), the difference being that in our case the user does not explicitly and consciously *teach* the system the desired mapping. In any case, supervised machine learning techniques as those proposed by Fiebrink and Cook could be applied in scenarios similar to ours.

7.7 Contribution 7: Articulation control through idiosyncratic gesture variation

In the second half of Chapter 5 we propose a system for control of articulation through idiosyncratic gesture variations based on the observations from the observation study. The proposed approach follows the principles of *Mapping by Demonstration* as defined by Françoise (2015): listening is the first step in the mapping design, and the mapping is explicitly defined by the user through gesture variation examples performed while listening to sound stimuli. We used soft classification based on GMM to achieve a sort of regression that allows the user to explore the whole space of articulation, from *legato* to *staccato*, through dynamic gesture variations.

Results from the user study with the system indicate that the model succeeds at providing control over articulation. Importantly, the model managed to learn intended articulations even if the way participants performed the articulations to train the system embedded idiosyncratic elements that were not shared across participants. Even though we imposed a particular base gesture (figure-eight) and tempo, variations in gesture execution were free. The results indicate that these variations were indeed idiosyncratic, meaning that a model learned on a user's set of data embeds her own expressive gesture qualities and may not be transferable to another user.

Interestingly, we found that the quality of the trained models, computed as the separability of different articulations in the representation space, did not have a significant effect on the results during performance. This, combined with the fact that musicians achieved better performance, suggests that participants were able to adapt their performance in an action-perception loop during the execution of the tasks.

This part of the work was done during a research stay at the Embodied Audiovisual Interaction (EAVI)[3] lab at Goldsmiths University of London and resulted in a joint peer-reviewed conference publication (Sarasúa et al., 2016a).

7.7.1 Limitations and future work

The fact that the quality of the models did not significantly affect the results during performance suggests that the continuous action-perception loop during execution allowed users to adapt their performance in order to achieve the task. The results from the questionnaire reveal that the audiovisual feedback was consistent with participants' expectations, indicating that they did not need to perform unnatural gestures to achieve the task, and that the audiovisual feedback was a reinforcing confirmation to their actions. Such aspects of sensorimotor learning that may enter into play constitute an important direction for future research. Note that in the case of the system for beat and loudness control the observation was in the opposite direction: the quality measures computed from the training stage were correlated with the performance during task execution. This suggests that investigating these aspects of sensorimotor learning during execution might need to consider the nature of the underlying model used for mapping as a factor affecting how users interact with the system.

Regarding the specific implementation of the proposed system, there are some limitations that are worth discussing. The scheme considers a single gesture at a fixed tempo, and the descriptors we used can be affected by changes in, for example, tempo, that should not affect articulation. The main strategies we foresee to address this issue are two.

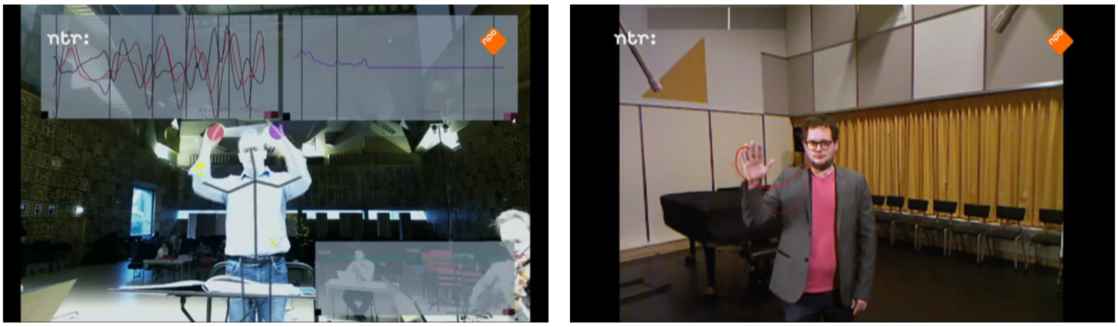
[3] http://eavi.goldsmithsdigital.com/

The first possible direction would be to build a representation space with descriptors that are more robust to these variations (for example, descriptors related to the shape of the gesture). Then, we believe that incorporating a temporal model of gesture could also be useful in this case. For example, continuous gesture recognition schemes based on dynamical systems as proposed by Caramiaux et al. (2014b) could afford the user the possibility to train the system to recognize different gestures and a set of potential variations which could then be dynamically explored, in performance, by the user. In this context, dynamic gesture variations estimated by the model could be used as a different representation space for variation, as opposed to using descriptors directly computed from positional data.

Also, the observational study revealed that articulation itself affects gesture timing. This is something we did not consider in our approach, where we focused on dynamic variations to drive articulation. Following this idea, future research should deal with how these techniques can be used to learn more complex *many-to-many* mappings. This is, while here we focused on learning a model for articulation, it would be desirable to learn, with as few training examples as possible, a model for a wider range of musical parameters. Human-centred approaches of machine learning, as discussed by Fiebrink and Caramiaux (2016), are particularly suitable to design such complex mappings through examples.

7.8 Contribution 8: Becoming the Maestro

Becoming the Maestro, the game developed during this work, has shown its potential to attract new audiences to classical music. In a context where it seems difficult for classical music to adapt to new music consumption habits, new technologies can help to create attractive experiences around this music genre. The PHENICX project has resulted in numerous applications pursuing this goal. In this context, the game exploits some of the technologies developed in this project, creating an experience that is not oriented to realistically replicate the role of the conductor during performance, but to provide the user with an entertaining experience that puts her in contact with this type of music. The game has been evaluated with a group of users who are not regular consumer of classical music, but who show a moderately positive attitude towards classical music. The game and the results form this evaluation have been presented in a peer-reviewed conference publication (Sarasúa et al., 2016b).



(a) Real-time MoCap feature visualizer.

(b) Presenter of the show controlling articulation.

Figure 7.1: Snapshots from *De Kennis van Nu* TV show.

7.8.1 Limitations and future work

The game has been developed to a prototype stage, so it is still necessary to take it to a more mature phase that allows its installation in possible institutions interested in it, or its commercialization through different platforms. Also, deploying the game in a public installation will allow us to better evaluate its possible real impact, beyond the user tests we can do in a lab environment. In addition, participants recruited for the presented evaluation generally showed a moderately positive attitude toward classical music. A challenge of the game is to be attractive to audiences with a more negative attitude towards the genre, so its deployment in a real environment will also allow to supplement information in this regard.

In addition, the game must incorporate some of the improvements indicated by users, such as including levels where they take control over different aspects of the performance. For this, the conclusions of the other prototypes developed in this work will be applicable, creating similar challenges to the tasks that we have used for their evaluation, and where the score is given depending on how the task is performed.

7.9 Non-academic impact

The work carried out throughout this thesis has not only resulted on the aforementioned contributions. It has also contributed to dissemination tasks, mostly related to the PHENICX project, sometimes gaining media attention. Here we mention some relevant events or media apparitions around this work:

- *Exponential Prometheus* at Singularity University (SU). The PHENICX consortium participated at the SU Summit in Seville (Spain), on March, 2015. In this event, we created real-time visualizations for a concert performed by the young *Sinfonietta de San Francisco de Paula*[4] playing the Overture to Beethoven's *The Creatures of Prometheus*. MoDe was used for the *Conductor* visualization, which we already mentioned in 6.1 as inspiration for the *Becoming the Maestro* visual design. Information about this event is available at the PHENICX project academic website[5].

- *Exponential Prometheus* at the 2015 International Society for Music Information Retrieval (ISMIR) conference. The same real-time visualizations were shown again in another concert with the same piece and orchestra, this time in ISMIR conference in Málaga (Spain), on October 2015.
- Appearance on *De Kennis van Nu*. On January 21st, 2016, the National Dutch TV science show *De Kennis Van Nu* ("current knowledge") aired a program about music conducting. They analyzed the figure of the conductor from different points of view, and we were invited to provide a scientific explanation of conducting movements. For this, we used MoDe to build a real-time MoCap descriptor visualizer that allowed to navigate through different joints and descriptors while discussing them. Figure 7.1a shows a snapshot of this visualizer during the show. Also, the presenter of the show demonstrated the system for articulation control through gesture variation presented in Chapter 5. A snapshot of this part of the show is depicted in Figure 7.1b. The program is available online[6], and the MoCap descriptor visualizer we developed is available as one of the examples in the MoDe library online repository.
- Public installation of *Becoming the Maestro* at *Arts Santa Mònica*. The final event of the PHENICX project was held at *Santa Mònica* arts centre in Barcelona (Spain). Assistants to this event could play *Becoming the Maestro*. Figure 7.2 shows a girl playing the game during this event.
- Appearance on *DeuWatts*. On March 10th, 2017, the Barcelona public TV science show *Deuwatts* featured *Becoming the Maestro* as an example of state-of-the-art technologies around music. The program is available online[7].
- *Becoming the Maestro* demonstration at *Fira Recerca en Directe* 2017. *Becoming the Maestro* will be shown at the *Fira Recerca en Directe* ("live science fair") on

[4] http://www.sinfoniettadesanfranciscodepaula.com/

[5] http://phenicx.upf.edu/SUPrometheus

[6] http://dekennisvannu.nl/site/media/De-Kennis-van-Nu—De-wetenschap-achter-dirigeren/5811

[7] http://beteve.cat/clip/deuwatts-tecnologia-musical/

Figure 7.2: A girl plays *Becoming the Maestro* at *Arts Santa Mònica* (Barcelona).

May 2017 at Caixaforum, Barcelona (Spain). This fair is an exhibition of disrupting technologies from catalan research institutions, open to the public.

7.10 Closing remarks

This dissertation has presented an in-depth study of interaction with systems based on the conductor metaphor. We have faced this analysis focusing on the potential that interface metaphors offer for interaction. We have analyzed the activity that inspires the metaphor (music conducting) and we have proven that users show idiosyncratic tendencies when they perform conducting movements. Accordingly, we have exploited this fact for user-specific DMI mapping design. During this research, done in the frame of the PHENICX project, we pursued the goal of creating applications attractive to potential new audiences of classical music. We have pursued this goal implicitly, by proposing these mapping strategies, but also explicitly, developing and evaluating a game with this goal in mind.

It is our belief that the outcomes of this thesis bring better understanding on how people with different musical expertise perform conducting movements, and that the proposed strategies allow to exploit this knowledge efficiently in an interactive context. We also believe that some of the outcomes of this thesis are useful for other interaction schemes using interface metaphors, as well as for gesture-based interaction. The good reception that the public demonstrations of the technologies developed in this project had makes us confident that we will be able to see applications beyond prototypes for a wide audience in the short and mid term.

Bibliography

Abdur Rahman, M., Qamar, A. M., Ahmed, M. A., Ataur Rahman, M., and Basalamah, S. (2013). Multimedia interactive therapy environment for children having physical disabilities. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval - ICMR '13*, pages 313–314, New York, New York, USA. ACM Press. 1.1.1

Ahnlund, J. (2016). Statistical Analysis of Conductor-Musician Interaction With a Focus on Musical Expression. Master's thesis, KTH, Computer Vision and Active Perception, CVAP. 2.4.2

Alemi, O., Pasquier, P., and Shaw, C. (2014). Mova: Interactive Movement Analytics Platform. In *Proceedings of the 2014 International Workshop on Movement and Computing*, MOCO '14, pages 37–42, Paris, France. ACM. 2.1.5

Alexiadis, D. S., Kelly, P., Daras, P., O'Connor, N. E., Boubekeur, T., and Moussa, M. B. (2011). Evaluating a dancer's performance using kinect-based skeleton tracking. In *Proceedings of the 19th ACM international conference on Multimedia - MM '11*, page 659. ACM Press. 1.1.1

Aristidou, A. and Chrysanthou, Y. (2014). Feature extraction for human motion indexing of acted dance performances. In *2014 International Conference on Computer Graphics Theory and Applications (GRAPP)*, pages 1–11. 2.1.3

Arzt, A., Goebl, W., Widmer, G., and Perception, C. (2015). Flexible Score Following: the Piano Music Companion and Beyond. In *Proceedings of the Third Vienna Talk on Music Acoustics*, pages 220–223. 6.1.2

Aschersleben, G. (2002). Temporal control of movements in sensorimotor synchronization. *Brain and Cognition*, 48(1):66–79. 4.4

Baba, T., Hashida, M., and Katayose, H. (2010). VirtualPhilharmony: A Conducting System with Heuristics of Conducting an Orchestra. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*, pages 263–270. 2.5.2

Bacot, B. and Féron, F.-X. (2016). The Creative Process of Sculpting the Air by Jesper

Nordin: Conceiving and Performing a Concerto for Conductor with Live Electronics. *Contemporary Music Review*, 35(4-5):450–474. 1.1.1, 2.5.2

Barbosa, J., Calegario, F., Teichrieb, V., Ramalho, G., and Cabral, G. (2013). A Drawing-Based Digital Music Instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 499–502, Daejeon, Republic of Korea. Graduate School of Culture Technology, KAIST. 1.1

Bender, T. and Hancock, C. (2010). The effect of conductor intensity and ensemble performance quality on musicians' evaluations of conductor effectivenes. *Journal of Band Research*, (46):13–22. 2.4.2

Bergen, S. (2012). *Conductor Follower: Controlling sample-based synthesis with expressive gestural input*. Master's thesis, Aalto University School of Science. 2.5.2, 4.1, 4.3.1, 4.3.3

Berlioz, H. (1856). *Le chef d'orchestre: théorie de son art*. Actes Sud. 2.4.1

Berthaut, F. and Knibbe, J. (2014). Wubbles: A Collaborative Ephemeral Musical Instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 499–500, London, United Kingdom. Goldsmiths, University of London. 1.1

Bevilacqua, F., Ridenour, J., and Cuccia, D. J. (2002). 3D motion capture data: motion analysis and mapping to music. In *Proceedings of the workshop/symposium on sensing and input for media-centric systems*, pages 562–569. IEEE, IEEE Computer Society Press. 2.1.5, 2.2.2

Bevilacqua, F., Schnell, N., Rasamimanana, N., Zamborlin, B., and Guédy, F. (2011). Online Gesture Analysis and Control of Audio Processing. In *Musical Robots and Interactive Multimodal Systems*, pages 127–142. Springer Berlin Heidelberg. 2.2.3

Bevilacqua, F., Zamborlin, B., Sypniewski, A., Schnell, N., Guédy, F., and Rasamimanana, N. (2010). Continuous Realtime Gesture Following and Recognition. In Kopp, S. and Wachsmuth, I., editors, *Gesture in Embodied Communication and Human-Computer Interaction: 8th International Gesture Workshop, GW 2009, Bielefeld, Germany, February 25-27, 2009, Revised Selected Papers*, pages 73–84. Springer Berlin Heidelberg, Berlin, Heidelberg. 2.2.2, 2.2.3, 4.3.3

Blackwell, A. F. (2006). The reification of metaphor as a design tool. *ACM Transactions on Computer-Human Interaction*, 13(4):490–530. 1.1, 1

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., and Serra, X. (2013). ESSENTIA: An audio analysis library for music information retrieval. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 493–498. 3.2.3, 4.2.2

Boie, B., Mathews, M. V., and Schloss, A. (1989). The Radio Drum as a Synthesizer Controller. In *International Computer Music Conference*, volume 1989, pages 42–45. Michigan Publishing, University of Michigan Library. 2.5.1, 4.1

Borchers, J., Hadjakos, A., and Mühlhäuser, M. (2006). MICON A Music Stand for Interactive Conducting. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 254–259. 2.5.2

Borchers, J. O., Samminger, W., and Mühlhäuser, M. (2002). Engineering a realistic real-time conducting system for the audio/video rendering of a real orchestra. In *Proceedings - 4th International Symposium on Multimedia Software Engineering, MSE 2002*, pages 352–362. IEEE Comput. Soc. 2.5.2, 7, 4.1, 4.3.1

Boulanger, R., Mathews, M., Vercoe, B., and Dannenberg, R. (1990). Conducting the MIDI Orchestra, Part 1: Interviews with Max Mathews, Barry Vercoe, and Roger Dannenberg. *Computer Music Journal*, 14(2):34–46. 2.5.1

Brecht, B. and Garnett, G. (1995). Conductor Follower. In *Proc. of the 1995 International Computer Music Conference.*, pages 185–186, San Francisco, California. International Computer Music Association. 2.5.2, 21, 4.1, 4.3.3

Burger, B. and Toiviainen, P. (2013). MoCap Toolbox-A Matlab toolbox for computational analysis of movement data. In *Proceedings of the 2013 Sound and Music Computing Conference 2013, SMC 2013*, pages 172–178, Stockholm, Sweden. Logos Verag Berlin. 2.1.5

Buxton, W., Reeves, W., Fedorkow, G., Smith, K. C., and Baecker, R. (1980). A Microcomputer-Based Conducting System. *Computer Music Journal*, 4(1):8. 2.5.1

Camurri, A., Hashimoto, S., Ricchetti, M., Ricci, A., Suzuki, K., Trocca, R., and Informatica, D. (2000). EyesWeb : Toward Gesture and Affect Recognition in Interactive Dance and Music Systems. *Computer Music Journal*, 24(1):57–69. 2.1.3

Camurri, A., Mazzarino, B., and Volpe, G. (2004). Analysis of expressive gesture: The eyesweb expressive gesture processing library. In *Gesture-based Communication in Human-Computer Interaction*, pages 460–467. Springer, Berlin, Heidelberg. 2.1.2, 2.1.2, 2.1.3, 2.1.5, 2.5.2

Caramiaux, B., Bevilacqua, F., and Schnell, N. (2010). Towards a Gesture-Sound Cross-Modal Analysis. In Kopp, S. and Wachsmuth, I., editors, *Embodied Communication and Human-Computer Interaction, volume 5934 of LNCS*, pages 158–170. Springer Berlin Heidelberg, Berlin, Heidelberg. 2.2.3

Caramiaux, B., Françoise, J., Schnell, N., and Bevilacqua, F. (2014a). Mapping Through Listening. *Computer Music Journal*, 38(3):34–48. 2.2.3, 4.3, 5.1

Caramiaux, B., Montecchio, N., Tanaka, A., and Bevilacqua, F. (2014b). Adaptive Gesture Recognition with Variation Estimation for Interactive Systems. *ACM Transactions on Interactive Intelligent Systems*, 4(4):1–34. 2.2.2, 4.3.3, 5.4, 6.2.2, 7.7.1

Caramiaux, B. and Tanaka, A. (2013). Machine Learning of Musical Gestures. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 513–518, Daejeon, Republic of Korea. 2.2.3

Clayton, A. M. H. (1986). *Coordination Between Players in Musical Performance*. PhD thesis, Edinburgh University. 2.4.2

Cornejo, R., Hernandez, D., Favela, J., Tentori, M., and Ochoa, S. (2012). Persuading older adults to socialize and exercise through ambient games. In *Proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcare*, pages 215–218. IEEE. 1.1.1

Dahl, S. and Friberg, A. (2004). Expressiveness of musician's body movements in performances on marimba. *Lecture Notes in Computer Science*, 2915(3):479–486. 2.4.2

Davidson, J. W. (1993). Visual Perception of Performance Manner in the Movements of Solo Musicians. *Psychology of Music*, 21(2):103–113. 2.4.2

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38. 5.3.1, 5.3.1

Diakopoulos, D., Trimpin, and Ludovic Morlot (2015). World premiere uses Kinect to conduct, puts audience inside the instrument. Online: https://news.microsoft.com/features/world-premiere-uses-kinect-to-conduct-puts-audience-inside-the-instrument/. 1.1.1, 2.5.2

Diesbach, S., Lacote, J., and Perrenoud, L. (2013). The computer orchestra. *University of Art and Design, Lausanne*. 2.5.2

Dixon, S. (2001). Automatic Extraction of Tempo and Beat From Expressive Performances. *Journal of New Music Research*, 30(1):39–58. 3

Dobson, M. C. and Pitts, S. E. (2011). Classical Cult or Learning Community? Exploring New Audience Members' Social and Musical Responses to First-time Concert Attendance. *Ethnomusicology Forum*, 20(3):353–383. 6.1.1

Erickson, T. D. (1995). Working with Interface Metaphors. In Baecker, R. M., Grudin, J., Buxton, W. A. S., and Greenberg, S., editors, *The Art of Human-Computer Interface*

Design, pages 147–151. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 1.1, 1

Fabiani, M. (2011). *Interactive computer-aided expressive music performance: Analysis, control, modification and synthesis*. PhD thesis, KTH. (document), 2.5.2, 2.8

Fails, J. A. and Olsen, D. R. (2003). Interactive machine learning. *Proceedings of the 8th international conference on Intelligent user interfaces IUI 03*, pages 39–45. 2.2.3

Fan, X. and Essl, G. (2013). Air Violin: A Body-centric Style Musical Instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 122–123, Daejeon, Republic of Korea. Graduate School of Culture Technology, KAIST. 1.1.1

Fiebrink, R. and Caramiaux, B. (2016). The Machine Learning Algorithm as Creative Musical Tool. *Handbook of algorithmic Music*. 2.2.3, 7.7.1

Fiebrink, R. and Cook, P. R. (2009). Play-Along Mapping of Musical Controllers. In *Proceedings of the International Computer Music Conference*, volume 2009, pages 61–64. Michigan Publishing. 2.2.3, 4.3, 7.6.1

Fiebrink, R. A. (2011). *Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance*. PhD thesis, Princeton, NJ, USA. 2.2.3

Françoise, J. (2015). *Motion-Sound Mapping by Demonstration*. PhD thesis, Université Pierre et Marie Curie. Paris, France. 2.2, 2.2.3, 5.1, 5.3, 7.7

Françoise, J., Schnell, N., Borghesi, R., and Bevilacqua, F. (2014). Probabilistic Models for Designing Motion and Sound Relationships. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 287–292, London, United Kingdom. Goldsmiths, University of London. 2.2.3, 4.3.3

Friberg, A. (2005). Home conducting - Control the overall musical expression with gestures. In *Proceedings of the 2005 International Computer Music Conference*, pages 479–482, San Francisco. International Computer Music Association. 2.5.2

Friberg, A., Bresin, R., and Sundberg, J. (2006). Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology*, 2(2-3):145–161. 2.5.2

Galkin, E. (1988). *A history of orchestral conducting : in theory and practice*. Pendragon Press. 2.4.1

Gambetta, C. (2005). *Conducting outside the box: Creating a fresh approach to conducting gesture through the principles of Laban Movement Analysis*. PhD thesis. 2.4.1

Garnett, G., Jonnalagadda, M., Elezovic, I., Johnson, T., and Small, K. (2001). Technological advances for conducting a virtual ensemble. In *International Computer Music Conference*, volume 2001, pages 167–169, Habana, Cuba. Michigan Publishing, University of Michigan Library. 2.5.2

Garnett, G. E., Malvar-Ruiz, F., and Stoltzfus, F. (1999). Virtual Conducting Practice Environment. In *Proceedings of the International Computer Music Conference*, pages 371–374. 2.5.2, 5.1

Gaver, W. W. (1991). Technology Affordances. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 79–84, New York, NY, USA. ACM. 1

Glette, K. H., Jensenius, A. R., and Godøy, R. I. (2010). Extracting action-sound features from a sound-tracing study. In *Proceedings of the second Norwegian Artificial Intelligence Symposium*, pages 63–66. 4.3.3

Godøy, R. I., Haga, E., and Jensenius, A. R. (2006). *Playing Air Instruments: Mimicry of Sound-Producing Gestures by Novices and Experts*, pages 256–267. Springer Berlin Heidelberg, Berlin, Heidelberg. 4.3.3

Gómez, E., Grachten, M., Hanjalic, A., Janer, J., Jordà, S., Julià, C. F., Liem, C. C. S., Martorell, A., Schedl, M., and Widmer, G. (2013). PHENICX: Performances as Highly Enriched aNd Interactive Concert Experiences. In *SMAC Stockholm Music Acoustics Conference 2013 and SMC Sound and Music Computing Conference 2013*, Stockholm, Sweden. 1.1.1, 1.1.1

Gonzalez-Jorge, H., Rodríguez-Gonzálvez, P., Martínez-Sánchez, J., González-Aguilera, D., Arias, P., Gesto, M., and Díaz-Vilariño, L. (2015). Metrological comparison between Kinect I and Kinect II sensors. *Measurement*, 70:21–26. 2.3.2

Grierson, M. and Kiefer, C. (2011). Maximillian: An easy to use, cross platform C++ Toolkit for interactive audio and synthesis applications. In *Proceedings of the International Computer Music Conference 2011*, pages 276–279, University of Huddersfield, UK. 5.3.2

Grull, I. (2005). *conga: A Conducting Gesture Analysis Framework*. Master's thesis, Universität Ulm. 2.5.2

Hachimura, K., Takashina, K., and Yoshimura, M. (2005). Analysis and evaluation of dancing movement based on LMA. In *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, volume 2005, pages 294–299. IEEE. 2.1.3

Hadjakos, A. and Grosshauser, T. (2013). Motion and Synchronization Analysis of Musical Ensembles with the Kinect. In *Proceedings of the International Conference* *on New Interfaces for Musical Expression*, pages 106–110, Daejeon, Republic of Korea. Graduate School of Culture Technology, KAIST. 1.1.1

Haflich, F. and Burnds, M. (1983). Following a conductor: The engineering of an input device. In *Proc. of the 1983 International Computer Music Conference, San Francisco.* 2.5.2, 18, 21, 4.1, 4.3.1

Hassenzahl, M. (2008). The Interplay of Beauty, Goodness, and Usability in Interactive Products. *Hum.-Comput. Interact.*, 19(4):319–349. 6.3, 6.3.1, 6.3.2

Huang, J.-D. and Jun-Da (2011). Kinerehab. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility - ASSETS '11*, page 319, New York, New York, USA. ACM Press. 1.1.1

Hunt, A. and Kirk, R. (2000). Mapping Strategies for Musical Performance. *Trends in Gestural Control of Music*, pages 231–258. 2.2.1

Ilmonen, T. and Takala, T. (1999). Conductor Following With Artificial Neural Networks. In *Proceedings of the International Computer Music Conference*, pages 367–370. 2.5.2, 4.1, 4.3.3

Jensenius, A. (2007). *Action-sound: Developing methods and tools to study music-related body movement.* PhD thesis, University of Oslo. 1.1.1, 2.1.3, 2.1.5, 4.4, 7.6.1

Jensenius, A. R. (2012). Motion-sound Interaction Using Sonification based on Motiongrams. In *Proceedings of the Fifth International Conference on Advances in Computer-Human Interactions*, pages 170–175, Valencia. 2.2

Jensenius, A. R. (2014). To gesture or Not? An Analysis of Terminology in NIME Proceedings 2001–2013. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 217–220, London, United Kingdom. Goldsmiths, University of London. 7

Jones, E., Oliphant, T., Peterson, P., and Others. SciPy: Open source scientific tools for Python. Online: https://www.scipy.org/. 3.2.3, 4.2.2

Jordà, S. (2005). *Digital Lutherie: Crafting musical computers for new musics' performance and improvisation.* PhD thesis, Universitat Pompeu Fabra. 1.1

Jordà, S. (2007). *Interactivity and Live Computer Music*, chapter 5, page 312. Cambridge Companions to Music. Cambridge University Press, Cambridge, UK. 1.1

Jungong Han, Ling Shao, Dong Xu, and Shotton, J. (2013). Enhanced Computer Vision With Microsoft Kinect Sensor: A Review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334. 1.1.1, 2.3.1

Kapadia, M., Chiang, I.-k., Thomas, T., Badler, N. I., and Kider Jr., J. T. (2013).

Efficient Motion Retrieval in Large Motion Databases. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '13, pages 19–28, New York, NY, USA. ACM. 2.1.2, 2.1.3

Karipidou, K. (2015). Modelling the body language of a musical conductor using Gaussian Process Latent Variable Models. Master's thesis, KTH, Computer Vision and Active Perception, CVAP. 2.4.2

Keane, D. and Gross, P. (1989). *The MIDI Baton.* Ann Arbor, MI: Michigan Publishing, University of Michigan Library. 2.5.2, 4.1, 4.3.1

Keane, D. and Wood, K. (1990). *The MIDI Baton II.* School of Music/Department of Electrical Engineering, Queen's University, Glasgow. 2.5.2

Keane, D. and Wood, K. (1991). *The MIDI Baton III.* School of Music/Department of Electrical Engineering, Queen's University, Glasgow. 2.5.2

Kolb, B. M. (2000). You call this fun? Reactions of young, first-time attendees to a classical concert. *Weissman D (ed.) Music Industry Issues and Studies*, 1(1):13–28. 6.1.1

Kolesnik, P. (2004). *Conducting Gesture Recognition, Analysis and Performance System.* Master's thesis, McGill University, Montreal. 2.2.2, 2.4.1, 2.5.2, 4.1, 4.3.3

Kumar, A. B. and Morrison, S. J. (2016). The Conductor As Visual Guide: Gesture and Perception of Musical Content. *Frontiers in psychology*, 7:1049. 2.4.2

Lago, N. and Kon, F. (2004). The quest for low latency. *Proceedings of the International Computer Music Conference*, pages 33–36. 4.3.3

Larboulette, C. and Gibet, S. (2015). A review of computable expressive descriptors of human motion. In *Proceedings of the 2nd International Workshop on Movement and Computing - MOCO '15*, pages 21–28, New York, New York, USA. ACM Press. 2, 2.1.2, 2.1.3

Lee, E., Kiel, H., Dedenbach, S., Grull, I., Karrer, T., Wolf, M., and Borchers, J. (2006). iSymphony: an adaptive interactive orchestral conducting system for digital audio and video streams. In *Proceedings of ACM CHI 2006 Conference on Human Factors in Computing Systems*, volume 2, pages 259–262, New York, New York, USA. ACM Press. 2.5.2, 2.5.2

Lee, E., Nakra, T. M., and Borchers, J. (2004). You're the conductor: a realistic interactive conducting system for children. In *Proceedings of the 2004 conference on New Interfaces for Musical Expression*, pages 68–73. National University of Singapore. 2.5.2, 2.5.2, 4.1, 4.3.1, 4.3.3, 6.1.1

Lee, E., Wolf, M., and Borchers, J. (2005). Improving orchestral conducting systems in public spaces: examining the temporal characteristics and conceptual models of conducting gestures. *Proceedings of ACM CHI 2005 Conference on Human Factors in Computing Systems*, 1:731–740. 2.5.2, 4.1, 4.2.4, 4.3.1, 4.3.2, 4.4

Lee, M., Garnett, G., and Wessel, D. (1992). An adaptive conductor follower. In *Proceedings of the 1992 International Computer Music Conference*, pages 454–455. International Computer Music Association. 2.5.2

Lewis, C., Mojsiewicz, K., Pettican, A., and Art, B. (2012). From Wunderkammern to Kinect: The Creation of Shadow Worlds. In *ACM SIGGRAPH 2012 Art Gallery*, SIGGRAPH '12, pages 330–337, New York, NY, USA. ACM. 1.1.1

Livingston, M. A., Sebastian, J., Ai, Z., and Decker, J. W. (2012). Performance measurements for the Microsoft Kinect skeleton. In *2012 IEEE Virtual Reality Workshops (VRW)*, pages 119–120. 2.3.1

Luck, G. (2000). Synchronising a Motor Response with a Visual Event: The Perception of Temporal Information in a Conductor ' s Gestures. In *International Conference on Music Perception and Cognition*. 2.4.2

Luck, G. and Sloboda, J. (2008). Exploring the Spatio-Temporal Properties of Simple Conducting Gestures using a Synchronization Task. *Music Perception: An Interdisciplinary Journal*, 25(3):225–239. 2.1.2

Luck, G. and Toiviainen, P. (2006). Ensemble musicians' synchronization with conductors' gestures: An automated feature-extraction analysis. *Music Perception*, 24(2):189–200. (document), 2.1.2, 2.2, 2.4.2, 3.2.3, 3.2.3, 3.2.3

Luck, G., Toiviainen, P., and Thompson, M. R. (2010). Perception of Expression in Conductors' Gestures: A Continuous Response Study. *Music Perception*, 28(1):47–57. 2.4.2

Marier, M. (2014). Designing Mappings for the Sponge: Towards Spongistic Music. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 525–528, London, United Kingdom. Goldsmiths, University of London. 1.1

Marrin, T. (2000). *Inside the Conductor's Jacket: Analysis, Interpretation and Musical Synthesis of Expressive Gesture*. PhD thesis, Massachusetts Institute of Technology. (document), 2.6, 2.5.2

Marrin, T. and Paradiso, J. (1997). The Digital Baton: a Versatile Performance Instrument. In *International Computer Music Conference*, volume 1997, pages 313–316. Michigan Publishing, University of Michigan Library. 2.5.2

Marrin, T. and Picard, R. (1998). The 'Conductors Jacket': A Device for Recording Expressive Musical Gestures. In *Proceedings of the International Computer Music Conference*, number 470, pages 215–219. 2.5.2

Marrin, T. A. (1996). *Toward an Understanding of Musical Gesture: Mapping Expressive Intention with the Digital Baton*. PhD thesis, Massachusetts Institute of Technology. 2.5.2

Mathews, M. V. (1976). The conductor program. In *Proceedings of the International Computer Music Conference*. 2.5.1, 4.1

Mathews, M. V. (1991). The radio baton and conductor program, or: pitch, the most important and least expressive part of music. *Computer Music Journal*, 15(4):37–46. 2.5.1, 4.1

Mathews, M. V. and Barr, D. (1988). The Conductor Program and Mechanical Baton. (document), 2.5.1, 2.5

Mathews, M. V. and Moore, F. R. (1970). GROOVE - a program to compose, store, and edit functions of time. *Communications of the ACM*, 13(12):715–721. (document), 2.5.1, 2.4

Mayor, O., Llop, J., and Maestre, E. (2011). RepoVizz: A multimodal on-line database and browsing tool for music performance research. In *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, Miami, USA. 3.2.1

McGrenere, J. and Ho, W. (2000). Affordances : Clarifying and Evolving a Concept. In *Graphics Interface*, number May, pages 1–8. 1

Melenhorst, M. S. and Liem, C. C. S. (2015). Put the Concert Attendee in the Spotlight. A User-Centered Design and Development Approach for Classical Concert Applications. In *Proceedings of the 16th International Conference on Music Information Retrieval (ISMIR'15)*, pages 800–806. 6.1.1

Miron, M., Carabias, J. J., and Janer, J. (2014). Audio-to-score alignment at the note level for orchestral recordings. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 125–130, Taipei, Taiwan. 6.3.1

Miron, M., Carabias, J. J., and Janer, J. (2015). Improving score-informed source separation for classical music through note refinement. In *Proceedings of the 16th International Society for Music Information Retrieval (ISMIR) Conference*, pages 448–454. 6.3.1

Mitchell, T. and Heap, I. (2011). SoundGrasp : A Gestural Interface for the Performance of Live Music. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 465–468, Oslo, Norway. 2.2.2

Modler, P. (2000). Neural networks for mapping hand gestures to sound synthesis parameters. *Trends in Gestural Control of Music*, 18. 2.2.2

Morita, H., Hashimoto, S., and Ohteru, S. (1991). A computer music system that follows a human conductor. *Computer*, 24(7):44–53. 2.5.2

Morita, H., Ohteru, S., and Hashimoto, S. (1989). Computer Music System Which Follows a Human Conductor. In *International Computer Music Conference Proceedings*, pages 207–210. Michigan Publishing, University of Michigan Library. 2.5.2, 4.1, 4.3.1

Morrison, S. J., Price, H. E., Smedley, E. M., and Meals, C. D. (2014). Conductor gestures influence evaluations of ensemble performance. *Frontiers in psychology*, 5:1 – 8. 2.4.2

Müllensiefen, D., Gingras, B., Musil, J., and Stewart, L. (2014). The musicality of non-musicians: An index for assessing musical sophistication in the general population. *PLoS ONE*, 9(2):e89642. 6.3.2

Müller, M. (2007). *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. 2.1.1, 2.1.3

Newlove, J. and Dalby, J. (2004). *Laban for all*. Taylor & Francis US. 2.1

Nymoen, K., Caramiaux, B., Kozak, M., and Torresen, J. (2011). Analyzing sound tracings - A Multimodal Approach to Music Information Retrieval. In *Proceedings of the 1st international ACM workshop on Music information retrieval with user-centered and multimodal strategies - MIRUM '11*, page 39, New York, New York, USA. ACM Press. 2.2.3

Paradiso, J. A. (1997). Electronic music: new ways to play. *IEEE Spectrum*, 34(12):18–30. 1.1

Place, T. and Lossius, T. (2006). Jamoma: A modular standard for structuring patches in max. In *In Proceedings of the International Computer Music Conference*, pages 143–146. 2.1.5

Platte, S. L. (2016). *The Maestro Myth: Exploring the Impact of Conducting Gestures on the Musician ' s Body and the Sounding Result*. Master's thesis, Massachusetts Institute of Technology. (document), 2.4.1, 2.4.2, 2.3, 5.1

Platz, F. and Kopiez, R. (2012). When the Eye Listens: A Meta-analysis of How Audio-visual Presentation Enhances the Appreciation of Music Performance. *Music Perception: An Interdisciplinary Journal*, 30(1):71–83. 2.4.2

Poggi, I. (2002). The lexicon of the Conductor's face. In *Language, Vision, and Music: Selected Papers from the 8th International Workshop on the Cognitive Science* *of Natural Language Processing, Galway, Ireland, 1999*, volume 35, page 271. John Benjamins Publishing. 7.4.1

Pressing, J. (1990). Cybernetic Issues in Interactive Performance Systems. *Computer Music Journal*, 14(1):12–25. 1.1

Price, H. E. and Mann, A. (2011). The effect of conductors on ensemble evaluations. *Bulletin of the Council for Research in Music Education*, (189):57–72. 2.4.2

Rodrigues, D. G., Grenader, E., Nos, F. d. S., Dall'Agnol, M. d. S., Hansen, T. E., and Weibel, N. (2013). MotionDraw: a tool for enhancing art and performance using kinect. *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13*, page 1197. 1.1.1

Rosa-Pujazon, A. and Barbancho, I. (2013). Conducting a virtual ensemble with a kinect device. In *Proceedings of the Sound and Music Computing Conference, Stockholm, Sweden*, pages 284–291. 2.5.2, 4.1, 4.3.1, 4.3.3

Rovan, J. B., Wanderley, M. M., Dubnov, S., and Depalle, P. (1997). Instrumental gestural mapping strategies as expressivity determinants in computer music performance. In *Proceedings of Kansei-The Technology of Emotion Workshop*, pages 3–4. 1.1, 2.2.1

Rudolf, M. (1980). *The grammar of conducting*. Schirmer Books. (document), 2.4.1, 5.1, 5.1

Sarasúa, Á., Caramiaux, B., and Tanaka, A. (2016a). Machine Learning of Personal Gesture Variation in Music Conducting. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3428–3432, San Jose, CA, USA. ACM. 7.7

Sarasúa, A. and Guaus, E. (2014a). Beat Tracking from Conducting Gestural Data: a Multi-Subject Study. In *Proceedings of the International Workshop on Movement and Computing*, pages 118–123, Paris, France. 7.5

Sarasúa, Á. and Guaus, E. (2014b). Dynamics in Music Conducting: A Computational Comparative Study Among Subjects. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 195–200, London, United Kingdom. Goldsmiths, University of London. 7.5

Sarasúa, Á., Melenhorst, M., Julià, C. F., and Gómez, E. (2016b). Becoming the Maestro - A Game to Enhance Curiosity for Classical Music. In *2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, pages 1–4. IEEE. 7.8

Sarasúa, Á., Urbano, J., and Gómez, E. (2017). Mapping by Observation: building a User-tailored Conducting System from Spontaneous Movements. *In preparation*. 7.6

Sawada, H. and Hashimoto, S. (1997). Gesture recognition using an acceleration sensor and its application to musical performance control. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 80(5):9–17. 2.2.2

Schacher, J. C. and Stoecklin, A. (2011). Traces – Body, Motion and Sound. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 292–295, Oslo, Norway. 2.2

Sell, J. and O'Connor, P. (2014). The xbox one system on a chip and kinect sensor. *IEEE Micro*, 34(2):44–53. 2.3.2, 4.3.3

Sentürk, S., Lee, S. W., Sastry, A., Daruwalla, A., and Weinberg, G. (2012). Crossole: A Gestural Interface for Composition, Improvisation and Performance using Kinect. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 185–188, Ann Arbor, Michigan. University of Michigan. 1.1.1

Skogstad, S. A., Holm, S., and Hovin, M. (2012). Digital IIR filters with minimal group delay for real-time applications. In *Engineering and Technology (ICET), 2012 International Conference on*, pages 1–6. IEEE. 2.1.2, 2.1.4, 2.1.5, 5.3.2, B.3

Skogstad, S. A. v. D., Nymoen, K., Høvin, M. E., Holm, S., and Jensenius, A. R. (2013). Filtering Motion Capture Data for Real-Time Applications. In *Proceedings of the 2013 Conference on New Interfaces for Musical Expression (NIME 2013)*, pages 142–147. 4.3.1, 4.3.3, 7.2

Stevens, S. S. (1975). *Psychophysics*. Transaction Publishers. 3.2.3, 4.2.2

Thompson, W. F., Graham, P., and Russo, F. A. (2005). Seeing music performance: Visual influences on perception and experience. *Semiotica*, 2005(156):203–227. 2.4.2

Tilmanne, J. and D'Alessandro, N. (2015). Motion machine: A new framework for motion capture signal feature prototyping. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 2401–2405. 2.1.5

Tits, M., Tilmanne, J., Wanderley, M., and D'Alessandro, N. (2015). Feature Extraction and Expertise Analysis of Pianists ' Motion-Captured Finger Gestures. In *Proceedings of the International Computer Music Conference (ICMC 2015)*, pages 102–105, Denton, Texas, USA. 2.1.5

Toh, L. W., Chao, W., and Chen, Y. S. (2013). An interactive conducting system using Kinect. In *Proceedings - IEEE International Conference on Multimedia and Expo*, pages 1–6. IEEE. 2.5.2, 4.1, 4.3.1, 4.3.3

Toiviainen, P., Luck, G., and Thompson, M. R. (2010). Embodied Meter: Hierarchical Eigenmodes in Music-Induced Movement. *Music Perception: An Interdisciplinary Journal*, 28(1):59–70. 2.1.5

Trail, S., Dean, M., Odowichuk, G., Tavares, T. F., Driessen, P., Schloss, W. A., and Tzanetakis, G. (2012). Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the Kinect. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan. University of Michigan. 1.1.1

Trajkova, M. and Cafaro, F. (2016). E-ballet: Designing for Remote Ballet Learning. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, pages 213–216, New York, NY, USA. ACM. 1.1.1

Tsui, C. K., Law, C. H., and Fu, H. (2014). One-man Orchestra: Conducting Smartphone Orchestra. In *SIGGRAPH Asia 2014 Emerging Technologies*, SA '14, pages 11:1—-11:2, New York, NY, USA. ACM. 2.5.2

Usa, S. and Mochida, Y. (1998a). A conducting recognition system on the model of musicians' process. *Journal of the Acoustical Society of Japan*, 4:275–287. 4.1, 4.3.1, 4.3.3

Usa, S. and Mochida, Y. (1998b). A Multi-Modal Conducting Simulator. In *International Computer Music Conference (ICMC)*, volume 1998, pages 25–32. Michigan Publishing, University of Michigan Library. 2.5.2, 5.1

Venkatesh, V., Morris, M. G., Davis, G. B., and Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, 27(3):425–478. 6.3, 6.3.1, 6.3.2

Venkatesh, V., Thong, J. Y. L., and Xu, X. (2012). Consumer Acceptance and Use of Information Technology: Extending the Unified Theory of Acceptance and Use of Technology. *MIS Q.*, 36(1):157–178. 6.3, 6.3.1, 6.3.2

Wagner, R. I. (1870). *Über das Dirigieren*. Tahnt. 2.4.1

Wanderley, M. (2001). *Performer-Instrument Interaction: Applications to Gestural Control of Sound Synthesis*. PhD thesis, University Pierre et Marie Curie-Paris VI. 1.1

Wanderley, M. M., Schnell, N., and Rovan, J. (1998). ESCHER-modeling and performing composed instruments in real-time. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, volume 2, pages 1080–1084 vol.2. 2.2.1

Yang, Q. and Essl, G. (2012). Augmented Piano Performance using a Depth Camera. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. 1.1.1

Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., and Presti, P. (2011). American sign language recognition with the kinect. In *Proceedings of the 13th international* *conference on multimodal interfaces - ICMI '11*, pages 279–286, New York, New York, USA. ACM Press. 1.1.1

Zhao, J., Yu, P. L., Shi, L., and Li, S. (2012). Separable linear discriminant analysis. *Computational Statistics & Data Analysis*, 56(12):4290–4300. 5.2.2

Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., and Harvill, Y. (1987). A Hand Gesture Interface Device. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, volume 18, pages 189–192, New York, New York, USA. ACM Press. 2.5.2

Appendix A

Complete beat analysis results from Chapter 3

Table A.1: Beat analysis from excerpt I1. p = precision; r = recall; F^* = F-measure without considering *lag*.

Descriptor	Max / Min	*lag* (s)	σ	p	r	F	F^*
x	max	-0.232	0.13	0.50	0.29	0.37	0.00
	min	-0.019	0.35	0.26	0.19	0.22	0.22
y	max	0.149	0.11	0.78	0.67	0.72	0.13
	min	-0.261	0.09	0.67	0.60	0.64	0.00
z	max	-0.058	0.32	0.48	0.33	0.40	0.17
	min	0.219	0.11	0.47	0.29	0.36	0.08
v_x	max	0.301	0.14	0.30	0.29	0.29	0.06
	min	0.026	0.25	0.31	0.29	0.30	0.37
v_y	max	-0.082	0.32	0.69	0.65	0.67	0.11
	min	0.209	0.10	0.62	0.60	0.61	0.08
v_z	max	0.327	0.15	0.51	0.54	0.53	0.08
	min	-0.014	0.28	0.33	0.33	0.33	0.29
a_x	max	-0.115	0.26	0.18	0.21	0.19	0.23
	min	-0.249	0.15	0.40	0.40	0.40	0.04
a_y	max	-0.348	0.12	0.65	0.67	0.66	0.02
	min	0.047	0.15	0.62	0.62	0.62	0.48
a_z	max	-0.049	0.26	0.20	0.25	0.22	0.30
	min	-0.220	0.13	0.40	0.44	0.42	0.02
v	max	0.162	0.22	0.26	0.23	0.24	0.18
	min	0.169	0.23	0.25	0.31	0.28	0.19
a	max	-0.061	0.23	0.21	0.21	0.21	0.21
	min	0.021	0.21	0.26	0.33	0.29	0.33
a_t	max	-0.135	0.22	0.25	0.31	0.28	0.15
	min	0.128	0.21	0.22	0.27	0.24	0.13

Table A.2: Beat analysis from excerpt I2. p = precision; r = recall; F^* = F-measure without considering *lag*.

Descriptor	**Max / Min**	*lag* **(s)**	σ	p	r	F	F*****
x	max	-0.267	0.22	0.28	0.21	0.24	0.12
	min	0.166	0.15	0.49	0.35	0.41	0.12
y	max	0.220	0.06	0.83	0.83	0.83	0.00
	min	-0.174	0.06	0.77	0.77	0.77	0.06
z	max	-0.011	0.33	0.40	0.33	0.36	0.30
	min	0.354	0.11	0.54	0.42	0.47	0.00
v_{x}	max	-0.365	0.21	0.44	0.44	0.44	0.08
	min	-0.039	0.18	0.50	0.48	0.49	0.32
v_y	max	-0.075	0.05	0.85	0.83	0.84	0.32
	min	0.299	0.07	0.73	0.73	0.73	0.00
v_{z}	max	-0.366	0.16	0.31	0.31	0.31	0.00
	min	-0.004	0.18	0.52	0.58	0.55	0.55
a_{x}	max	0.165	0.12	0.47	0.54	0.50	0.21
	min	-0.218	0.12	0.39	0.44	0.41	0.06
a_y	max	-0.277	0.08	0.71	0.73	0.72	0.02
	min	0.070	0.31	0.63	0.65	0.64	0.23
a_{z}	max	0.200	0.16	0.33	0.38	0.35	0.16
	min	-0.181	0.19	0.28	0.31	0.30	0.16
v	max	0.193	0.20	0.23	0.25	0.24	0.22
	min	-0.168	0.24	0.20	0.23	0.21	0.27
a	max	-0.160	0.22	0.17	0.21	0.19	0.21
	min	0.171	0.18	0.23	0.29	0.26	0.26
a_{t}	max	0.091	0.24	0.20	0.25	0.22	0.15
	min	-0.064	0.20	0.20	0.25	0.22	0.21

Table A.3: Beat analysis from excerpt I3. p = precision; r = recall; F^* = F-measure without considering lag.

Descriptor	**Max / Min**	lag **(s)**	σ	p	r	F	F*****
x	max	0.234	0.28	0.22	0.10	0.14	0.14
	min	0.192	0.18	0.47	0.31	0.38	0.10
y	max	0.193	0.06	0.83	0.81	0.82	0.02
	min	-0.199	0.11	0.75	0.69	0.72	0.02
z	max	-0.207	0.22	0.24	0.17	0.20	0.17
	min	0.144	0.24	0.31	0.17	0.22	0.19
v_{x}	max	0.316	0.19	0.40	0.40	0.40	0.06
	min	-0.084	0.21	0.21	0.23	0.22	0.24
v_y	max	-0.086	0.16	0.83	0.81	0.82	0.19
	min	0.267	0.07	0.71	0.71	0.71	0.00
v_{z}	max	0.226	0.14	0.39	0.44	0.41	0.10
	min	0.169	0.24	0.15	0.15	0.15	0.23
a_{x}	max	-0.138	0.29	0.13	0.15	0.14	0.20
	min	0.137	0.28	0.10	0.12	0.11	0.22
a_y	max	-0.319	0.10	0.63	0.65	0.64	0.02
	min	0.069	0.28	0.75	0.75	0.75	0.33
a_{z}	max	-0.089	0.30	0.18	0.21	0.19	0.19
	min	0.286	0.19	0.24	0.27	0.25	0.06
v	max	0.170	0.22	0.25	0.25	0.25	0.21
	min	-0.144	0.27	0.13	0.15	0.14	0.25
a	max	-0.133	0.24	0.22	0.21	0.21	0.19
	min	-0.094	0.22	0.11	0.15	0.13	0.22
a_{t}	max	0.119	0.25	0.11	0.12	0.12	0.23
	min	-0.103	0.19	0.20	0.25	0.22	0.11

Table A.4: Beat analysis from excerpt I4. p = precision; r = recall; $F*$ = F-measure without considering *lag*.

Descriptor	**Max / Min**	*lag* **(s)**	σ	p	r	F	F*****
x	max	-0.109	0.24	0.23	0.19	0.21	0.34
	min	0.232	0.15	0.43	0.27	0.33	0.05
y	max	0.264	0.08	0.60	0.38	0.46	0.03
	min	-0.004	0.28	0.40	0.40	0.40	0.38
z	max	-0.040	0.31	0.50	0.33	0.40	0.23
	min	0.208	0.18	0.29	0.19	0.23	0.10
v_x	max	-0.306	0.18	0.33	0.33	0.33	0.10
	min	-0.008	0.28	0.37	0.38	0.37	0.41
v_y	max	-0.008	0.21	0.46	0.44	0.45	0.47
	min	0.336	0.13	0.55	0.54	0.55	0.04
v_z	max	0.292	0.18	0.27	0.29	0.28	0.10
	min	0.073	0.18	0.25	0.27	0.26	0.26
a_x	max	0.185	0.14	0.36	0.44	0.40	0.08
	min	-0.174	0.22	0.29	0.31	0.30	0.18
a_y	max	-0.216	0.15	0.42	0.46	0.44	0.08
	min	0.186	0.14	0.52	0.52	0.52	0.12
a_z	max	0.161	0.15	0.28	0.33	0.30	0.19
	min	-0.183	0.18	0.26	0.29	0.28	0.20
v	max	0.232	0.19	0.28	0.23	0.25	0.14
	min	-0.111	0.25	0.20	0.25	0.22	0.24
a	max	-0.111	0.24	0.19	0.17	0.18	0.11
	min	0.122	0.22	0.15	0.19	0.17	0.13
a_t	max	0.119	0.23	0.19	0.25	0.22	0.16
	min	-0.152	0.19	0.25	0.31	0.28	0.19

Table A.5: Beat analysis from excerpt V1. p = precision; r = recall; $F*$ = F-measure without considering lag.

Descriptor	**Max / Min**	lag **(s)**	σ	p	r	F	F*****
x	max	-0.217	0.21	0.32	0.25	0.28	0.09
	min	0.145	0.20	0.56	0.40	0.46	0.15
y	max	0.208	0.03	0.95	0.88	0.91	0.00
	min	-0.093	0.31	0.88	0.88	0.88	0.02
z	max	0.014	0.29	0.35	0.25	0.29	0.29
	min	0.278	0.15	0.35	0.25	0.29	0.07
v_{x}	max	0.293	0.14	0.38	0.38	0.38	0.04
	min	-0.013	0.19	0.37	0.35	0.36	0.38
v_y	max	-0.055	0.07	0.96	0.96	0.96	0.94
	min	0.276	0.04	0.85	0.83	0.84	0.00
v_{z}	max	-0.271	0.13	0.47	0.48	0.47	0.10
	min	-0.019	0.18	0.45	0.44	0.44	0.42
a_{x}	max	0.137	0.14	0.43	0.44	0.43	0.21
	min	-0.199	0.17	0.32	0.33	0.33	0.14
a_y	max	-0.229	0.06	0.78	0.79	0.78	0.00
	min	0.084	0.28	0.89	0.85	0.87	0.09
a_{z}	max	0.204	0.13	0.45	0.44	0.44	0.11
	min	0.113	0.26	0.16	0.17	0.16	0.22
v	max	0.223	0.13	0.40	0.44	0.42	0.12
	min	0.184	0.17	0.37	0.40	0.38	0.30
a	max	0.141	0.16	0.24	0.25	0.24	0.28
	min	0.067	0.18	0.23	0.27	0.25	0.27
a_{t}	max	0.118	0.21	0.19	0.21	0.20	0.08
	min	-0.154	0.21	0.21	0.23	0.22	0.28

Table A.6: Beat analysis from excerpt V2. p = precision; r = recall; F^* = F-measure without considering lag.

Descriptor	**Max / Min**	lag **(s)**	σ	p	r	F	F*****
x	max	-0.291	0.17	0.28	0.17	0.21	0.08
	min	-0.031	0.34	0.21	0.15	0.17	0.20
y	max	0.175	0.13	0.95	0.88	0.91	0.00
	min	-0.167	0.03	1.00	0.88	0.93	0.00
z	max	0.009	0.27	0.16	0.10	0.13	0.15
	min	0.178	0.26	0.19	0.10	0.13	0.19
v_x	max	0.233	0.16	0.42	0.44	0.43	0.06
	min	-0.101	0.17	0.43	0.42	0.43	0.26
v_y	max	-0.094	0.01	1.00	1.00	1.00	0.02
	min	0.251	0.06	0.79	0.79	0.79	0.00
v_z	max	-0.243	0.21	0.10	0.10	0.10	0.19
	min	0.023	0.23	0.38	0.38	0.38	0.44
a_x	max	0.005	0.23	0.33	0.38	0.35	0.29
	min	-0.214	0.16	0.33	0.35	0.34	0.10
a_y	max	-0.263	0.05	0.83	0.83	0.83	0.00
	min	0.065	0.22	0.83	0.83	0.83	0.46
a_z	max	0.137	0.20	0.49	0.48	0.48	0.08
	min	-0.193	0.19	0.31	0.33	0.32	0.12
v	max	0.200	0.20	0.29	0.29	0.29	0.19
	min	0.180	0.18	0.32	0.33	0.33	0.37
a	max	0.074	0.20	0.32	0.31	0.32	0.29
	min	-0.164	0.21	0.34	0.38	0.36	0.26
a_t	max	0.107	0.18	0.31	0.33	0.32	0.26
	min	-0.103	0.20	0.19	0.21	0.20	0.24

Table A.7: Beat analysis from excerpt C. p = precision; r = recall; F^* = F-measure without considering *lag*.

Descriptor	**Max / Min**	*lag* (s)	σ	p	r	F	F*
x	max	-0.065	0.29	0.23	0.20	0.22	0.19
	min	-0.333	0.14	0.33	0.27	0.29	0.02
y	max	-0.263	0.07	0.74	0.71	0.73	0.02
	min	0.206	0.09	0.60	0.55	0.57	0.04
z	max	0.202	0.12	0.46	0.39	0.42	0.09
	min	-0.200	0.14	0.44	0.29	0.35	0.00
v_{x}	max	0.009	0.26	0.27	0.33	0.29	0.29
	min	0.294	0.16	0.35	0.37	0.36	0.12
v_y	max	0.295	0.08	0.61	0.61	0.61	0.02
	min	0.066	0.37	0.06	0.06	0.06	0.18
v_{z}	max	-0.033	0.32	0.36	0.41	0.38	0.32
	min	0.313	0.13	0.31	0.37	0.34	0.04
a_{x}	max	-0.239	0.17	0.19	0.24	0.21	0.12
	min	0.002	0.26	0.23	0.29	0.25	0.25
a_y	max	0.035	0.26	0.36	0.41	0.38	0.38
	min	-0.367	0.14	0.52	0.59	0.55	0.04
a_{z}	max	-0.265	0.17	0.27	0.37	0.31	0.03
	min	-0.032	0.24	0.17	0.20	0.19	0.26
v	max	0.128	0.29	0.08	0.10	0.09	0.14
	min	0.191	0.18	0.29	0.41	0.34	0.09
a	max	0.241	0.19	0.26	0.37	0.30	0.08
	min	-0.095	0.19	0.19	0.29	0.23	0.20
a_{t}	max	-0.223	0.24	0.20	0.29	0.24	0.10
	min	-0.112	0.21	0.18	0.24	0.21	0.17

Table A.8: Beat analysis from excerpt T. p = precision; r = recall; $F*$ = F-measure without considering *lag*.

Descriptor	**Max / Min**	*lag* **(s)**	σ	p	r	F	F*
x	max	-0.249	0.33	0.07	0.06	0.07	0.11
	min	-0.203	0.21	0.35	0.23	0.28	0.18
y	max	-0.066	0.40	0.33	0.21	0.26	0.05
	min	0.319	0.16	0.46	0.45	0.45	0.00
z	max	0.314	0.20	0.29	0.21	0.24	0.05
	min	-0.122	0.20	0.22	0.17	0.19	0.19
v_x	max	0.033	0.24	0.29	0.36	0.32	0.36
	min	-0.318	0.21	0.29	0.34	0.31	0.04
v_y	max	0.419	0.18	0.54	0.57	0.56	0.06
	min	-0.001	0.19	0.52	0.57	0.55	0.55
v_z	max	-0.027	0.21	0.15	0.19	0.17	0.22
	min	-0.318	0.22	0.26	0.32	0.29	0.13
a_x	max	-0.179	0.21	0.22	0.30	0.25	0.16
	min	0.180	0.21	0.27	0.34	0.30	0.19
a_y	max	-0.085	0.31	0.07	0.09	0.08	0.25
	min	-0.218	0.13	0.42	0.47	0.44	0.12
a_z	max	-0.173	0.20	0.19	0.23	0.21	0.23
	min	0.189	0.18	0.26	0.32	0.29	0.17
v	max	0.085	0.27	0.20	0.19	0.20	0.22
	min	-0.190	0.23	0.18	0.26	0.21	0.14
a	max	0.132	0.24	0.13	0.15	0.14	0.18
	min	-0.093	0.23	0.14	0.21	0.17	0.12
a_t	max	0.092	0.23	0.21	0.32	0.25	0.20
	min	0.105	0.22	0.12	0.17	0.14	0.17

Appendix B

Summary of published data

The following datasets are available online at

http://mtg.upf.edu/download/datasets/phenicx-conduct

B.1 Dataset of conductor movement during performance

- **Correspondence in dissertation:** Chapter 3
- **Included data:**
  - 32 microphones audio
  - Video and audio from a videocamera capturing the whole orchestra
  - From Kinect V1 facing the conductor (all at 30 fps):
    - MoCap with 3D position of nine joints: head, neck, torso, shoulders, elbows and hands.
    - RGB video (640×480 pixels).
    - RGBD video (320×240 pixels).
  - Aligned score
  - Audio descriptors computed by Essentia
  - Scores of fragments analyzed in this thesis

B.2 Dataset of spontaneous conducting movements

- **Correspondence in dissertation:** Chapter 4
- **Included data** (for each participant):
  - Audio and scores of excerpts from Beethoven's 3$^{\mathrm{rd}}$ Symphony (*Eroica*) 1st Movement performed by the Royal Concertgebouw Orchestra
  - MoCap data captured by Kinect V1 with 3D position of fifteen joints: head, neck, torso, shoulders, elbows, hands, hips, knees and feet.
  - Beat annotation (ground truth)
  - Beat prediction from hand acceleration data
  - Loudness prediction from body movement
  - Computed MoCap descriptors

B.3 Dataset of conducting movements performed with different articulation

- **Correspondence in dissertation:** Chapter 5
- **Included data** (for each participant):
  - Audio of synthesized musical excerpts
  - MoCap data captured by Kinect V1 with 3D position of fifteen joints: head, neck, torso, shoulders, elbows, hands, hips, knees and feet.
  - Beat annotation (ground truth)
  - Beat prediction from hand acceleration data
  - Computed MoCap descriptors

Appendix C

The MoDe library

MoDe is an open-source, LGPL-licensed C++ library for real-time MoCap feature extraction meant for creative purposes. As such, it has been developed to be easily compatible with OpenFrameworks[1], and it can be compiled as an addon for this creative suite.

It is available online at https://github.com/asarasua/MoDe/, where different examples for Mac OSX and Windows using Kinect V1 and V2 devices are provided. Its main characteristics are:

- Real-time differentiation from positional data using optimal filters proposed by proposed by Skogstad et al. (2012).
- Compatibility with any MoCap device. The library Application Programming Interface (API) allows to provide positional data using standard C++ containers, so it does not require to use any specific device or library.
- Easy handling of temporal events. The library allows to *subscribe* to events detected in the descriptors. For example: it is possible to get notifications when a given descriptor reaches a local maxima or changes its sign. This can be useful for triggering actions based on such events.

C.1 API description

All MoDe classes and constants are accessible within the `MoDe` namespace.

C.1.1 MoDeExtractor

The `MoDeExtractor` class is used to compute descriptors for a body or set of joints.

```
MoDe::MoDeExtractor modeExtractor;
```

[1] http://openframeworks.cc/

MoDeExtractor computes a number of features from all joints. It does not only give access to the current value of descriptors, but also to statistics such as the mean, standard deviation and RMS values. The number of stored frames from which this computation is done can be defined using the setup method. Default is 30 (1 second at 30 fps).

```
modeExtractor.setup(30);
```

Every time a new MoCap data frame arrives from the device, the update method must be called passing a map<int, MoDePoint> as argument. This map contains pairs of joints IDs and positional data for the received frame. The MoDePoint class is built on top of the ofVec3f class in OpenFrameworks[2]. It has has three member variables, x, y, and z, which allow to conveniently store 3D data such as position, velocity, or acceleration. It can cast vector<double> and vector<float> objects. Assuming we have an object device where the position of the jth joint at the current frame can be accessed through device.getJoint(j).getPosition(), the way to make ModeExtractor compute the descriptors for the current frame would be:

```
map <int, MoDe::MoDePoint> joints;
for (int j = 0; j < device.getNumJoints(); j++) {
  joints[j] = device.getJoint(j).getPosition();
}
modeExtractor.update(joints);
```

C.1.2 MoDeJoint

The MoDeJoint class objects contain all information from a joint, including the computed descriptor. If, for example, the right hand joint is associated with a constant RIGHT_HAND, it can be accessed using getJoint as:

```
MoDe::MoDeJoint rh = modeExtractor.getJoint(RIGHT_HAND);
```

C.1.3 MoDeDescriptor

Descriptors can correspond to a single joint (joint descriptors) or can be computed combining information from different joints (body descriptors).

Body descriptors

Body descriptors can be accessed directly from the MoDeExtractor object using constants defined in the MoDe namespace:

[2] http://openframeworks.cc/documentation/math/ofVec3f/

```
MoDeDescriptor qom = modeExtractor.getDescriptor(MoDe::DESC_QOM);
```

Joint descriptors

Joint descriptors are computed from a single joint. These can be accessed with `getDescriptor` from the corresponding `MoDeJoint` using constants defined in the `MoDe` namespace:

```
MoDe::MoDeDescriptor rh_vel = rh.getDescriptor(MoDe::DESC_VELOCITY);
```

This `MoDeDescriptor` object now contains different information accessible for the right hand velocity descriptor:

```
// MoDePoint with current value of the descriptor
rh_vel.getCurrent();
// double with current velocity in the y axis
rh_vel.getCurrent().y;
// MoDePoint with mean value during the configured nr of frames
rh_vel.getMean();
// double with current magnitude of the velocity vector
rh_vel.getMagnitude();
```

C.1.4 Event subscription: MoDeEvent and MoDeListener

MoDe provides easy handling of temporal events based on the values of the descriptors. This is done using two classes: `MoDeEvent` and `MoDeListener`.

`MoDeListener` is an abstract class. If we want a class `MyListener` to get events from a `MoDeExtractor` it must be declared as:

```
class MyListener : public MoDe::MoDeListener {
public:
  void newEvent(MoDe::MoDeEvent event){
    // do something with event
  }
}
```

When an object of class `MyListener` is instantiated, it can subscribe to the events of a `MoDeExtractor` object:

```
MoDe::MoDeExtractor modeExtractor;
MyListener listenerObject;
listenerObject.setExtractor(modeExtractor);
```

Once it is subscribed to the events, the newEvent method in MyListener will be called every time there is an event (local maxima, local minima, zero cross) on any of the descriptors computed by modeExtractor. The passed MoDeEvent object contains information about the detected event (type, joint, axis, value, feature). For instance, if we want MyListener to print a message every time the vertical velocity in the right hand joint reaches a local maximum, it must be declared as:

```
class MyListener : public MoDe::MoDeListener{
public:
  void newEvent(MoDe::MoDeEvent event){
    if (event.joint == RIGHT_HAND &&
      event.axis == MoDe::AXIS_Y &&
      event.type == MoDe::EVENT_MAXIMUM){
        cout << "Max at RH: " << event.value << " m/s" << endl;
      }
  }
}
```

Appendix D

Publications by the author

- Álvaro Sarasúa, Baptiste Caramiaux, and Atau Tanaka. Machine learning of personal gesture variation in music conducting. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pages 3428–3432, San Jose, CA, USA, 2016. doi: 10.1145/2858036.2858328.
- Álvaro Sarasúa, Mark Melenhorst, Carles F. Julià, and Emilia Gómez. Becoming the maestro: a game to enhance curiosity for classical music. In Proceedings of the 2016 International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), pages 1–4, Barcelona, Spain. 2016. doi: 10.1109/VS-GAMES.2016.7590338.
- Álvaro Sarasúa and Enric Guaus. Dynamics in music conducting: A computational comparative study among subjects. In Proceedings of the 2014 International Conference on New Interfaces for Musical Expression, pages 195–200, London, United Kingdom, 2014.
- Álvaro Sarasúa and Enric Guaus. Beat tracking from conducting gestural data: A multi-subject study. In Proceedings of the 2014 International Workshop on Movement and Computing (MOCO), pages 118–123, Paris, France, 2014. doi: 10.1145/2617995.2618016.
- Álvaro Sarasúa, Cyril Laurier, and Perfecto Herrera. Support vector machine active learning for music mood tagging. In Proceedings of the 9th International Symposium on Computer Music Modeling and Retrieval (CMMR), London, United Kingdom, 2012.
- Álvaro Sarasúa, Julián Urbano, and Emilia Gómez. Mapping by observation: building a user-tailored conducting system from spontaneous movements. In preparation, 2017.