Interactive computer-aided expressive music performance

Analysis, control, modification, and synthesis methods

MARCO FABIANI

KTH Computer Science and Communication

Doctoral Thesis in
Speech and Music Communication
Stockholm, Sweden, 2011

KTH Computer Science
and Communication

Interactive computer-aided expressive music performance

Analysis, control, modification, and synthesis methods

MARCO FABIANI

Doctoral Thesis
Stockholm, Sweden 2011

TRITA-CSC-A 2011:12
ISSN-1653-5723
ISRN-KTH/CSC/A–11/12-SE
ISBN 978-91-7501-031-1

KTH School of Computer Science
and Communication
SE-100 44 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framläggs till offentlig granskning för avläggande av teknologie doktorsexamen i datalogi onsdagen den 15 juni 2011 klockan 10.00 i F3, Sing-Sing, Kungl Tekniska högskolan, Lindstedtsvägen 26, Stockholm.

Tryck: AJ E-print AB

Abstract

This thesis describes the design and implementation process of two applications (*PerMORFer* and *MoodifierLive*) for the interactive manipulation of music performance. Such applications aim at closing the gap between the musicians, who play the music, and the listeners, who passively listen to it. The goal was to create computer programs that allow the user to actively control how the music is performed. This is achieved by modifying such parameters as tempo, dynamics, and articulation, much like a musician does when playing an instrument. An overview of similar systems and the problems related to their development is given in the first of the included papers.

Four requirements were defined for the applications: (1) to produce a natural, high quality sound; (2) to allow for realistic modifications of the performance parameters; (3) to be easy to control, even for non-musicians; (4) to be portable. Although there are many similarities between *PerMORFer* and *MoodifierLive*, the two applications fulfill different requirements. The first two were addressed in *PerMORFer*, with which the user can manipulate pre-recorded audio performance. The last two were addressed in *MoodifierLive*, a mobile phone application for gesture-based control of a MIDI score file. The tone-by-tone modifications in both applications are based on the KTH rule system for music performance.The included papers describe studies, methods, and algorithms used in the development of the two applications.

Audio recordings of real performance have been used in *PerMORFer* to achieve a natural sound. The tone-by-tone manipulations defined by the KTH rules first require an analysis of the original performance to separate the tones and estimate their parameters (IOI, duration, dynamics). Available methods were combined with novel solutions, such as an approach to the separation of two overlapping sinusoidal components. On the topic of performance analysis, ad-hoc algorithms were also developed to analyze DJ scratching recordings.

A particularly complex problem is the estimation of a tone's dynamic level. A study was conducted to identify the perceptual cues that listeners use to determine the dynamics of a tone. The results showed that timbre is as important as loudness. These findings were applied in a partly unsuccessful attempt to estimate dynamics from spectral features.

The manipulation of tempo is a relatively simple problem, as is that of articulation (i.e. *legato-staccato*) as long as the tone can be separated. The modification of dynamics on the other hand is more difficult, as was its estimation. Following the findings of the previously mentioned perceptual study, a method to modify both loudness and timbre using a database of spectral models was implemented.

MoodifierLive was used to experiment with performance control interfaces. In particular, the mobile phone's built-in accelerometer was used to track, analyze, and interpret the movements of the user. Expressive gestures were then mapped to corresponding expressive music performances. Evaluation showed that modes based on natural gestures were easier to use than those created with a top-down approach.

Acknowledgments

It was little less than five years ago when I moved back to Stockholm to start my PhD studies. Although it feels like yesterday, it has in fact been a pretty long part of my life. All the people I met in these years contributed, more or less directly, to this work. Some of them helped me with concrete suggestions, critics, solutions, and all that was related to my work. Some of them kept me sane by taking me out of my apartment to have some fun and distraction. And some of them did both. Here, without further ado, I would like to thank them all, and in particular ...

... my supervisor Anders Friberg. His precious suggestions, ideas for improvement, constructive criticism, and help with manuscripts have been a decisive contribution to my work. And although sometimes I would have preferred to have more strict deadlines from him, his positive and easy going attitude, and the fact that his door was always open made me feel very lucky in comparison to other PhD students I know.

... my co-supervisor Roberto Bresin. Without his help I would probably have never been a PhD student in the first place. It was him who first talked me into applying for the position and who found the money to pay for my salary. He also complemented Anders's more scientific mentoring with practical researcher's skills such as the art of navigating through EU project deliverables, meetings, reviews, and what not.

... the senior members of the Music Acoustics Group (Anders Askenfelt, Svante Granqvist, Erik Jansson, Sten Ternström, Johan Sundberg). Their strong dedication to music- and acoustics-related research has been a source of continuous inspiration.

... Olov Engval and Björn Flemsäter for reading my manuscript and giving me many useful suggestions for improvement.

... my fellow (ex)-PhD students. Thanks to their experience, they helped me cope with many difficulties, and at the same time we had a really good time, both in and out the office. My "musical supervisor" Kjetil for introducing me to music I had never heard, and making me play things I would have never imagined I could play. Anick, for the moral support and encouragement during some difficult times. Petko for the conversations about signal processing, bikes, and life in general. Giampi for the discussions over lunch about computers, LaTex, and Italian politics. My office colleagues Erwin, for all the precious tips about Matlab, and Glaucia, for the fun and open-hearted conversations. Gaël for helping with

the incredibly hard to program Nokia phones. And Kahl for teaching me so many funny things about Swedish culture.

...all the guest researchers and master students that visited the lab over the years. Here is a partial list, I apologize to those I might have forgotten: Anna, Ann-Christine, Birgitta, Clara, Dominic, Enrico, Engin ("lo scudiero"), Fabio, Matt, Matthias, Matyas, Smilen, ...it was fun to spend time with you all.

...all the people working at TMH (the Speech group, the administration, the Language Unit, the technical support, and the Director Musices Gunnar Julin) for creating such a nice environment, for all the *fikabröd*, *Luciagröt* and Christmas parties, for the *innebandy* matches (Daniel E., Kjel, Mats, Preben, ...), and for the "concerts" with the Formant orchestra. Being at TMH has been almost like being part of a big family.

...Jordi Bonada, Esteban Maestre, and all the other researchers I met at UPF's Music Technology Group in Barcelona during my visit in 2009, for all the ideas and suggestions, so many that I still haven't had the time to try them all.

...the musicians who played the samples in my recordings.

...the participants to the listening and evaluation tests.

...all the bright persons I met while working in the BrainTuning and SAME projects, at conferences, symposia, and summer schools.

...the Slobe: Kjetil, Björn and Maria.

...my neighbors and friends Alberto and Shilpa.

...La Dolce Vita team (Enrico, Federico, Gabriele, Luca, Magnus A., Magnus S., Mats, Pablo, Raffaele, Roberto, Tristan).

...my fiancé Su, for having been here with me in the last months, for waiting for (almost) all these years for me to "finish this PhD thing", and for the cover art!

...my parents Piera and Giampietro, for always giving me their full support, no matter what I chose to do and, the hardest part of all, where I chose to be, in this case 1938 km[1] away from home.

...and everyone else who does not appear in this list but feels nevertheless part of my effort. Thank you all!

This work was in part supported by the EU-funded projects BrainTuning (FP6-2004-NEST-PATH-028570) and SAME (FP7-ICT-STREP-215749).

[1] Google Maps, shortest route by car

Contents

Papers included in the thesis

This thesis is the original work of the candidate except for commonly understood and accepted ideas or where explicit references have been made. The dissertation consists of eight papers, and an introduction. The papers will be referred to by capital letters.

The principal contributions to all the papers, excluding Paper D, were made by the candidate. These include study design, data collection and analysis, algorithms design, software implementation, and manuscript preparation. Coauthors' contributions are stated below.

The papers are not contained in the electronic version of the thesis. Please refer to the author or the copyright owners for reprints.

Paper A

Marco Fabiani, Anders Friberg and Roberto Bresin. 2011. Systems for interactive control of computer generated music performance. To appear in: Alexis Kirke and Eduardo Miranda (Eds.), *Computer Systems for Expressive Music Performance*. Springer, Berlin.

Anders Friberg wrote parts of sections 2 and 4. Roberto Bresin wrote parts of section 2, and the entire section 5.

Paper B

Marco Fabiani. 2010. Frequency, phase and amplitude estimation of overlapping partials in monaural musical signals. In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-10)*. Graz, Austria.

Anders Friberg contributed to manuscript authoring.

Paper C

Marco Fabiani and Anders Friberg. 2011. Influence of pitch, loudness, and timbre on the perception of instrument dynamics. Submitted to *Journal of the Acoustic Society of America, Express Letters*.

Anders Friberg contributed to the experiment design and statistical data analysis, and to manuscript authoring.

Paper D

Kjetil Falkenberg Hansen, Marco Fabiani and Roberto Bresin. 2011. Analysis of the acoustics and playing strategies of turntable scratching. *Acta Acustica united with Acustica*, Vol. 97, pp. 303–314.

Kjetil Hansen performed the main part of the work. Marco Fabiani designed and implemented the algorithms for data extraction, and wrote the corresponding part of the manuscript. Roberto Bresin contributed to the planning of the experiment, while both Marco Fabiani and Roberto Bresin contributed to the analysis, interpretation and discussion of the results.

Paper E

Marco Fabiani. 2009. A method for the modification of acoustic instrument tone dynamics. In *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-09)*. Como, Italy.

Anders Friberg contributed to manuscript authoring.

Paper F

Marco Fabiani, Roberto Bresin and Gaël Dubus. 2011. Interactive sonification of expressive hand gestures on a handheld device. To appear in *Journal of Multimodal User Interfaces*.

Gaël Dubus contributed to the application design and development, and to the evaluation experiment design and data collection. Roberto Bresin contributed to the statistical data analysis and manuscript authoring.

Paper G

Marco Fabiani. 2011. *PerMORFer*: interactive rule-based modification of audio recordings. Submitted to *Computer Music Journal*.

Anders Friberg contributed to manuscript authoring.

Paper H

Marco Fabiani, Gaël Dubus and Roberto Bresin. 2011. *MoodifierLive*: interactive and collaborative music performance on mobile devices. In *Proceedings of the 2011 International Conference on New Interfaces for Musical Expression (NIME2011)*. Oslo, Norway.

Gaël Dubus contributed to the application design and development, in particular the collaborative mode. Roberto Bresin contributed to the manuscript authoring.

Other publications by the author

Marco Fabiani. 2011. Rule-based expressive analysis and interactive re-synthesis of music audio recordings. In Heinz von Loesch and Stefan Weinzierl, editors, *Gemessene Interpretation - Computergestützte Aufführungsanalyse im Kreuzverhör der Disziplinen*, volume 4 of *Klang und Begriff*. Schott, Mainz (in press).

Marco Fabiani, Gaël Dubus, and Roberto Bresin. 2010. Interactive sonification of emotionally expressive gestures by means of music performance. In Roberto Bresin, Thomas Hermann, and Andy Hunt, editors, *Proceedings of ISon 2010 - Interactive Sonification Workshop*, Stockholm, Sweden.

Marco Fabiani and Anders Friberg. 2007a. Expressive modifications of musical audio recordings: preliminary results. In *Proceedings of the 2007 International Computer Music Conference (ICMC07)*, volume 2, pages 21–24, Copenhagen, Denmark.

Marco Fabiani and Anders Friberg. 2007b. A prototype system for rule-based expressive modifications of audio recordings. In Aaron Williamon and Daniela Coimbra, editors, *Proceedings of the International Symposium on Performance Science (ISPS 2007)*, pages 355–360, Porto, Portugal.

Marco Fabiani and Anders Friberg. 2008. Rule-based expressive modifications of tempo in polyphonic audio recordings. In *Computer Music Modeling and Retrieval. Sense of Sounds*, volume 4969 of *LNCS*, pages 288–302, Berlin, July 2008. Springer.

Giovanna Varni, Sami Oksanen, Gaël Dubus, Gualtiero Volpe, Marco Fabiani, Roberto Bresin, Vesa Välimäki, and Jari Kleimola. 2011. Interactive sonification of synchronization of motoric behavior in social active listening of music with mobile devices. To appear in *Journal on Multimodal User Interfaces - Special Issue on Interactive Sonification*.

Part I

From design to implementation

CHAPTER 1

Introduction

Historically, there has always been a strong separation between those who play music (the musicians), and those who listen to it (the listeners). Until the advent of recording techniques at the end of the 19th century, music listening was limited to public spaces such as concert halls or theaters. Exceptions were those people with the means to have a small chamber orchestra play in their living rooms, or having a musician in the family. Recording made music accessible to a much wider audience, virtually at any time, in the comfort of the living room. Nowadays, digital music allows people to have entire musical collections available in the palm of their hand. Another important consequence of recorded music is the freedom for the user to choose what to listen to.

But even after more than a hundred years of developments in the field of music recording, production, and distribution, the separation between listeners and musicians has remained almost unchanged. On one hand, music listening can be seen as a passive process: the listeners may now have the luxury of being able to choose among millions of tracks, but they cannot influence the way in which the music sounds. On the other hand, playing music is an active process: even if musicians read a score composed by somebody else, they have the ability to change its performance in many ways. By simply modulating some of the basic musical parameters such as tempo and dynamics, a musician can easily express different styles and emotions, and make it sound more personal. Unfortunately, not everyone has the means and time to become a musician. Mastering a musical instrument is a process that requires a lot of practice (and perhaps some talent too). Besides, even musicians do not have the possibility of having an orchestra play in their living room.

The main goal of the work presented in this thesis is to help reducing the gap between playing and listening to music, between musicians and listeners. In other words, to allow non-musicians to feel as an active part of the music they are listening to, in a simple and intuitive way. The term *active listening* has been used in recent years to define the possibility for the listener to interactively control how the music sounds (Goto, 2007a,b). In a general sense, active listening can go from simple equalization, to the control of how different instrument tracks are mixed, to

more advance modifications to the interpretation of the musical piece. In this thesis, the latter category will be used to define active music listening: the possibility to influence, in a simple and intuitive way, the performance of a piece of music in a way that for example mirrors the mood of the listener, much like a musician is able to do. In practical terms, this means providing the listener with what I call a system for computer-based interactive control of music performance.

1.1 Interactive performance systems

Interest in the area of interactive control of music performance has been growing both inside and outside academia in recent years. The worldwide success of music-related video games such as *Guitar Hero*, *Rock Band*, and *WiiMusic* shows that the general public does not only like to listen to music, but also to play (with) it. However, the freedom to control a musical performance in these commercial examples is relatively limited. To score points, one has to hit the correct button or do the right movement at the right time, where "right" normally means matching the original performance. More unbounded expression is sometimes allowed through the so-called "jam" play modes.

Even though such commercial applications only appeared a few years ago, research in academia dates back more than thirty years. A large number of interactive music systems are described in the literature. These range from basic MIDI sequencers controlled by a few parameters, to systems for improvisation that map for example complex gestures to synthesis parameters or to robots performing music. This work focuses on the limited category of systems with which the user controls how a predefined musical score is performed.

How many have not, at least once, mimicked the orchestra conductor's gestures while listening to a piece of classical music? Questions like these led many researchers to develop interactive music performance systems. It is also the reason why most of these systems follow the well-established orchestra conducting paradigm. This paradigm appears to be very suitable to describe the type of interaction that such systems aim at providing: the computer is a virtual orchestra, which the user conducts by deciding how the score this orchestra is playing is supposed to be performed. The first paper included in the thesis (Paper A) offers a detailed overview of the systems based on the orchestra conducting paradigm. The aspects discussed in Paper A that are more relevant for understanding the design solutions adopted in the present work are briefly summarized in the following paragraphs.

Let us first look at how a generic interactive performance system works, with the help of Figure 1.1. Typically, but not necessarily, the user controls the performance through gestures, which are picked up by devices fitted with sensors or by videocameras. Various parameters describing the user's intentions are obtained from the analysis of the collected data. The gesture parameters are subsequently mapped to musical parameters that define how the performance should sound

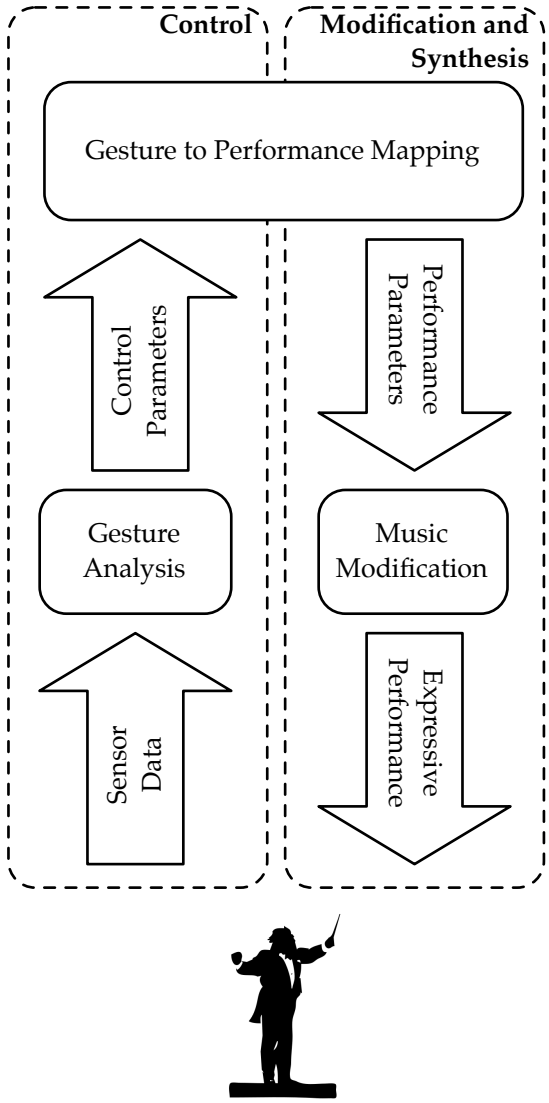


Figure 1.1: Block diagram of a generic interactive performance system.

like, according to the user's intentions. The performance is finally synthesized and played back, all in real time. This series of operations can be logically divided into two groups, i.e. those related to the interactive control and those to the performance modification and synthesis. The two groups are linked together through the mapping between the respective parameters. Within the two groups, several different design options are available, although some choices on one side can limit those on the other side.

To get a better idea of what the control and performance parameters are in practice, it might be useful to adopt the four-level classification based on the level of complexity/abstraction proposed by Camurri *et al.* (2001), and shown in Figure 1.2. At the bottom of the scale is the physical level, i.e. the signal (audio or sensor data); at the second level we find those cues that can be extracted directly from the signal, e.g. energy or direction for gestures, or tone's intensity and pitch for music; at an intermediate level we find descriptors related to segmented patterns, such as complex gestures, or tempo and phrasing for music; at the highest level are cues also known as *semantic* descriptors (Leman *et al.*, 2005), representing abstract qualities, such as emotions.

A parallel can easily be drawn between the general scheme shown in Figure 1.1 and the four levels of abstraction defined in Figure 1.2. In the control part of the system, we start from physical level signals (i.e. sensor data), and analyze them to extract control parameters at different levels of abstraction. These are mapped to performance parameters of the same level. The changes to the performance parameters are then propagated to the lower abstraction levels until the audio signal is synthesized.

Probably the most important choice relating to the control part of the system concerns the level at which the user can influence the performance, i.e. what is directly controlled by the user and what is automatically taken care of by the system (see Paper A and Chapter 3 for more details on the topic). Regarding the performance modification and synthesis, the main question seems to be whether to play symbolic scores through a synthesizer, or manipulate audio recordings of real performance (see again Paper A, and Chapters 2 and 4). Depending on the choices in these two key areas, different control and performance parameters become available, and different mappings are required.

1.2 Objectives and requirements

To concretize the goal of providing an *active listening* experience to the user, a lot of work has been dedicated to the development of applications for interactive music performance. Four main requirements were set for such applications. These requirements are listed here, together with the approach followed to fulfill them. The specific problems encountered during the development of such applications have been addressed using a combination of novel methods and algorithms, which are described in the included papers, and already available ones.

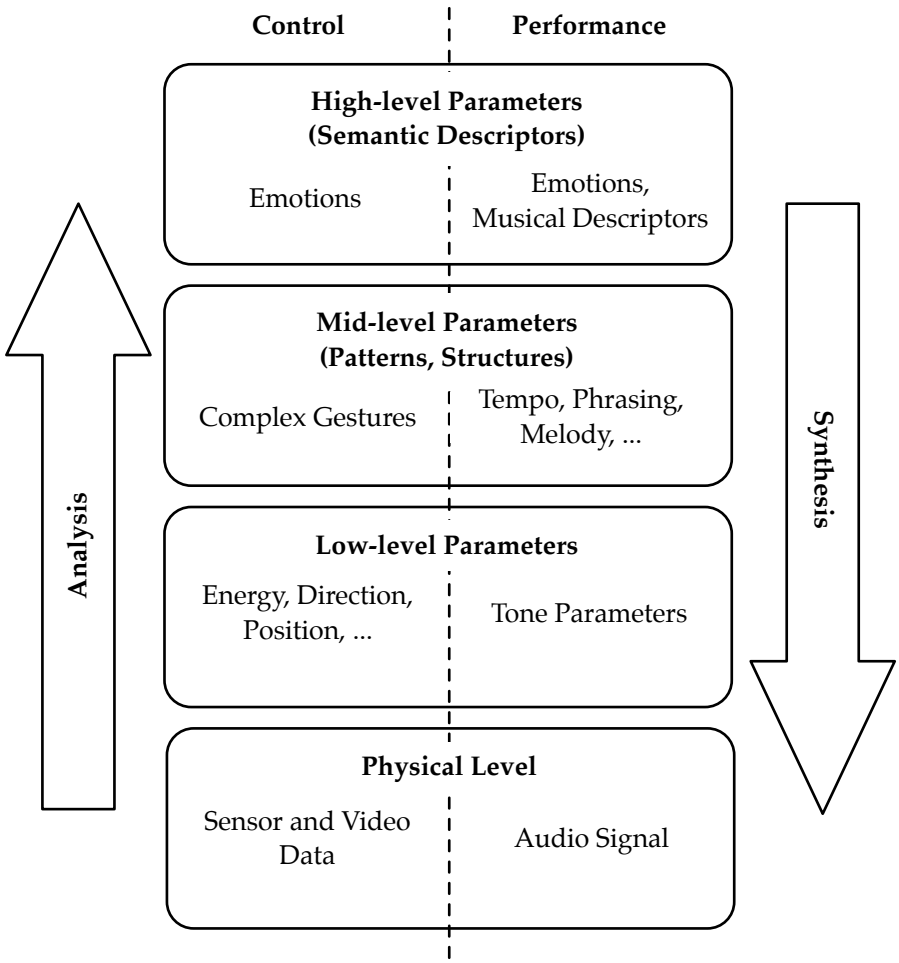


Figure 1.2: Multi-layered conceptual framework in which both performance and control parameters are divided into different levels of abstraction (adapted from Camurri *et al.*, 2001; Leman *et al.*, 2005).

1. Natural and realistic sound

Approach: Manipulation of real audio recordings

Motivation: Although high quality MIDI synthesizers are available on the market, they tend to be very expensive. Furthermore, a lot of tweaking is required in most cases to obtain realistic sounding results. In normal circumstances (i.e. consumer computers), the naturalness of the music produced by the default synthesizer is very poor. Audio recordings should solve this problem as they are consistent in all situations (i.e. they do not depend on the available synthesizer), and they contain small variations, nuances, and environmental noises that help to create a realistic experience.

Related problems: The modification of audio recordings, on the other hand, is far more complex than the modification of a score representation. Several advanced audio signal processing techniques are required. Three are the main problems with this approach. First, to be able to compute the new performance parameters' values, their original values should be known (i.e. performance analysis is required). Second, modifications should not introduce disturbing artifacts that would spoil the original quality of the audio recording. Third, the audio processing should be optimized to run in real time to make the system interactive.

More details: Chapters 2 and 4; Papers B and D

2. Easy and intuitive performance control

Approach 1: Rule-based performance modifications

Motivation: The direct control of low-level performance parameters requires musical knowledge to create coherent performance constructs (e.g. phrasing). The KTH rule system (Friberg *et al.*, 2006, and Section 3.2) allows even non-musicians to easily achieve pleasant performances with good musical standards by modeling the way in which musicians control mid- and low-level parameters during their performances.

Related problems: Modifications at the tone level are required to obtain the best results with the KTH rule system. As a consequence, the low-level characteristics of each tone to be modified need to be estimated. This is a trivial task for symbolic music, but very complex when audio recordings are used (see Requirement 1).

More details: Chapters 3 and 4; Paper A

Approach 2: Control based on semantic descriptors and expressive gestures

Motivation: The KTH rule system contains several parameters that might be difficult to control simultaneously in real-time. Semantic descriptors, such as adjectives communicating an emotional state, can be used in combination with gesture-based control interfaces that automatically recognize and map expressive gestures to the appropriate sets of performance parameters to allow the user a more immersive musical experience.

Related problems: The extraction of semantic features from gesture data requires advanced analysis methods, based for example on machine learning and pattern recognition. The use of semantic descriptors also implies the design of more complex mappings between control and performance parameters than those required by the use of low-level descriptors.

More details: Chapter 3; Paper F

3. Effective and realistic performance modifications

Approach: Modifications based on real instrument models

Motivation: As previously mentioned, the use of the KTH rule system implies tone level modifications to the audio signal. It is well known that when a tone is played on an acoustic instrument, its character (e.g. timbre, attack, sustain, release) changes depending on the values of the performance parameters. More realistic results can be obtained by modifying not only basic performance parameters such as duration and sound level, but also more advanced tone characteristics such as timbre, using for example specific instrument models.

Related problems: The use of such instrument models requires first of all an accurate estimation of the original tone's parameters (see Requirement 1). Furthermore, the models need to be as general as possible, but at the same time seek to preserve the original character of the particular instrument used for the recording.

More details: Chapter 4; Papers C and E

4. Portability

Approach: Multi-platform applications based on generic hardware and software.

Motivation: The experimental systems described in literature usually require specific hardware and software, as well as *ad-hoc* musical material to function, although some attempts have been made to create open frameworks on which to build extensible platforms. The applications developed within the present work

are based as much as possible on hardware, software, and musical material readily available to everyone. Furthermore, the code has been written with scalability and portability in mind, to be available on both computers and mobile devices.

Related problems: Certain compromises have to be accepted depending on the chosen platform. For example, the use of audio recordings is not feasible on mobile devices because of their limited computational power. Regarding the use of real audio, although the long term objective is to be able to use any available recording, the KTH rule system requires information currently only available in a symbolic score, which needs thus to be available alongside the audio data.

More details: Chapter 5; Papers G and H

1.3 System design overview

During the time spent working on the present thesis, I developed a few prototype systems in which I tried to incorporate some, if not all, the requirements contained in the previous list. The first prototypes were implemented in Matlab (Fabiani and Friberg, 2007a,b) and although they allowed the transformation of audio recordings using the KTH performance rules, they lacked real-time gesture-based control, and were not very portable. Another prototype application, named *PyDM*, was then designed with portability and real-time performance in mind. It was substantially a Python reimplementation of Friberg's *pDM* (developed in the Pure Data environment). *PyDM* supported real-time control of MIDI file using the KTH rules and semantic descriptors, but not audio manipulation. It worked on computers only, and did not support expressive gestures as control input.

As previously mentioned, considering the currently available technology, we have to reach a compromise between the use of advanced signal processing techniques required by the manipulation of audio recordings and the portability allowed by mobile devices. For this reason, two separate applications were finally developed by combining different parts of the earlier prototypes: *PerMORFer* (Paper G) and *MoodifierLive* (Paper H). The two applications somehow complement each other: the former focuses on the sound quality, i.e. audio recordings manipulation and the creation of natural sounding performances; the latter focuses on the ease of use, i.e. gesture-based control, and portability, i.e. works on mobile devices.

Conceptually, the two applications follow the general scheme adopted by previous systems (see Figure 1.1), including the prototypes from which they derive. The user's intentions are interpreted from control data (e.g. expressive gestures) and mapped to the performance parameters (in this case using the KTH rule system). The modified performance parameters are finally used to synthesize the performance. Nevertheless, as previously pointed, combining the manipulation of audio recordings with tone-level modifications based on the performance rules

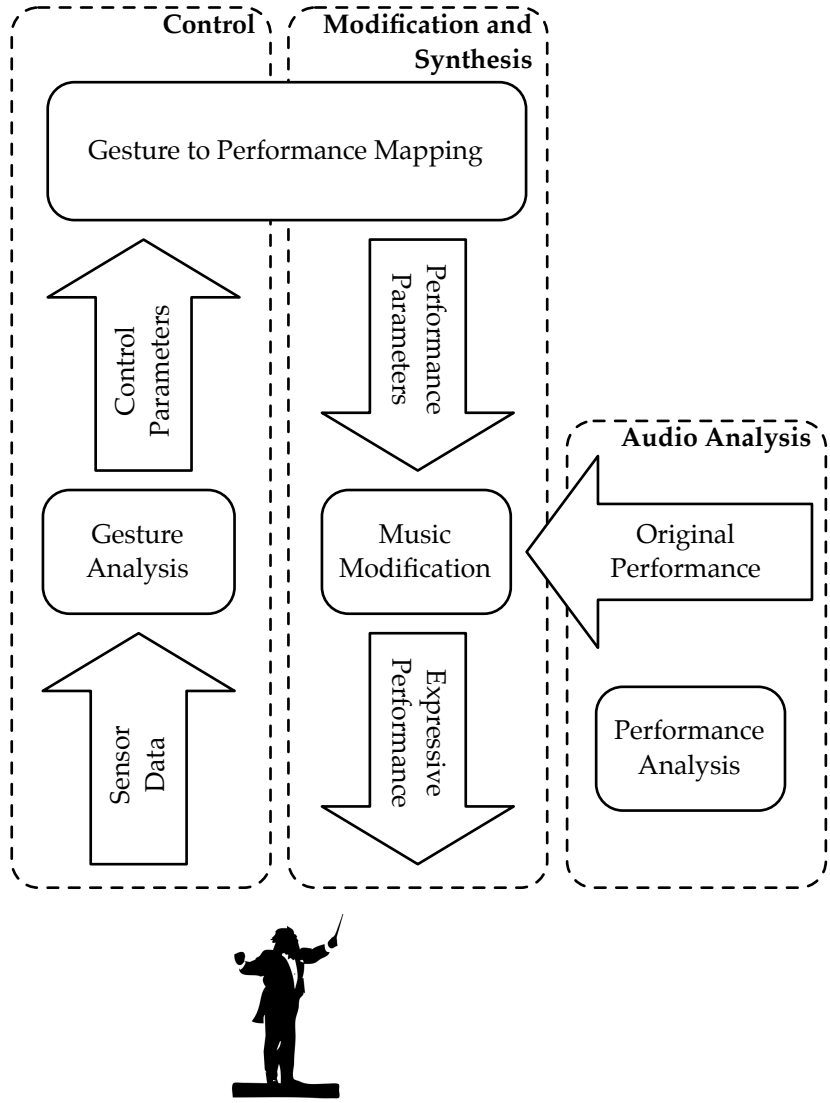


Figure 1.3: General block diagram of the interactive performance systems proposed in this thesis, using rule-based transformations of audio recordings.

requires an extra passage to estimate the parameters of the original performance. This is not necessary in case a symbolic representation of the music is used because the original performance is the nominal performance described by the score itself.

The design of *MoodifierLive*, which can only play MIDI files, thus follows Figure 1.1. On the other hand, *PerMORFer* includes an additional analysis stage, as shown in Figure 1.3. A similar block diagram was described by Amatriain *et al.* (2003) in a paper where the concepts of content-based transformations of musical audio signals was discussed from a theoretical point of view. Among the different scenarios described in the paper, systems similar to *PerMORFer* were also envisioned for the future.

1.4 Outline

The following chapters give some essential background information to help the reader see the content of the included papers in a more organic context. However, it is beyond the scope of this first part of the thesis to give a complete overview of all the fields related to the present work.

The structure of the thesis follows the same order in which the three main operations shown in Figure 1.3 (i.e. analysis, control, and modification/synthesis) are carried out in *PerMORFer*. In Chapter 2, I will first discuss the analysis of the audio signal for estimating the original performance parameters. Rule- and gesture-based performance control is discussed in Chapter 3. Note that the work presented in this chapter has been mostly carried out during the development of *MoodifierLive* but, as we will see later, it also applies to *PerMORFer*. In fact, *MoodifierLive* and *PerMORFer* share a large amount of code in the control section. I discuss then in Chapter 4 how modifications to the audio signal that match the desired performance parameters are obtained, and how the new performance is synthesized. The synthesis of a MIDI performance is trivial and is thus not discussed here. In Chapter 5, the two applications are summarized and compared to other systems. Finally, some general conclusions and ideas for future developments are presented.

CHAPTER 2

Analysis

The most noticeable difference between *PerMORFer* and the earlier systems, both MIDI- and audio-based, is the audio performance analysis section. The need for this analysis stage is a consequence of the kind of tone level modifications we want to achieve, as mentioned in Chapter 1. In MIDI-based systems (e.g. *pDM* Friberg, 2006) the parameters of the tone (i.e. pitch, dynamics, duration, IOI) are available in the symbolic score, and modifications to their nominal values can easily be obtained by changing the content of the MIDI messages sent to the synthesizer. In previous audio-based systems, changes are instead only made to the overall tempo and sound level, and not at the tone level. Some reference points, such as beat positions, are sufficient for this type of modifications.

The KTH rule system is used in *PerMORFer* to compute tone level modifications to an audio recording (see Section 3.2 for examples of the available rules and their function). A default modification value for each note and each rule is computed using the *Director Musices* computer program (Friberg, 1995) and saved in the score, along with the note duration and dynamic level. The effect of a single rule on the overall performance is controlled by a weighting factor, which can be changed by the user in real-time. If the score is played back through a MIDI synthesizer, the performance parameters (i.e. tone duration and dynamics, and tempo) are computed by combining the note parameters in the score with the weighted sum of the rule values.

When manipulating an audio recording, the values of the tone parameters in the original performance usually differ from the nominal values in the score. Consequently, they need to be estimated before applying the performance rules. For example, assuming that in the target performance a tone has to sound *mezzo forte*, different changes might apply depending on the original level of the tone: if it was played *fortissimo*, dynamics are lowered, if it was played *pianissimo*they will be raised, and if it was played *mezzo forte*they will be left untouched. The tone level performance modification in *PerMORFer* can be also seen as a two-stage process.

The original performance is first reduced to a so-called *deadpan* version matching the nominal performance found in the score. The exact same modification values used in the MIDI performance playback are then applied to the deadpan audio

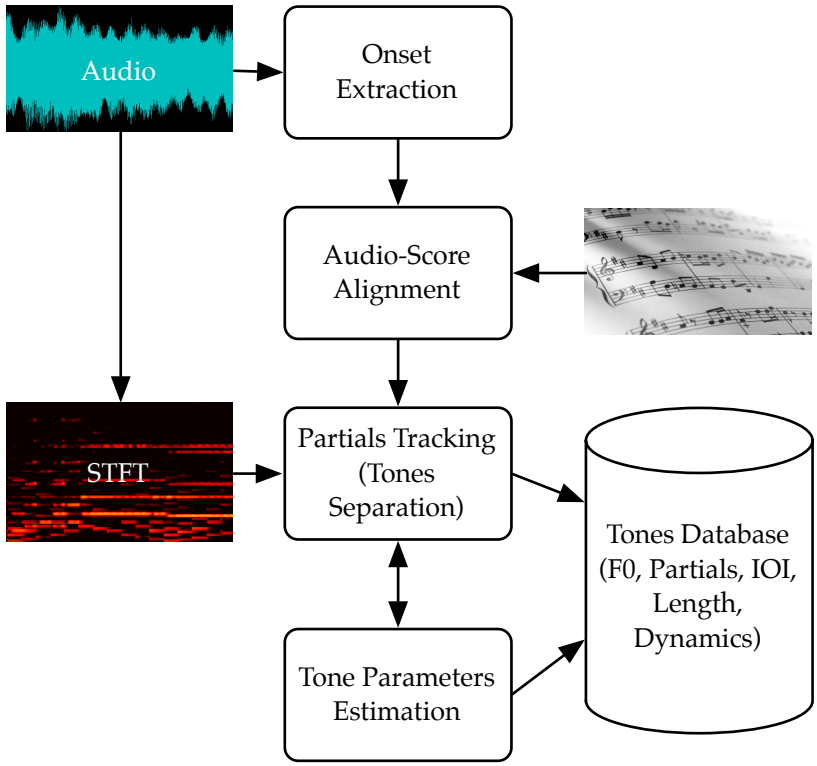


Figure 2.1: Block diagram showing the five main stages of the audio analysis process: onsets extraction, audio-to-score alignment, Short Time Fourier Transform (STFT) computation, partials tracking, and tones parameter estimation (adapted from Paper G).

signal. The two-stages are in practice merged by going directly from the original to the modified performance, thus bypassing the deadpan version. The audio signal performance analysis is a one-time operation, unless some of the analysis settings are changed, and it does not require real-time performance.

Let us now briefly look at how the analysis of the audio signal is executed in *PerMORFer* (for more details see Paper G). Figure 2.1 summarizes the five main stages in the analysis process. The tone onsets in the audio signal are first detected. Several methods are available to automatically extract tone onsets, some of which are summarized in Section 2.1. For the time being, the onsets are estimated outside *PerMORFer* using available tools and a list of their positions in time is fed to the program before the analysis begins. Although very accurate detection is possible, some manual corrections are almost always required.

Tone onsets are used here for the purpose of aligning the audio performance

with the corresponding score, which is required because it contains the default KTH performance rule values. Since the score is already available for the previous reason, it is also exploited in the audio analysis process, and in particular for partials tracking and the separation of tones (see Section 2.4 and Paper G). Several advanced methods for audio-to-score alignment are available as an alternative to tone onset matching. A short summary of these methods is presented in Section 2.2.

The final two steps in the analysis procedure are the estimation of the tones parameters (i.e. pitch, duration, dynamics, IOI), and the tracking and separation of their partials from the signal. This is done using a time-frequency representation of the audio signal, i.e. the Short Time Fourier Transform (STFT). A fundamental step that influences both the choice of techniques for parameter estimation and the final performance synthesis is the selection of what I call here the audio analysis/synthesis framework, which are introduced in Section 2.3. Although analysis and synthesis are strongly related, the latter is discussed more in detail in Chapter 4 together with tempo modifications. This choice is dictated by two reasons. The first is that tempo modifications are an integral part of the synthesis process; the second is to follow the chronological order of operations performed on the audio signal.

A common problem in parameter estimation and partials tracking is overlapping partials, i.e. sinusoidal components that have frequencies so close to each other that they are not distinguishable in the time-frequency representation of the signal. This problem is presented in Paper B together with a possible solution (see Section 2.4 for a summary).

Apart from being used to compute an expressive performance, the results from the audio signal analysis can also be studied to understand the mechanisms behind music performance in itself. Paper D describes a study where an unusual kind of music performance, e.g. DJ scratching, has been analyzed in order to link both acoustical and gesture low-level parameters (see Figure 1.2) to adjectives expressing emotions. This kind of music performance analysis is briefly discussed in Section 2.5, followed by a summary of Paper D.

2.1 Onset detection

Music can be seen as a succession of small events, the tones. Their onsets are thus a fundamental feature to be extracted from a recording. A comprehensive overview of the topic is given by Bello *et al.* (2005) and complemented by Dixon (2006). This section is slightly more detailed because onset detection has not been discussed elsewhere in the included papers.

Before talking about the different approaches to onset detection, it is useful to define, with the help of Figure 2.2, three important terms that are often confused: the tone's *attack*, *transient*, and *onset*. The *attack* of a tone is the time interval during which the amplitude envelope increases. The *transient* is usually a short interval

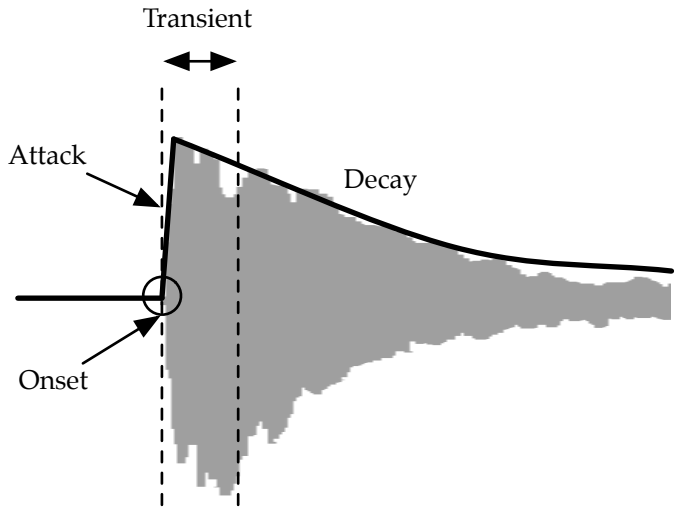


Figure 2.2: Attack, transient and onset in the ideal case of an isolated tone (adapted from Bello *et al.*, 2005).

during which the signal evolves quickly in some non-trivial or unpredictable way. The *onset* of the tone is a single instant that, in most cases, coincides with the beginning of the transient. In more realistic cases, signals are noisy and polyphonic and the distinction of transients from other parts of the tone is more difficult.

The standard procedure for onset detection can be broken down into three parts: signal pre-processing, used to enhance certain characteristics of the sound, such as impulsive parts; reduction of the signal to a *detection function*, which usually represents the level of *transientness*; extraction of the peaks in the detection function, which represent the onsets' estimated positions. The various techniques differ from each other mostly in how the detection function is computed.

Bello *et al.* identify two main approaches to pre-processing. In the first approach, based on psychoacoustics, the signal is divided into multiple frequency bands that approximate the behavior of the human auditory system (Duxbury *et al.*, 2002; Hainsworth and Macleod, 2003; Klapuri, 1999; Pertusa *et al.*, 2005; Rodet and Jaillet, 2001). It follows from the observation that humans can easily detect tone onsets while listening even to very complex sounds. After pre-processing, the bands can all be analyzed with the same method, or with different frequency-dependent ones. The second approach to pre-processing attempts to separate transients from steady-state parts of the signal. It is usually performed by analysis-synthesis techniques (see Section 2.3).

The concept of reduction, which is the most crucial part of the onset detection process, refers to the transformation of the signal into a detection function that better represents the occurrence of transients in the original signal. Three main

approaches for obtaining such a detection function have been used.

The first and most evident sign of the occurrence of a transient is a sudden increase in the signal's amplitude, which can be detected using for example an adaptive threshold on the amplitude's envelope, preferably if measured on a logarithmic scale (see Klapuri, 1999). Narrower peaks are also obtained using the derivative of the energy. Amplitude envelopes can also be used to detected tone offsets. For this purpose, Brosbol and Schubert (2006) use sparse coding decomposition, while Friberg *et al.* (2007) combine amplitude and pitch variations to improve the detection of slow attacks and releases. Tone offsets are detected in *PerMORFer* from the amplitude envelope using a threshold that adapts to the envelope's maximum value.

Other features indicating transientness can be found in the spectral structure of the signal. A transient is usually characterized by a wide band event in the frequency domain. The effect is more noticeable at higher frequencies, since the energy of the signal is normally concentrated in the low frequencies band. Frequency dependent weighting of the STFT, or the distance between successive STFT windows, also known as spectral flux, has been used to emphasize these spectral variations (Duxbury *et al.*, 2002; Masri, 1996). Besides energy variations, transients also produced time and frequency dependent phase variations in the STFT that can be exploited to produce detection functions (Bello and Sandler, 2003; Duxbury *et al.*, 2002).

More advanced techniques are based on alternative types of transforms (e.g. Wavelet), and on probability models where onsets detection is based on the assumption that a transient is usually an abrupt change that does not follow a predefined model (see for example Basseville and Benveniste, 1983).

The identification of peaks in the detection function is the final step of the onset detection process. Results from different detection functions can be combined, as in Friberg *et al.* (2007); Holzapfel *et al.* (2010). Bello *et al.* offers some guidelines for choosing the right detection function according to the application's requirements. Time domain methods are adequate for percussive sounds, while for strongly pitched signals spectral methods are more appropriate. Spectral difference is a good general choice, although more computationally demanding. For high precision detection, methods based on wavelet transforms are suggested. The best overall results are given by the statistical methods, but the computational load is much higher.

Since onset detection is not integrated in *PerMORFer*, any of the above methods can be used. However, the estimated onset positions are normally corrected by hand before being used in the program, because 100% estimation accuracy is hard to achieve. Wrong positions compromise the audio-to-score alignment and thus both the tone separation and parameter estimation (see following sections). Wrong tone parameter estimation consequently causes mistakes in the calculation of the performance transformations (see Chapter 4).

2.2 Audio-to-score alignment

Audio-to-score alignment refers to the procedure that seeks to find a correspondence between a symbolic representation of a piece of music, and an audio performance of the same piece. This means identifying and pairing the positions of relevant events in both the score and the audio signal. Note that not always a one-to-one correspondence between events exists, as musicians might miss a tone, or improvise.

As pointed out in a short but informative overview of the subject by Dannenberg and Raphael (2006), there are two different versions of this matching problem: an *offline* version, usually called score alignment, and an *online* version, commonly known as score following. The main difference between the two is that in the offline case, information about future events can be used in the matching, while only the past events are available in score following. I will focus here on offline methods because they are most relevant for *PerMORFer*, although large parts of the discussion can be extended to online methods as well.

According to Ewert *et al.* (2009), the process of matching an audio recording to a score can be broken down into three steps: feature extraction, feature distance computation, and optimal alignment estimation. To be able to compare the two representations, one must first identify and extract common features between the score and audio signal, such as for example onsets and pitches. Heuristic rules can be used in order to handle missing tones, as well as simultaneous onsets in the score that are separated in the audio signal.

The problem with such simple features is that the extraction accuracy is not always high enough, especially with polyphonic recordings. Alternatively, spectral features extracted from the STFT of the audio signal can be used. The same features are obtained from the score with a variety of methods. For example, an audio representation of the score produced with a MIDI synthesizer is analyzed by Woodruff *et al.* (2006) with the same techniques used on the audio signal. Models of the spectrum based on the assumption that tones have harmonic spectra have also been proposed (Raphael, 2008). Another popular approach is to compare the chromagrams derived from the audio signal and from the score (Hu *et al.*, 2003).

The second step in the process consists in the creation of the so-called *cost matrix*. The distance between all values from the two features vectors previously extracted from the score and the audio recording are computed. Different distances have been used, such as the Euclidean distance (Loscos *et al.*, 1999), or the cosine distance (Ewert *et al.*, 2009).

Finally, an optimal, minimum-cost path to go through the cost matrix is estimated. The most common approach to achieve this goal is by dynamic programming. In particular, Dynamic Time Wrapping is a standard method for aligning two given sequences (see e.g. Dixon and Widmer, 2005; Orio and Schwarz, 2001, for a detailed explanation). Unfortunately, dynamic programming is not suitable for online alignment, as it requires the complete cost matrix.

Hidden Markov Models (HMMs) on the other hand can be used both in offline

and online alignment (Loscos *et al.*, 1999; Raphael, 1999). Each note or chord in the score is represented by a node in the model. The optimal time of transition to the next node is computed based on the distance between the features vectors. A drawback with HMMs is the fact that they need to be trained.

Dannenberg and Raphael provide a list of some typical applications for audio-to-score alignment. Online score following is suitable for automatic accompaniment of solo musicians, for example for practice purposes. Several commercial systems are available for this purpose. For offline matching, they envision applications that would allow audio engineers to automatically identify and edit specific tones in a recording. For example, score-assisted source separation that relies on accurate audio-to-score alignment has been proposed by Raphael (2008). The onset-based score alignment implemented in *PerMORFer* might be substituted in future releases with one of the above-mentioned more advanced techniques.

2.3 Analysis-Synthesis frameworks

The choice of an analysis-synthesis framework in which to perform low-level analysis and manipulation of the audio signal is an important step in the design of an interactive system for automatic music performance. Amatriain *et al.* (2003) present an interesting theoretical analysis of what they define as *content-based transformation* of musical audio signals. They identify two possible approaches to such transformations: one in which the information obtained from the analysis of the signal is fed back to a transformation block working directly on the input signal, and another in which the analysis data is modeled and used as input to a synthesis block which generates a new signal.

The Phase Vocoder and sinusoidal modeling are popular options respectively for the first and second approach. Both frameworks take advantage of a time-frequency representation of the signal, normally the STFT, although other transforms such as Wavelets or Discrete Cosine Transforms have also been used. To compute the STFT, the sampled signal is first divided into chunks of constant length, usually overlapping in time. These chunks are multiplied by an analysis window (e.g. Hanning, Hamming, and Sine windows), and the Fast Fourier Transform (FFT) of the resulting signal is finally computed. The magnitude of the STFT is also known as the *spectrogram*.

The analysis-synthesis framework used in *PerMORFer* combines the Phase Vocoder-based approach, where the input data is modified and re-synthesized, with some techniques borrowed from sinusoidal modeling, such as partials tracking. Therefore, both frameworks are briefly summarized here.

Phase Vocoder

Thanks to its simplicity and the fact that in its basic form it is relatively easy to implement, the Phase Vocoder (Dolson, 1986; Flanagan and Golden, 1966) is one of the most popular frameworks for audio signal manipulation. The Phase Vocoder

is traditionally understood in two different ways: as a filter bank, or as a block-by-block FFT model (see for example Dolson, 1986; Zolzer, 2002). These two models are translated in the time-frequency plane as horizontal and vertical lines, respectively.

Looking at the Phase Vocoder from the FFT point of view, it is easy to see that what we call the analysis process is in fact the computation of the STFT with an analysis window $h_a(n)$ and frame overlap ratio (or step size) S_a between data chunks. The synthesis process consists of the computation of the Inverse FFT of each frame, followed by multiplication with a synthesis window $h_s(n)$, and overlap-add of the resulting signal chunks with a synthesis ratio S_s. Perfect reconstruction of the original signal is achieved if no changes are made to the STFT, $S_a \equiv S_s$, and appropriate analysis and synthesis windows are used.

As mentioned earlier, time scale modifications are an integral part of the synthesis process: to perform time stretching without changing the pitch of the tones, different frame overlap ratios are used for analysis and synthesis (i.e. $S_a \neq S_s$). For more details regarding time scale modifications the reader is referred to Chapter 4. Changes to the pitch of the tones as well as more advanced effects can also be obtained with the Phase Vocoder by manipulating the STFT (see for example Laroche and Dolson, 1999b). However, to avoid artifacts such as a reverberant sound or the so-called transients smearing (i.e. loss of sharpness in the tones onsets), the phase of the STFT needs to be corrected before the inverse transform can be computed. This follows from the fact that the phase of a sinusoid changes over time as a function of its frequency. If the frequency is changed (i.e. pitch-related transformations), or the time scale of the STFT is altered, the original relation between frequency and phase is lost and needs to be corrected. The phase is usually only corrected around the energy peaks in the spectrogram, since they correspond to the sinusoidal component (see for example the Phase-locked Vocoder in Laroche and Dolson, 1999a).

The re-synthesis of the modified audio signal is obtained in *PerMORFer* with *spectrogram inversion* (i.e. reconstruction of the time signal from the STFT's magnitude only) and not by correcting the phase of the modified STFT. This choice was made in order to avoid tracking all the phase changes introduced by the performance modification section (see Chapter 4).

Spectral modeling

A different approach to analysis and re-synthesis of audio is based on parametrical models of the spectrum of the signal. Sinusoidal models, first used for speech by McAulay and Quatieri (1986), are based on Fourier theory, which decomposes a signal into the sum of time-varying sinusoids. This model is well suited to musical signals since music is composed by tones with a more or less constant pitch and an harmonic spectrum. The model has been extended to accommodate parts of the signal that are not harmonic, such as transients and noise, which for example are represented by stochastic models (Serra and Smith, 1990). An overview of some

of the analysis-synthesis frameworks developed over the years can be found in Wright *et al.* (2000) and Beauchamp (2006). Examples include SMS by Serra and Smith (1990), SNDAN by Beauchamp (1993), and the IRCAM's analysis/synthesis suite (Rodet, 1997).

The analysis of the signal in spectral modeling involves the estimation of the model parameters. For the harmonic part of the signal, these parameters are the time-varying frequency, phase, and amplitude of the sinusoidal components. Section 2.4 discusses the problems related to the estimation of such parameters, as well as the tracking and separating of the single sinusoidal components. These operations are discussed separately because they are not only used in sinusoidal modeling but also for other purposes, such as in performance analysis (see Section 2.5), and source separation, which aims at isolating the different instruments in a polyphonic recording. Spectral modeling can be seen as a possible approach to source separation, as long as the partial tracks are grouped according to the instrument that produced them.

Compared to the Phase Vocoder, where the analysis is basically the computation of the STFT, spectral modeling requires more advanced techniques to accurately estimate the model parameters. On the other hand, transformations are easily obtained with sinusoidal modeling by altering the model parameters. The more accurate the estimation of these parameters, the more realistic the re-synthesized signal, which is normally obtained with additive synthesis.

2.4 Tone parameters estimation

The three parameters that mainly define a tone are pitch, duration, and dynamics. Accurate estimation of these parameters is an important task for many reasons. In sinusoidal modeling it is essential in order to correctly reconstruct the signal with the model. It is also the base for a correct performance analysis, which in turn is essential for successfully applying rule-based modifications such as those described in the present work. Finally, the tone parameters are also interesting for musicological or psychological studies, as we will see in Section 2.5.

Articulation estimation

Articulation can mean different things for different persons. A very general definition is offered by Cooper (1985):

> In music, an articulation is a sign, direction, or performance technique that indicates or affects the transition or continuity between notes or sounds. Articulations include ties, slurs, phrase marks, *staccati*, *staccatissimi*, accents, *sforzandi*, *rinforzandi*, and *legati*.

In the present work, though, articulation is used to define only the relative duration of a tone with respect to the IOI (i.e. articulation ratio), and is described by

such adjectives as *legato* and *staccato*. This is in line with the definition used in the KTH rule system.

Articulation is important for the expressive character of a performance. For example, Bresin and Battel (2000) studied the strategies used by different pianists performing the same score in order to have an overalpp *legato* or *staccato* perception. The key overlap depends on the duration of the note, and is usually 40% of the IOI for *staccato* articulation, while repeated notes were played with 60% duration.

Section 2.1 describes several methods to detect the onset of a tone, some of which also provide an estimate of the offset position. The duration of the tone is obtained by simply computing the difference between the onset and offset times. In Brosbol and Schubert (2006), an overall articulation index was defined as the total time that at least one tone is sounding, divided by the total length of the piece. Maestre and Gomez (2005) automatically extract expressive features from a monophonic saxophone recording by first segmenting the signal into different events (attack, sustain, release, and transition segments), and subsequently computing an amplitude and pitch contour, from which articulation and dynamics information can be extracted.

As explained in Paper G, articulation is estimated in *PerMORFer* by detecting the offset of each tone from its amplitude envelope. Some basic attempts were made to automatically identify the attack and release parts of the tone by fitting a segmented line to the envelope, but in the current version of the system, they are defined by a fixed length.

Pitch estimation

The term pitch refers to the perceptual sensation of how high or low a certain tone is. Although mainly related to the fundamental frequency, its perception also depends on the partials of which complex harmonic sounds are composed. Indeed, studies have shown that in certain cases it is possible to recognize a pitch even if the fundamental frequency is missing (see e.g. Schmuckler, 2004). Pitch is a mid-level feature, according to the classification described in Chapter 1. To estimate such mid-level features, one does normally start from the estimation of the physical low-level ones.

Pitch estimation can be broken down into three steps: estimation of the peak frequencies in the spectrogram, which usually represent sinusoidal components in the signal; partials tracking, i.e. grouping of these peaks according to the evolution of their frequency and amplitude in time; and estimation of the fundamental frequency of the tone according to the frequency relations among these partials. To define a sinusoidal model, one can limit the analysis to the partials tracking, unless pitch-related modifications are required. Fundamental frequency estimation is necessary for automatic music transcription (see Klapuri and Davy, 2006, for an overview). Since the score is already available in *PerMORFer*, only frequency estimation and partials tracking are needed.

Frequency estimation

The number of instruments playing simultaneously is a crucial factor influencing pitch estimation accuracy and the choice of methods to employ. Accurate frequency estimation is relatively easy to achieve in monophonic recordings, as long as the tones have a stable pitch. If a time-frequency representation of the signal is available, accurate results can be obtained by interpolating the real position of the energy peak within a frequency bin (see for example Ferreira and Sinha, 2005; Serra and Smith, 1990, as well as Paper B).

Frequency estimation in polyphonic recordings, on the other hand, represents a more challenging task. The main problem in this case is that of overlapping partials. As a consequence of the quasi-rational relation between tones in Western music scales, the partials that constitute different tones often have frequencies very close, if not identical, to each other. Although classic methods such as peak interpolation still work well in polyphonic recordings for non-overlapping partials, they are not accurate when the frequency of the two components are very close to one another.

The problem of overlapping partials has been address in Paper B, to which the reader is also referred for a more complete overview of the already available methods. The method described in this paper is summarized in the following section. An extension is also proposed here, and described in more detail in Appendix X. The extended version of this algorithm is used for frequency estimation in *PerMORFer*.

Summary of Paper B

The algorithm described in Paper B starts from the assumption that the STFT of a stationary sinusoidal component corresponds to the FFT of the analysis window, shifted and centered around the frequency of the sinusoid. The window used in this case was the sine window. The STFT of the sum of two overlapping components (i.e. that have a frequency falling in the same FFT frequency bin) is modeled as the sum of two window transforms, which are functions of the frequency, amplitude, and phase of the two sinusoids. The complex values of the three frequency bins centered around the peak in the STFT corresponding to the two overlapping partials are used to estimate the model parameters by solving a system of six equations (i.e. the real and imaginary parts of the three bins) in six unknowns (i.e. the frequency, phase, and amplitude of the two components). In Paper E, the system of equations was solved numerically using a grid search.

Evaluation using synthetic sinusoids showed that the method can accurately estimate the parameters of two overlapping partials even if their frequencies are moderately modulated, and for Signal-to-Noise ratios down to 30 dB. The algorithm can also be used to estimate the frequency of a single sinusoid. In this case, it performs better than a similar method (Ferreira and Sinha, 2005).

The method described in Paper B has two limitations. First, as the approach by

Ferreira and Sinha (2005), it was developed using the Odd Discrete Fourier Transform (ODFT). Nevertheless, it can be adapted to work with the more common DFT. Second, solving the system of equations numerically makes it very computationally demanding. A solution to this problem is proposed in Appendix X, where an analytical solution to the system of six equations is derived for the sine window case. The same procedure can be adapted to derive analytical solutions for other window types, such as the Generalized Hamming window family.

Partials tracking

Partials (or frequency) tracking consists in grouping the peaks identified in the STFT into tracks in which frequency, phase, and amplitude coherently evolve in time. Each track represents a partial, i.e. a time-varying sinusoid. Looking at the spectrogram, these tracks clearly appear as horizontal lines, but for a computer system the task of identifying them is not trivial. A short but informative overview of partials tracking can be found in Beauchamp *et al.* (2006).

The concept of frequency tracking was first introduced in speech analysis by McAulay and Quatieri (1986), and extended by Smith and Serra (1987) for musical applications. Peaks in consecutive time frames are connected following several heuristic rules, such as limits to the maximum variations of frequency and amplitude between two successive frames. Rules are used to decide when a track begins and ends (since tracks do not extend to the entire recording): for example, a track is deemed to be over only if the absence of a matching peak persists for several frames.

More advanced partials tracking methods try to look at the problem from a wider perspective, taking into consideration large time spans. Difficult problems to solve in frequency tracking involve fast or large frequency variations, such as vibrato or sinusoidal components crossing each other in frequency. Methods involving for example Hidden Markov Models (Depalle *et al.*, 1993) or linear predictors (Lagrange *et al.*, 2007), showed improved results over standard methods in these more complex situations.

Partials tracking can be simplified if some *a priori* knowledge about the signal is available. This is the case if an aligned score is provided along with the audio recording. Score-assisted partials tracking has been used for example by Raphael (2008) for separating the melody and the accompaniment lines in a polyphonic recording. The purpose of this method was not properly partials tracking, but source separation, i.e. the separation of different instruments in a polyphonic recording. The two topics are related because if we are able to group the estimated tracks based on the instrument that generated them, we have in fact achieved source separation.

However, having monophonic recordings of each instrument makes the pitch estimation much easier. For example, former systems for interactive modification of audio recordings used monophonic multitrack recordings (Borchers, 2009), even though they did not attempt tone-level modifications. An intermediate solution

could be to use advanced source separation techniques to pre-process polyphonic recordings and thus obtain separate tracks for each instrument.

An approach similar to the one proposed by Raphael has been used in *PerMORFer*, as explained in Paper G. A mask based on the fundamental frequency (determined by the score) and on the assumption that the tones have harmonic spectra is combined with the previously mentioned heuristic rules in order to identify which peaks in the spectrogram belong to the same sinusoidal component.

Dynamics estimation

The term dynamics is used here to make a distinction between the physical quantity that is the amplitude of the tone, and the composite perceptual quantity that is better described by musical terms such as *piano* or *forte*. Automatic dynamics estimation is a complex problem because, unlike duration and pitch, several factors contribute to its perception. Paper C presents a study that investigated these factors by systematically varying three parameters: pitch, timbre, and loudness. The results from this paper are interesting for at least two reasons. First of all, they justify the complex manipulations used in *PerMORFer* and described in Paper E for dynamics modification. Second, we can use them to create algorithms for dynamics estimation based on the same acoustical cues involved in its perception by human listeners. Because of the second reason, a summary of the paper is included here, followed by the description of an attempt at developing such an algorithm for dynamics estimation.

Summary of Paper C

An experiment is described which aimed at studying the effect of three perceptual tone features, i.e. pitch, loudness, and timbre, on the perception of the dynamics of isolated acoustical instrument tones. Tones from five instruments (i.e. clarinet, flute, trumpet, violin, piano) were used in a full factorial design. The features were varied over three values: pitch over three octaves (four for piano), loudness over three levels (i.e. low, mid, and high), and timbre over three spectral qualities (*pp*, *mf*, *ff*). After listening each stimulus only once the subjects were asked to judge as fast as possible its dynamics on a six-step scale between *pp* and *ff*.

The results for each instrument were separately analyzed using a three-way analysis of variance (ANOVA) with repeated measures. Significant main effects for all the three factors were found for all the instruments. The main effects' sizes, although in general quite large, varied appreciably among instruments. For example, the effect of timbre was markedly higher for clarinet than for flute. Pitch was found to be not so important for violin and trumpet. Although we found some significant interactions among the factors, their effect size was considerably smaller than that of the main effects. In general, our hypothesis that both loudness and timbre are equally important in the perception of dynamics, and that pitch plays a lesser role, was confirmed by the data from the experiment.

Dynamics estimation based on spectral variations

Because of its complexity, automatic dynamics estimation is a relatively neglected topic. Only in the last few years have computational methods been introduced in the study of music performance. They are used almost exclusively for tempo analysis through algorithms for onset detection and tempo estimation.

Although the sound level of a tone is normally used as an estimate for dynamics, it might not be a good estimate for several possible reasons. First of all, the radiation patterns of musical instruments introduce large variations in the measured sound level, depending on the position of the microphone. Room acoustics and the distance from the microphone are other influential factors. Furthermore, the relative levels between instruments might be altered during the mixing of multitrack recordings used in modern studios. Effects such as compressors and limiters also alter the sound level.

The results of Paper C show that timbre plays an equally important role in the perception of dynamics. It was previously shown by Nakamura (1987) that listeners could identify the intended dynamics of tones even if these did not have any relation with their sound level. As shown by several measurements (e.g. Luce, 1975), there are clear differences in the spectrum of tones played at different dynamics.

Compared to sound level, timbre appears to be a more constant factor. For this reason, I attempted to estimate the dynamic level of a tone from its spectral characteristics only. This study is summarized in the following section, although for many reasons it did not lead to particularly good results, and was thus not published.

A lot of attention in the past have been given to automatic identification of instruments (see for example Kitahara, 2007) based on their timbre characteristics. The methods described in literature build on previous perceptual studies on the similarity of instrumental tones. Probably the best known study of this kind is described in Grey (1977). In general, subjects are asked to rate the similarity between different tones. The ratings are then analyzed using Multidimensional Scaling (MDS) in order to place the tones on a multidimensional space based on their relative similarity. The dimensions of such a space are then related to acoustical features that can be extracted from the sound, for example spectral features like brightness, spectral slope, flux, and tilt, or the cepstral coefficients. These features can then be used to train machine learning algorithms, such as for example Support Vector Machines (SVM), to achieve automatic instrument classification.

These methods inspired my approach to dynamics estimation. There are, as we have seen, timbre differences between tones played with different dynamics on the same instrument, although these differences might be much smaller than those between two different instruments. The first acoustical features to be considered for this purpose were the features normally used for instrument classification. Other features were also tested, for example down sampled spectral envelopes, and the ratios between each partial and the fundamental frequency.

To experiment with this approach, the recordings collected for the studies presented in Papers C and E were used. Features vectors were extracted for all the tones and their dimensionality reduced using Principal Components Analysis (PCA) to in part reduce redundancy. The reduced feature vectors were used to train both SVM classifiers with different numbers of classes, and a SVM regressor. The sample labels used for training were the dynamics that the musicians were asked to perform. Several sets of features, as well as several different training data sets were tested. For example, tones were divided into octave groups in order to reduce the influence of pitch on the spectral features, as in Paper E.

Note that there is a difference in the dynamics estimation of impulsive (e.g. piano, guitar) and sustained tones (e.g. bowed instruments, woodwinds, brass instruments). For the former, a single measure is sufficient, since the dynamics cannot be modified once the tone has been played. For the latter, dynamics can vary over time. In this study, only spectral timbre features were used, while transient features, i.e. those related to the attack of a tone, were not considered. Since dynamics perception in impulsive tones relies more on the attack of tone (see Paper C) I focused here on sustained instruments. Features vectors were extracted for every time frame in the STFT representation of the signal. However, since the KTH rule system does not define variations within a tone, single values for a tone were obtained by either taking the median of the classification results, or the most recurring level.

Although in general the results from cross-validation were promising, with around 80% correct matches, the results deteriorated considerably (down to 60% or even lower) when the system was used to classify tones from a different instrument specimen. This suggests that despite cross-validation the classifier suffered from over-fitting.

The poor results of this study might be caused by several factors. First, the reduced number of samples and musicians used in the training is certainly one cause for over-fitting. Second, the correctness of the labels used for training the classifier had not been verified: they were based on what the musicians were asked to play during the recording, but significant uncertainty in the accuracy and consistency of a single musician. The regressor was trained using the same labels, and not a continuos measure of the level, as would have been more appropriate. A measure of loudness, based on the ITU-R BS.1770-1 standard, was also tried to train the regressor, but the results did not significantly improve.

The feature vectors that gave the best results were down-sampled and averaged spectral envelopes. My explanation, however not directly supported by actual data, is that these features represent a good compromise between the detail of high-resolution spectral envelopes, and the coarseness of classic features, such as those used to measure brightness, which can be used to distinguish between different instruments, but fail to point out differences within instruments.

2.5 Overall performance analysis

Tone-level performance modifications such as those achieved by *PerMORFer* are only a particular application for the estimation of low- and mid-level tone parameters with the methods described in the previous sections. These parameters are used in other fields related to music performance. In musicology they are used for example to study and compare different interpretation styles, and how they evolve over the years. Psychologists and neuroscientist look for connections between these parameters and the perception for example of emotions (Juslin and Timmers, 2010).

Paper D is included here as an example of the use of low- and mid-level parameters in the study of expressivity in music. Another reason for its inclusion is the fact that the type of audio and sensor data required developing specific analysis techniques because of their unique nature.

Summary of Paper D

This is a study that aimed at understanding if, and how, it is possible to create expressive performances with DJ scratching. Three DJs were asked to express a number of emotions through their scratching performances, and two kinds of data were collected. An audio signal of the performances was also recorded. Furthermore, sensors were installed on the record player and on the mixer's crossfader in order to detected the record's direction and speed of rotation, and the crossfader's position.

Because of the uniqueness of DJ scratching audio signals (e.g. extremely short events, absence of a defined pitch), standard techniques for audio features extraction, such as those implemented in the MIR Toolbox (Lartillot and Toiviainen, 2007), were found unsuitable in this situation. Algorithms that combine audio and sensors data were developed by the author in order to extract mechanical, acoustical, and overall performance features. Mechanical features include the crossfader's onset and offset times, the record's direction and speed, and the position, at any given instant, of the player's needle within the sample used for scratching. Acoustical features include tone onsets and offsets, sound level, spectral centroid, attack slope, and pitch. Overall features include gesture coordination and quantity of motion.

The analysis of the features extracted from the sensor data showed that the expressive practices used by DJs in their performances were very similar to those used by more traditional musicians, when they attempt to convey different emotions. However, the resulting audio representation of these performances was very far from that of common musical instruments, thus the need for custom analysis methods. Perceptual studies are still required in order to asses if listeners are able to understand the intended emotion only from the audio recording.

CHAPTER 3

Control

This chapter presents and discusses the problems related to the interpretation of the user's expressive intentions and their mapping to meaningful transformations of the performance parameters. Section 3.1 is a summary, and an extension, of Paper A. This paper contains a broad overview of the control aspects of interactive systems for automatic music performance, in particular those referred to as *virtual conducting*. Section 3.2 briefly summarizes the KTH rule system for music performance, on which both *PerMORFer* and *MoodifierLive* are based. Section 3.3 gives a brief overview of the relation between music and emotions, and how it can be used to create high-level performance control paradigms. Finally, Section 3.4 describes how the limited resources available in a mobile phone were used to build a gesture-based performance control interface (Paper F).

3.1 Control mode categories

In Paper A, a classification of the different performance control approaches found in the literature is proposed, based on a combination of two concepts. On one side, the considerations made by Buxton *et al.* (1980) regarding the balance between the level of control given to the user and the level of automatization of the system. On the other side, the concepts behind the levels of abstraction used to classify the performance and gesture parameters that were presented in Chapter 1. Three categories were identified in Paper A that directly correspond to the three top levels of abstraction in Figure 1.2:

Direct control. The user is given direct control to the low- and mid-level performance features, for example the overall tempo and dynamics, and the articulation of the single tones.

Model-based control. The system automatically takes care of low- and mid-level performance parameters using models of high-level performance features such as phrasing (i.e. tempo and dynamics arches), and micro-variations in articulation and dynamics. The user controls the model's parameters rather than the low- and mid-level performance parameters.

Control based on semantic descriptors. The user defines the overall character of the performance through semantic descriptors such as emotional or musical adjectives (e.g. sad, happy, angry; *con brio*, *allegro*, *maestoso*). These adjectives can be mapped to the performance models, or directly to low-level performance parameters.

The boundaries between these categories are rather loosely defined. For example, systems adding a fixed delay to the beat defined by the conductor's gestures could be classified either as direct control or rule-based control: the user directly controls the tempo, although the introduced delay models part of the behavior of a real orchestra.

Based on the user group that the application targets, one or more of these approaches are normally chosen. Users can also be grouped depending on their level of musical knowledge, as described in Friberg (2006). These groups are:

Conductor level. The group includes conductors who know what it means and how it feels to direct an orchestra. Most of the systems reviewed in Paper A have the more or less explicit goal of providing an emulation of orchestra conducting as realistic as possible. These systems have an obvious pedagogic application, both for students and expert composers who want to be able to practice at any time. Solutions adopted to achieve this goal include for example automatic recognition of standard conducting gestures. Unobtrusive methods that allow the use of real batons has been emphasized because it was observed that other control devices create a sense of discomfort in the users (Marrin Nakra, 2000). The systems were not always formally evaluated with real conductors, but it seems that some achieved acceptable results (Baba *et al.*, 2010).

Musician level. This category includes those who have never conducted an orchestra but are familiar with musical concepts such as phrasing, *ritardando*, *staccato*, and so on. These users might be less sensitive to such details as the weight of the baton or the response time of the orchestra, but they know how to perform a piece of music so that it sounds good. For this category of users, both direct and model-based control have been used because the users understand the meaning of the parameters of a performance model, and can use them directly to fine-tune their performance. This kind of control could also be used by conductors to experiment with their performances before meeting with the real orchestra, as suggested in Hashida *et al.* (2009).

Naïve level. Adults and children with no musical experience belong to this user group. Naïve users are considered here as those who enjoy music but don't know anything about the previously mentioned musical concepts. To allow them to easily achieve pleasant and musically correct performances (e.g. without sudden tempo and dynamics changes, or stops in unexpected positions), semantic descriptors have been previously used in conjunction with

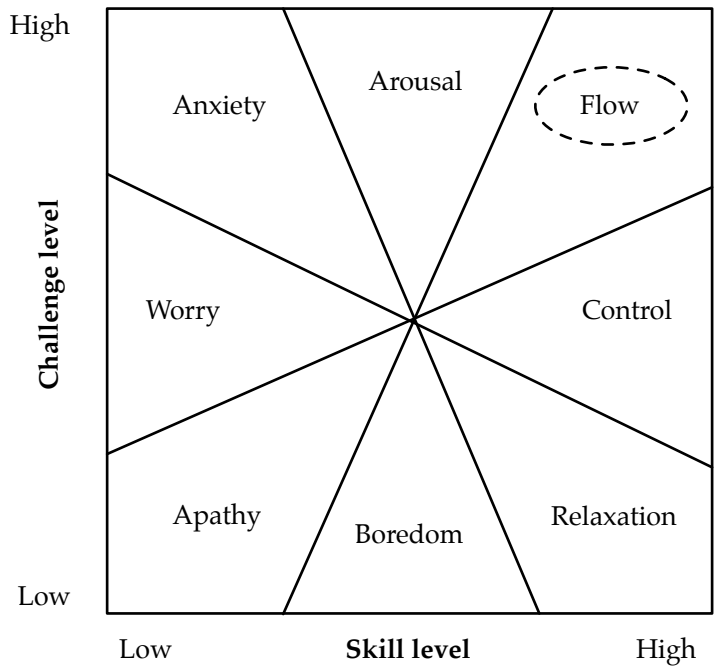


Figure 3.1: Mental states as a function of the skill of the user and the challenge level of the task (adapted from Csikszentmihalyi, 1998).

the KTH rule system (Bresin and Battel, 2000; Friberg, 2006). By employing for example emotional descriptors to define the desired performance, the relative underlying variations of the performance parameters are hidden from the user. These variations and their relation to semantic descriptors, which are well known to musicians, have been previously studied and modeled (see Section 3.3 for an overview).

Nothing prevents conductors to use semantic descriptors or a naïve user to use low-level control modes. These control levels can be looked at from the point of view of the *flow* theory (Csikszentmihalyi, 1998), which has applications in game design, education, and music, among other things. The flow theory was only referred to in Paper A, and it is thus briefly summarized here.

The flow theory states that in normal conditions only about one third of the mental capacity of an individual is used by the task at hand, so that we are able to decide what to focus on. But when we find ourselves in a state of flow, all the attention is allocated to the current task, and we loose awareness of everything else. According to Csikszentmihalyi, several mental states can be defined by the relation between the level of challenge and the level of skill for the task at hand,

as perceived by the user (see Figure 3.1). To enter in the state of flow, the levels of challenge and skill must both be above the average.

Applications can be design to adapt the level of challenge to the skills of the user, which improves with training, either continuously or step by step, as for example in computer games. The same approach has been used in the performance systems based on the KTH rule system, including *PerMORFer* and *MoodifierLive*. The naïve users can start with the semantic descriptors to get an idea of how a performance should sound. Once they get accustomed, they can step up and explore single performance rules and how they influence the performance. Finally they can create their personal performances taking either control of the single rules or of the low-level parameters. In this way it would be possible to maintain the state of flow for a longer period of time. If unskilled users are presented with a difficult task from the beginning, the risk is that they become frustrated.

PerMORFer and *MoodifierLive* mainly target the naïve group of users, following the requirements described in Section 1.2. They use both the KTH rule system (Section 3.2) and semantic descriptors (Section 3.3).

3.2 The KTH rule system

The performance control in both *PerMORFer* and *MoodifierLive* is based on the KTH rule system for music performance, which is thus briefly summarized here. For a complete overview, see for example Friberg *et al.* (2006).

The performance rules define contextual modifications to the low-level performance parameters such as tempo, note duration and dynamics. Some of the rules are applied automatically to each note based on their pitch and neighboring notes, while other rules require some additional information specified by the user, for example phrase boundaries or the definition of the melody line.

The following rules are a selection of the most commonly used ones:

High-Loud: increases the dynamic level as a function of the fundamental frequency.

Duration Contrast Articulation: inserts a micro pause between two consecutive notes if the first note has an Inter Onset Interval (IOI) of between 30 and 600 ms.

Duration Contrast: increases the contrast between long and short notes, i.e. comparatively short notes are shortened and softened, while comparatively long notes are lengthened and made louder.

Punctuation: automatically locates small tone groups and marks them with a lengthening of the last note and a following micro-pause.

Phrase Arch: each phrase is performed with an arch-like tempo and dynamic level curve, starting slow/soft, becoming faster/louder in the middle, followed by a *ritardando-decrescendo* at the end.

Final Ritardando: introduces a *ritardando* at the end of the piece.

Tempo: changes the overall tempo of the score by a constant percentage.

Sound Level: decreases or increases the dynamic level of all notes in the score by a constant value.

As previously mentioned (Chapter 2), default modification value for each note and each rule is saved in the score file, and controlled by a weighting factor k_n, which can be changed by the user in real-time. For example, given the nominal note duration d_S, the performance note duration d_P is obtained as (Friberg, 2006):

$$d_P = d_S + \sum_{n=1}^{N} k_n \Delta r_n \tag{3.1}$$

where Δr_n is the default deviation value of the nth rule, and N is the number of rules. The weighting factors can also be mapped to predefined semantic descriptors, as explained in the following section.

3.3 Emotions as semantic performance descriptors

There is undoubtedly a strong connection between music and emotions, which has been the subject of numerous studies (see Juslin and Sloboda, 2010, for an overview). Emotions have been previously used as semantic performance descriptors to simplify the control of automatic music performance in combination with KTH rules (see for example Bresin and Friberg, 2000; Friberg, 2005, 2006), an approach that has been followed also in the present work. I present here a short summary of the theories about music and emotions behind the development of the control paradigms described in Paper F.

Before discussing the connection between music and emotions, two apparently similar but distinct concepts need to be defined (Juslin and Sloboda, 2001): emotions *induced* and emotion *expressed* by music. Emotions can be *induced* by music, i.e. the listener's emotional state is modified. On the other hand, with the term *expressed* we mean instead those emotions that emerge from a detached analysis of the performance, and that can be easily recognized by the listener. Often, induced and expressed emotions do not coincide. For example, a piece of music can sound sad or happy, but not make the listener sad or happy as a consequence. A further distinction can be made between the emotion intrinsically expressed by the piece, and that expressed by the musician during the performance. The former is something controlled mainly by the composer for example by using the major or minor mode. The latter is what a system for interactive music performance can control. The validity of the first notion (i.e. emotions induced by music) is still under debate (Scherer and Zentner, 2001). On the other hand, there is wide agreement regarding the fact that music can express emotions. However, there is less agreement on *which* emotions can be expressed and how.

Which emotions?

Research about how music can communicate emotions has a long history that followed very often the most popular theories about emotional reactions at that particular time. Scherer (2004) points out that there is not yet a common procedure, or an established methodology, to run listening test in order to verify which emotions can be expressed with music. Three approaches are generally available to define emotions for such experiments: lists of basic emotions, the valence-arousal dimensions, and eclectic emotions inventories. According to Scherer, none of them is perfectly suitable for the task because they lack a theoretical analysis of the underlying process. However, from a practical point of view they are usually suitable to create high-level control mappings thanks to their inherent simplicity.

List of basic emotions. This theory is based on the assumption, derived from Darwin's evolutionary theory, that emotions can be reduced to between seven and fourteen basic emotions that are experienced in everyday life, for example anger, joy, or sadness. The model is widely used because it is particularly suitable for designing listening tests. With a limited number of answers to choose from, the analysis of the results becomes much easier. It also simplifies the definition of semantic descriptors for both performance and gestures. The experiment described in Fabiani *et al.* (2010), as other experiments previously carried out by my colleagues, employes such a model.

Nevertheless, the use of a small set of emotions in an empirical experiment can bias the results according to the designer's choice of adjectives to be included. Scherer identifies two other reasons for concern regarding the basic emotions approach. First, reducing the wide spectrum of emotions reported by scientific and fictional accounts to a small number of basic emotions does not adapt well to a musical context. Some theorists suggest the idea of emotions blending, but such a theory was never verified. Second, according to the basic emotions theory the cause of emotional reactions can be found in some prototypical situations like loss or threat, something that does not account for the main reason for listening to music, which is for pleasure.

Valence-arousal dimensions. Another widely adopted model which works well for listening tests is based on the idea that emotions can be described by the combination of two properties, i.e. the activity, also know as arousal, and the valence (Russell, 1980). While the activity represents the energy connected to an emotion (e.g. happiness is a high activity emotion, sadness is a low energy one), the valence distinguishes between positive (e.g. happiness) and negative (e.g. anger) emotions. The values of activity and valence define the position of the different emotions in a two-dimensional plane, which allows to easily determine how close they are to each other. It has been shown that this representation is also suitable for controlling emotionally expressive music performances (Canazza *et al.*, 2003).

According to Scherer, this model is more realistic than the previous one because it accounts for more nuances of emotions. However its resolution, i.e. the ability to distinguish similar emotions, is still relatively low. Furthermore, a two-dimensional space theory lacks a theoretical base that can explain the underlying mechanisms of emotions arousal.

Eclectic emotions inventories. A third approach to the empirical study of emotional reactions is to create a very large collection of different adjectives for the participants to choose from to explain their emotional state. Although this can enhance the resolution of the model, it is almost certain that it will not be possible to compare studies performed by different researchers. Despite the fact that the risk of bias is smaller than for the basic emotions approach, it might also happen that the terms used by the researcher are not exactly suitable, or that they are not fully understood by the listener.

Before running any experiment, it is very important to clearly define if the experiment is design to test *expressed* or *induced* emotion. The experiment's participants should be instructed appropriately. Induced emotions are usually subtler to determine than expressed ones. For induced emotions, the listener might get sidetracked by what he might be expected to feel, which is usually the expressed emotion. In case we are testing expressed emotions, a good idea is to use music unknown to the participants: this should reduce the effect of other factors related to emotion induction, such as particular memories connected to a known song.

Mapping performance rules to emotions

After choosing one of the models described in the previous section, its parameters need to be mapped to performance parameters so that changes in the emotions model are reflected in the expressive music performance. Both the basic emotions model and the activity-valence model have been used in *PerMORFer* and *MoodifierLive*. In particular, emotional models and performance parameters are mapped through the KTH performance rules, as in Bresin and Friberg (2000); Friberg (2006).

Macro-rules defined by sets of performance rule values are used to create performances that express the different basic emotions. Two approaches are available to create such prototypical expressive performances: extract the correct parameters from previous performances that have been judged to express the desired emotion in specific listening tests, or ask a group of musicians to produce those emotions by changing the rule values accordingly.

Following the first approach, Gabrielsson (1994, 1995) and Juslin (1997a,b) created a list of expressive cues that seem characteristic of some basic emotions (fear, anger, happiness, sadness, tenderness, solemnity). These cues, described in qualitative terms, concern tempo, sound level, articulation, tone onsets and decays, timbre, IOI deviations, vibrato, and final *ritardando*. Bresin and Friberg (2000) used this qualitative information as a starting point for defining the values of their expressive performances. Once the macro-rules had been set up, they were validated

with listening tests where the participants were asked to rate the emotional content of the synthetic performances.

The second approach has been used more recently by Bresin and Friberg (2011). A group of expert musicians was given the task to create several different expressive performances of five short musical pieces by setting the values of seven rules from the KTH rule system (i.e. tempo, sound level, articulation, phrasing, register, timbre, attack speed). The results showed strong agreement between the musicians, and were also similar to those used in the previous approach. The values for the prototypical performances obtained in this study have been used in both *PerMORFer* and *MoodifierLive*.

The basic emotion model is connected to the activity-valence one since the basic emotions can be described by their activity and valence values. Friberg (2006) proposed to define the activity-valence space in terms of the emotions at its four corners (i.e. joy, anger, sadness, and tenderness), using the macro-rules from the basic emotions approach. Intermediate emotional levels are obtained by interpolation.

3.4 Gesture control

We have seen how different performance control methods can be designed in relation to the skills of the user, and how this influences the available control parameters and their level of abstraction (Figure 1.2). Although point and click interfaces can be used to set the values of the control parameters, arguably the best way to create an interactive experience is to use gestures. The first reason is that conductors use gestures to direct an orchestra. The second is that it is easier in this way to control several parameters simultaneously and in real-time.

Paper A gives both a brief overview on how gestures have been mapped to different performance control approaches in previous systems, as well as on the devices available to collect the gesture data. To fulfill the portability requirement mentioned in Chapter 1, the work in this thesis regarding gesture-based control of music performance focused on the use of mobile phones. This requirement was fulfilled by developing an application working entirely on a mobile phone. Another reason for using such a devices is the fact that they are both easily available to everyone, and many contain an accelerometer that can be used to track gestures. However, the limited computational power and accelerometer resolution posed a limit to the complexity of the gesture analysis.

Mapping two heterogeneous quantities such as gestures and music requires a common set of parameters. In the case of *PerMORFer* and *MoodifierLive*, these common parameters are the semantic emotion descriptors described in Section 3.3, where their mapping to the music performance parameters was already described. In Paper F, summarized in the following section, a possible mapping between gesture data and semantic descriptors is described.

Summary of Paper F

Control modes Paper F (see also Fabiani *et al.*, 2010) describes two mappings between gesture data (i.e. accelerometer data from a Nokia N95 mobile phone) and emotion descriptors (i.e. activity-valence and basic emotions). These mappings were implemented as two control modes in *MoodifierLive*: the *Marbles in a Box* mode, and the *Free movement* mode. The data is processed directly on the phone, although the same control parameters can also be sent to a computer running *PerMORFer* using Open Sound Control (OSC) messages.

The *Marbles in a box* mode design was inspired by the *Home Conducting* system by Friberg (2005), in which hand gestures are captured by a webcam. The video data is analyzed to extract two features, i.e. the quantity of motion and the position of the hands in the space. These features are directly mapped to the activity and valence of the music performance, respectively. The rationale behind the valence mapping is that positive emotions are usually expressed with hands up in the air, while negative ones with hands down.

In the same way, the energy extracted from the accelerometer data was mapped in the *Marbles in a box* mode to the activity axis, while the vertical tilt of the phone to the valence. The tilt was chosen as an approximation of the position of the phone, since the actual position of the phone in the space cannot be accurately estimated from the accelerometer data only.

It was concluded from informal observations that it was difficult for users to obtain the desired performance if the mapping in the *Marbles in a box* mode is not explained beforehand. We thus decided to create the *Free movement* control mode, which is based on the analysis of expressive gestures that are freely performed by the user. In a pilot study (Fabiani *et al.*, 2010) eight subjects were asked to produce gestures that expressed one of the previously mentioned four basic emotions while holding a mobile phone. The accelerometer data were logged and subsequently analyzed.

By observing the subjects we noticed that all produced very similar gestures, although with different intensities. Energy and jerkiness, which describes the impulsiveness of a gesture, were extracted from the data and used as features to train a simple emotion classifier. The classification tree was chosen because it is simple enough to be implemented with the limited resources of a mobile phone. The outcome of the classification was then mapped to the macro-rules defining the four basic expressive performances by Bresin and Friberg (2011).

Evaluation The evaluation experiment is presented from the point of view of sonification, i.e. the use of non-speech audio to convey information (Barrass and Kramer, 1999). Sonification is useful to represent large amounts of data in a compact way, on several simultaneous channels, thanks to the discriminatory properties of the human auditory system. Interactive sonification is concerned with the representation of real-time data with sounds. According to this definition, we think that mapping complex expressive gestures to variations in a music perfor-

mance is, in a way, a form of interactive sonification.

The two modes were evaluated using a short task-based experiment. The participants were asked to shake and move the phone so that the resulting performance expressed a given emotion. The subjects were allowed to first test how the two modes reacted to their movements, but were not given any explanation regarding the mapping from gestures to performance parameters. Once they felt confident, their gestures and the corresponding activity-valence values were logged. After each task they were also asked to answer two simple questions regarding the success in obtaining the request musical performance, and the agreement between the performance and the gesture.

In general the answers showed that the users judged their effort quite successful with both modes, although with a preference for the *Free movement*. These results were confirmed by the analysis of the accelerometer data: the activity-valence values obtained with the *Free movement* were closer to the target emotions that for the *Marbles in a box*. This confirms the informal observation that lead to the design of the *Free movement* mode. The different results for the two modes might be explained by observing that while the *Free movement* mode is based on natural gestures performed by real persons, the *Marbles in a box* was create with a top-down approach.

CHAPTER 4

Modification and synthesis

Amatriain *et al.* (2003) point out that, although analytical studies have been carried out to determine the rules of expressive music performance (see Chapter 3), there is a lack of research on how to change the musical meaning of a recorded piece of music taking advantage of such rules. In this chapter, I will discuss some of the aspects related to the manipulation of the audio signal required to match the transform audio signal to the performance parameters. These operations are performed by the *Modification and Synthesis* functional block shown in Figure 1.3. This part of the work is aimed at achieving the third requirement listed in Chapter 1, i.e. to obtain effective and realistic performance modifications. The contributions to this goal are contained in Paper E and partly in Paper G.

Previous research has shown that variations of tempo, dynamics, and articulation account for about 90% of the communication of emotions in expressive music performance (Juslin and Laukka, 2003). The KTH rule system, and thus *PerMORFer* and *MoodifierLive*, focuses on the modification of these three parameters (see Section 3.2). Tempo and dynamics are the two parameters that can normally be controlled in the previous interactive systems based on audio recordings (see Paper A). While tempo modifications have been studied extensively and successfully implemented in many applications, dynamics transformations have only been superficially approached, usually by changing the sound level of the overall signal. As was shown in Paper C, timbre plays a role as important as loudness (i.e. mainly sound level) in the perception of dynamics. For this reason, an important part of the present work has been dedicated to studying dynamics transformations that could mimic the behavior of real instruments. Articulation modification received little attention in previous research. Even in the present work, this problem was only partially addressed.

Modifications of the properties of a tone are similar to the methods used in sample-based sound synthesis. Assuming that we can obtain perfect separation of each tone during analysis, the resulting signal is a sample which can be manipulated with the techniques used for example in samplers. The difference between a sampler and the approach proposed in this thesis is that while in the former a different sample is chosen depending on the desired tone's parameters, in the lat-

ter the original tone is kept and modified. An hybrid approach was proposed by Maestre *et al.* (2009), in which a performance is decomposed into separate tones that are recombined to obtain a new performance.

4.1 Dynamics modification

It is intuitively clear and also verified by measurements (e.g. Luce, 1975), that the timbre of acoustical instruments vary in relation to the dynamics at which they are played. Paper C also showed that these timbre variations play an important role to the perception of dynamics by the listener. Let us assume that we have an estimate of the dynamics of a tone in a recording, obtained for example with the method described in Section 2.4. How do we then manipulate this tone to obtain a realistic change of dynamics? The assumption made is that to obtain realistic transformations, one needs to modify both the sound level and the timbre of the tone in order to match the behavior of real instruments. A technique based on models of these timbre (i.e. spectral) variations is described in Paper E.

Summary of Paper E

The paper describes a model-based technique, later included in *PerMORFer*, for the modification of the dynamics of single tones. The transformation models, available for four instruments (violin, trumpet, clarinet, and flute) were obtained by analyzing a large number of tones played at several different dynamic levels, which were recorded by the author specifically for this purpose. The recordings were also used in the experiment presented in Paper C, and for the dynamics estimation approach described in Section 2.4.

The musicians were instructed to play about 30 pitches at five dynamic levels (*pp*, *p*, *mp*, *mf*, *f*, *ff*), with a sustain part of about 3 to 5 five seconds. The harmonic and residual parts of each tone were separated using the partials tracking techniques described in Section 2.4. The average spectral envelopes of the sustain section for the harmonic and residual parts of the tone were computed. To obtain more generic models and reduce the number of envelopes, tones belonging to the same octave were once again averaged. The loudness of each tone was also estimated using the algorithm specified by the ITU-R standard BS-1770-1 (ITU-R, 2007), and averaged over each octave and each dynamic level. The model thus consists of a table of average logarithmic spectral envelopes, one for each instrument, octave, and dynamic level.

Note that shaping the spectrum of the tone to be modified with the model's envelope would mean removing the character of the instrument that played the original tone, as well as the small variations appearing during the evolution of the tone. To preserve the original character of the tone, a *differential* envelope is obtained from the difference between the model envelopes corresponding to the original and the target dynamics. The differential envelope is then added to the spectrum of the original tone. This is based on the assumption that the tone's

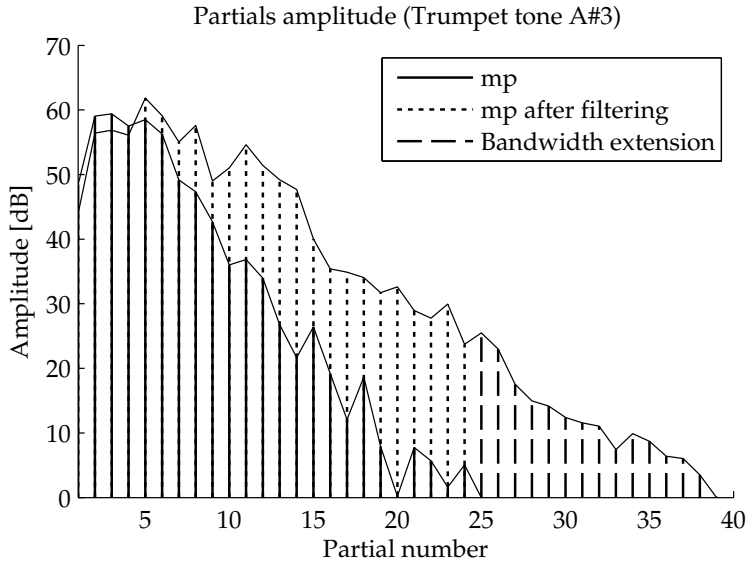


Figure 4.1: Amplitude of the partials of a *mezzo piano* trumpet tone before (solid lines) and after (dotted lines) applying a differential envelope to transform it to *forte*. Additional partials (dashed lines) have been added using the bandwidth extension technique (adapted from Paper E).

spectrum is shaped by two filters, one characteristic of the instrument specimen that does not depend on dynamics, and one that depends on dynamics and is common to all the specimens of that instrument. By computing the differential envelope, we are left with only the latter. At the same time, using a differential envelope allows us to preserve the time-varying small fluctuations present in the original tone.

The spectrum of the residual can also be modified using the differential envelopes. The two parts of the tone were studied separately because it was observed that they often vary in opposite ways. In flute tones for example the level of the residual increases while the dynamic level decreases as a consequence of the breathiness typical of soft tones. However,residual modifications can only be applied in monophonic recordings since it is not possible to distinguish the residuals of different tones in polyphonic recordings.

An improvement to the method, called *spectral extension*, was also introduced after observing that high frequency partials are not detected in the analysis of tones with low dynamics, since they are below the noise level. A differential envelope to raise the level of such tones would only influence the amplitude of the low frequency detected partials. However, high frequency partials are very important for the perception of a high dynamics because they contribute to the brightness of

the timbre. With the spectral expansion technique, synthetic high frequency partials are added to the original tone. Their frequency is computed by assuming that the spectrum is perfectly harmonic, and their amplitudes are based on the model's envelopes. The result of transforming a *mp*trumpet tone into a *f*one is shown in Figure 4.1.

4.2 Articulation modification

According to the definition that was given in Section 2.4, articulation includes such tone characteristics as its duration relative to the IOI, as well as the shape of the attack and release, and the timbre. In the specific case of the KTH rule system, articulation is only related to the duration of the tone.

Not many examples of articulation modifications in audio recordings can be found in the literature. One example is *SaxEx* by Arcos *et al.* (1997), a system that allows for case-based expressive transformations of saxophone solo recordings. An inexpressive musical phrase is transformed according to the character of the most similar expressive phrase retrieved from a database. Audio manipulations are applied in a sinusoidal modeling framework (i.e. SMS Serra, 1989). Observe that *SaxEx* produced modifications to both the duration of the tone, i.e. *staccato-legato*, and the position of its attack (i.e. swing), although not to its envelope.

Articulation has been only briefly addressed in the present work. *Staccato-legato* transformations are obtained by simply removing or duplicating parts of a tone while leaving the overall length of the signal unchanged, as explained in detail in Paper G.

4.3 Tempo modification

Expressive modifications of tempo are described e.g. in Fabiani and Friberg (2008); Gouyon *et al.* (2003); Grachten (2006); Janer *et al.* (2006). Such transformations to the time-scale of a signal rely on time stretching algorithms. Because of its various practical and commercial applications, time scale modifications of audio recordings have been extensively studied over the years. A rudimental way to change the tempo of a recording is to play it at a different speed. However, with this method the pitch of tones is also affected. Time scale modification methods that preserve the pitch are briefly summarized here. These approaches can be grouped into three categories: time domain methods, Phase Vocoder-based methods, and model-based methods.

The most simple time domain method is based on the segmentation of the signal into very small data chunks. To lengthen or shorten the signal, segments of data are discarded or repeated, respectively. Artifacts can occur at the junction between segments that were not previously adjacent, because of phase and amplitude discontinuities. This problem was addressed by using overlapping data segments. Thus, better amplitude and phase continuity is obtained when the seg-

ments are overlap-added (OLA), especially if they are correctly aligned, as for example in the Synchronous OLA methods. Time domain techniques are simple to implement and require less computational power, but the quality of the results are quite limited.

Time scale modifications based on the Phase Vocoder (see Section 2.3) also use the overlap-add approach, but in the time-frequency domain (i.e. STFT). Time stretching is obtained by using different hop sizes (i.e. the amount of overlap between frames) during the analysis and synthesis of the signal. There are two main problems with this approach. The first is the artifacts caused by the same amplitude and phase discontinuities encountered in time domain methods. To obtain a constant amplitude, appropriate combinations of analysis/synthesis windows and hop sizes are used. Phase propagation is used instead to maintain the continuity of phase in time, i.e. the phase of each bin in the STFT is corrected based on its instantaneous frequency. The second problem affecting the original Phase Vocoder approach is transient smearing (transients are not as sharp as in the original signal), and "phasiness" (a sort of reverberation effect). These two artifact are caused by the loss of *vertical phase* coherence, i.e. the phase relations between bins in the same time frame. The Phase-locked Phase Vocoder approach address these problems by concentrating the phase correction only around the energy peaks in the STFT (normally sinusoidal components), locking the phase of the remaining bins to that of the closest peak. Further improvements were introduced by Bonada (2000): higher stretching ratios were obtained by discarding or duplicating certain frames, and to further reduce smearing, the transients were detected and left unchanged. These solutions proved to be very effective in a series of listening tests (Honing, 2006), where the quality of the time stretched version was comparable to that of the original recording.

Finally, if the audio signal is modeled using for example sinusoidal modeling, the tempo can be modified by simply changing the model's parameters. This method works very well for the harmonic part of a signal, but not for transients. Verma *et al.* (1997) proposed an extension to Serra's SMS sinusoidal modeling method that includes a model of the transients.

It appears clear that time scale modifications are achieved by first analyzing a signal, and then altering the value of some parameters, for example the hop size, during the re-synthesis. The Phase Vocoder approach, combined with the preservation of transients and the removal and duplication of frames, was also used for time scale modifications in *PerMORFer*. However, the approach to the re-synthesis of a time domain signal defers from that of the Phase Vocoder, as explained in the following section.

4.4 Synthesis

We have seen in Section 2.3 how to synthesize a time domain sound using the Phase Vocoder and sinusoidal modeling. We have also seen how time scale modi-

fications with the Phase Vocoder require keeping track of the phase changes in the STFT using phase propagation (Section 4.3). However, several transformations are performed on the STFT in the three modification stages of *PerMORFer*. Keeping track of all the different changes in order for example to use the phase-locked Phase Vocoder seemed rather cumbersome. To obtain a coherent time domain signal from the modified STFT, Spectrogram Inversion, i.e. a technique that attempts to estimate the phase of the STFT from its magnitude only, was used in *PerMORFer*, as explained in Paper G. The part of the paper concerned with tempo modifications is summarized here.

Spectrogram Inversion

The spectrogram inversion goal is to iteratively estimate the phase of the STFT of a real signal starting from its magnitude only. The G&L algorithm (Griffin and Lim, 1984) is probably the first and best known method for spectrogram inversion. However, this algorithm is not suitable for real-time applications because it requires the complete spectrogram of the signal, which is not always available, and it is very demanding in terms of computational power.

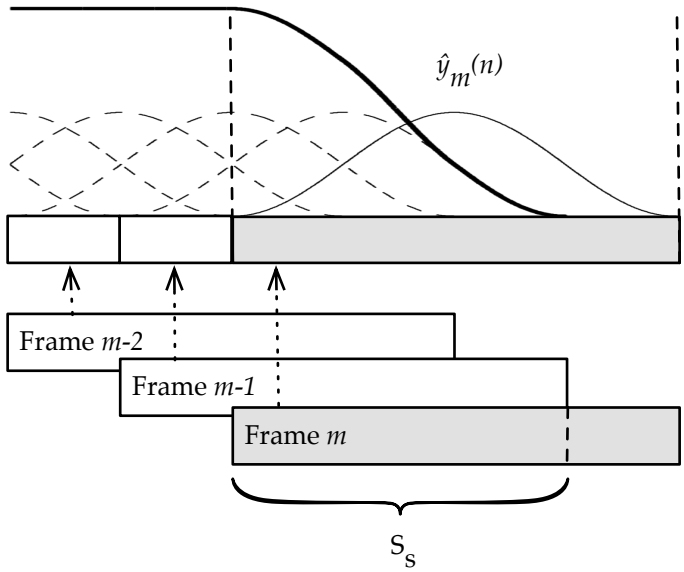
Zhu *et al.* (2007) implemented a real-time version of the G&L algorithm, the so-called Real-Time Iterative Spectrum Inversion with Look-Ahead (RTISI-LA). To compute an estimate of the current frame's phase, the RTISI-LA algorithm only uses information from the previous frames, and eventually in a small number of the following ones (Look-Ahead option).

When the percent of overlap between frames is kept unchanged from analysis to synthesis (i.e. $S_a = S_s$) and no Look-Ahead is used, the basic RTISI algorithm works as follows. Let us assume that we want to estimate the phase of the STFT's mth frame from its magnitude $|X_m(k)|$. Let us define the previously reconstructed and overlap-added signal, i.e. until frame $m-1$, as $y_{m-1}(n)$ (see Figure 4.2a, bold line). The current frame's initial phase estimate is computed by placing an analysis window on $y_{m-1}(n)$ and centering it at the position of frame m. If we assume $S_a = S_s = 75\%$, then the last 25% of the resulting data is all zeros. The FFT of this data, $Y_m(k)$, is then computed and its phase combined with $|X_m(k)|$ using the magnitude constraint (Griffin and Lim, 1984) to obtain $\hat{X}_m(k)$, a first estimate of the frame's transform:

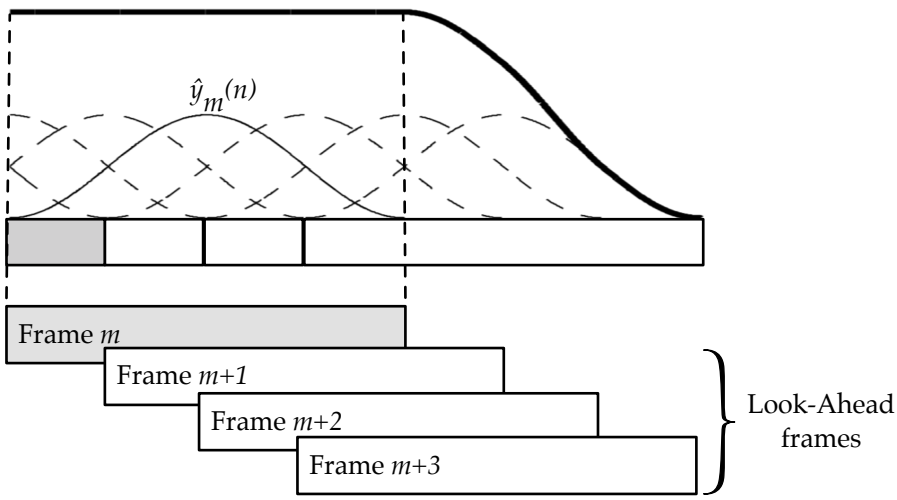
$$\hat{X}_m(k) = Y_m(k)\frac{|Y_m(k)|}{|X_m(k)|} \tag{4.1}$$

The inverse FFT of $\hat{X}_m(k)$ is overlap-added to $y_{m-1}(n)$ to obtain an estimate of the current reconstructed signal $y_m(n)$. A new $Y_m(k)$ is obtained this time by placing an analysis window on $\hat{y}_m(n)$, $\hat{X}_m(k)$ is updated, and a new estimated of the reconstructed frame is added to $y_{m-1}(n)$. This sequence of operations is repeated for a fixed number of iterations.

The fact that the information in $\hat{y}_m(n)$ comes only from previous frames (see Figure 4.2a) reduces the accuracy of the estimation. This problem is addressed by



(a) RTISI



(b) RTISI-LA

Figure 4.2: Partially reconstructed signal using the simple RTISI algorithm (a) and using the Look-Ahead variant (b). S_s indicates the percent of overlap between frames (in this case, S_s=75%). Observe the difference between $\hat{y}_m(n)$ in RTISI (asymmetric), and in RTISI-LA (symmetric). Figure adapted from Zhu *et al.* (2007).

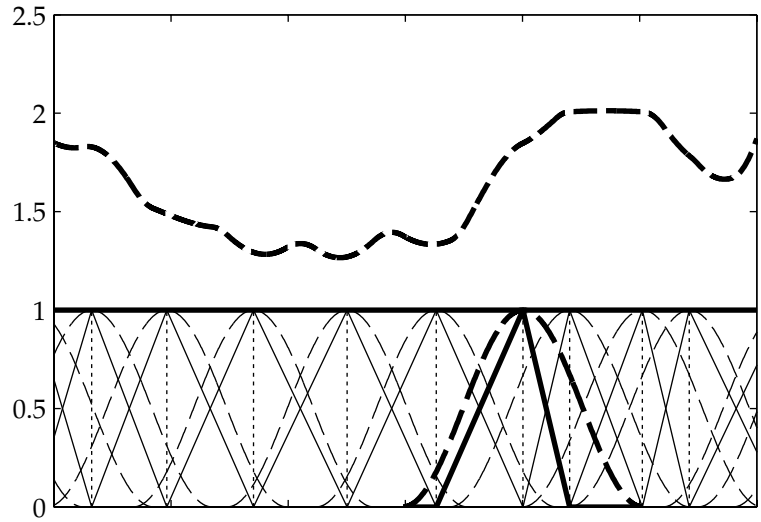


Figure 4.3: Amplitude envelopes resulting from overlap-adding Hanning windows (dashed line) and asymmetric triangular windows (continuous line) as those described in Paper G, when the overlap ratio varies in time. Note how the latter result in a constant amplitude.

the Look-Ahead strategy. Substantially, the previously mentioned iteration is used to reconstruct all the frames up until $m + z$, although at the end only the mth frame is overlap-added to the output signal. In this way, $\hat{y}_m(n)$ contains information from all the frames that overlap with it, as shown in Figure 4.2b. At the end of the iterations, only frame m is overlap-added to the output.

Time stretching with the RTISI-LA algorithm follows the Phase Vocoder approach, i.e. by using different hop sizes for analysis and synthesis. To avoid problems with amplitude fluctuations, the synthesis hop size is held constant at for example 75%, while the analysis hop size is varied according to the desired time stretching ratio. This implies that the analysis and synthesis are both performed on the fly. This approach is unpractical in *PerMORFer*, since the analysis is more complex than simply computing the STFT of the signal. To use a variable synthesis hop size and maintain a constant amplitude, a different solution was found, which is explained in detail in Paper G. Briefly, an asymmetric triangular window is applied to the RTISI-LA reconstructed data frames before overlap-adding them to the output signal. These triangular windows are computed on-the-fly to add up to a constant value, unlike symmetric windows, which do this only for certain overlap values (see Figure 4.3).

The quality of the reconstruction based on the RTISI-LA algorithm also depends on the synthesis hop size, i.e. more overlap means better sound quality.

As in several previous methods, in *PerMORFer*'s implementation of the RTISI-LA frames are duplicated or discarded in order to maintain a high sound quality and yet achieve high stretching ratios. Furthermore, frames that contain a transient are not duplicated, and their synthesis hop size is left unchanged. This addresses two problems: first, it avoids the risk of creating double attacks; and second, it reduces transient smearing, a problem encountered also in the Phase Vocoder (see Section 4.3).

CHAPTER 5

Results and conclusions

The declared final goal of the present thesis (see Chapter 1) was the convergence of the musician's and listener's roles in music listening through the implementation of applications for interactive computer-aided music performance. In Chapter 1, the requirements that were set for the applications to be developed were presented in general terms together with a brief discussion of the choices that have to be made in the early stages of design (see also Paper A). Chapters 2, 3, and 4 discuss more in detail the three main areas of research on which the thesis focused in order to concretely achieve those requirements. These three research topics are the analysis of musical signals to extract performance parameters, the interactive control of the system through the analysis of sensor data, and the rule-based modification of the original performance.

In this final chapter, I will briefly present *PerMORFer* and *MoodifierLive*, the two applications, extensively described in Papers G and H respectively, that represent the synthesis of the work presented in the previous chapters. The applications are then compared to previous systems to point out the novelties introduced by the present work. Finally, a short discussion regarding the present work is followed by a summary of the main contributions of the included papers and by possible future work topics.

5.1 Final applications

PerMORFer (summary of Paper G)

PerMORFer is a multi-platform application that allows for real-time manipulation of a music performance based on the KTH rule system (Friberg *et al.*, 2006). The unique feature that differentiates *PerMORFer* from previous applications based on the KTH rule system (i.e. *Home Conducting* and *pDM* Friberg, 2005, 2006) is the possibility to manipulate audio recordings instead of MIDI files.

To achieve expressive audio transformations, *PerMORFer* was designed following the general block diagram shown in Figure 1.3. In contrast with MIDI-based system, the audio signal needs first to be analyzed to estimate the original

performance parameters (see Chapter 2). The signal is first transformed to the time-frequency domain (STFT), and aligned with a corresponding score file. The tones in the recording are then separated using score-assisted partials tracking and their low-level parameters (pitch, duration, dynamics) estimated. These are required to achieve tone-level transformations define by the KTH rule system. A residual STFT is also computed by subtracting the spectra of the separated tones from the original STFT.

The tone parameters corresponding to the expressive performance are obtained by combining the weighted sum of the default rule values and the nominal note parameters, both stored in the aligned score file. The weighting factors for each rule are controlled by the user in real-time, either directly with sliders, or indirectly using high-level semantic descriptors such the Activity-Valence space (see Chapter 3). The coordinates of a circle whose color and size vary according to the emotion expressed by the performance represent the Activity-Valence values in a separate window. All the parameters can be controlled externally via OSC messages, for example using a mobile phone as a gesture-based controller.

The audio signal transformations are performed in the time-frequency domain on an onset-by-onset basis (see Chapter 4). The STFT of each separated tone is modified to achieve the desired duration and dynamics. To achieve more realistic dynamics modification, the spectrum of the tones is transformed using the model-based approach described in Paper E. The resulting modified tones are added to the residual STFT. The amount of overlap between the frames of the STFT is varied during the reconstruction of the time domain signal to achieve tempo modifications. Spectrogram inversion, i.e. the reconstruction of a time domain signal from only the magnitude of the STFT, is used to synthesize the audio performance in real time.

PerMORFer also supports MIDI playback. In this respect, it is a reimplementation of *pDM* as a stand-alone application. It is also possible to save the performance parameters (i.e. rule weighting factors and activity-valence values) as metadata, so that the performance can be reproduced using MIDI or a different audio recording of the same piece.

MoodifierLive (summary of Paper H)

MoodifierLive is a scaled down, mobile version of *PerMORFer*, with which it shares the control and MIDI sequencer parts. It supports only MIDI reproduction through the phone's built-in synthesizer. *MoodifierLive*, unlike *PerMORFer*, has several built-in gesture-based control modes that exploit the accelerometer found in the mobile phone. These modes can be also used to control *PerMORFer* via OSC messages.

There are two gesture-based modes in *MoodifierLive*, i.e. the *Marbles in a box* and *Free movement*, which were designed for non-expert users. Expressive gestures are detected with the accelerometer and mapped to corresponding expressive performances (see Paper F). With the *Navigate the performance* mode, the user controls the

Table 5.1: Main differences between *PerMORFer* and *ModifierLive*

	PerMORFer	*ModifierLive*
Platform	Mac, Win†, Linux‡	Nokia S60
Language	Python, PyQT, C++	Python (PyS60)
Musical Material	MIDI, Audio	MIDI
Number of built-in control modes	2	5

† Only tested with MIDI
‡ Not tested

Table 5.2: Summary of the requirements met by the two proposed applications.

Objective	*PerMORFer*	*ModifierLive*
Natural sound	•	
Intuitive control	•	•
Realistic modifications	•	
Portability		•

Activity-Valence values, as in *PerMORFer*. Finally, sliders are available to directly control four main performance rules, i.e. tempo, phrasing, overall dynamics, and articulation. A collaborative mode is also available. One phone acts as a server that receives control messages via bluetooth from client phones and reproduces the performance. Each client can control one of the four available rules from the sliders mode.

Comparison between *PerMORFer* and *ModifierLive*

Table 5.1 illustrates the main differences between *PerMORFer* and *ModifierLive*. The use of Python for implementing most of the two applications allowed for a high degree of portability across platforms. The two applications share large parts of the code, namely the performance control modules, the performance modification modules, and the MIDI sequencer. The audio analysis, modification, and synthesis are only available in *PerMORFer*. To achieve real time performance, the more demanding algorithms were written in C++. However, the porting of *ModifierLive* to other mobile platforms (e.g. iOS, Android, Windows Mobile) is limited by the availability of a Python interpreter and of an accessible built-in MIDI synthesizer. From Table 5.1 we can also see that while *PerMORFer* mainly focused on the manipulation of audio recordings, the work carried out on *ModifierLive* was addressed more towards the performance control, as indicated by the number of available built-in control modes.

Table 5.2 compares the two applications with respect to the requirements listed

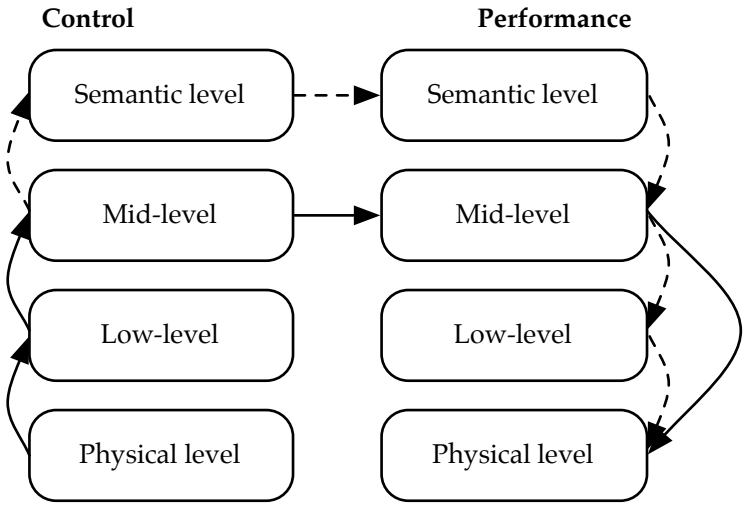


Figure 5.1: Continuos arrows connect the abstraction levels (see Chapter 1) touched by earlier conducting systems. Dashed arrows show the levels reached, on top of the previous ones, by the systems based on the KTH rule system, both MIDI and audio.

in Chapter 1. Combining the two applications, all the requirements were addressed. In this case, portability indicates the fact that *MoodifierLive* is a mobile phone application.

Comparison with previous systems

With the help of Figure 5.1, I will try to point out some general aspects that make *PerMORFer* unique with respect to similar systems described in the literature (e.g. Borchers, 2009; Bruegge *et al.*, 2007; Marrin Nakra *et al.*, 2009; Murphy *et al.*, 2004). Figure 5.1 is based on the levels of abstraction for the performance parameters (right side) and control parameters (left side), previously described in Chapter 1. The continuous arrows point out the levels of abstraction touched by previous systems (in particular the audio-based ones), while the dashed arrows connect the levels included in *PerMORFer*.

The first difference to be pointed out is the use of semantic descriptors in combination with the KTH rules, which allows non-expert users to easily and intuitively control the music performance, in contrast with the simple control of mid-level parameters such as tempo and overall sound level. Although the KTH rules and semantic descriptors were previously used (e.g. *pDM*, *Home Conducting*), they were never applied to the control of an audio recording. *MoodifierLive*, on the other hand, is different from the previous KTH rules-based systems because it is the first

to work on a mobile device.

The use of the KTH rules requires tone level modifications of the audio recording. This implies that in order to obtain the desired performance, we need to pass through the mid- and low-level parameters of the audio signal, as shown in Figure 5.1. In contrast, the modifications to the overall tempo and sound level produced by previous system could be applied directly on the physical level. This also means that the performance analysis described in Chapter 2 was not required for these systems.

5.2 Key contributions

The key contributions of each paper included in the thesis are briefly summarized here.

Paper A

- A comprehensive overview of thirty years of research in conducting systems
- A proposal for the classification of such systems
- Suggestions for a common approach to their evaluation
- A starting point for developing new systems

Paper B

- Improvement to the accuracy of frequency estimation of a sinusoidal component over similar approaches (e.g. Ferreira and Sinha, 2005)
- A new method for separating two overlapping components

Paper C

- An experimental verification that loudness and timbre are both important in the perception of dynamics
- The results suggests that realistic dynamics transformations should include timbre manipulation (see Paper E)
- The results also indicate that timbre could be used for dynamics estimation

Paper D

- Specific algorithms for the analysis of DJ scratching performance data (audio and gesture)

Paper E

- A model-based method for the modification of dynamics that includes timbre changes
- Modifications of dynamics achieved with the new method are more realistic that those obtained by simply changing the sound level of the tone

Paper F

- An application of music performance to the sonification of expressive gestures
- Basic expressive gestures are common to the majority of the people and can be used to intuitively control a music performance
- With a single, low resolution accelerometer, acceptable results can still be achieved

Paper G

- The first application that uses the KTH rule system for music performance to interactively manipulate an audio recording

Paper H

- The first real time mobile phone implementation of the KTH rule system
- Collaborative use of the rule system

5.3 General conclusions

The design and implementation of complex applications such as *PerMORFer* and *MoodifierLive* was a challenging task. The range of problems to be solved is wide and multidisciplinary, from simple technical difficulties such as making a piece of code work on a mobile phone, to the design of perceptual studies, from the development and evaluation of novel algorithms, to the implementation and integration of already available ones.

This work was approached with the final applications in mind, not focusing on single specific aspects. This in part explains the variety of contents in the included papers. As a consequence, the proposed solutions might not always be as advanced as if efforts had been concentrated on a single topic, for example the analysis of the audio signal.

Selecting and integrating already available algorithms was a task almost as difficult as designing new ones. Choices made early in the design process influenced the approach to the following parts. For example, using a Phase Vocoder-style analysis-synthesis framework instead of sinusoidal modeling led to the use

of spectrogram inversion for synthesis. Although with hindsight some of the early choices did not appear to be optimal, changing would have meant re-implementing a lot more than a single part of such a complex system as *PerMORFer*.

The content of each included paper, summarized in the previous section, represents a small but important step towards the ideal active music listening system envisioned at the beginning of this work. Despite the fact that the problems addressed by this work were rather specific to the design of a system for interactive music performance, the single results can be of interest to other related areas of research, for example musicology and affective psychology, as mentioned in Section 2.5.

However, several problems remain to be solved, especially with regard to the analysis and manipulation of audio recordings. The estimation of the performance parameters, and in particular the estimation of dynamics, is far from being solved. This would be particularly useful for automatic performance analysis because current studies mostly focus on tempo, which is easier to estimate. Despite the fact that the attempt at using spectral attributes to estimate dynamics was in part unsuccessful, I think this approach might have some good potential and thus deserves more work. *PerMORFer* would also benefit in general from advances in source separation, and in particular in the separation of overlapping partials. In a more distance future, the ideal system could be made completely automatic by including for example score transcription and automatic performance analysis (e.g. musical phrase segmentation).

An important aspect of the performance modification was not taken into consideration in the development of *PerMORFer*, mainly for lack of time: the manipulation of the tones' transients in connection with both dynamics and articulation transformations. A model-based approach similar to the one proposed in Paper E could be used, although separating the attack transients of several tones in a polyphonic signal is much more difficult that separating their partials. As mentioned in Paper G, preserving the vibrato rate would also increase the quality of the performance manipulation. This problem could be solved by using a sinusoidal modeling analysis-synthesis framework.

The use of semantic descriptors in combination with gesture analysis allowed for the creation of intuitive and easy to use control paradigms. However, the use of mobile phones as control devices limited the expressive possibilities of such paradigms (e.g. only three-four emotions are available in *MoodifierLive*). These possibilities could be extended by using either more powerful handheld devices, or other types of sensors, such as cameras or motions capture systems.

Finally, a thorough evaluation of the systems is required, from the interaction and usability, as well as from the audio quality point of view. In this respect, only single parts were formally evaluated, for example the gesture-based control modes in *MoodifierLive*.

Bibliography

Xavier Amatriain, Jordi Bonada, Alex Loscos, Josep Lluis Arcos, and Vincent Verfaille. 2003. Content-based transformations. *Journal of New Music Research*, 32(1): 95–114.

Josep Lluis Arcos, Ramon Lopez de Mantaras, and Xavier Serra. 1997. Saxex: a case-based reasoning system for generating expressive musical performances. *Journal of New Music Research*, 27:194–210.

Takashi Baba, Mitsuyo Hashida, and Haruhiro Katayose. 2010. "VirtualPhilharmony": a conducting system with heuristics of conducting an orchestra. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME10)*, pages 263–270, Sydney, Australia.

Steven Barrass and Gregory Kramer. 1999. Using sonification. *Multimedia Systems*, 7:23–31.

Michele Basseville and Albert Benveniste. 1983. Sequential detection of abrupt changes in spectral characteristics of digital signals. *IEEE Transactions on Information Theory*, 29(5):709–724.

James W. Beauchamp. 1993. Unix workstation software for analysis, graphics, modification, and synthesis of musical sounds. In *94th Convention of the Audio Engineering Society, preprint no. 3479*, Berlin, Germany.

James W. Beauchamp, editor. 2006. *Analysis, Synthesis, and Perception of Musical Sounds: The Sound of Music*. Springer Verlag, Berlin/Heidelberg.

James W. Beauchamp, Andrew B. Horner, Hans-Friedrich Koehn, and Mert Bay. 2006. Multidimensional scaling analysis of centroid- and attack/decay-normalized musical instrument sounds. *Journal of the Acoustical Society of America*, 120(5):3276–3276.

Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark Sandler. 2005. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047.

Juan Pablo Bello and Mark Sandler. 2003. Phase based note onset detection for music signals. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP2003)*, Hong Kong.

Jordi Bonada. 2000. Automatic technique in frequency domain for near-lossless time-scale modification of audio. In *Proceedings of the International Computer Music Conference (ICMC00)*, Berlin, Germany.

Jan Borchers. 2009. Personal Orchestra. URL `http://hci.rwth-aachen.de/po`.

Roberto Bresin and Giovanni Umberto Battel. 2000. Articulation strategies in expressive piano performance. Analysis of legato, staccato, and repeated notes in performances of the andante movement of Mozart's sonata in G major (K 545). *Journal of New Music Research*, 29(3):211–224.

Roberto Bresin and Anders Friberg. 2000. Emotional coloring of computer-controlled music performances. *Computer Music Journal*, 24(4):44–63.

Roberto Bresin and Anders Friberg. 2011. Emotion rendering in music: Range and characteristic values of seven musical variables. To appear in *CORTEX*.

Jens Brosbol and Emery Schubert. 2006. Calculating articulation in solo music performances. In *Proceedings of the 9th International Conference on Music Perception & Cognition*, Bologna, Italy.

Bernd Bruegge, Christoph Teschner, Peter Lachenmaier, Eva Fenzl, Dominik Schmidt, and Simon Bierbaum. 2007. Pinocchio: Conducting a virtual symphony orchestra. In *Proceedings of the International Conference on Advances in Computer Entertainment Technology (ACE2007)*, Salzburg, Austria.

William Buxton, William Reeves, Guy Fedorkow, Kenneth C. Smith, and Ronald Baecker. 1980. A microcomputer-based conducting system. *Computer Music Journal*, 4(1):8–21.

Antonio Camurri, Giovanni De Poli, Mark Leman, and Gualtiero Volpe. 2001. A multi-layered conceptual framework for expressive gesture applications. In *Proceedings of the Workshop on Current Research Directions in Computer Music*, Barcelona, Spain.

Sergio Canazza, Giovanni De Poli, Antonio Rodà, and Alvise Vidolin. 2003. An abstract control space for communication of sensory expressive intentions in music performance. *Computer Music Journal*, 32(3):281–294.

Helen Cooper. 1985. *Basic Guide to How to Read Music*. Perigee Trade. Retrieved from: http:/encyclopedia.thefreedictionary.com/Articulation+(music).

Mihaly Csikszentmihalyi. 1998. *Finding Flow: The Psychology of Engagement with Everyday Life*. Basic Books.

Roger B. Dannenberg and Christopher Raphael. 2006. Music score alignment and computer accompaniment. *Commununications of the ACM*, 49(8):38–43.

Philippe Depalle, Guillermo Garcia, and Xavier Rodet. 1993. Traching of partials for the additive sound synthesis using Hidden Markov Models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-93)*, Minneapolis, MN, USA.

Simon Dixon. 2006. Onset detection revisited. In *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, Montreal, Canada.

Simon Dixon and Gerhard Widmer. 2005. MATCH: a music alignment tool chest. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR05)*, London, UK.

Mark Dolson. 1986. The phase vocoder: A tutorial. *Computer Music Journal*, 10(4): 14–27.

Chris Duxbury, Mark Sandler, and Mike Davies. 2002. A hybrid approach to musical note onset detection. In *Proceedings of the 5th International Conference on Digital Audio Effects (DAFx-02)*, Hamburg, Germany.

Sebastian Ewert, Meinard Müller, and Roger B. Dannenberg. 2009. Towards reliable partial music alignments using multiple synchronization strategies. In Marcin Detyniecki, Ana García-Serrano, and Andreas Nürnberger, editors, *Proceedings of the 7th International Conference on Adaptive Multimedia Retrieval: Understanding Media and Adapting to the User*, number 6535 in LNCS, pages 35–48, Berlin/Heidelberg. Springer Berlin / Heidelberg.

Marco Fabiani, Gaël Dubus, and Roberto Bresin. 2010. Interactive sonification of emotionally expressive gestures by means of music performance. In Roberto Bresin, Thomas Hermann, and Andy Hunt, editors, *Proceedings of ISon 2010 - Interactive Sonification Workshop*, Stockholm, Sweden.

Marco Fabiani and Anders Friberg. 2007a. Expressive modifications of musical audio recordings: preliminary results. In *Proceedings of the 2007 International Computer Music Conference (ICMC07)*, volume 2, pages 21–24, Copenhagen, Denmark.

Marco Fabiani and Anders Friberg. 2007b. A prototype system for rule-based expressive modifications of audio recordings. In Aaron Williamon and Daniela Coimbra, editors, *Proceedings of the International Symposium on Performance Science (ISPS 2007)*, pages 355–360, Porto, Portugal.

Marco Fabiani and Anders Friberg. 2008. Rule-based expressive modifications of tempo in polyphonic audio recordings. In *Computer Music Modeling and Retrieval. Sense of Sounds*, volume 4969 of *LNCS*, pages 288–302, Berlin, July 2008. Springer.

Anibal J.S. Ferreira and Deepen Sinha. 2005. Accurate and robust frequency estimation in the ODFT domain. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, October 2005.

James L. Flanagan and Richard M. Golden. 1966. Phase vocoder. *The Bell Systems Technical Journal*, pages 1493–1509.

Anders Friberg. 1995. *A Quantitative Rule System for Musical Performance*. Phd thesis, KTH Royal Institute of Technology, Stockholm, Sweden.

Anders Friberg. 2005. Home Conducting: Control the overall musical expression with gestures. In *Proceedings of the International Computer Music Conference (ICMC2005)*, pages 479–482, Barcelona, Spain, September 2005.

Anders Friberg. 2006. pDM: an expressive sequencer with real-time control of the KTH music-performance rules. *Computer Music Journal*, 30(1):37–48.

Anders Friberg, Roberto Bresin, and Johan Sundberg. 2006. Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology, Special Issue on Music Performance*, 2(2-3):145–161.

Anders Friberg, Erwin Schoonderwaldt, and Patrik N. Juslin. 2007. CUEX: An algorithm for extracting expressive tone variables from audio recordings. *Acta Acustica united with Acustica*, 93:411–420.

Alf Gabrielsson. 1994. Intention and emotional expression in music performance. In Anders Friberg, editor, *Proceedings of the 1993 Stockholm Music Acoustics Conference*, pages 108–111, Stockholm, Sweden.

Alf Gabrielsson. 1995. *Music and the Mind Machine: The Psychophysiology and the Psychopathology of the Sense of Music*, chapter Expressive Intention and Performance. Springer Verlag, Berlin / Heidelberg.

Masataka Goto. 2007a. Active music listening interfaces based on signal processing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2007)*, volume 4, pages 1441–1444.

Masataka Goto. 2007b. Active music listening interfaces based on sound source separation and F0 estimation. *Journal of the Acoustical Society of America*, 122(5): 2988.

Fabien Gouyon, Lars Fabig, and Jordi Bonada. 2003. Rhytmic expressiveness transformations of audio recordings: Swing modifications. In *Proceedings of the International Conference on Digital Audio Effects (DAFX03)*, London, UK.

Maarten Grachten. 2006. *Expressivity-aware Tempo Transformations of Music Performances Using Case Based Reasoning*. PhD thesis.

John M. Grey. 1977. Multidimensional perceptual scaling of musical timbres. *Journal of the Acoustical Society of America*, 61(5):1270–1277.

Daniel W. Griffin and Jae S. Lim. 1984. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2).

Stephen Hainsworth and Malcolm Macleod. 2003. Onset detection in musical audio signals. In *Proceedings of the International Computer Music Conference (ICMC2003)*, Singapore.

Mitsuyo Hashida, Sunji Tanaka, and Haruhiro Katayose. 2009. Mixtract: a directable musical expression system. In *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction (ACII2009)*, Amsterdam, the Netherlands.

André Holzapfel, Yannis Stylianou, Ali C. Gedik, and Bariş Bozkurt. 2010. Three dimensions of pitched instrument onset detection. *IEEE Transactions on Audio, Speech and Language Processing*, 18:1517–1527.

Henkjan Honing. 2006. Evidence for tempo-specific timing in music using a web-based experimental setup. *Journal of Experimental Psychology: Human Perception and Performance*, 32(3):780–786.

Ning Hu, Roger Dannenberg, and George Tzanetakis. 2003. Polyphonic audio matching and alignment for music retrieval. In *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 185–188, New Paltz, NY, USA.

ITU-R. 2007. Recommendation ITU-R BS.1770-1 algorithms to measure audio programme loudness and true-peak audio level. ITU-R recommendation, ITU-R, Switzerland.

Jordi Janer, Jordi Bonada, and Sergi Jordà. 2006. Groovator - an implementation of real-time rhythm transformations. In *Proceedings of 121st Convention of the Audio Engineering Society*, San Francisco, CA, USA.

Patrik N. Juslin. 1997a. Emotional communication in music performance: A functionalist perspective and some data. *Music Perception*, 14(4).

Patrik N. Juslin. 1997b. Perceived emotional expression in synthesized performances of a short melody: Capturing the listeners judgment policy. *Musicae Scientiae*, 1(2):225–256.

Patrik N. Juslin and Petri Laukka. 2003. Communication of emotions in vocal expression and music performance: Different channels, same code? *Psychological Bulletin*, 129(5):770–814.

Patrik N. Juslin and John A. Sloboda, editors. 2001. *Music and emotion: theory and research*. Oxford University Press, Oxford (UK).

Patrik N. Juslin and John A. Sloboda, editors. 2010. *Handbook of Music and Emotion: Theory, Research, Applications*. Oxford University Press.

Patrik N. Juslin and Renee Timmers. 2010. *Handbook of Music and Emotion: Theory, Research, Applications*, chapter Expression and communication of emotion in music performance, pages 453–489. Oxford University Press.

Tetsuro Kitahara. 2007. *Computational Musical Instrument Recognition and Its Application to Content-based Music Information Retrieval*. PhD thesis, Kyoto University, Japan.

Anssi P. Klapuri. 1999. Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of the the IEEE International Conference On Acoustics, Speech and Signal Processing (ICASSP99)*, Phoenix, AZ, USA.

Anssi P. Klapuri and Manuel Davy, editors. 2006. *Signal Processing Methods for Music Transcription*. Springer Verlag, New York.

Mathieu Lagrange, Sylvain Marchand, and Jean-Bernard Rault. 2007. Enhancing the tracking of partials for the sinusoidal modeling of polyphonic sounds. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1625–1634. ISSN 1558-7916.

Jean Laroche and Mark Dolson. 1999a. Improved Phase Vocoder time-scale modification of audio. *IEEE Transactions on Speech and Audio Processing*, 7(3):323–332.

Jean Laroche and Mark Dolson. 1999b. New phase-Vocoder techniques for pitch-shifting, harmonizing and other exotic effects. In *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA.

Olivier Lartillot and Petri Toiviainen. 2007. A Matlab toolbox for musical feature extraction from audio. In *Proceedings of the 10th International Conference on Digital Audio Effects (DAFx-07)*, Bordeaux, France, September 2007.

Mark Leman, Valery Vermeule, Liesbeth De Voogdt, Dirk Moelants, and Micheline Lesaffre. 2005. Prediction of musical affect using a combination of acoustic structural cues. *Journal of New Music Research*, 34(1):39–67.

Alex Loscos, Pedro Cano, and Jordi Bonada. 1999. Score-performance matching using hmms. In *Proceedings of the International Computer Music Conference (ICMC1999)*, pages 441–444, Beijing, China.

David A. Luce. 1975. Dynamic spectrum changes of orchestral instruments. *Journal of the Audio Engineering Society*, 23(7):565–568.

Esteban Maestre and Emilia Gomez. 2005. Automatic characterization of dynamics and articulation of expressive monophonic recordings. In *Proceedings of the 118th Convention of the Audio Engineering Society*, Barcelona, Spain.

Esteban Maestre, Rafael Ramírez, Stefan Kersten, and Xavier Serra. 2009. Expressive concatenative synthesis by reusing samples from real performance recordings. *Computer Music Journal*, 33(4):23–42.

Teresa Marrin Nakra. 2000. *Inside the Conductor's Jacket: Analysis, Interpretation and Musical Synthesis of Expressive Gesture*. PhD thesis, MIT Massachusetts Institute of Technology, Boston, MA, USA.

Teresa Marrin Nakra, Yuri Ivanov, Paris Smaragdis, and Chris Adult. 2009. The UBS Virtual Maestro: an interactive conducting system. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME09)*, Pittsburgh, PA, USA, June 2009.

Paul Masri. 1996. *Computer Modelling of Sound for Transformation and Synthesis of Musical Signals*. PhD thesis, University of Bristol - Department of Electrical and Electronic Engineering, Bristol, UK.

Robert J. McAulay and Thomas F. Quatieri. 1986. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(4):744–754.

Declan Murphy, Tue Haste Andersen, and Kristoffer Jensen. 2004. Conducting audio files via computer vision. In Antonio Camurri and Gualtiero Volpe, editors, *Gesture-Based Communication in Human-Computer Interaction*, volume 2915 of *LNAI*, pages 529–540. Springer, Heidelberg.

Toshie Nakamura. 1987. The communication of dynamics between musicians and listeners through musical performance. *Attention, Perception, & Psychophysics*, 41:525–533.

Nicola Orio and Diemo Schwarz. 2001. Alignment of monophonic and polyphonic music to a score. In *Proceedings of the International Computer Music Conference (ICMC2001)*, pages 155–158, Havana, Cuba.

Antonio Pertusa, Anssi P. Klapuri, and Jose Maria Inesta. 2005. Note onset detection using semitone bands. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR2005)*, London, UK.

Lennart Råde and Bertil Westergren. 2004a. *Mathematics handbook for science and engineering*, 5th edition, chapter Trigonometric functions, page 127. Studentlitteratur, Lund, Sweden.

Lennart Råde and Bertil Westergren. 2004b. *Mathematics handbook for science and engineering*, 5th edition, chapter Cubic equations, pages 64–65. Studentlitteratur, Lund, Sweden.

Christopher Raphael. 1999. Automatic segmentation of acoustic musical signals using hidden markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):360–370.

Christopher Raphael. 2008. A classifier-based approach to score-guided source separation of musical audio. *Computer Music Journal*, 32(1):51–59.

Xavier Rodet. 1997. Musical sound signal analysis/synthesis: sinusoidal-plus-residual and elementary waveform models. *Applied Signal Processing*, 4(3):131–141.

Xavier Rodet and Florent Jaillet. 2001. Detection and modelling of fast attack transients. In *Proceedings of the International Computer Music Conference (ICMC2001)*, Havana, Cuba.

James A. Russell. 1980. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178.

Klaus Scherer. 2004. Which emotions can be induced by music? What are the underlying mechanisms? And how can we measure them? *Journal of New Music Research*, 33:239–251.

Klaus Scherer and Marcel Zentner. 2001. *Music and emotion: theory and research*, chapter 16 - Emotional Effects of Music: Production Rules, pages 361–392. Oxford University Press, Oxford/New York.

Mark Schmuckler. 2004. *Ecological Psychoacoustics*, chapter Pitch and Pitch structures, pages 271–315. Elsevier.

Xavier Serra. 1989. *A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition*. PhD thesis, Stanford University.

Xavier Serra and Julius O. Smith. 1990. Spectral modeling synthesis: A sound analysis/synthesis based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14:12–24.

Julius O. Smith and Xavier Serra. 1987. PARSHL: An analysis/synthesis program for non-harmonic sound based on a sinusoidal representation. Technical report, CCRMA, Dept. of Music, Stanford University, Stanford, CA, USA.

Tony S. Verma, S. Levine, and Teresa H. Y. Meng. 1997. Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals. In *Proceedings of the International Computer Music Conference (ICMC1997)*, Thessaloniki, Greece.

John Woodruff, Bryan Pardo, and Roger Dannenberg. 2006. Remixing stereo music with score-informed source separation. In *Proceedings of the 7th International Symposium on Music Information Retrieval (ISMIR2007)*, Victoria, Canada.

Matthew Wright, James Beauchamp, Kelly Fitz, Xavier Rodet, Axel Röbel, Xavier Serra, and Gregory H. Wakefield. 2000. Analysis/synthesis comparison. *Organized Sound*, 5(3):173–189.

Xinglei Zhu, Gerald T. Beauregard, and Lonce Wyse. 2007. Real-time signal estimation from modified Short-Time Fourier Transform magnitude spectra. *IEEE Transactions on Audio, Speech and Language Processing*, 15(5):1645–1653.

Udo Zolzer, editor. 2002. *DAFX: Digital Audio Effects*. John Wiley and Sons.

APPENDIX X

Extension to Paper B

The frequency estimation problem described in Paper B was solved numerically, as explained in Section B.2. An analytical solution to the problem is proposed here. Note that this is only a preliminary result that requires to be verified and evaluated. Equation numbers preceded by the letter B refer to equations found in Paper B, while those preceded by the letter X refer to this Appendix.

First, let us isolate Q_1 in Equations B.29-B.30. Equating the two resulting expressions gives us:

$$X_R(l+1)(C_2 i_1 + S_2 r_1) - X_I(l+1)(C_2 r_1 - S_2 i_1) + Q_2(r_1 i_2 - i_1 r_2) = 0 \qquad \text{(X.1)}$$

Substituting Equations B.21-B.24 into Equation X.1, and reordering the terms, we obtain an expression of the form:

$$aP_1 + bP_2 + cP_1P_2 + dP_2^2 + eP_1Q_2 + FP_2Q_2 = 0 \qquad \text{(X.2)}$$

where

$$\begin{aligned}
a &= (C_1C_2 + S_1S_2)(X_R(l)X_I(l+1) - X_I(l)X_R(l+1)) + \\
&\quad (S_1C_2 - C_1S_2)(X_R(l)X_R(l+1) + X_I(l)X_I(l+1)) && \text{(X.3)}\\
b &= -a && \text{(X.4)}\\
c &= C_2(X_R(l-1)X_I(l+1) - X_I(l-1)X_R(l+1)) - \\
&\quad S_2(X_R(l-1)X_R(l+1) - X_I(l-1)X_I(l+1) && \text{(X.5)}\\
d &= -c && \text{(X.6)}\\
e &= C_1(X_R(l-1)X_I(l) - X_I(l-1)X_R(l)) - \\
&\quad S_1(X_R(l-1)X_R(l) + X_I(l-1)X_I(l) && \text{(X.7)}\\
f &= -e && \text{(X.8)}
\end{aligned}$$

Substituting the expressions for P_1, P_2, and Q_2 (Equations B.13 and B.25) into X.2, and after reorganizing the terms, we obtain:

$$A|H(\omega_2(l-1))| + B|H(\omega_2(l))| + |H(\omega_2(l+1))| = 0 \qquad \text{(X.9)}$$

where $A = a/e$ and $B = c/e$.

The expression for $|H(\omega_2(k))|$, as functions of Δl_2, can now be substituted into Equation X.9 in order to find an analytical solution to the frequency estimation problem.

Substituting ω as given in Equation B.12 into the normalized magnitude of the sine window's transform (Equation B.4) we obtain:

$$|H(\omega_2(l-1))| = 2\sin\frac{\pi}{2N}\sin(\pi\Delta l_2)\left(\frac{1}{\sin\left(\frac{\pi}{N}\Delta l_2\right)} - \frac{1}{\sin\left(\frac{\pi}{N}(\Delta l_2+1)\right)}\right) \quad \text{(X.10)}$$

$$|H(\omega_2(l))| = 2\sin\frac{\pi}{2N}\sin(\pi\Delta l_2)\left(\frac{1}{\sin\left(\frac{\pi}{N}\Delta l_2\right)} - \frac{1}{\sin\left(\frac{\pi}{N}(\Delta l_2-1)\right)}\right) \quad \text{(X.11)}$$

$$|H(\omega_2(l+1))| = 2\sin\frac{\pi}{2N}\sin(\pi\Delta l_2)\left(\frac{1}{\sin\left(\frac{\pi}{N}(\Delta l_2-2)\right)} - \frac{1}{\sin\left(\frac{\pi}{N}(\Delta l_2-1)\right)}\right) \quad \text{(X.12)}$$

Note that the absolute value signs were removed from the right side of the previous expressions after verifying that they are positive in the interval $0 < \Delta l_2 < 1$.

Using sum-difference trigonometric identities (Råde and Westergren, 2004a), we can now separate $\frac{\pi}{N}\Delta l_2$ from the constant part of the argument of the sines in the previous expressions. We can then insert Equations X.10-X.12 into X.9. After simplifying and reorganizing the terms of the resulting expression, we obtain:

$$\sin(\pi\Delta l_2)\left[A\left(\frac{C_1\sin x + S_1\cos x - \sin x}{\sin x(C_1\sin x + S_1\cos x)}\right) + B\left(\frac{C_1\sin x - S_1\cos x - \sin x}{\sin x(C_1\sin x - S_1\cos x)}\right) + \right.$$
$$\left. + \frac{C_1\sin x - S_1\cos x - C_2\sin x + S_2\cos x}{(C_2\sin x - S_2\cos x)(C_1\sin x - S_1\cos x)}\right] = 0 \quad \text{(X.13)}$$

where $x = \frac{\pi}{N}\Delta l_2$. Let us set aside trivial solutions $\Delta l = \{0, 1, 2, \ldots\}$. Grouping the three terms of the second part of Equation X.13 with the least common denominator, and dividing the result by $\cos^3 x$, gives us a numerator that is a third order polynomial in $\tan x$:

$$m_3\tan^3 x + m_2\tan^2 x + m_1\tan x + m_0 = 0 \quad \text{(X.14)}$$

where

$$m_0 = (A+B)S_1^2S_2 \quad \text{(X.15)}$$
$$m_1 = -S_1[(A+B)S_1C_2 + (A-B-1)S_2 + S_1] \quad \text{(X.16)}$$
$$m_2 = C_1S_2[(A+B)(1-C_1)+1] + S_1C_2(A-B+1) \quad \text{(X.17)}$$
$$m_3 = C_1[C_2(C_1-1)(A+B) + C_1 - C_2] \quad \text{(X.18)}$$

The roots of the polynomial can be found using the cubic formula (see e.g. Råde and Westergren, 2004b). Of the possible solutions, the ones that give $0 < \Delta l_2 < 1$

are the estimates of the frequencies of the two components. In case only one solution in this interval is found, we assume there is only one component.

This analytical formulation gives a real solution in the desired interval in the ideal case in which noise is absent. It was preliminary observed that when noise is added, approximations might be required as the roots of Equation X.14 can become complex. Other estimation problems might arise if the two frequencies are exactly the same, or for the trivial solutions $\Delta l = \{0, 1, 2, \ldots\}$. A systematic analysis of the method's performance in different situations is still required to better understand the limitations of this approach.

The method described here was directly derived from the method presented in Paper B, which makes use of the ODFT. The general procedure can be adapted to the more common DFT. Furthermore, it can also be extended to use Generalized Hamming windows (e.g. Hanning and Hamming). In this case, the solutions are found by solving a fourth order polynomial.

Part II

Included Papers

TRITA-CSC-A 2011:12
ISSN-1653-5723
ISRN-KTH/CSC/A–11/12-SE
ISBN 978-91-7501-031-1

www.kth.se