Separation of musical sources and structure from single-channel polyphonic recordings

Mark Robert Every

B. Sc. (Physics, Applied Mathematics) University of the Witwatersrand

B. Sc. Hons. (Physics) University of the Witwatersrand

M. Sc. (Music Technology) University of York

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy to

University of York

Department of Electronics

February 2006

Abstract

The thesis deals principally with the separation of pitched sources from single-channel polyphonic musical recordings. The aim is to extract from a mixture a set of pitched instruments or sources, where each source contains a set of similarly sounding events or notes, and each note is seen as comprising partial, transient and noise content. The work also has implications for separating non-pitched or percussive sounds from recordings, and in general, for unsupervised clustering of a list of detected audio events in a recording into a meaningful set of source classes. The alignment of a symbolic score/MIDI representation with the recording constitutes a pre-processing stage. The three main areas of contribution are: firstly, the design of harmonic tracking algorithms and spectral-filtering techniques for removing harmonics from the mixture, where particular attention has been paid to the case of harmonics which are overlapping in frequency. Secondly, some studies will be presented for separating transient attacks from recordings, both when they are distinguishable from and when they are overlapping in time with other transients. This section also includes a method which proposes that the behaviours of the harmonic and noise components of a note are partially correlated. This is used to share the noise component of a mixture of pitched notes between the interfering sources. Thirdly, unsupervised clustering has been applied to the task of grouping a set of separated notes from the recording into sources, where notes belonging to the same source ideally have similar features or attributes. Issues relating to feature computation, feature selection, dimensionality and dependence on a symbolic music representation are explored. Applications of this work exist in audio spatialisation, audio restoration, music content description, effects processing and elsewhere.

Acknowledgement

I wish to acknowledge firstly, my supervisor John Szymanski, who has constantly been enthusiastic, supportive, eager to embrace and suggest new ideas, and indispensable in providing the research context for this work in the Department of Electronics, University of York. I would also like to thank Tony Tew and Damian Murphy for their references and advice along the way.

I am very grateful for the support of the Overseas Research Students Awards Scheme, without which I would certainly not be writing this now. To the Royal Academy of Engineering and Digital Music Research Network, I thank both bodies for their generous assistance in travelling to conferences and for a secondment to the Music Technology Group (MTG) at Universitat Pompeu Fabra. My appreciation also extends to my hosts at MTG: Emilia Gómez, Fabien Gouyon, Salvador Gurrera and Perfecto Herrera. Also, to all those in D013, so long and take it easy on those cookies and sweets.

To my Mum and Dad who have patiently been watching their son overseas nibbling away at their wine, exotic dining and travel fund, thank you, and I suggest you try the Neethlingshof Sauvignon Blanc, 2002 was a particularly good year... Finally, to Lucy, for sharing it all and for making a good experience a complete and unforgettable one, my gratitude is boundless.

Table of Contents

List of Figures

List of Tables

Author's Declaration

Except where references are made to other sources, the work presented in this thesis is the sole contribution of the author. Parts of chapter 4 have been presented before in:

M.R. Every and J.E. Szymanski, Separation of synchronous pitched notes by spectral filtering of harmonics, to be published in *IEEE Trans. Speech and Audio Processing*, 2006.

M.R. Every and J.E. Szymanski, A Spectral-Filtering Approach to Music Signal Separation, *Proc. 7th Int. Conf. on Digital Audio Effects (DAFx'04)*, (Naples, Italy), pp. 197–200, Oct. 2004.

Section 5.2 was presented before as:

M.R. Every and J.E. Szymanski, Separation of overlapping impulsive sounds by band-wise noise interpolation, *Proc. 8th Int. Conf. on Digital Audio Effects (DAFx'05)*, (Madrid, Spain), Sep. 2005.

Selected parts of chapter 5 were previously presented or modified from:

M.R. Every, Separating harmonic and inharmonic note content from real mono recordings, *Proc. Digital Music Research Network Summer Conf.*, (Glasgow, U.K.), pp. 9–13, July 2005.

Chapter 1

Introduction

> "For me, every sound has its own minute form– is composed of small flashing rhythms, shifting tones, has momentum, comes, vanishes, lives out its own structure."
>
> **- Annea Lockwood (1939–)**

The work in this thesis can be described broadly as separating musical structure from single-channel polyphonic recordings. The term 'musical structure' can be interpreted in several different ways, from the most minute embellishments to large scale compositional structure or even the musician's intentions. Even at a level of comparable complexity, a myriad of different ways of describing a piece of music exists. In terms of physical properties, an example of a low-level structure is a set of harmonics; these give rise to the sensation of pitch. An extended description regards music as a combination of partials, transients and noise content. It should not be forgotten though, that these 'structures' are simplifications of the real audio signal that can be difficult to define precisely but make it easier to converse about the nature of the signal. As we step backwards other structures emerge: continuity over time of a certain timbre, natural and articulatory sounds, notes, the spatial position of these elements in the sound environment, and then motifs, harmonies, melodies, repeated themes, compositional structures, crescendos, and then we enter a realm where higher level structures are more difficult to define.

The majority of this work has been directed at separating what is loosely 'mid-level' structure. Specifically, separating pitched and non-pitched notes from polyphonic music and, following on from this, organising these notes into sources or instrumental parts. However, the means to extract mid-level structure involves much low-level signal processing.

So, it has been stated that the recordings are single-channel or 'mono', and polyphonic. 'Polyphony' means that more than one pitch can sound simultaneously. This specification has been made so that the work has general applicability to all recording conditions and

music in general. Typically, in a commercial recording the number of simultaneously sounding pitched sources at any instant, which will be referred to as the degree of polyphony, is greater than one. As a single voice or monophonic recording is much easier to deal with, then the requirement of polyphony is all encompassing. The single-channel specification ensures applicability of these methods to all recording formats. It is always possible to reduce a multi-track recording into a single channel, or to apply methods designed for mono files separately to each channel of the multi-track recording. Of course, this would be a crude way to deal with spatial information, which could be used much more efficiently to assist the separation. A logical development of these methods would be to adapt them to make better use of spatial information in those situations where it is available.

The use of the term 'separation' must still be clarified, as our usage of this word may differ from others. When a particular musical structure is separated from a recording, there is as little as possible perceived trace of it left in the residual signal, the residual being the original minus the separated signal. At the same time, the perceived quality of the separated signal must be optimised. Whilst it may seem that a high fidelity separated signal directly implies an accurate residual, this is not always true. Some analysis/synthesis methods attempt to re-synthesise the desired structure within the signal, with the intention of achieving the best perceptual quality of this structure, but by discarding phase information for example, an accurate residual is not obtained. If the residual is to be subjected to further analysis, such as in an iterative subtraction scheme, then the quality of the residual is important to avoid error propagation between consecutive iterations.

The notion of prior information must also be introduced, as this can have a large effect on overall quality. This consists of any information we have about the recording prior to analysis, in the form of actual data, models or expectations. The main source of prior information in this work is a symbolic score containing rough estimates of pitches, and onset and offset times of notes in the recording. In practice, the amount of prior information depends on the particular application and often the size of the data set. If the objective is to restore a recording to maximum fidelity, we might use a transcription or score of the recording, a list of instruments/voices playing, the degree of polyphony at all times, knowledge of recording conditions, and so on. If we are only interested in this single recording, it might even be worthwhile to do the restoration painstakingly by hand. This amount of prior information would generally not be available for a large database of recordings, which would require a more automatic system for retrieving musical content. However, even when trying to make the analysis as general as possible, expectations about the nature of the signal are unavoidable. This work has tried to make balanced modelling assumptions informed by the behaviour and characteristics of typical musical signals in

different visual representations and by listening. However, given the complexities of a real musical signal, it seems an impossibly difficult task to construct a practical model of sufficient complexity and flexibility. This is the principle reason that an adaptive filtering methodology has been adopted rather than a parameterised signal model. The hope is that filtering removes existing content or structures from the mix, rather than creating new content or artifacts in the case that the recording is not well described by the signal model. Much of the work tries to make these filters adaptive to the signal. However, we are still working with the assumption that the recording is built from individual notes, which is restrictive, especially when applied to recordings containing drone-like sources, feedback or resonance interaction between different sound sources, gradual timbral changes as opposed to localised notes, and synthetically sculpted sounds.

1.1 Applications

The quality of source separation determines the types of applications that this work might lead towards. For the purpose of music information retrieval, a reduction of the audio signal into a small set of descriptive parameters is usually the end goal. Thus, if we use a separation of the signal to aid the extraction of these descriptors, the quality of separation need not necessarily be of the highest quality. On the other hand, if the separated structure is intended to be heard in isolation, then a much higher fidelity may be required. Applications in which a separated source is scaled slightly and then re-mixed with the original recording require a fidelity somewhere between these two limits, as fortunately, there is a tendency for artifacts of the separation to be masked when re-mixing. A few potential applications of source separation from music are listed below.

1.1.1 Spatialisation

Audio spatialisation is the processing of sound in a manner that creates the perceived effect of sources coming from positions in 3-dimensional space around the listener. Once a set of sources has been extracted from a mono recording, it is possible to re-mix them into a multi-track recording by including spatial information. There is a multitude of classic music recordings and film soundtracks only available in mono, which could potentially be spatialised in this way. The separation quality required for this type of application is likely to depend partly on how far apart the extracted components are positioned spatially.

1.1.2 Audio Restoration

Audio restoration involves the removal of localised disturbances like clicks or sudden bursts of noise, and global degradations such as background noise from corrupted audio. Under conditions in which the degrading signal is additive, the opposite process of separating the desired audio signal from the degraded mix could result in improvements of audio quality. In an interesting example of audio restoration culminating in the compilation *Italian Songs*[3], the desired signal was the voice of renowned tenor Enrico Caruso (1873-1921), and the unwanted signal was his original orchestral accompaniment, both of very poor audio quality. New recordings were produced by filtering and equalising Caruso's voice from the original, and then superimposing it on a modern orchestral accompaniment.

1.1.3 Music Content Description

Music content description and information retrieval are of increasing importance due to a shift in the nature of music production, distribution and consumption, the need to efficiently manage large databases of music recordings and samples, and to compute similarity between different pieces of music. The possibilities of a heightened interaction with the music contents in terms of classification, browsing, recommendation, retrieval, rendering, personalising and editing are numerous, and are being, or have been, explored by projects such as CUIDADO[4], Semantic HIFI[5] and SIMAC[6].

The extraction of musically meaningful features or descriptors from a raw recording can sometimes be assisted by low-level separations of the audio, such as transient, steady-state and noise decompositions. For example, rhythmic descriptors can be computed on the isolated transient component, and harmonically-related descriptors on the steady-state component. Alternatively, given a higher-level separation of the recording into notes, chapter 5 shows how different source attributes can be used to group notes into source types. In theory, this makes it possible to extract a particular instrument or source from the mix, which in turn allows the computation of source or instrument-specific descriptors. The process could even be reflexive; separation algorithms could then be modified and informed by extracted high-level information.

1.1.4 Creative Musical Applications

The applications of music signal separation that are possibly the most difficult to foresee are the creative ones, as these depend on each individual composer/performer's intentions. A potential creative application is 'targeted effects processing'. That is, applying a musical effect to a structured component of a mix as opposed to processing the mix as a whole. Some examples within this framework could be: a compressor that compresses only certain

kinds of percussive onsets, a karaoke-like system that suppresses any desired instrument in the mix, and an equaliser that amplifies different structures within the mix rather than specific frequency bands. Another creative possibility is sample-based music: music that re-uses segments of existing audio material in new compositions. Rather than re-using a complete segment of a recording, the sound artist may wish to extract only a particular structure from within that segment.

1.2 Overview of the thesis

The content of the thesis is split into three main chapters: the first (chapter 4) treats the extraction of harmonic content of pitched notes from polyphonic mixes. The second (chapter 5) deals with the separation of non-partial content when multiple pitched or percussive notes are overlapping in time. The third section (chapter 6) discusses some feature-based clustering experiments for grouping sets of separated or segmented notes into different source types. In unison, these chapters describe an approach to source extraction from polyphonic mixes. The block diagram of the implemented system is shown in fig. 1.1b, and an envisaged completely automatic system is given in fig.1.1a. In addition, chapter 2 establishes the basic theoretical foundations of subsequent chapters and gives a general review of the context of this work when not covered in the above chapters. Chapter 3 discusses the alignment of the recording with a MIDI representation, which is the main source of prior information. Although MIDI data is an essential pre-processing component of the current system, in future it is intended to replace this by an automatic music transcription system. What follows is a break down of each chapter in slightly more detail.

Chapter 2 - This begins by reviewing alternative ways of representing sound other than the standard time-domain waveform. This is important as the audio representation is the substrate from which musical structures are recognised and isolated during separation. An overview of time-frequency and time-scale representations with an emphasis placed on the short-time Fourier transform (STFT) and wavelet analysis will be given. The chapter also discusses various methods for de-constructing sound into elementary structures or building blocks, such as partials, noise envelopes, transients, time-frequency masks or atomic decompositions.

Chapter 3 - Prior information consisting of note pitches, onset times and offset times are required to 'find' the notes within the recording, as it will be assumed in later chapters that this has already been performed. Whilst a fairly robust multi-pitch estimator has been developed for simple test samples, processing real recordings can be tricky. Automatic transcription is a difficult problem in its own right, and it is chiefly for this reason that it is being avoided, thereby allowing all effort to be directed at other interesting areas.

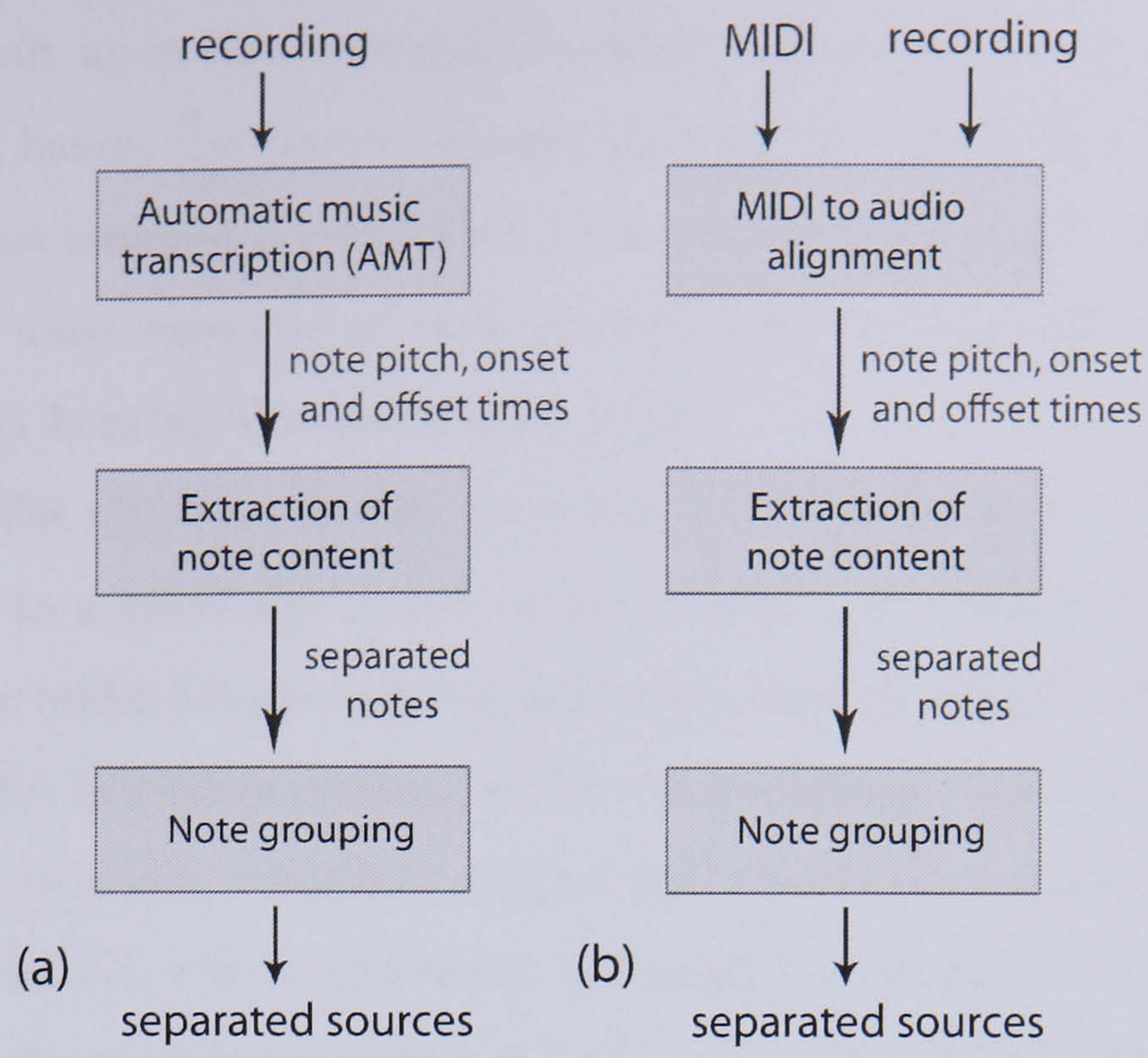

Figure 1.1: (a) An automatic system for source separation from musical recordings, (b) The system implemented here using prior MIDI data.

This chapter describes how user-improvised MIDI data is aligned with recordings. Human timing errors and limitations of the MIDI representation can be partially overcome by aligning MIDI notes to the audio recording. A note onset detector is used to estimate a set of note attack times from the recording, which are then matched to MIDI notes using a dynamic programming algorithm, and MIDI pitches are refined using a pitch estimation process.

Chapter 4 - This chapter describes the separation of harmonic content of pitched notes from polyphonic mixes by adaptive filtering in the spectral-domain. The process is described in detail, starting with parameter estimation of spectral peaks in the STFT, continuing with the tracking of note harmonics over time, and the design of adaptive filters for separating overlapping notes. Particular emphasis is placed on the separation of harmonics that are overlapping in the spectral-domain, and a review of previous methods for separating overlapping partials is given. Results are also provided that compare adaptive filtering with sinusoidal extraction techniques, validating the use of adaptive filtering for source separation under certain conditions. Spectral subtraction of harmonics has also been tested as a means of reducing filtered noise in noisy spectra.

Chapter 5 - This chapter treats the transient and noise components of the signal, and describes three techniques for separating transient and noise content of notes in polyphonic recordings. These include: a time-domain linear predictive method for transient extraction, which is intended to retain the transient characteristics of isolated note attacks. Secondly, a bandwise energy interpolation method[7] for separating overlapping and decaying noisy

onsets. This is built upon three alternative signal representations: the DFT followed by processing in Bark bands, the discrete wavelet transform and the wavelet packet transform. The third technique assumes a correlation between the shape of the harmonic amplitude envelope and the noise envelope of each pitched note, and attempts to split the noise envelope of the mix between the overlapping notes.

Chapter 6 - The objective in this chapter is to find an automatic way to separate all content belonging to a particular source or instrument type from the mix, as opposed to de-constructing the audio file into a set of unlabelled note waveforms. This will allow us to extract or transform individual sources within the recording. An unsupervised clustering approach is taken, in which notes are grouped into clusters in a multidimensional feature space. Feature selection will be discussed, although the detailed feature derivations are given in appendix A, and the clustering algorithm uses model-based clustering (MBC)[8]. Results are reported both when the number of clusters or sources is provided *a priori*, and when this must be determined from the data. A comparative study of clustering performance when computing features on raw note segments versus separated notes from the recording will also be given.

Chapter 7 - This contains a summary of the main contributions of the thesis, reflections, and suggestions for further work.

Chapter 2

Music signal representation and modelling

> "The whole problem can be stated quite simply by asking, 'Is there a meaning to music?' My answer would be, 'Yes.' And 'Can you state in so many words what the meaning is?' My answer to that would be 'No.'"
>
> **- Aaron Copland (1900–1990)**

We are facing the task of separating musical structures arising from multiple sources from a mono recording. As these structures are in general overlapping in both the time and frequency domains, it might be useful to find one or more alternative representations in which these structures are more easily separable. This leads naturally to a two-dimensional representation with both time and frequency axes. Section 2.1 reviews three representations which correspond to different samplings or tilings of the time-frequency plane. Firstly, the discrete short-time Fourier transform (STFT), which is the main representation used for separation of harmonic content in chapter 4, and has also been used for separating non-partial content in chapter 5. Secondly, the discrete wavelet transform (DWT), and thirdly, the wavelet packet transform (WPT), both of which have been used in chapter 5 for separating overlapping non-partial content.

Although the separation methods described in chapters 4 and 5 extract different musical structures directly from the representation, it is a set of signal modelling assumptions that informs how the energy in the representation is to be distributed between the sources. For this reason, a review of some common signal models that have been applied to music and speech processing is given in section 2.2. These include sinusoidal modelling and partial tracking, which have much in common with the methods used for tracking harmonics of pitched notes in section 4.3. Noise modelling is discussed in section 2.2.2, introducing the idea of a time and frequency dependent noise power envelope. This idea is continued in

section 5.2 where it is applied to the separation of decaying noisy onsets, and in section 5.3 where it is implicitly used in a technique for separating overlapping noise from multiple sources. Models and extraction of transient content are discussed in section 2.2.3, providing some context for the time-domain autoregressive (AR) method for separating transient events proposed in section 5.1.

2.1 Music signal representation

If we plan to separate a musical structure from a recording, it makes sense to use a representation of the sound in which this structure is evident, and can be seen to be separable from other types of musical structure occurring simultaneously. Furthermore, for the representation to be useful as a musical tool, sounds of perceptual importance should also be salient in this sound representation, since it is generally these structures over which the user wishes to have some control. It does not make much sense to use an audio representation in which the kinds of structure that are easily separable are difficult to interpret or do not have much musical significance. Unfortunately, there is seldom a single audio representation that concisely displays or sparsely represents all structures of interest. Even for the well-known spectrogram, a compromise must be decided upon with regards to the quality of representation of steady-state content and partials versus rapidly time-varying content and transients. This motivates the use of multiple signal representations, or at least multi-resolution and even adaptive signal representations, tailored for displaying a particular structure of interest. Multi-resolution methods such as wavelet analysis[9, 10], the constant-Q transform[11, 12], and multi-resolution filter banks and decomposition trees[13] have become fairly popular in music signal processing. The use of these methods is motivated by factors such as the desire to mimic the human auditory system, the fact that our frequency sensitivity is more appropriately described by a constant relative frequency resolution ($Q = \frac{f}{\Delta f}$, where f = centre frequency, Δf = frequency resolution at f) than constant absolute frequency resolution (Δf), the fact that musical scales follow a logarithmic frequency pattern, and the knowledge that in addition to the signal's frequency envelope, its phase information and shape of the time-domain waveform are perceptually significant as well.

If the representation is itself to be subjected to further processing, an important factor is whether it is possible to re-synthesise sound from the transformed representation, and what artifacts might be introduced by this transformation. For this reason there is interest in perfect reconstruction analysis/synthesis methods.

A further desirable characteristic of the representation is that it is a linear system in the following sense. If $\mathbf{x_1}$ and $\mathbf{x_2}$ are two sampled signals in vector notation, $F\{\mathbf{x_1}\}$ and

$F\{\mathbf{x_2}\}$ are their transformed representations, and α_1 and α_2 are two constants, then the principle of superposition holds:

$$F\{\alpha_1\ \mathbf{x_1} + \alpha_2\ \mathbf{x_2}\}\ =\ \alpha_1\ F\{\mathbf{x_1}\} +\ \alpha_2\ F\{\mathbf{x_2}\}. \tag{2.1}$$

This ensures that the opposite process of subtracting the exact representation of one of the signals, let's say $F\{\mathbf{x_1}\}$, from the mixture $F\{\mathbf{x_1} + \mathbf{x_2}\}$, to produce a residual representation $F\{\mathbf{x_r}\}$, satisfies:

$$F\{\mathbf{x_r}\}\ =\ F\{\mathbf{x_1} + \mathbf{x_2}\} - F\{\mathbf{x_1}\}\ \equiv\ F\{\mathbf{x_2}\} \tag{2.2}$$

and so the inverse (perfect reconstruction) transformation of the residual representation:

$$F^{-1}\{F\{\mathbf{x_r}\}\}\ \equiv\ F^{-1}\{F\{\mathbf{x_2}\}\}\ =\ \mathbf{x_2} \tag{2.3}$$

yields the second signal $\mathbf{x_2}$ without introducing any interference terms. In other words, if the chosen source is subtracted from the original representation so that an inverse transformation allows the perfect reconstruction of this source, the subtraction will not produce artifacts that could hinder the extraction of further sources from the residual representation. Although we prefer to base our analyses on a linear representation for the reasons of ease of interpretation and simplicity in terms of signal processing, there is evidence to suggest that human listening is nonlinear. For example, the sum of two identical tones does not sound twice as loud; if this were the case it might be very difficult to hear a solo violin playing with a full orchestral background accompaniment.

In some representations such as the spectrogram or Cohen's class of bilinear energy time-frequency distributions[14], if $\mathbf{x_1}$ and $\mathbf{x_2}$ are sufficiently far apart in time and frequency, the principle of superposition is nearly satisfied. However, this is generally not the case given that sources in music are often overlapping or nearly overlapping in both time and frequency domains. Cohen's class of time-frequency energy distributions[14] is the class of distributions that measure the spread of signal energy in time and frequency which are time and frequency covariant, i.e. a time or frequency translation of the signal produces a corresponding simple translation of the energy distribution. The class includes the Wigner-Ville distribution (WVD)[14], and smoothed versions of this such as the pseudo-Wigner-Ville distribution[15], smoothed-pseudo-Wigner-Ville distribution[15], and modal distribution[16]. Whilst the WVD is capable of providing simultaneously very high time and frequency resolutions, and the modal distribution is designed specifically for the estimation of the instantaneous frequencies of a multi-component sinusoidal model, this comes at a cost. The representations do not satisfy eqn. 2.1, and produce interference terms which can make them difficult to interpret visually for multi-component signals. Fig. 2.1 compares the spectrogram and WVD for a sum of two chirp signals, one logarithmically

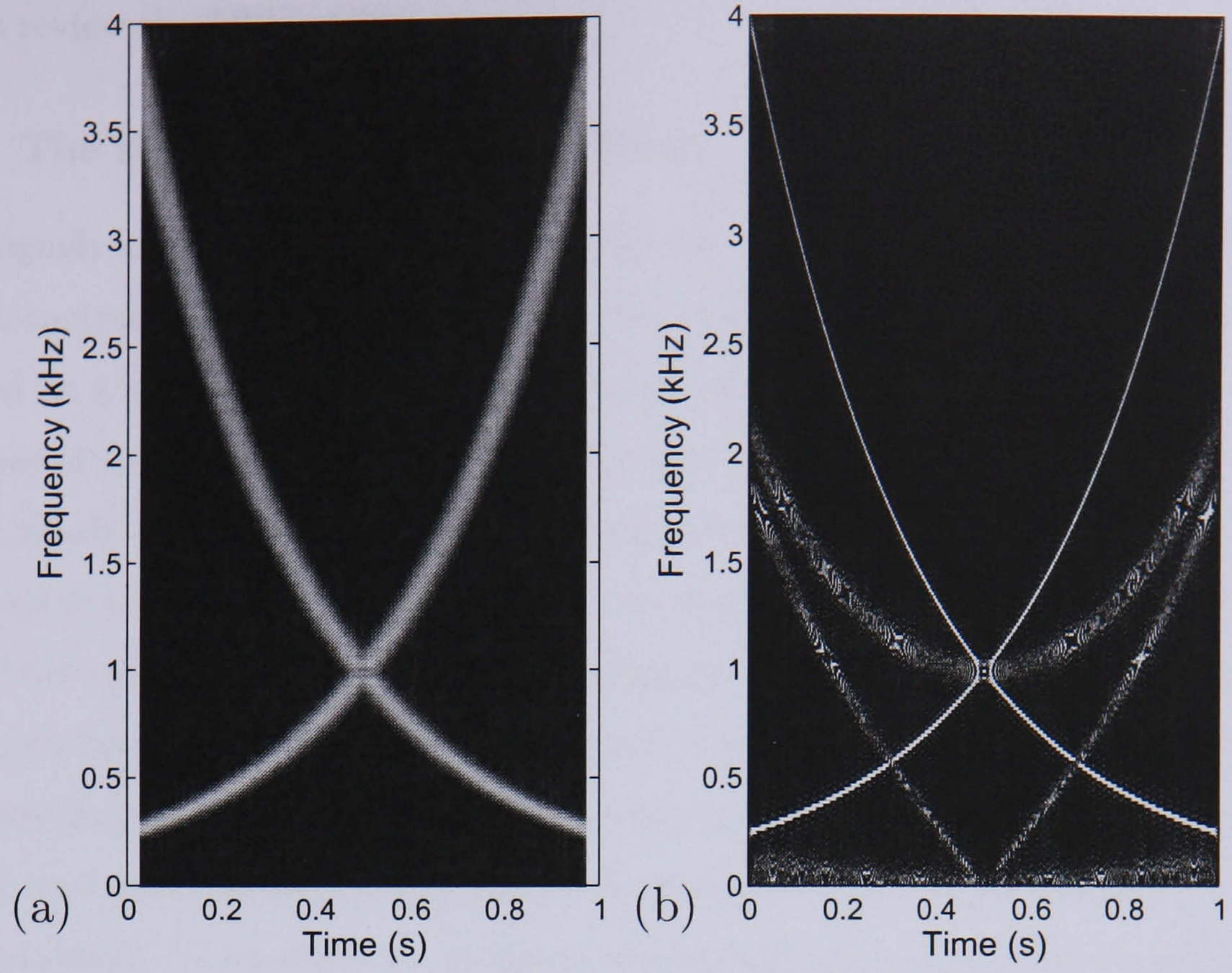

Figure 2.1: A sum of two chirp signals (one logarithmically increasing and the other decreasing in frequency) viewed in the: **(a)** spectrogram, **(b)** Wigner-Ville distribution (WVD)

increasing in frequency and the other decreasing in frequency. Both the excellent simultaneous time and frequency resolution, and average and difference frequency interference terms of the WVD can be seen clearly. A review of time-frequency representations for the analysis of musical signals is given in [16, 17].

One of the most popular music signal representations that satisfies eqn. 2.1, and from which a perfect reconstruction can be obtained under appropriate sampling conditions, is the discrete form of the STFT. This is an excellent way of simultaneously displaying the crude time structure and resonance structure of the signal. Furthermore, many useful signal transformations can be made in the STFT time-frequency-domain. A closely related technique, the phase vocoder[18, 19, 20], is well established as an analysis/re-synthesis tool for speech and music signals, allowing a variety of transformations such as pitch-shifting and time-stretching. It was thus decided to do the majority of audio processing within the STFT representation except where this proved to be inadequate, such as for highly non-stationary content. Wavelet analysis was used in section 5.2; this encodes the shape of the time-domain waveform as a sum of 'bites' of information at a set of sequentially larger time scales. Both the DWT and WPT can be implemented efficiently using filter bank structures (sections 2.1.4 and 2.1.5). Wavelet packet analysis allows the construction of a signal dependent basis providing an optimal multi-resolution decomposition of the signal. Both the DWT and WPT satisfy eqn. 2.1 and are invertible transformations. The following

sections review the STFT, DWT and WPT.

2.1.1 The short-time Fourier transform

Music signals are characteristically non-stationary, meaning that the sound evolves over time. In contrast, a stationary signal is one whose deterministic component can be exactly modelled as a sum of sinusoids of constant amplitude and frequency, and the statistical properties of its stochastic component do not vary over time either. In reality, music signals are nearly always non-stationary, but over sufficiently short periods of time are often considered to be 'quasi-stationary'. That is, the sinusoidal parameters and statistical properties of the stochastic component change negligibly over a short duration. This property of local stationarity suggests the discrete short-time Fourier transform (STFT) as a signal representation. With a suitably chosen analysis window function, the STFT measures the local time and frequency behaviour of the signal over durations in which the signal is quasi-stationary. Fortunately, the weighted overlap-add method[21] allows a perfect reconstruction to be obtained from the STFT, and the STFT is also versatile to performing a host of temporal and spectral transformations.

The origins of the STFT in Fourier theory are well known and so we will skip over these and simply state the result. The (continuous) STFT of a signal $x(t)$ measured at some time instant τ and angular frequency ω is:

$$\mathrm{STFT}(\tau,\omega) \;=\; \int_{-\infty}^{\infty} x(t)\; h^*(t-\tau)\; e^{-i\omega t} dt \tag{2.4}$$

where $h(t)$ is the analysis window function. If $h(t)$ is chosen to be concentrated about $t = 0$, then we can interpret the above as a measure of the signal content in a time-frequency region centred around τ and ω. Eqn. 2.4 can also be seen as a filtering operation, where the impulse response of the filter is the window function modulated at the frequency ω.

In the real discrete signal case, $x(t) \rightarrow x[n]$ with $n = 0, \ldots, L-1$, it is usual to sample the continuous STFT at discrete time and frequency intervals. The time axis is segmented into time frames, which can be overlapping, with a step size between consecutive frames of N_{hop} samples. We denote $r = 0, \ldots, R-1$ as the time frame index. The frequency axis is sampled at equidistant frequencies $\omega_k = 2\pi k/N$, where $k = 0, \ldots, N-1$, and N is the transform length, whose meaning will soon become clearer. This yields the discrete STFT:

$$S[k,r] \;=\; \sum_{n=-\infty}^{\infty} x[n] \cdot h[n - rN_{hop}]\; e^{-i\omega_k n} \tag{2.5}$$

where $h[n]$ is the discrete and real-valued analysis window function. Substituting $s = n - rN_{hop}$, and specifying that $h[n]$ is of non-zero length at most N:

$$\begin{aligned} h[n] \;&\geq\; 0 \quad ;\; n = 0, \ldots, N-1 \\ h[n] \;&=\; 0 \quad ;\; \text{otherwise} \end{aligned} \tag{2.6}$$

then eqn. 2.5 can be rewritten as:

$$\begin{aligned} S[k,r] &= \sum_{s=0}^{N-1} x[s+rN_{hop}] \cdot h[s]\ e^{-i\omega_k(s+rN_{hop})} \\ &= e^{-i\omega_k rN_{hop}} \sum_{s=0}^{N-1} x[s+rN_{hop}] \cdot h[s]\ e^{-i\omega_k s} \\ &= e^{-i\omega_k rN_{hop}} \cdot F_r[k] \end{aligned} \tag{2.7}$$

where

$$F_r[k] = \sum_{s=0}^{N-1} x[s+rN_{hop}] \cdot h[s]\ e^{-i\omega_k s} \tag{2.8}$$

is the discrete Fourier transform (DFT) of the r^{th} block of N samples multiplied by the window function $h[s]$. This is convenient, as the STFT is now written in a form that has a direct implementation using the DFT of consecutive segments of the signal, as we slide the window function across the waveform taking snapshots every N_{hop} samples. Once again, N is the transform length, which is chosen to be at least as large as the window length. When N is larger than the width of the window, this is known as zero-padding, which can be used to increase the density of sampling of the frequency axis, since $\Delta\omega = \omega_{k+1} - \omega_k$ is inversely proportional to N.

If a windowed segment of $x[n]$ in the r^{th} time frame is defined as:

$$\hat{x}_r[n] = x[n] \cdot h[n - rN_{hop}] \tag{2.9}$$

and N_{hop} is chosen to be smaller than the width of the window function, then the non-zero portions of $\hat{x}_r[n]$ are overlapping, as shown in fig. 2.2.

Synthesis from the STFT representation can be performed in a similar manner. Let $\tilde{S}[k,r]$ denote the STFT after some transformation has been applied in the time-frequency-domain, and let $\tilde{F}_r[k]$ be defined as:

$$\tilde{F}_r[k] = e^{i\omega_k rN_{hop}} \cdot \tilde{S}[k,r]. \tag{2.10}$$

The transformed signal $\tilde{x}[n]$ can be obtained by weighted overlap-add synthesis[21] with a synthesis window $\tilde{h}[n]$ as follows:

$$\begin{aligned} \tilde{x}[n] &= \frac{1}{N} \sum_{r=0}^{R-1} \tilde{h}[n - rN_{hop}] \sum_{k=0}^{N-1} \tilde{S}[k,r]\ e^{i\omega_k n} \\ &= \frac{1}{N} \sum_{r=0}^{R-1} \tilde{h}[n - rN_{hop}] \sum_{k=0}^{N-1} \tilde{F}_r[k]\ e^{i\omega_k(n - rN_{hop})} \\ &= \sum_{r=0}^{R-1} \tilde{h}[n - rN_{hop}]\ \tilde{x}_r[n] \end{aligned} \tag{2.11}$$

where $\tilde{x}_r[n]$ is the inverse discrete Fourier transform (DFT^{-1}) of $\tilde{F}_r[k]$, translated to the r^{th} frame:

$$\tilde{x}_r[n] = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{F}_r[k]\ e^{i\omega_k(n - rN_{hop})}. \tag{2.12}$$

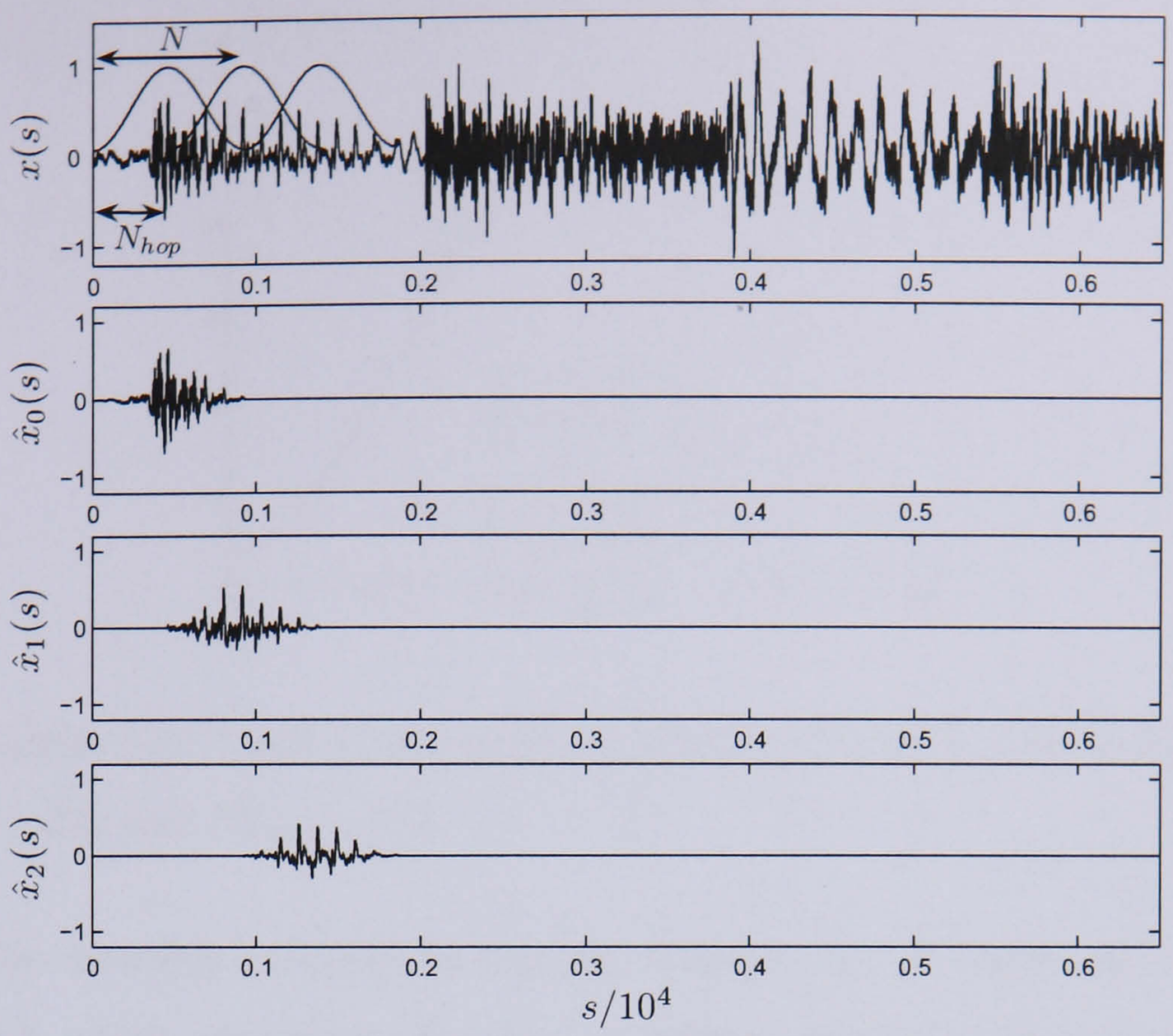

Figure 2.2: Calculation of the discrete STFT of a music signal using a sliding window function. The lower three figures show overlapping windowed segments of the original waveform. The hop size between frames, N_{hop}, is half the transform length, N.

$\tilde{x}_r[n]$ and $\hat{x}_r[n]$ would be identical in the absence of any transformations to the STFT data. Similarly to $\hat{x}_r[n]$ in fig. 2.2, $\tilde{x}_r[n]$ is a windowed and potentially transformed segment of the signal. It is only necessary to calculate $\tilde{x}_r[n]$ for sample values within the r^{th} frame, i.e. $n - rN_{hop} = 0, \ldots, N-1$, as it is later multiplied by the synthesis window function $\tilde{h}[n - rN_{hop}]$ in eqn. 2.11 which is zero elsewhere. The condition on the analysis and synthesis window functions for perfect reconstruction in the absence of any transformations, i.e. $x[n] = \tilde{x}[n]$, is:

$$\sum_{r=0}^{R-1} \tilde{h}[n - rN_{hop}]\, h[n - rN_{hop}] \;=\; 1, \qquad \forall\, n = 0, \ldots, L-1. \tag{2.13}$$

It happens that $h[n]$ was chosen as a Hamming window of length N due to other considerations, one being to obtain accurate estimates of spectral peak parameters (section 4.2). Thus it was convenient to choose:

$$\tilde{h}[n] \;=\; \begin{cases} \frac{2 \cdot N_{hop}}{N} \frac{t[n]}{h[n]} & ;\; n = 0, \ldots, N-1 \\ 0 & ;\; \text{elsewhere} \end{cases} \tag{2.14}$$

where $t[n]$ is the triangular window function:

$$t[n] \;=\; \begin{cases} \frac{2n+1}{N} & ;\; 0 \le n \le N/2 - 1 \\ \frac{2(N-n)-1}{N} & ;\; N/2 \le n \le N-1 \\ 0 & ;\; \text{elsewhere} \end{cases} \tag{2.15}$$

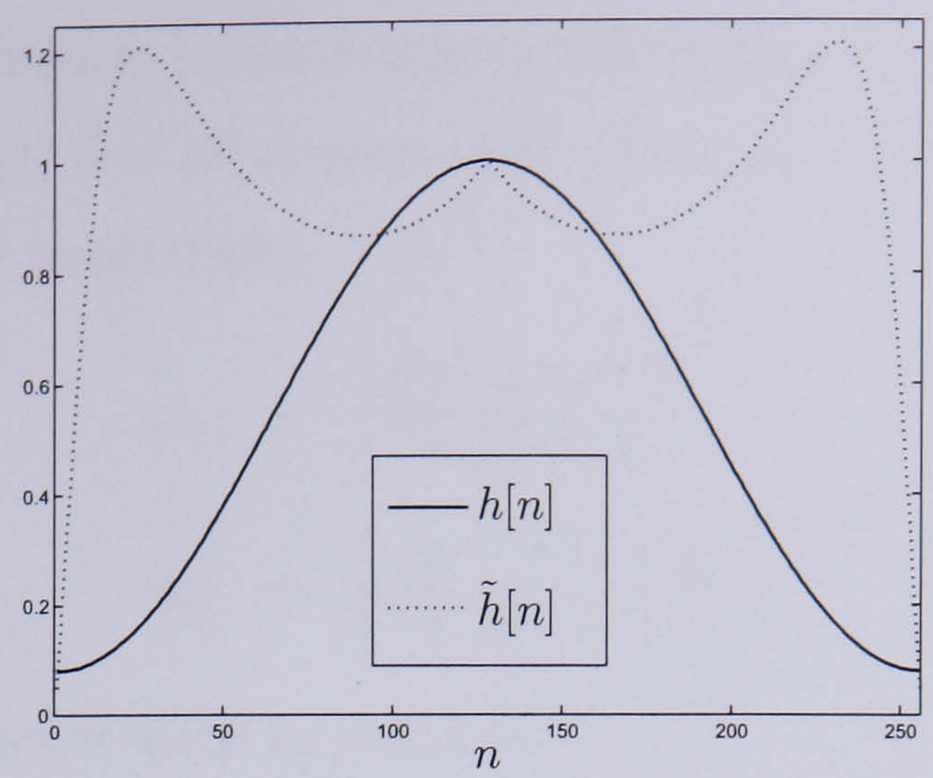

Figure 2.3: Analysis ($h[n]$) and synthesis ($\tilde{h}[n]$) windows used for calculating the discrete STFT with $N = 256$ and $N_{hop} = N/2$

$\tilde{h}[n]$ now has the desirable property of tending towards zero at the frame boundaries, as shown in fig. 2.3, which minimises edge discontinuities when spectral transformations are made to $S[k, r]$. Actually, eqn. 2.13 is not entirely satisfied for $n < N$ and $n \geq L - N$, but this is not of much concern as the signal can be buffered with short segments of silence on each end. It is worth mentioning that eqn. 2.13 can also be satisfied for certain combinations of the window function and overlap factor (N/N_{hop}) with identical analysis and synthesis windows. For example, a simple sine window $h[n] = \tilde{h}[n] = \sin(\pi n/N),\ n = 0, \ldots, N-1$ with with an overlap factor of $m \geq 2,\ m \in \mathbb{Z}$ would satisfy eqn. 2.13, as would a Hanning window with $m \geq 3,\ m \in \mathbb{Z}$.

The DFT can be calculated very efficiently using a fast Fourier transform (FFT) algorithm[22]. There are a number of different implementations of the FFT that depend on how the transform length N is reduced to smaller factors called radices. It is common in audio processing to choose N to be a power of 2 samples long. The radix-2 FFT algorithm requires $O(N \log_2 N)$ arithmetical operations in comparison to direct computation of the DFT in $O(N^2)$ operations. N_{hop} was chosen as $N/2^m$, where $m \in \mathbb{N}$, and no zero-padding has been used.

Finally, the familiar spectrogram is simply the modulus squared of the STFT, i.e. $|S[k, r]|^2$. This is the spectral energy density of the signal within the time-frequency plane. Some spectrograms of musical instrument sounds are given in figs. 4.6, 4.7 and 4.8.

2.1.2 Time-frequency resolution and multi-resolution approaches

The STFT was proposed as a way of representing the local behaviour of the non-stationary musical signal, and it was shown how it can be calculated efficiently using the DFT and synthesised from using the DFT^{-1}. A consequence of using a finite length analysis window is that the window is itself of non-zero bandwidth, and must obey the uncertainty principle.

This is convenient to formulate in the continuous-time case. If the window function and its Fourier transform are $h(t)$ and $H(\omega)$ respectively, then the time spread and bandwidth of the window function are respectively:

$$\Delta_t = \left[\frac{\int t^2 \ |h(t)|^2 dt}{\int |h(t)|^2 dt}\right]^{1/2} \tag{2.16}$$

$$\Delta_\omega = \left[\frac{\int \omega^2 \ |H(\omega)|^2 d\omega}{\int |H(\omega)|^2 d\omega}\right]^{1/2} \tag{2.17}$$

and the uncertainty principle must be satisfied:

$$\Delta_t \Delta_\omega \geq \frac{1}{2}. \tag{2.18}$$

The equality is only achieved for Gaussian window functions. This is sometimes written in terms of frequency rather than angular frequency, i.e.:

$$\Delta_t \Delta_f \geq \frac{1}{4\pi}. \tag{2.19}$$

Eqn. 2.8 is the DFT of the product of the signal and the analysis window, which due to a well known property of the DFT, can also be written as the convolution of the DFT of the signal with the DFT of the window. A result of this convolution and the fact that $\Delta_\omega > 0$ for any finite window length, is a broadening of all spectral lines. In other words, since the window function is of finite spectral width, then when it is convolved with the DFT of the signal, any spectral line in the original un-windowed signal would be replaced by the spectral shape of the window function after convolution. At a given frequency ω_k, eqn. 2.5 can alternatively be interpreted as filtering the signal with a bandpass filter having a finite impulse response $h[-n]$ modulated at the frequency ω_k. Both views indicate that $S[k, r]$ measures the signal's behaviour in a finite neighbourhood around a time-frequency point, i.e. roughly $[t_r - \Delta_t, t_r + \Delta_t]$, $[w_k - \Delta_\omega, w_k + \Delta_\omega]$, where t_r is situated within the r^{th} analysis frame. Since only a single type of analysis window is being used, and as N_{hop} is constant, then t_r and w_k are spaced equally apart on a rectangular time-frequency grid as illustrated in fig. 2.4a.

According to eqn. 2.18, there is clearly a trade-off between time and frequency resolution. The better localised the window function is in time, the worse the frequency resolution of the STFT, and vice versa. This property is not circumvented by over-sampling the STFT or zero-padding, and so must be used to greatest effect for the signal of interest with an appropriate choice of the window function. The implications for musical signals are that long window functions are required to accurately represent slowly time-varying partials, and short window functions are required to represent quickly time-varying or non-stationary segments. To make things worse, the nature of music is that both quickly time-varying and

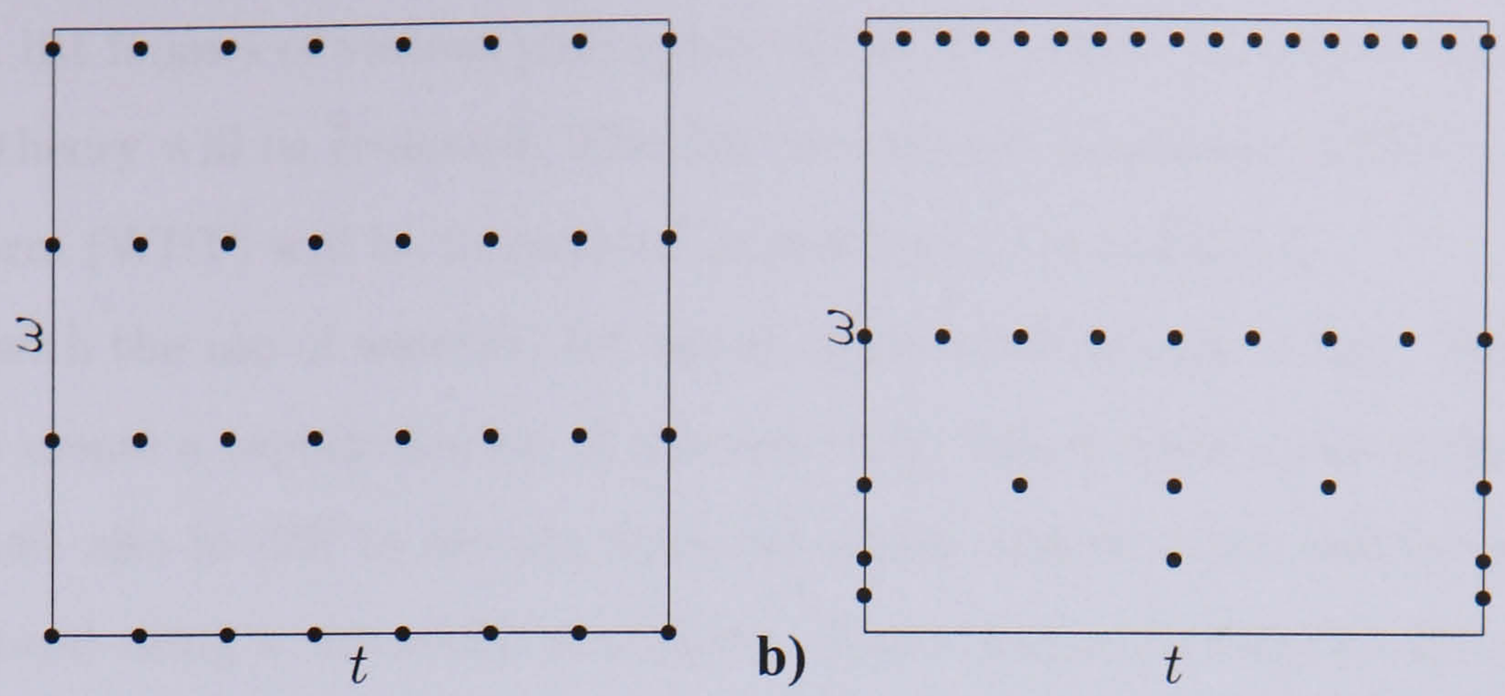

Figure 2.4: The time-frequency sampling grid for **(a)** the STFT, **(b)** the DWT

slowly time-varying content often occur simultaneously. It is therefore difficult to find a single STFT representation that captures all aspects of the signal's behaviour.

Various multi-resolution representations exist, i.e. where Δ_t and Δ_ω vary with frequency and/or time, such as the discrete wavelet transform (DWT). The DWT (section 2.1.4) has a dyadic time-frequency sampling grid as shown in fig. 2.4b, which can sometimes provide a better representation of the signal. A signal-dependent and optimal representation of the signal can also be obtained using wavelet packets (section 2.1.5). Another approach is to use an adaptive representation such as the pitch-synchronous wavelet transform[23], or a pitch-synchronous Fourier representation[24], which adapts to the periodicity of the signal (assuming there is at most one dominant periodic source at any instant, which is limiting for polyphonic music). Even so, multi-resolution and adaptive representations are, on their own, still single viewpoints of the signal, which don't always provide concise representations of all the different kinds of content one typically encounters in music signals, especially when different structures are overlapping in both time and frequency. Ideally, multiple representations of the signal, or a multidimensional representation, would be better suited to encoding all of the structured components of the signal. Of course, there would be technical issues concerned with analysing, re-synthesising or extracting information from such a representation. In this work, different signal representations have been used for extracting different kinds of musical structure, including the time-domain waveform, STFT, DWT and WPT. Each produces its own characteristic artifacts upon re-synthesis if any modifications are made within the transformed representation.

2.1.3 Wavelet analysis

Wavelet theory has become increasingly conspicuous in speech and music signal processing, not to mention in a variety of other applications such as image processing, applied mathematics, quantum physics and seismic geology. Introductory material on wavelet theory can be found in [9, 10], and there are a number of books available on the subject, such as

[25, 26, 27]. A list follows of various past applications of wavelets to music, after which the basic wavelet theory will be reviewed. The discrete wavelet transform (DWT) and wavelet packet transform (WPT) will be introduced in sections 2.1.4 and 2.1.5.

We begin with the use of wavelets for signal representation and coding. Wavelets were used in [28] to create a representation of musical audio based upon a sinusoidal plus transient model, and also in [29] to encode transient audio content after steady-state content had been removed using a sinusoidal model[30]. Multi-resolution filter-banks and wavelet representations were also suggested for improved sinusoidal modelling in [13], particularly to avoid the shortcomings of using fixed duration basis expansions such as the STFT, which are inadequate at representing transient content. An adaptive switched filter bank scheme was described in [31] for audio coding, that uses the discrete cosine transform (DCT) for encoding stationary time frames and the DWT for encoding non-stationary frames. The pitch-synchronous wavelet transform [23] and its extension, the harmonic-band wavelet transform[32], provide a frequency decomposition of a periodic signal in such a way that large scales encode the average periodic behaviour of the waveform and small scales encode the fluctuations from the average periodicity at different rates. A 1/f-like noise model for the spectral harmonic sidebands was discussed in [32, 33]. It was shown that the harmonic-band wavelet transform provides a convenient multi-resolution decomposition for estimating the characteristics of the 1/f decay, and creating synthetic sounds driven by white noise with similar harmonic behaviour. It is also able to separate the deterministic from the noise components of the sound.

One of the first accounts of applying the wavelet transform to speech and music processing was given in [34], where it was suggested that sound transformations such as pitch shifting, filtering and cross-synthesis could be accomplished by altering wavelet coefficients. It was shown in [35] how simple linear musical effects such as filtering and delay could be implemented accurately in the wavelet-domain using wavelet tables. De-noising of audio signals was performed by thresholding DWT coefficients[36] and thresholding using complex wavelets[37]. Wavelet transform derived features have also been used for instrument[38] and genre[39] classification, and general audio classification and beat detection[40].

The continuous wavelet transform (CWT) interprets the signal as a sum of time-translated dilations and contractions of a single prototype basis function or mother wavelet, $\psi(t)$. A scaled and translated copy of $\psi(t)$ is:

$$\psi_{\tau,a}(t) = \frac{1}{\sqrt{a}}\,\psi\left(\frac{t-\tau}{a}\right) \tag{2.20}$$

where τ is the time translation, a is the scale factor and the factor $\frac{1}{\sqrt{a}}$ has been added to

ensure that the wavelet is also of unit energy:

$$\int_{-\infty}^{\infty} |\psi_{\tau,a}(t)|^2 \, dt \;=\; 1. \tag{2.21}$$

The CWT of $x(t)$ is defined as:

$$W(\tau, a) \;=\; \int_{-\infty}^{\infty} x(t) \; \psi^*_{\tau,a}(t) \; dt. \tag{2.22}$$

Like the STFT, $W(\tau, a)$ measures the similarity between $x(t)$ and the basis function $\psi_{\tau,a}(t)$. However, for the STFT, the basis functions were translated and modulated versions of the window function, i.e. did not involve any time scaling. Eqn. 2.22 can be interpreted as a convolution of $x(t)$ with the impulse response of a bandpass filter $\psi^*_{\tau,a}(-t)$. If, for example, we choose the mother wavelet to be the modulated window or Gabor atom (strictly speaking this refers to a Gaussian shaped window) used in the STFT case:

$$\psi(t) \;=\; h(t) \; e^{i\omega_k t} \tag{2.23}$$

then:

$$\psi_{0,a}(t) \;\propto\; h\left(\frac{t}{a}\right) \; e^{i\frac{\omega_k}{a}t} \;=\; h\left(\frac{t}{a}\right) \; e^{i\omega t}. \tag{2.24}$$

This indicates that scale is inversely proportional to frequency: $a = \omega_k/\omega$. The act of expanding the mother wavelet ($a > 1$) would effectively decrease the modulation frequency, i.e. $\omega < \omega_k$. However, although the modulation frequency of a basis function has a well-defined meaning in the case of the STFT, in general there is no similar interpretation for a wavelet, i.e. ω is rather meaningless on its own. Hence, scale is a more meaningful variable than frequency for wavelet analysis, and the CWT is referred to as a time-scale representation rather than a time-frequency representation.

Associated with the CWT is the reconstruction formula:

$$x(t) \;=\; c_\psi^{-1} \int_{-\infty}^{\infty} \int_{0}^{\infty} W(\tau, a) \; \psi_{\tau,a}(t) \; \frac{da}{a^2} \; d\tau \tag{2.25}$$

which is valid as long as the admissibility criterion is satisfied:

$$c_\psi \;=\; \int_{0}^{+\infty} \frac{|\Psi_{\tau,a}(\omega)|^2}{\omega} \, \mathrm{d}\omega < \infty \tag{2.26}$$

where $\Psi_{\tau,a}(\omega)$ is the Fourier transform of $\psi_{\tau,a}(t)$. Eqn. 2.26 implies that $\Psi_{\tau,a}(0) = 0$, hence the wavelet has a bandpass property.

The scaling property of wavelets (eqn. 2.20) and the inverse proportionality of scale with frequency, means that shorter wavelets will be used to represent the signal behaviour at higher frequencies, and longer wavelets will be used to model lower frequency components. This also means that the time resolution of the CWT is better/worse at higher/lower frequencies respectively, and conversely, the frequency resolution is better/worse at lower/-higher frequencies respectively. In fact, the CWT is an example of a constant-Q analysis.

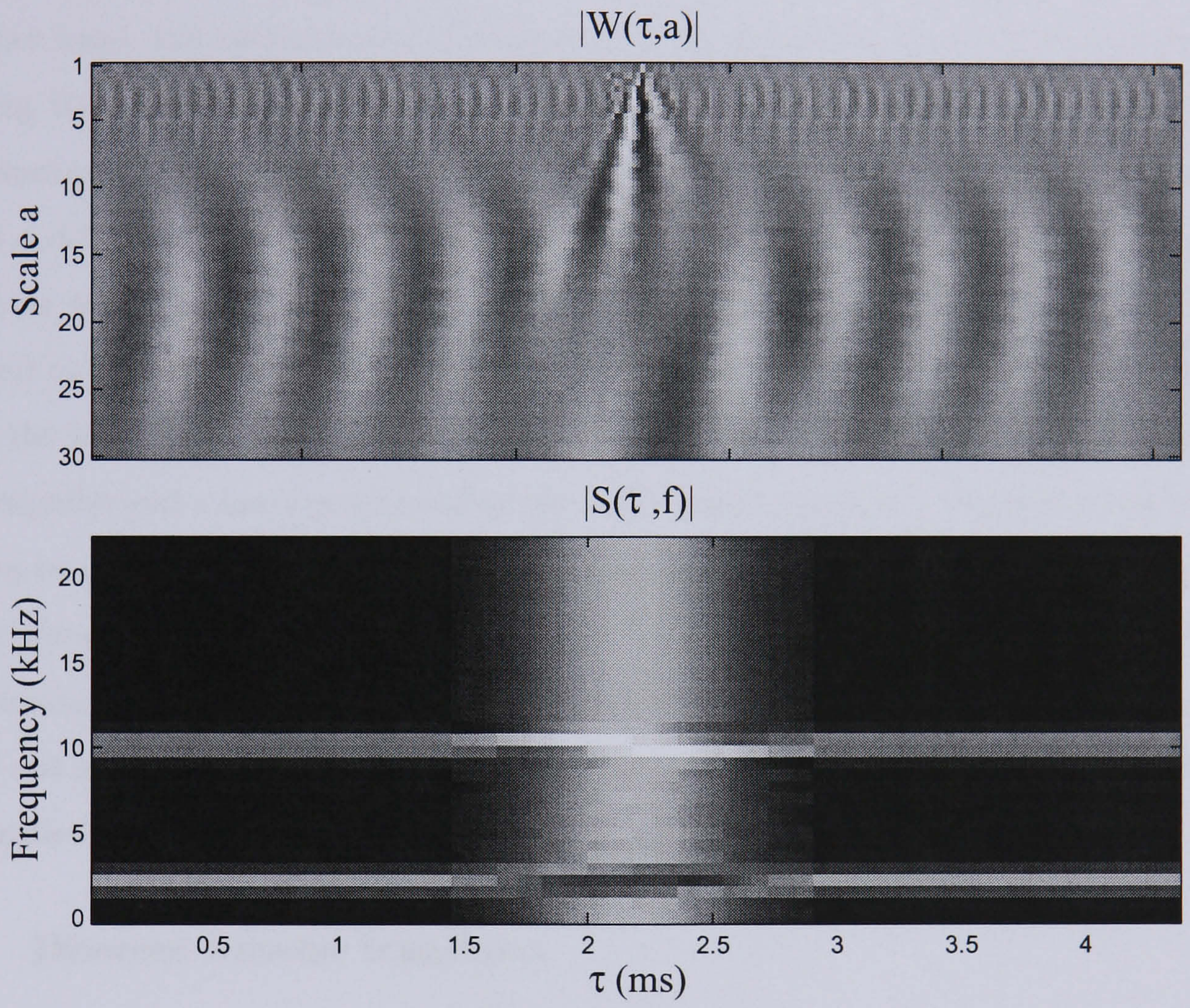

Figure 2.5: Magnitude of the CWT and STFT of a test signal containing a sum of a delta function and two sinusoids of frequencies 2 and 10 kHz, respectively (f_s = 44.1 kHz, a 'db–6' wavelet was used to calculate the CWT, and for the STFT, N = 64)

where $Q = \omega/\Delta_\omega$ measures the relative frequency resolution of the representation at the frequency ω.

One could argue that a constant-Q representation is well suited to musical signals. We know that the distribution of critical bands in the human auditory system is roughly logarithmic above about 1 kHz, and that the frequencies of notes in a chromatic scale are equally distributed on a logarithmic scale. These are both indications that less frequency resolution is required at higher frequencies. At small scales the CWT is able to represent very fine structure, which makes wavelet analysis particularly suited to modelling of transient signals[41], whilst higher scales simultaneously encode larger temporal structures. On the other hand, the extraction of stationary content or partials from audio is made easier by using the same time and frequency resolution at all frequencies. Fig. 2.5 compares the advantages/disadvantages of the CWT versus the STFT for a test signal containing a sum of a delta function which is non-zero only at one sample, plus two sinusoids of equal amplitude with frequencies of 2 and 10 kHz respectively (a Daubechies-6 ('db–6') wavelet was used to calculate the CWT and the window length of the STFT was roughly chosen to obtain the best overall representation of the signal). We see that excellent time localisation of the impulse and a fairly good localisation of the higher frequency sinusoid along the scale axis can be achieved from the CWT at small scales, although the lower frequency sinusoid is very spread out along the scale axis. The STFT is unable to discern details that are much smaller than the time resolution of its window function, and so the delta function is spread out in time. However, both sinusoids are represented very similarly and are fairly well localised in frequency in the STFT.

2.1.4 Discrete wavelet transform

The CWT given in eqn. 2.22 is highly redundant in the sense that a one dimensional continuous signal is mapped onto a continuous two dimensional time-scale plane. Thus, in practice, the CWT is regularly sampled at discrete time and scale positions. We represent the corresponding discrete set of wavelets as $\{\psi_{j,k}(t)\ ;\ j,k \in \mathbb{Z}\}$ with:

$$\psi_{j,k}(t) \ = \ \frac{1}{\sqrt{a_0^j}}\ \psi\left(\frac{t - k\ \tau_0\ a_0^j}{a_0^j}\right) \tag{2.27}$$

$a_0 > 1$ is the fixed dilation step, and $\tau_0\ a_0^j$ is the time step, which depends on the scale j. Similarly to eqn. 2.22, the wavelet coefficients at scale j and translation k are defined as:

$$W_{j,k} \ = \ \int_{-\infty}^{\infty} x(t)\ \psi_{j,k}^*(t)\ dt. \tag{2.28}$$

Two competing factors arise in sampling: to reduce redundancy the CWT should be sampled sparsely, but as perfect reconstruction is required, the sampling should not be too

sparse so as not to be able to reconstruct the signal from its set of wavelet coefficients. There exist certain families of orthonormal wavelet bases for which dyadic sampling is capable of perfect reconstruction. Dyadic sampling corresponds to $a_0 = 2$, and this non-uniform sampling of the time-frequency plane is illustrated in fig. 2.4b. Reconstruction is achieved from the wavelet coefficients using:

$$x(t) = \sum_{j,k \in \mathbb{Z}} W_{j,k} \, \psi_{j,k}(t). \tag{2.29}$$

Once again, eqn. 2.28 can again be interpreted as filtering the signal using a bandpass filter with impulse response $\psi^*_{j,k}(-t)$. Each time j is incremented, the scale of $\psi^*_{j,k}(-t)$ doubles and its frequency support halves. Thus, the dyadic sampling of the CWT resembles an octave spaced filter bank. For reconstruction we require that these bandpass filters provide sufficient covering of the frequency axis. Thus they should be at least slightly overlapping in frequency.

As j increases, the scale of the wavelet $\psi_{j,k}(t)$ increases, i.e. the corresponding wavelet coefficients represent larger scale features. Therefore, let us define the following that encodes the detail contained in the signal at level j:

$$D_j(t) = \sum_{k \in \mathbb{Z}} d_{j,k} \, \psi_{j,k}(t) \tag{2.30}$$

where $d_{j,k} \equiv W_{j,k}$ are the detail or wavelet series coefficients. It follows from eqn. 2.29 that the signal is the sum of all the details:

$$x(t) = \sum_{j \in \mathbb{Z}} D_j(t). \tag{2.31}$$

We can also say that at some level J, $A_J(t)$:

$$A_J(t) = \sum_{j > J} D_j(t) \tag{2.32}$$

is the sum of all details of scales larger than j, i.e. it is an approximation to the signal lacking the finer structure of the smaller scale details $j \leq J$. Clearly the signal is equal to a sum of the approximation at scale J plus all finer details:

$$x(t) = A_J(t) + \sum_{j \leq J} D_j(t). \tag{2.33}$$

It is also evident that the approximations at levels J and $J+1$ are related via:

$$A_J(t) = A_{J+1}(t) + D_{J+1}(t). \tag{2.34}$$

We return briefly to the idea of an octave spaced filter bank. Instead of computing an infinite series of bandpass filtered components where the centre frequencies of the bandpass filters

are monotonically decreasing, at some point it may be useful to terminate this operation and simply lump all further bandpass components into one low-pass filtered component. This low-pass approximation to the signal is $A_J(t)$. Now, similarly to the set of wavelets $\{\psi_{j,k}(t);\ k \in \mathbb{Z}\}$ being an orthonormal basis for the detail at scale j, we define a set of scaling functions, $\{\phi_{j,k}(t);\ k \in \mathbb{Z}\}$, that are an orthonormal basis for the j^{th} approximation. Hence, similarly to eqn. 2.30, the approximation at scale j can be written:

$$A_j(t) \ = \ \sum_{k \in \mathbb{Z}} a_{j,k}\ \phi_{j,k}(t) \tag{2.35}$$

where the approximation coefficients are:

$$a_{j,k} \ = \ \int_{-\infty}^{\infty} x(t)\ \phi^*_{j,k}(t)\ dt. \tag{2.36}$$

We wish to go now from the continuous time case to the discrete signal case. This is not, however, a simple case of replacing t by kT where T is the sampling period. If this was the case, we can see in eqn. 2.27 that wavelets obtained for $j > 0$ would yield non-integer sample numbers. However, it is fortunate that the computation of the discrete wavelet transform (DWT) does not explicitly require the wavelets or scaling functions. The link between the continuous and discrete time cases was provided in the multi-resolution theory of [42] and [43]. It was shown that the DWT could be computed using a pyramidal filter bank, depicted in fig. 2.6, by convoluting the signal with a pair of quadrature mirror filters (QMFs)[44]. Due to a property of the scaling function known as the two-scale relation, it turns out that the approximation coefficients: $\{a_{j+1}[k];\ k \in \mathbb{Z}\}$, can be computed as a weighted sum of approximation coefficients encoding the next finer level of detail: $\{a_j[k];\ k \in \mathbb{Z}\}$. Incidentally, the notation has been changed from $a_{j,k}$ to $a_j[k]$ to indicate discrete signals, in line with the filter bank interpretation of the DWT. We have:

$$a_{j+1}[k] \ = \ \sum_{n \in \mathbb{Z}} g[n - 2k]\ a_j[n]. \tag{2.37}$$

This resembles a convolution, and is equivalent to filtering $a_j[n]$ with a filter having an impulse response $g[-n]$, and then down-sampling the result by a factor 2. The detail coefficients at scale $j + 1$ can similarly be obtained from the approximation coefficients at level j:

$$d_{j+1}[k] \ = \ \sum_{n \in \mathbb{Z}} h[n - 2k]\ a_j[n]. \tag{2.38}$$

If we define the low-pass filter $\tilde{g}[n] = g[-n]$ and the high-pass filter $\tilde{h}[n] = h[-n]$, then together they form a pair of QMFs of length L (even) which are related according to:

$$\tilde{h}[L - 1 - n] \ = \ (-1)^n\ \tilde{g}[n] \quad ;\ n = 0, \ldots, L - 1. \tag{2.39}$$

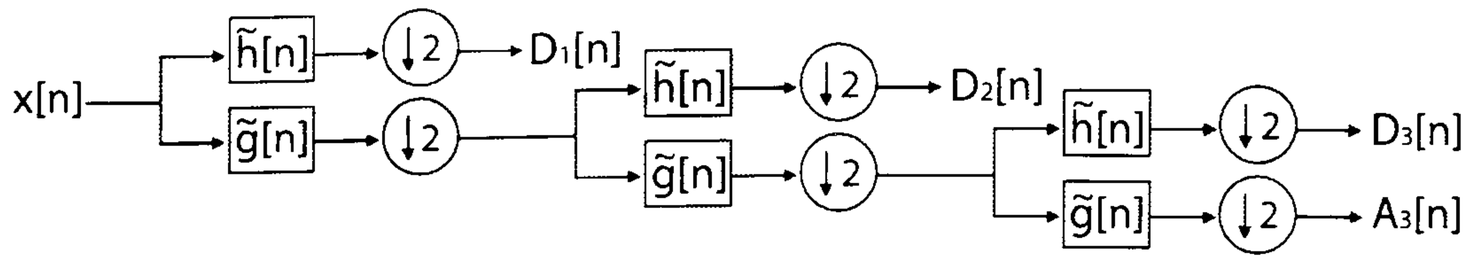

Figure 2.6: Digital filter bank implementation of the discrete wavelet transform (DWT)

If $\tilde{H}(\omega)$ and $\tilde{G}(\omega)$ are the DFTs of $\tilde{h}[n]$ and $\tilde{g}[n]$ respectively, the filters satisfy the condition of power complementarity:

$$|\tilde{H}(\omega)|^2 + |\tilde{G}(\omega)|^2 \;=\; 2. \tag{2.40}$$

It can be shown that the approximation coefficients at the highest level of detail necessary for a sampled signal are roughly equal to the sampled signal itself. If we initialise $j = 0$ at the finest level of detail, then the sequence of filtering and down-sampling operations can be initialised using $a_0[n] = x[n]$. From eqns. 2.37 and 2.38, the approximation and detail coefficients can be computed at any scale by a sequence of filtering and down-sampling by factor 2 operations on the original signal $x[n]$. This is illustrated in fig. 2.6 and is known as the discrete wavelet transform (DWT), where the word 'discrete' indicates that both time and scale are discretely sampled. At each node in the decomposition tree the approximation at this level is filtered into a low-pass signal encoding the larger scale features, and a high-pass signal which is the difference between the two approximations. At some point the sequence of filtering operations can be stopped, resulting in a set of detail signals at each scale, and a final low-pass approximation to the signal, analogous to eqn. 2.33.

The decomposition tree in fig. 2.6 can be inverted to perfectly reconstruct the signal from its approximation or detail coefficients. Like eqn. 2.34, it is possible to reconstruct the approximation coefficients at level j from a sum of detail and approximation coefficients at level $j + 1$:

$$a_j[k] \;=\; \sum_{n\in\mathbb{Z}} g[k-2n]\; a_{j+1}[n] + \sum_{n\in\mathbb{Z}} h[k-2n]\; d_{j+1}[n]. \tag{2.41}$$

Again this can be seen as a filtering process, but this time the approximation or detail coefficients at scale $j + 1$ are first up-sampled by a factor 2 by inserting zeros between consecutive samples, and then filtered using $g[n]$ or $h[n]$ accordingly. The reconstruction is shown in fig. 2.7 and is called the inverse discrete wavelet transform (DWT^{-1}).

Up until now, we have not actually described what the wavelets look like and have only used their properties. Fig. 2.8 shows a few sample wavelets from the Daubechies, symlets and coiflets wavelet families. As expected from the variation of shape of these wavelets in the time-domain, the choice of wavelet has a significant effect on the output of a wavelet-based analysis/synthesis system. There are a number of factors influencing the choice of wavelet. From the point of view of computational efficiency, wavelets characterised by short filter

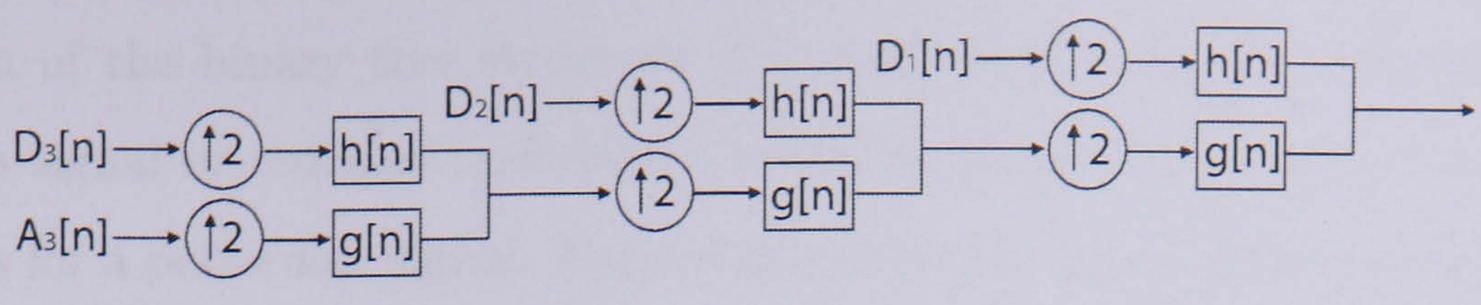

Figure 2.7: Digital filter bank implementation of the inverse discrete wavelet transform (DWT^{-1})

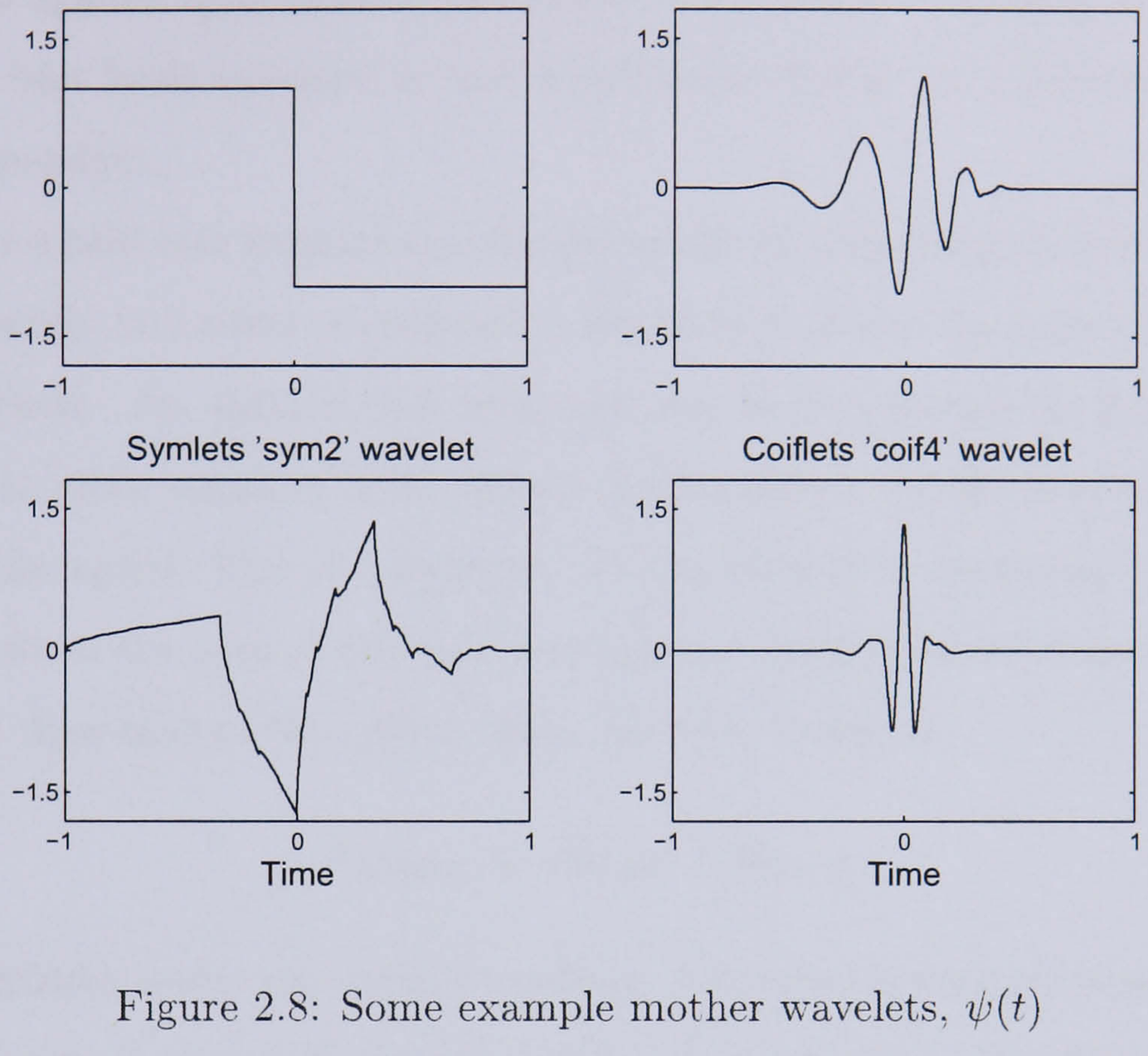

Figure 2.8: Some example mother wavelets, $\psi(t)$

impulse responses, $h[n]$ and $g[n]$, are advisable. However, longer impulse responses allow sharper transition band widths. Other factors to take into consideration are the number of vanishing moments, smoothness, symmetry, linear phase and time support of the wavelet. For coding purposes, a 'sparse' representation of the signal (i.e. one in which the signal is encoded using a minimal number of non-zero wavelet coefficients) would be preferable, leading to a different choice of wavelet than might be chosen if signal transformation or separation was the main objective. Thus, the selection of a 'best' wavelet basis is application and signal dependent.

2.1.5 Wavelet packet transform (WPT)

An obvious extension of the wavelet decomposition tree or pyramidal filter bank in fig. 2.6 is a structure in which a two-channel sub-band decomposition is applied to both low-pass/approximation and highpass/detail coefficients, as opposed to only the approximation coefficients in the DWT case. This is illustrated in fig. 2.9, and as the filter bank structure is equivalent to wavelet packet analysis, we will refer to this as the wavelet packet transform (WPT). An advantage of this decomposition is that a huge amount of flexibility is afforded

in the design of the binary tree structure (i.e. each node is either split into a high-pass and low-pass signal or remains undivided), resulting in the notion of a 'best tree' or best wavelet basis for a particular signal. For coding purposes, the best tree would be the one in which the most energy compaction and decorrelation of the transformed signal is achieved. In [45] it is described how to determine a best basis for decomposition from the point of view of compression. A Lagrangian cost function is computed at each node which trades off coding rate against quantisation distortion at a particular 'quality factor'. However, the particular best basis measure or cost function attributed to a decomposition is again application dependent.

A search for a best tree requires two things: a way of computing the cost associated with a particular branch, and a fast procedure for searching through the huge number of possible tree configurations. An optimal tree structure can be determined by initially expanding the tree fully to some maximal level, which can be chosen by the user or determined by the length of the signal. The tree structure is then pruned by removing 'child nodes' of a 'parent node' when the sum of the cost functions or entropy-based measures of the child nodes is larger than that of the parent node. In other words, if:

$$E_{parent} \leq E_{child1} + E_{child2} \tag{2.42}$$

then the two children nodes are merged together. A number of entropy-based measures that can be used in eqn. 2.42 and having an additive property exist. The Shannon entropy[46] has been used here:

$$E_s = -\sum_n s^2[n] \log\left(s^2[n]\right) \tag{2.43}$$

where $s[n]$ are the approximation/detail coefficients at a particular node.

Whilst wavelet packets have been shown to arise naturally from the filter bank structures of section 2.1.4, there is a dual interpretation of the WPT as a projection of the signal onto an orthogonal wavelet basis. Suppose we represent the band-pass signal at any node within the filter bank structure as $c_{j,n}[k]$, where j is the depth within the tree, n indexes the set of nodes at a particular depth, and k is the translation coefficient. $c_{j,n}[k]$ actually measures the projection of the original signal onto the time and frequency translated wavelet at scale j centred at time $t = 2^j\ k$:

$$\psi_{j,k,n}(t) = 2^{-j/2}\ \psi_n(2^{-j}\ t - k). \tag{2.44}$$

$\psi_n(t)$ has the same scale as the mother wavelet $\psi(t)$, but is translated in frequency according to the value of n. In summary, wavelet packets allow highly non-uniform tilings of the time-frequency plane, and thereby an expansion of the signal into a signal-dependent and orthogonal basis. Like the DWT^{-1}, an inverse filter bank structure exists for wavelet packets

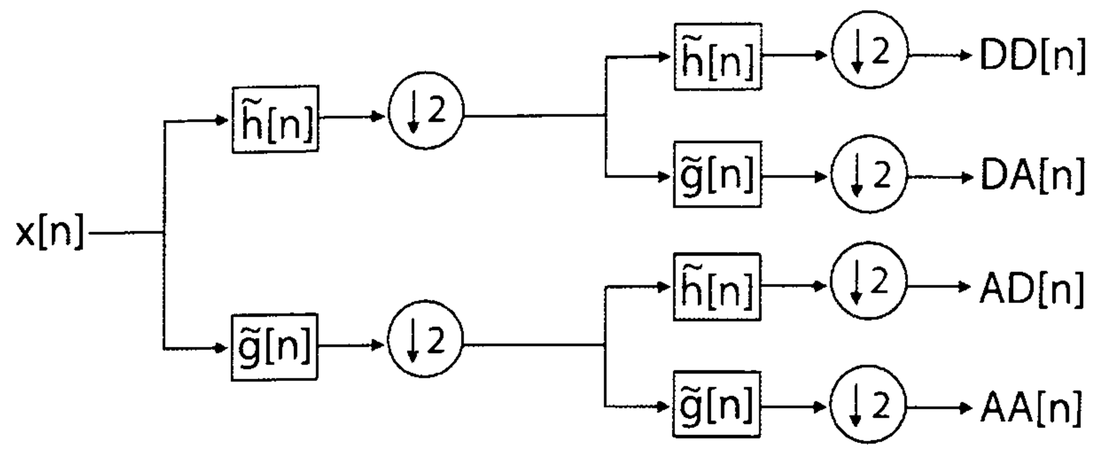

Figure 2.9: The wavelet packet transform (WPT)

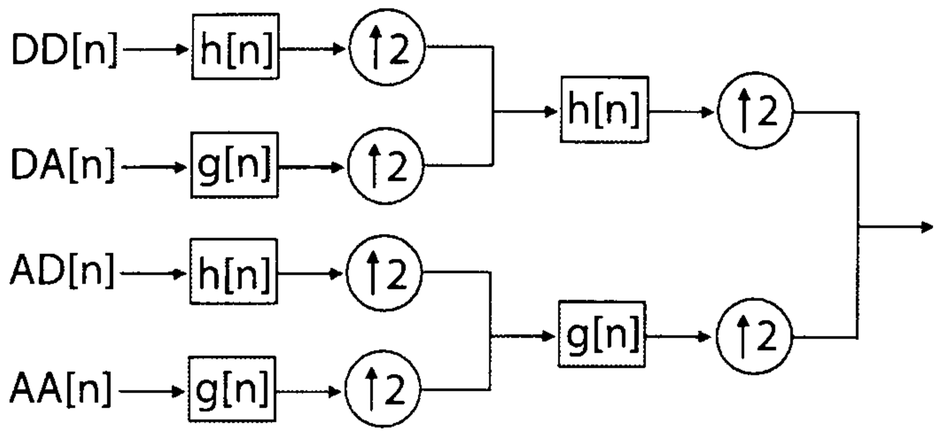

Figure 2.10: Reconstruction from the WPT using the inverse wavelet packet transform (WPT^{-1}).

allowing perfect re-construction of the signal, which will be referred to as the inverse wavelet packet transform (WPT^{-1}). This is shown in fig. 2.10.

2.2 Music signal modelling

Section 2.1 introduced the idea of a music signal representation, focusing on three alternative views of the signal: the STFT, DWT and WPT. We now turn to the problem of modelling the musical signal or extracting information from this representation.

It is natural to think of music as being a highly structured combination of smaller sonic elements such as notes. For the human listener, these elements can usually be identified fairly easily with their respective sources or instrument types, which seems to support the notion that elements produced by a particular source are acoustically similar in some sense. If these elements are notes, each one can be characterised using convenient terms such as 'attack', 'sustain', 'decay', and be viewed as containing distinct components such as partials, transients and noise. By modelling each of these components separately, we construct a complex model of the musical signal. However, there are a number of inadequacies in this approach. Firstly, it is debatable how structured the musical signal is even at a cognitive level[47], let alone at the signal level. Although the music usually 'makes sense' at some level or has some recognisable high-level structure, which usually goes in hand with a satisfactory listening experience, as soon as we try to quantify the elements of this structure, we ignore all signal content that does not fit these criteria. In relation to our description of the signal in terms of notes, we have probably failed to account for reverberation and other effects

processing, other non-instrumental sources in the recording environment, instrument sounds produced by unconventional gestures or perhaps muffled or shortened, and synthesised or acoustic ambient-like sources that are better described as continually changing timbres rather than sounds of finite duration. The existence of a potentially infinite variety of synthesised sounds and the reasons above makes any useful symbolic description of the signal in terms of a standardised set of 'sound events' incomplete. Even if we discard this idea, and regard the signal as containing distinct and mutually-exclusive non-symbolic components such as transients, partials and noise, there are still problems. For example, when exactly does the transient excitation that eventually settles into a stable vibrational mode become a partial? Also, if transient content is characterised by rapid increases in broad-band energy, when is the energy considered to have subsided enough so as to be labelled noise rather than transient content? Thus, an all-encompassing and descriptive music signal model is realistically either unattainable or ambiguous. A sum of several musical sources in a recording, where the number of channels is less than the number of sources, is accompanied by a loss of information except in the most trivial of cases. Given that we cannot extrapolate this lost information after mixing by using a musical signal model whose parameters have (hypothetically) been estimated perfectly, the proposition of perfectly separating a set of overlapping sources from a mono recording is implausible. With these cautions in mind though, the question that should really be asked is whether the musical signal model is sufficiently flexible and accurate to be useful in a particular application, for example one of those listed in section 1.1, where the requirements of fidelity differ from application to application.

Although signal representation and signal modelling are described here as separate entities, there is some overlap between them. The STFT for instance, although generally considered a representation, can also be seen as the estimation of model parameters in which the signal is modelled as a weighted sum of Gabor-like time-frequency atoms. Similarly, the DWT finds the coefficients of a set of wavelets modelling the signal. If the signal is reduced into a sub-set of atoms or dictionary elements as opposed to a complete basis, such as when using the matching pursuit algorithm[48], it is even more difficult to say whether a representation or model of the signal has been obtained. Here, a representation is considered to be complete in the sense that it provides a complete covering of the time-frequency plane. All the representations used here will also be invertible, allowing perfect reconstruction of the signal. Conversely, a signal model is incomplete in its covering of the time-frequency plane, and is able to produce a perfect reconstruction only in trivial cases.

As there seems to be sufficient evidence of qualitatively different structures in the musical signal (e.g. partials, transients and noise), although their exact distinction is not very

clear as mentioned above, different separation algorithms will be used to separate different structures within the sound, rather than designing a unified signal model or separation method. Iterative subtraction of multiple structures is performed to avoid separating the same content more than once. This means that for each structure that is separated from the recording, a residual is calculated, and all further structures are extracted directly from the residual.

A thorough review of various approaches to audio signal modelling is given in [13]. These include sinusoidal modelling using multi-resolution representations and adaptive time segmentation, perceptual models of the non-sinusoidal residual, complete and over-complete basis expansions, atomic decompositions calculated using the matching pursuit algorithm, and pitch-synchronous methods. We continue with a description of sinusoidal modelling for the extraction of partials from audio (section 2.2.1). Then, noise modelling of the non-sinusoidal residual (section 2.2.2) will be discussed, and transient modelling is reviewed in section 2.2.3. Signal modelling in terms of atomic decompositions (section 2.2.4) and masking of time-frequency cells (section 2.2.5) will also be mentioned.

2.2.1 Sinusoidal modelling

The sinusoidal model is probably the most widely used signal model in speech and music processing, see for example [30, 49, 50, 51]. Any acoustic source that has resonance frequencies or vibrational modes, or synthetic source containing a deterministic component is a good candidate for sinusoidal modelling. In fact, any source at all can be modelled as an infinite sum of sinusoids with coefficients given by its Fourier transform, but for practical reasons the number of sinusoids is limited so that only the quasi-stationary deterministic component of the sound is modelled.

The deterministic component of a sound can be modelled as a sum of $M \equiv M(t)$ sinusoids with time-varying amplitudes $a_m(t)$ and frequencies $f_m(t)$:

$$x(t) \;=\; \sum_{m=1}^{M} a_m(t)\;\cos\left(\phi_m(t)\right) \;=\; \sum_{m=1}^{M} \frac{a_m(t)}{2}\;\left[\; e^{i\phi_m(t)} + e^{-i\phi_m(t)} \;\right]. \tag{2.45}$$

The phase of the m^{th} sinusoid is measured relative to an initial phase $\phi_m(t_0)$ at time t_0:

$$\phi_m(t) \;=\; 2\pi \int_{t_0}^{t} f_m(t')\; dt' + \phi_m(t_0) \tag{2.46}$$

and the instantaneous frequency of the m^{th} sinusoid is the time derivative of its phase:

$$f_m(t) \;=\; \frac{1}{2\pi}\;\frac{d}{dt'}\;\phi_m(t')\Big|_t \;. \tag{2.47}$$

The amplitudes $a_m(t)$ should be allowed to be sufficiently time-varying to model the attack and decay of sinusoids and any amplitude modulations. Similarly, the frequency trajectories

$f_m(t)$ should allow for small amounts of frequency modulation which could occur due to vibrato, glissando or other effects. In [52] it is advised that the frequency behaviour should be formulated in terms of variations of relative slope of the trajectory, rather than relative value. The sinusoidal frequencies, amplitudes, and phases are typically measured within each time frame with a hop size between consecutive frames of N_{hop} samples. In the McAulay-Quatieri (MQ) method[30] the signal was split into time frames, and the DFT was computed in each time frame after windowing the signal with a Hamming window. It was shown that for perfectly voiced speech in idealised conditions, an optimal estimator of the sinusoidal frequency is the corresponding DFT magnitude peak frequency. Given the estimated sinusoidal frequency, its amplitude and phase can then be estimated from the complex value of the DFT at the peak frequency. Given that in practice, spectral peaks almost never occur exactly at a frequency bin, methods will be described in section 4.2 for estimating spectral peak parameters in the non-ideal case. These estimates can be used to describe the evolution of the sinusoidal parameters over time.

A sinusoidal subtraction or synthesis method requires that the sinusoidal parameters be interpolated across frame boundaries. A sinusoidal modelling method designed for speech analysis and synthesis, the MQ algorithm[30], uses a cubic phase interpolation function:

$$\phi_m(t) = \zeta + \gamma t + \alpha t^2 + \beta t^3 \tag{2.48}$$

which results in a quadratic frequency interpolation according to eqn. 2.47. The interpolation coefficients are determined by matching sinusoidal frequencies at consecutive time frames with an additional requirement that the phase interpolation function be 'maximally smooth'. Although the MQ model has been used on many occasions to good effect, it is often the case that the variation of partial frequencies is closer to a sinusoidal evolution than a polynomial one. Thus, it is important to choose a small enough hop size that a quadratic approximation of the partial frequency between consecutive time frames is valid.

It should also be mentioned that artifacts of the sinusoidal model such as pre-echo distortion (this occurs when an event that occurs at the end of an analysis time frame is spread across the entire frame after a spectral transformation and upon re-synthesis) and smoothing of transient events can partially be avoided by using analysis and synthesis windows that are synchronised in time with transient event boundaries[53]. Adaptive time segmentation for sinusoidal modelling is also discussed in [13], and it is suggested how appropriate time-frequency tradeoffs be applied in different regions of the signal by using variable window lengths. Thus, shorter windows are used in transient regions and longer windows are used in stationary regions.

Partial tracking

It is implicit in eqn. 2.45 that the number of sinusoids at any particular time, M_t is variable, i.e. the sinusoidal model allows for the birth and death of sinusoids. Partial tracking algorithms are used to track the sinusoidal parameters from frame to frame, and to determine when new partials begin and existing ones terminate. They should be robust to noise, as the presence of noise and side-lobes can give rise to DFT peaks which can be misconstrued as sinusoidal content. They also encompass rules which govern the allowable frequency variation of partials between frames, particularly in the situation that multiple sinusoids cross or become very close in frequency and the continuation of the trajectory of each partial is not obvious. In [30] a simple rule-based system was used to track DFT peak frequencies across consecutive time frames. A similar rule-based algorithm[54] predicts sinusoidal frequencies in future time frames using a linear predictive model computed on the frequency evolution of partials in past time frames. The partial tracking algorithm in [55] projects sets of spectral peaks in consecutive time frames into states of a hidden Markov model (HMM), and the optimum sequence of states is determined using the Viterbi algorithm. Other approaches to partial tracking include synchronising adaptive oscillators to the output of an auditory model[56], Kalman filtering[57], and a pinching plane method applied to the spectrogram[58].

Partial tracking algorithms can be aided using peak selection procedures that attempt to discriminate between sinusoidal and stochastic spectral peaks. A sinusoidal likeness measure is given in [52] which quantifies by computing a spectral correlation, the similarity between a spectral peak and the shape of the Fourier transform of the window function. As this method is not very robust to non-stationary sinusoids, a phase derived sinusoidality measure designed with a model of linear frequency variation was given in [59]. To discriminate modulated sinusoids from stochastic peaks, [60] used as a sinusoidality measure the correlation between the measured spectrum and the spectrum of a frequency modulated sinusoid.

Grouping of partial tracks

The previous section reviews the extraction of partials or sinusoids from a speech or music recording. If the intention is to separate the partial or harmonic content of a single source from the mix, some way of finding the subset of estimated sinusoidal tracks belonging to the desired source must be devised. Some extracted sinusoidal tracks may have been due to spurious spectral peaks, whilst others may have arisen from interfering sources. One approach is to use Gestalt grouping cues[61] to measure the similarity between different partial tracks, where it is assumed that partials from the same source are more similar

than those from different sources. Some perceptual grouping cues are common onset and offset time, common amplitude/frequency modulation, spectral proximity and spatial proximity. If the desired source is pitched, then we can also exploit the fact that its harmonics are distributed at roughly integer ratios of the fundamental frequency or pitch. In [62] the perceptual distance or similarity between pairs of sinusoids was calculated using three measures: the mean square error between the normalised frequency trajectories, the mean square error between the normalised amplitude trajectories, and a measure of harmonic concordance. The set of detected sinusoids was then split into classes having a minimum total error between trajectories within the same class. Elsewhere[58], onset synchrony of sinusoids was used for grouping frequency components into notes in a system for hierarchical description of music. In [63] the similarity between adjacent partial amplitude and frequency envelopes was used to estimate a de-mixing matrix for overlapping partials from multiple sources. A statistical *a posteriori* probability estimator is described in [64] for estimating a set of note events, given a partition of partial tracks into note events and those not associated with any note event. A likelihood function reflects how well a particular note event is described by a set of partial tracks, and exploits grouping cues such as onset/offset synchrony, harmonicity, and partial density or support. It also favours the presence of the first and second harmonic and an overall larger number of harmonics, and penalises missing partials.

2.2.2 Sinusoidal + noise decompositions

Whilst the parameterised form of the sinusoidal model allows a variety of interesting musical transformations such as time-stretching, pitch-shifting and timbral modifications, it is not ideal for modelling noisy signals. Although theoretically it is possible to represent a noise signal as a sum of sinusoids, it is impractical as noise potentially consists of components at all frequencies within the band limits. This is the motivation behind the deterministic plus stochastic decomposition known as spectral modeling synthesis (SMS)[50, 65, 66]. This general analysis/synthesis method can be used for processing or transforming existing sounds, or for generating new sounds based upon instrument models. In SMS, the deterministic component of the sound is modelled as a sum of time-varying sinusoids, and the stochastic component is approximated as white noise shaped by a time-varying filter. Synthesis of the deterministic component is achieved by additive synthesis, i.e. by summing a set of oscillator outputs with time-varying amplitudes and frequencies. The deterministic component is then subtracted from the original sound in the time-domain[50] using the McAulay-Quatieri algorithm[30] or in the magnitude spectral-domain[66], to produce a residual signal, which is assumed to be completely stochastic. The time-domain subtraction, although more com-

putationally expensive, is favoured in [50]. It facilitates the use of a shorter window length for analysing the stochastic component, than that which is necessary for obtaining sufficient frequency resolution in the analysis of the deterministic component. The stochastic component is regarded within each time frame as a frequency dependent power spectral density, i.e. it is assumed that only magnitude information needs to be preserved in the residual component. A noise magnitude envelope is obtained from a line-segment approximation to the residual spectrum, but could equally well be approximated using another curve fitting technique or linear predictive coding. The re-synthesised stochastic component is obtained by applying the DFT^{-1} to the noise envelope with added random phase, and then using an overlap-add technique on the resulting time segments to avoid discontinuities at frame boundaries.

The approximation of the stochastic or residual component as white noise shaped by a frequency dependent noise envelope is a theme which will reappear in chapter 5. In section 5.2 the noise component is split into frequency bands, and we will assume that the noise content is adequately described by the energy envelope within each band. This is supported by simple auditory models of noise perception when these bands are spaced according to the critical bands of the human hearing system[13]. Section 5.3 also considers the noisy residual to be a frequency dependent energy envelope in much the same way.

2.2.3 Modelling transients

Whilst the sinusoidal plus noise decomposition known as SMS[50, 65, 66] overcame some of the inadequacies of sinusoidal modelling by explicitly modelling the non-sinusoidal component of the sound, it is founded upon the assumption that the residual or non-sinusoidal component is purely stochastic, which at times is invalid. For example, the non-sinusoidal component may be more complex or 'textured' than random noise, and we would therefore expect that the phase content of the residual spectrum is also important. Secondly, at times the sinusoidal subtraction is imperfect, and so a small component of the partial content can remain in the residual. Furthermore, the processing of transients within the SMS framework can be unconvincing. Transients modelled as filtered white noise loose their sharpness of attack and suffer from the artifacts of using finite window lengths. For this last reason it is of interest to build an independent model for transient signals. Although a precise definition of a 'transient' does not exist, it is usually used to describe rapid increases in the temporal envelope of the waveform, visible in a time-frequency representation as an increase in broad-band noise energy. This is usually followed by a slower decay of broad-band energy after the initial attack. In an acoustic instrument a transient component is often present at the note attack, and constitutes a perceptually significant part of the note. Without the

transient attack a note can sound quite dull, and its use in instrument identification by humans has been noted (section 6.2).

The SMS framework was extended in [67] with a flexible model of transient signals, and renamed transient modeling synthesis. Transient modeling synthesis incorporates a transient signal model directly into the SMS framework by using the existing techniques for sinusoidal modelling, but applying them in the discrete cosine transform (DCT)-domain instead of the time-domain. It is argued that, just as slowly varying sinusoids are impulsive in the frequency-domain, transient signals, which are impulsive in the time-domain, should be oscillatory in a properly chosen frequency-domain. In fact, the location of the transient signal within a block of audio determines the sinusoidal frequency in the DCT-domain. The full analysis system of transient modeling synthesis begins by subtracting sinusoidal content from the original waveform, resulting in a residual containing transient and noise content. The transient content is then extracted from the residual as described above, forming a second residual which contains only the noise component. However, it is not always necessary to perform the three way decomposition unless there is actually evidence that all three components exist in a section of audio. For this reason, a tonality criterion is used to detect when sinusoidal or transient content is present. It was discussed in [67] how time-scaling and pitch modifications could be performed, where separate control of transient and sinusoidal content is necessary to retain the integrity of the signal.

We continue with some other approaches to transient modelling. Exponentially damped sinusoids have been used to provide an efficient audio model for coding[68, 69, 70]. The exponentially damped sinusoid (EDS) model[68] is of the form:

$$x[n] = \sum_{m=1}^{M} a_m \, e^{\gamma_m n} \cos\left(w_m n + \phi_m(t_0)\right) \tag{2.49}$$

where γ_m is the damping factor for the m^{th} sinusoid, and $a_m \, e^{\gamma_m t_0}$ is its initial amplitude. It is clear that the stationary sinusoidal model is a special case obtained when $\gamma_m = 0 \; \forall \; m$. The advantage of the EDS model is that attacks or fast time-varying signals can be modelled efficiently with damped sinusoids. In the EDS model, an adaptive segmentation of the signal is advisable to ensure that transient events occur near the beginning of the segment $x[n]$. This facilitates an efficient representation as a sparse set of decaying sinusoids. The damped and delayed sinusoidal (DDS) model[70] extended the EDS model to avoid artifacts such as pre-echo distortion by introducing a delay parameter for each component. The partial damped and delayed sinusoidal model[69] is a special case of the DDS model, which groups together DDS components with the same time-delay in order to model transient attacks.

Overcomplete dictionaries have also been used for modelling transient components[41]. providing an efficient decomposition of the signal using the matching pursuit algorithm

into a set of dictionary elements. In [41] the dictionary elements are the wavelet functions that implement a wavelet packet filter bank. A wavelet packet decomposition is also used to encode the non-sinusoidal residual signal in [29], and it is explained how the residual component can be split into high-frequency wavelet coefficients encoding transient edges, and wavelet coefficients determined by a noise model which encode the remaining noise content.

Transient attacks have been represented elsewhere as aggregates of time-frequency bins within a STFT representation[71]. Non-steady-state content (transients plus noise) was separated from steady-state deterministic content in [72] by applying a threshold to the phase increment between frequency bins of the STFT in adjacent time frames. The phase increment of a frequency bin containing mainly partial content would be expected to vary less than if it contained non-stationary or stochastic content.

2.2.4 Atomic decompositions and the matching pursuit algorithm

The STFT can be thought of as an expansion of the signal in terms of frequency and time translated atoms, each atom being a modulated version of the window function. It is natural to extend this idea to other types of atoms or basis functions. An overcomplete or redundant dictionary of atoms allows the coding of the signal in terms of a minimal set of atoms that provide an optimal fit to the signal. Some examples of atomic dictionaries are dictionaries of Gabor atoms[73, 74], complex exponentials[75, 76], wavelets[41], real sinusoids[77] and damped sinusoids[13].

The expansion of the signal into a finite sum of dictionary elements can be achieved using the matching pursuit (MP) algorithm[48]. This is a greedy algorithm in the sense that the residual at each iteration is projected onto the dictionary element with which it has the closest match. The residual in the following iteration is the difference between the residual at the present iteration and the projection of the current residual onto the best matching element. The method will now be described in slightly more detail.

Let $d_{j(m)}[n]$ be the dictionary element from within a set of unit norm dictionary elements $\mathcal{D}$ that best matches the m^{th} residual $r_m[n]$. The notation $j(m)$ shows explicitly that the best dictionary element j is specific to the iteration number m. By the orthogonality principle, it can be shown that $d_{j(m)}[n]$ is the element that maximises the magnitude of the projection:

$$\arg\max_{d_k \in \mathcal{D}} |\langle r_m, d_k \rangle| \;=\; \arg\max_{d_k \in \mathcal{D}} |\alpha_k| \,. \tag{2.50}$$

To clarify, the constant α_k measures the projection of the dictionary element d_k onto the residual r_m. The $(m+1)^{th}$ residual $r_{m+1}[n]$ can then be computed as previously stated:

$$r_{m+1}[n] \;=\; r_m[n] - \alpha_m \, d_{j(m)}[n]. \tag{2.51}$$

The iterative process is initialised with $r_0[n] = x[n]$. At each iteration the residual decreases according to:

$$\|r_{m+1}\|^2 \;=\; \|r_m\|^2 - |\alpha_m|^2 \tag{2.52}$$

where we have used the fact that the dictionary elements are of unit norm. It can also be seen that the dictionary element chosen in eqn. 2.50 minimises the 2-norm of the residual $\|r_{m+1}\|^2$. Providing that the dictionary is complete, the residual decreases at each iteration and gradually tends to zero. After a number of iterations M, the process can be terminated according to a threshold criterion on the residual energy or a maximum preset number of iterations. Finally, the signal can be approximated as a weighted sum of dictionary elements:

$$x[n] \simeq \sum_{m=1}^{M} \alpha_m \; d_{j(m)}[n]. \tag{2.53}$$

Variations on the MP algorithm exist in which a sparse decomposition of the signal is required in terms of the dictionary elements that have the most perceptual significance. These include the weighted matching pursuit algorithm[75] in which the dictionary elements are allowed to have non-unit norms, and the psychoacoustic-adaptive matching pursuit algorithm[77].

Whilst sparse atomic decompositions provide compact signal representations which have obvious benefits for audio coding, they do not offer the same flexibility for music processing as parametric models such as the sinusoidal and noise model. They have been included here to make the discussion of musical signal models more complete.

2.2.5 Masking/grouping of t-f cells

We complete the section by mentioning an approach to musical signal modelling that involves masking or grouping of time-frequency cells in an invertible representation such as the STFT. A component of a signal can be represented by applying a mask to the STFT which is non-zero only at those cells in the STFT representation in which the signal is expected to be present. The component can then be re-synthesised from the masked STFT using an overlap-add technique. On the one hand, this can be viewed as a sparse atomic decomposition where the dictionary elements are the basis functions of the STFT (i.e. windowed exponentials). Alternatively, we may think of it as a limited or filtered depiction of the signal. The shape of the mask, i.e. where it is non-zero, can be determined by fitting a signal model or template to the STFT. This is basically the approach taken in chapter 4, where the harmonic content of a particular note is extracted by multiplying the DFT coefficients in each time frame by a mask or comb-like filter. Alternatively, spatial information in multi-track recordings has been used to identify those cells in which a particular source is dominant[78, 79, 80].

The simplest case is a binary mask, meaning that the coefficients of the mask are either zero or one. It is then assumed that the different sources or components in the signal are disjoint in the time-frequency plane, i.e. each cell contains energy from at most one source. This condition is referred to as W-disjoint orthogonality[80], and is generally more common in speech signals than in music signals. This is due mainly to the fact that music sources often play harmonically related notes, resulting in a fairly large incidence of overlapping harmonics in the time-frequency plane. In [81] low amplitude drum sounds were separated from percussion tracks using binary time-frequency masks, aided by a prior drum transcription and statistical instrument models. It was found that high-frequency percussive attacks, such as hi-hats and cymbals, when overlapping with kick drums, could be considered W-disjoint orthogonal.

Section 4.6 deals with the case where W-disjoint orthogonality is not satisfied, by constructing weighted masks at overlapping harmonics, which share the energy in a particular time-frequency cell between the interfering sources. Along the same line, in [63] time-frequency regions in a STFT representation containing one or multiple overlapping partials from different sources are identified. These are then used to determine a mixing matrix for a multi-channel mixture that allows the estimation of the individual STFTs of each source. Elsewhere, aggregates or clusters of time-frequency cells which appeared at close temporal locations were associated with transient events, and were used for extracting transient content[71].

2.3 Conclusions

This review chapter has established some foundations for further work, in particular the STFT, DWT and WPT. It has also set the research context for much of the work in this thesis, a large part of which is focused on music signal modelling. A number of different signal representations and modelling frameworks for music signals have been mentioned. The broad perspective is that the nature of music is often unpredictable and multi-faceted, and is best understood by the use of multiple representations or modelling frameworks. We have established the basic principles of partial, transient and noise modelling, which reappear in chapters 4 and 5.

Chapter 3

MIDI to Audio Alignment

"I don't know anything about music. In my line you don't have to."

- Elvis Presley (1935–1977)

This chapter focuses on the inclusion of prior knowledge into the separation problem, specifically note timing and pitch information. Without this prior information some means of locating note segments within the recording would be necessary, and the note pitches are also a pre-requisite for extracting harmonic content in chapter 4. When dealing with polyphonic music, segmentation and multi-pitch estimation are difficult problems in their own right. Automatic music transcription (AMT) is the area of research concerned with automatically extracting a musical score or transcription from the recording. The transcription usually consists of a set of notes with estimated onset times, durations and pitches, very much like the conventional western stave notation or MIDI representation. It should be noted though, that existing polyphonic AMT systems rarely identify or label each note with a particular source or instrument track, whereas this identification is implicit in a MIDI/score-based representation. We will return to this point in chapter 6, where clustering methods will be implemented for automatically grouping unclassified notes within the recording into different source types.

Several approaches to AMT can be found in the literature, and a few of these will be mentioned. One of the earlier approaches is a frequency tracking algorithm for separating duet voices[82] based upon sinusoidal modelling using the McAulay-Quatieri algorithm[30]. The systems in [83, 58, 84] integrate bottom-up signal driven processing with high-level prior information or expectations using blackboard architectures. In [64] a model of partial behaviour is used to guide segmentation algorithms. The systems in [85, 86] incorporate prior information into a statistical hierarchical Bayesian framework for estimating the parameters of a music signal model. A system designed specifically for transcription of the dominant melody and bass line is described in [87]. Transcription of percussive instruments

was dealt with in [81] by combining statistical blind separation techniques such as independent component analysis with prior knowledge. Notes were identified as salient features when applying non-negative sparse coding to STFT spectra[88], and [89] reviews several transcription methods based upon blackboard models, multiple-cause models. independent component analysis and sparse coding. Two iterative subtractive methods were presented in [90] for polyphonic pitch estimation and musical meter estimation. Finally, [91] gives a comparative review of several transcription systems, focusing on polyphonic pitch estimation and musical meter estimation. Although a quantitative comparison of different complete AMT systems was not found, comparative results are available for some polyphonic pitch estimation and meter estimation systems[92]. In test mixes of randomly selected notes, the multi-pitch estimator in [93] produced note error rates (number of pitches estimated in error divided by the total number of pitches in the reference transcription) of 1.8%, 3.9%, 6.3%, 9.9%, 14% and 18% for polyphonies ranging from one to six notes when the degree of polyphony was known in advance, although we could expect these to be lower for real recordings.

The task of AMT would require a large effort on its own to fully explore, so instead, the intention has been to concentrate on separation algorithms given a prior transcription, score or MIDI representation of the corresponding audio. The accuracy of the various transcription systems above indicate that an AMT pre-processing system would in future be a feasible alternative to the inclusion of this prior information. To be precise, a MIDI representation has been used, although one could easily adapt the system to input a score in standard Western music notation.

The alignment of audio with a symbolic music notation is known as score following. We wish to provide a temporal alignment of the MIDI information with the audio, given that the MIDI and audio file may differ with respect to tempo and note durations. Two main approaches to time-warping exist, the first uses dynamic programming[94, 95, 96, 97] or dynamic time warping algorithms[98], and the other uses hidden Markov models (HMMs)[99, 100]. However, it is usually not possible to obtain a prior score or MIDI information for a given popular recording, so it was not deemed to be worthwhile to pay too much attention to the score following task, given that the intention in the longer term is to replace this with an AMT system. Therefore, it was decided that the user would improvise a MIDI accompaniment for each instrumental part to be separated, whilst concurrently listening to the recording. The accompaniment is thus a near replica of each instrumental part contained within the recording. Practically speaking, the user does not produce a perfect transcription, i.e. there are normally slight note timing inaccuracies. Also, due to the fact that a MIDI note pitch is itself a static representation of what is generally a

time-varying pitch envelope, it is still necessary to make some slight adjustments to the MIDI data. We wish to refine the timing and pitch information in the MIDI data to more closely resemble the corresponding audio.

Although the user-improvised MIDI accompaniment has been suggested as a way of concentrating on the separation problem rather than on AMT, and a fully automatic AMT-based separation system has not yet been developed here, the present system is already useful in applications for which user-input is not a hindrance. There are certain applications in which a user could realistically be expected to create a MIDI accompaniment for one or more source/s, such as remastering a classic recording, or extracting a short sample from an existing recording to re-use in a new composition. For any given application, there is often an optimal balance between the amount of prior information that can be included and the required fidelity, so the compulsory inclusion of prior information should not necessarily be seen as a weakness of the system, as long as it is capable of yielding better results than an equivalent fully automatic system.

The chapter proceeds by discussing two aspects of the MIDI to audio alignment: aligning note onset times (section 3.2) and transforming MIDI pitch values into time-varying pitch envelopes (section 3.3). Section 3.2 involves the use of the note onset detector described in section 3.1.

3.1 Note onset detection

The alignment of a set of user-improvised MIDI note onset times with the audio recording involves two stages: a note onset estimation stage, and an alignment stage which matches a set of MIDI notes with the estimated note onsets, the latter of which is described in section 3.2. Many alternative methods for note onset detection exist, for example [53, 101, 102, 103], with comparative reviews given in [104, 105]. A general consensus is that the accuracy of different onset detectors depends on the characteristics of the recording and differs from genre to genre. It was decided to use the complex-domain onset detector described in [106, 107, 108], which is fairly robust and has been compared rigorously with other methods in [104]. It is based upon a complex spectral difference estimator, and is capable of detecting both percussive onsets, characterised by sharp increases in energy, and softer 'tonal' onsets, which are characterised by a sudden change in timbre or tonality. The salient features of the method will be summarised here[108].

The onset detection function is constructed using the complex spectral difference in each frequency bin between consecutive time frames of the STFT. Let $S_k[r] = R_k[r]\ e^{j\varphi_k[r]}$ denote the complex value of the k^{th} bin of the discrete STFT (section 2.1.1) in time frame r. written in terms of the bin magnitude $R_k[r]$ and unwrapped phase $\varphi_k[r]$. The instantaneous

frequency of a steady-state component within each frequency bin should remain relatively constant, and therefore, a nearly constant phase difference would result between consecutive frames:

$$\Delta\varphi_k[r] \;=\; \varphi_k[r] - \varphi_k[r-1] \;\simeq\; \varphi_k[r-1] - \varphi_k[r-2]. \tag{3.1}$$

Therefore, we can predict the phase in frame r based upon the phases in the past frames. $r-1$ and $r-2$:

$$\hat{\varphi}_k[r] \;=\; 2\,\varphi_k[r-1] - \varphi_k[r-2]. \tag{3.2}$$

The deviation:

$$d_k[r] \;=\; \text{princarg}\,[\varphi_k[r] - \hat{\varphi}_k[r]] \tag{3.3}$$

measures the difference between the predicted and measured phase in frame r, and its amplitude is likely to be large for frequency bins containing non-stationary or noisy content. The function 'princarg' maps the phase difference to the range $[-\pi,\ \pi]$. $d_k[r]$ was used elsewhere for separating steady-state from non-steady-state content[72]. We now turn from phase information towards the STFT amplitude information. A simple predictor of the STFT amplitude in frame r is the measured amplitude in the previous frame:

$$\hat{R}_k[r] \;=\; R_k[r-1]. \tag{3.4}$$

Combining eqns. 3.2 and 3.4, a prediction of the complex amplitude in the r^{th} frame is:

$$\hat{S}_k[r] \;=\; \hat{R}_k[r]\; e^{i\hat{\varphi}_k[r]}. \tag{3.5}$$

A measure of stationarity within the k^{th} bin is thus obtained by the complex difference:

$$\Gamma_k[r] \;=\; \left|S_k[r] - \hat{S}_k[r]\right| \tag{3.6}$$

which reduces to the amplitude difference $\left|R_k[r] - \hat{R}_k[r]\right|$ when the phase deviation $d_k[r] = 0$. An onset detection function can be obtained simply by summing over all frequency bins $k = 0, \ldots, K$, where K has been chosen as the Nyquist frequency $(N/2)$:

$$\eta[r] \;=\; \sum_{k=0}^{K} \Gamma_k[r]. \tag{3.7}$$

In [107] a similar method was applied within a multi-resolution decomposition framework, where frequency sub-bands were implemented using a constant-Q filter bank. This resulted in a detection function $\eta_b[r]$ for each band b. These can be combined by emphasising the time-resolution capabilities of the higher sub-bands, and the more reliable onset detection behaviour of the lower sub-bands. For simplicity we have stuck to the single resolution approach. An adaptive threshold for onset detection can be obtained by filtering $\eta[r]$ using a weighted median filter of length H:

$$\delta[r] \;=\; \gamma + \lambda \text{ median } \eta[n] \quad ;\; n \;=\; \left\{r - \frac{H}{2} + 1 \ldots, r + \frac{H}{2}\right\} \tag{3.8}$$

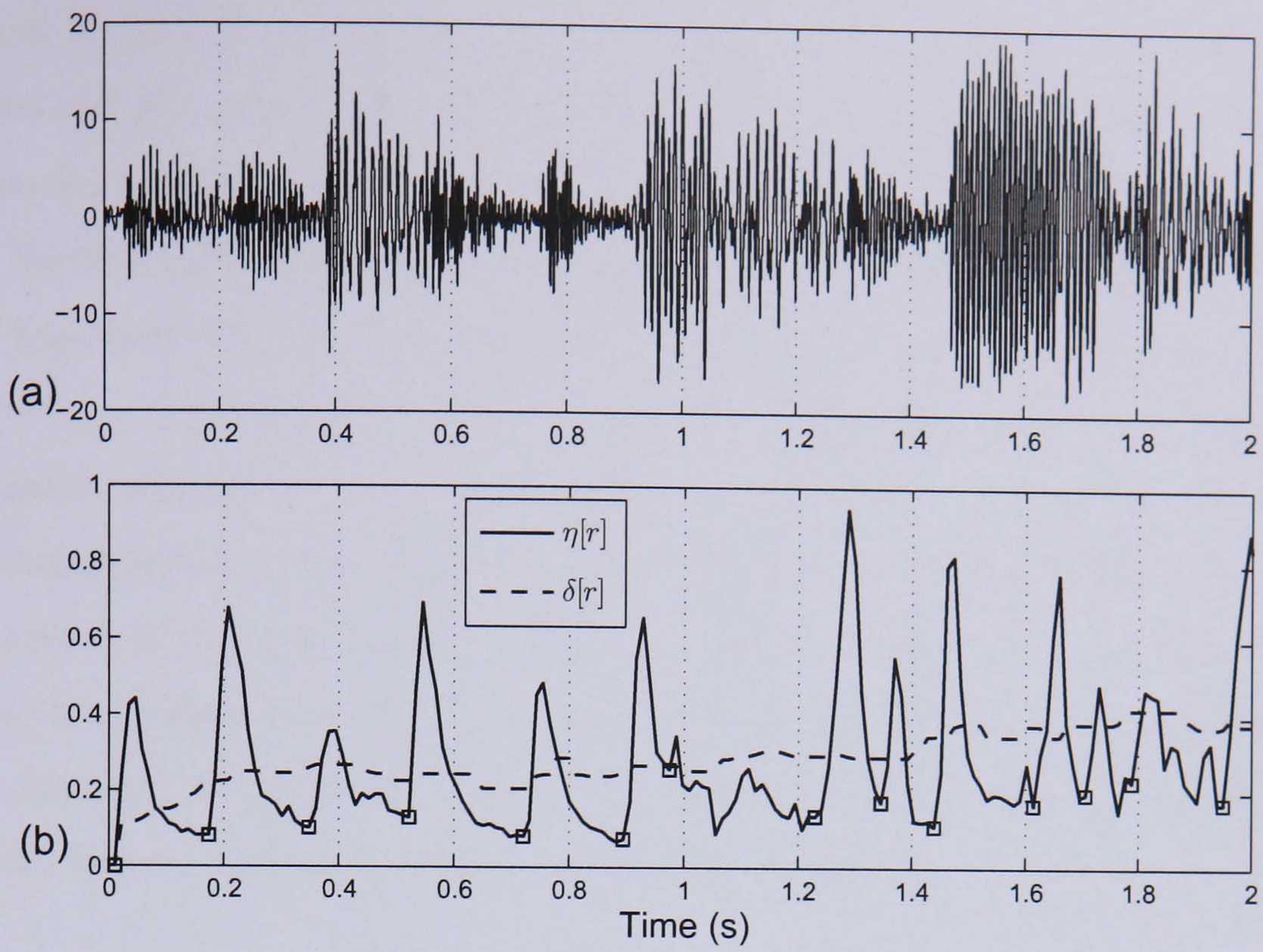

Figure 3.1: **(a)** An audio waveform, **(b)** The corresponding note onset detection function $\eta[r]$ and threshold $\delta[r]$. The squares show the estimated note onset times. ($N = 1024$, $N_{hop} = \frac{N}{2}$)

where the constants $\gamma = 0$ and $\lambda = 1.4$ have been chosen. The median filter length was chosen as half a second ($H = 0.5\ f_s/N_{hop}$, with $N = 1024$ and $N_{hop} = N/2$).

Fig. 3.1 shows the onset detection function $\eta[r]$ and threshold $\delta[r]$ for a sample recording containing both tonal and percussive instruments. The onset times $\{\tau_k;\ k = 1, \ldots, K\}$ where K is the total number of detected onsets, were found by first locating all peaks in the detection function above the threshold, then finding the first minimum within a limited range to the left of each peak, and then moving to the right of this point until a minimum onset gradient was exceeded.

3.2 Note onset alignment

The Needleman-Wunsch algorithm[109] is a numerical method for computing similarity and finding an optimal global alignment between two sequences, and was initially developed for comparing amino acid sequences in two protein molecules. It is a dynamic programming algorithm since it solves a global problem based upon the solutions to subproblems. With a slight modification, it is fairly straightforward to apply the algorithm to aligning the set of MIDI onset times, $\{T_j;\ j = 1, \ldots, J\}$, with the set of note onset times estimated using the note onset detection function above, $\{\tau_k;\ k = 1, \ldots, K\}$. The alignment should be flexible enough to make allowance for potential errors produced by the note onset detector.

Aside from limitations in time-resolution, the note onset detector can on occasion detect false notes and skip real notes. Thus in general J and K are different, and the alignment algorithm should allow for gaps in both time sequences.

The Needleman-Wunsch algorithm constructs a matrix W, from which it is possible to trace backwards from the cell $W_{j,k}$ the optimal alignment of the two partial sequences $\{T_{j'};\ j' = 1, \ldots, j\}$ and $\{\tau_{k'};\ k' = 1, \ldots, k\}$. The mechanism behind this is that the set of all possible alignments of the two sequences is constantly reduced to a smaller set of sub-optimal alignments, from which the best alignment can easily be chosen.

The matrix W is filled using an iterative process starting at the top-left cell $W_{0,0}$. We introduce three parameters that characterise the update procedure: w_T penalises a gap in the first sequence, w_τ penalises a gap in the second sequence, and s_{jk} is a match award between T_j and τ_k. The matrix edges are initialised as follows:

$$\begin{aligned} W_{j,0} &= W_{j-1,k} + w_T \quad ;\ j > 0 \\ W_{0,k} &= W_{j,k-1} + w_\tau \quad ;\ k > 0. \end{aligned}$$

The update procedure that fills the matrix is:

$$W_{j,k} = \max\{W_{j-1,k-1} + s_{jk},\ W_{j-1,k} + w_T,\ W_{j,k-1} + w_\tau\} \quad ;\ j > 0,\ k > 0. \qquad (3.9)$$

As a simple example, table 3.1 shows the matrix W computed with $w_T = w_\tau = -1$ and s_{jk} such that $T_j = \tau_k \rightarrow s_{jk} = 5$ and $T_j \neq \tau_k \rightarrow s_{jk} = 0$. The two sequences to be aligned are $\{1, 2, 3, 6\}$ and $\{1, 2, 2, 3, 5, 6\}$.

Table 3.1: The alignment of two finite sequences using the Needleman-Wunsch algorithm. The sequences are $\{1, 2, 3, 6\}$ and $\{1, 2, 2, 3, 5, 6\}$.

		1	**2**	**2**	**3**	**5**	**6**
	0	-1	-2	-3	-4	-5	-6
1	-1	5	4	3	2	1	0
2	-2	4	10	9	8	7	6
3	-3	3	9	10	14	13	12
6	-4	2	8	9	13	14	18

Once the matrix has been filled using eqn 3.9, the optimal alignment of the two sequences is determined by tracing the path from the global maximum ($W_{4,6}$=18) back to $W_{0,0}$. It can occur that multiple paths exist when there is more than one equally likely path to a cell $W_{j,k}$ from $W_{j-1,k-1}$, $W_{j-1,k}$ or $W_{j,k-1}$ using eqn. 3.9, and in this case it is arbitrary how we back-trace from this cell. The optimal alignment of the two sequences in table 3.1

is given by the shaded cells, and can be represented as:

$$\begin{array}{cccccc} 1 & - & 2 & 3 & - & 6 \\ | & & | & | & & | \\ 1 & 2 & 2 & 3 & 5 & 6 \end{array} . \tag{3.10}$$

An equally valid path would be:

$$\begin{array}{cccccc} 1 & 2 & - & 3 & - & 6 \\ | & | & & | & & | \\ 1 & 2 & 2 & 3 & 5 & 6 \end{array} . \tag{3.11}$$

Normally s_{jk} can have one of only two possible values as in the example above. However, as the task is to align a sequence of MIDI note onsets with estimated note onsets, for which a 'correct match' should allow for some slight deviation in timing, we instead define a continuous measure $s_{jk} = \kappa\, P_{jk}$, where P_{jk} is defined as the probability of a match between a pair of onsets from alternate sequences:

$$P_{jk} \;=\; 1 - \min\left\{1, \frac{|T_j - \tau_k|}{T^{max}}\right\}. \tag{3.12}$$

κ is the maximum match award between two onsets when $P_{jk} = 1 \rightarrow T_j = \tau_k$, and T^{max} measures the maximum delay between T_j and τ_k for the pair to have any non-zero probability of being connected, and was set at 100 ms. It is of minor consequence if there are gaps in the alignment of the MIDI or detected note onset sequences. If a MIDI onset remains unmatched with a detected note onset, its value is simply not modified. Thus, it was decided to penalise gaps only slightly in comparison to the maximum match award ($\kappa = 5$) by choosing $w_T = w_\tau = -1$.

Fig. 3.2 shows the note onset detection function $\eta[r]$, threshold $\delta[r]$, and detected note onsets $\{\tau_k\}$ for a short excerpt from a commercial piano recording. The original MIDI note onset times $\{T_j\}$ are also shown as black triangles, some of which have been adjusted by the alignment so that their new positions are indicated by the white triangles. The table below shows around two seconds of the alignment of the two sequences, with the adjusted MIDI onset times denoted by T'_j.

T_j	–	0.13	0.38	0.78	1.03	–	1.26	1.29	1.51	1.75	1.86	2.00	2.18
		|	|				|		|	|		|	|
τ_k	0.02	0.09	0.39	0.51	0.67	0.93	1.18	1.42	1.60	1.79	–	1.95	2.14
T'_j	–	0.09	0.39	0.78	1.03	–	1.18	1.29	1.60	1.79	1.86	1.95	2.14

3.3 Multi-pitch refinement

The second task in aligning a MIDI representation to an audio recording is to transform the static MIDI pitch values into time-varying pitch envelopes for each note. Normally a note

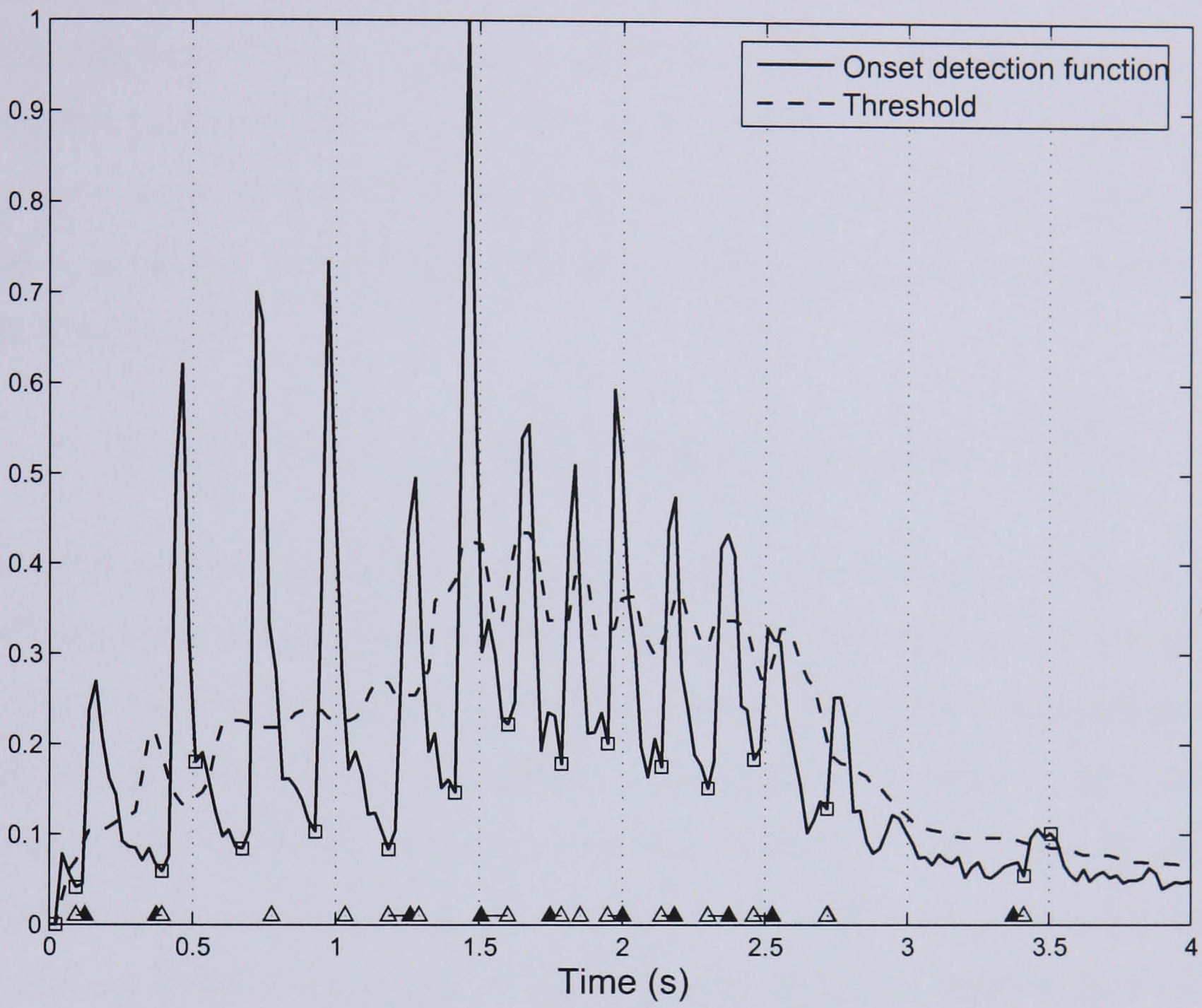

Figure 3.2: The alignment of MIDI data to detected note onsets. The black/white triangles are the original/aligned MIDI onset times respectively, and the squares are the estimated note onset times. A connecting line between a black and a white triangle indicates that the MIDI note onset has been adjusted to its aligned value by at most $T^{max} = 100$ ms. White triangles without any connecting line show that the MIDI note onset is not matched to an estimated note onset and remains at its original value after the alignment procedure.

is labelled in the MIDI format with a single pitch value on an integer scale of semitones ranging from 0 to 127. This is obviously an inadequate representation when the true pitch does not fall exactly on a standard MIDI pitch value. Furthermore, it neglects expressive pitch variation due to vibrato, glissando and other factors. In some instruments there is actually a natural pitch fluctuation occurring immediately after the note/transient onset. However, as the MIDI pitch value provides a rough estimate of the true time-varying pitch envelope, the difficulty of the pitch alignment is much reduced in comparison to multi-pitch estimation without any prior information.

Similar methods were used for pitch refinement and harmonic tracking, the latter of which will be discussed in section 4.3. As the method for pitch refinement is a frame-by-frame process, it will be assumed that all variables apply within a single frame. The pitch f_0^p of note p is estimated by finding a weighted linear least-squares error (LSE) fit to the harmonic frequencies:

$$\arg\min_{f_0^p} \{ E(f_0^p) \} = \arg\min_{f_0^p} \left\{ \frac{1}{|M|} \sum_{m \in M} a_m^p \ |m\ f_0^p - f_m^p|^2 \right\} \tag{3.13}$$

where m is the harmonic number, and f_m^p and a_m^p are the detected frequency and amplitude of the m^{th} harmonic of note p respectively. M is the set of all harmonics of note p that were identified as being non-overlapping, and $|M|$ is the number of elements in this set. This last point ensures that estimates of harmonic frequencies or amplitudes that could possibly be inaccurate due to interference from a harmonic of another instrument do not bias f_0^p. It was decided to use at most the first four harmonics in the pitch refinement, i.e. $m \leq 4$, and the method is robust as long as at least one of the first four harmonics of each note is non-overlapping and detectable.

The pitch refinement process clearly requires knowledge of the harmonic frequencies f_m^p and amplitudes a_m^p. These were found using an iterative spectral peak matching process. Spectral peak picking is described in detail in section 4.2. Let f_v denote the interpolated frequency of the spectral magnitude peak v that is closest to a predicted harmonic frequency $\hat{f}_m^p$. It will be explained later how these predicted frequencies are obtained. f_v is matched uniquely to $\hat{f}_m^p$ if $|f_v - \hat{f}_m^p| < \delta^p \ \hat{f}_m^p$, but we require further that this peak must not be capable of being matched to any harmonics of other notes using the same formula. In other words, we must be able to confidently say that the peak v is the m^{th} harmonic of note p, and then we set $f_m^p = f_v$. The relative frequency range δ^p has been initialised to one semitone: $\delta^p = \delta = 2^{1/12} - 1$. However, the LSE fit in eqn. 3.13 suggests an adaptive value for δ^p that depends on the goodness-of-fit of the ideal harmonic series to the matched harmonic frequencies. In other words, if we are confident that the ideal model is accurate up to harmonic m, then there will be less uncertainty about the predicted frequencies of

higher harmonics, and so δ^p can be reduced. δ^p was set as a measure of the relative spread of the error function $E(f)$ around its minimum f_0^p:

$$\delta^p = \frac{1}{f_0^p}\left[\frac{\sum_n (f[n] - f_0^p)^2 \, E(f[n])^{-1}}{\sum_n E(f[n])^{-1}}\right]^{1/2} \tag{3.14}$$

where $f[n]$ was incremented by small intervals within the range $[(1-\delta)f_0^p, (1+\delta)f_0^p]$. δ^p can be updated each time eqn. 3.13 is used. If no match was made above, we have no choice but to use the predicted frequency for the m^{th} harmonic: $f_m^p = \hat{f}_m^p$. We then attempt to match the next harmonic, and so define the predicted frequency of the $m+1$ harmonic as:

$$\hat{f}_{m+1}^p = h(f_0^p, m+1) = (m+1)\, f_0^p. \tag{3.15}$$

f_0^p above is determined using eqn. 3.13, which has been calculated using all matched harmonics up to m. $h(f_0, m)$ is given as a trivial function of the two input parameters, but will be referred to again later. As the procedure relies on knowing at all times whether the peak v is within range of one or multiple harmonics, the iterative process had to be followed simultaneously for all notes. The sequence of iterations is decided by choosing as the candidate for the next iteration, the note corresponding to the minimum of the set $\{\hat{f}_{m+1}^p \; ; \; p = 1, \ldots, P\}$. To clarify, $m \equiv m(p)$, i.e. it denotes the m^{th} harmonic of note p. To begin with $m(p) = 0 \; \forall \; p$, and $m(p)$ is incremented with each iteration of note p. The iterative process begins with $\hat{f}_1^p = f_i^p$, where f_i^p are either the initial rough MIDI pitches converted to Hz, or the aligned pitches from the preceding time frame.

In summary, the iterative process forms a joint estimate of the pitch and $(m+1)^{th}$ harmonic frequency given all harmonics up to m that have been matched uniquely to spectral peak frequencies. It then looks for a spectral peak uniquely matched to the $(m+1)^{th}$ predicted harmonic frequency. The method in its current form allows slight deviations of the harmonic frequencies from perfect harmonicity using $\delta^p > 0$, which is a fairly common phenomenon observed in acoustic instruments, and is also necessary to overcome the limited frequency resolution of the STFT representation. Furthermore, the running estimates of f_0^p and δ^p ensure that any inaccuracy in the initial pitch estimate is not compounded when multiplying by m to find the m^{th} harmonic frequency.

By modifying the simple linear fit in eqn. 3.13 and the harmonic prediction function $h(f_0, m)$ in eqn. 3.15, the procedure can easily be adapted to account for alternative models of inharmonicity or partial spacings, such as the physical 1-d model of a stiff string[110]:

$$f_m = h(f_0.m) = m \, f_0 \sqrt{1 + m^2 B}. \tag{3.16}$$

B is the inharmonicity coefficient and is determined by the physical dimensions and stiffness of the string. This will be returned to in the harmonic tracking stage (section 4.3) as

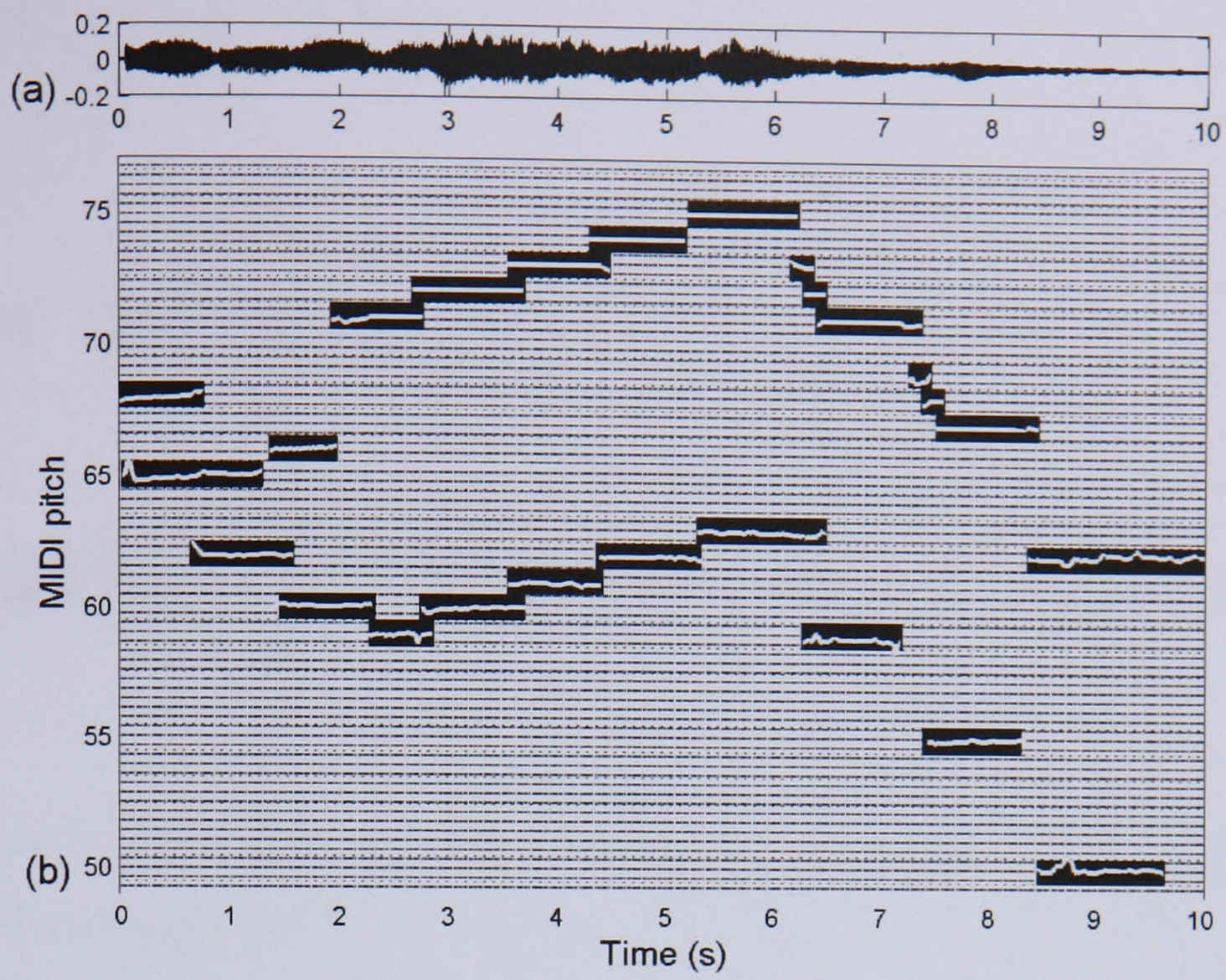

Figure 3.3: Alignment of MIDI data with an audio recording. (a) Audio, (b) original MIDI sequence overlaid with refined pitch envelopes of each note.

inharmonicity is typically not particularly pronounced for the first four harmonics, which are the only ones being used for pitch alignment.

Finally, fig. 3.3 shows a sample MIDI to audio alignment for a recording of two melodies played by saxophone and trumpet. The original MIDI notes in fig. 3.3b, which were improvised by listening to the recording, are overlaid with the aligned pitch envelopes calculated using the above method.

3.4 Conclusions

We have described a pre-processing stage incorporating prior information that extracts note pitch envelopes, onset times and offset times from polyphonic recordings. This information is essential for later work in separating the harmonic and non-harmonic content of these notes from the recording. The prior information is comprised of MIDI data which is improvised by the user (here, the author) whilst concurrently listening to the recording. Due to human timing errors and limitations of the MIDI representation it is necessary to perform a fine alignment of this MIDI data with the audio. A note onset detector and sequence alignment method using a dynamic programming algorithm have been used, which produce an optimal matching between detected note onsets and MIDI note onset times. Following this, a multi-pitch alignment algorithm was described that transforms static MIDI pitch values into time-varying pitch envelopes.

Chapter 4

Separation of Harmonic Content

> "Out of clutter, find Simplicity. From discord, find Harmony. In the middle of difficulty lies Opportunity."
>
> **- Albert Einstein (1879–1955)**

We wish to separate a set of pitched sources from a polyphonic recording, and within our modelling framework, we assume that each source produces a set of notes. So, what is basically required is to separate each pitched note from the mixture, which possibly contains other simultaneously sounding notes. It is clear that a note is a complex event containing both deterministic and stochastic content, stationary and time-varying, and given that every note is different, there is some difficultly in distinguishing what component of the mixture arises from a particular note. However, one structure that is significant in all pitched notes is a set of harmonics, and as these have a regular pattern in the frequency-domain, it is relatively easy to separate the harmonic structures belonging to different notes. A set of harmonics is the most obvious and perhaps the only common characteristic of all pitched notes, and so it makes sense to build our model around this. The more challenging task of separating the remaining non-harmonic content of the note from the recording is discussed in chapter 5.

Each note is characterised by an onset time, offset time and time-varying pitch envelope, and the procedure for determining these was described in chapter 3. The note timing and pitch information provides a convenient starting point to locate the harmonic content of each note within the STFT representation (section 2.1.1). The separation method essentially aligns a harmonic template of each note to spectral peaks in the DFT spectrum, and then constructs a filter for each note that filters content from the mix around the locations of the harmonics in this aligned template. Spectral peaks are detected using peak-picking algorithms (section 4.1), and their amplitudes, frequencies and phases can be improved over the rough estimates obtained by direct sampling of the DFT spectrum (section 4.2). The

template matching procedure is described in section 4.3, which basically performs tracking of the harmonics over time, and is adaptable to non-uniform models of harmonicity. The harmonic content of each note is then separated by constructing comb-like filters with unity-amplitude narrow band-pass filters centred at the positions of the tracked harmonics (section 4.5). These filters share any spectral content in frequency regions where more than one harmonic is overlapping (section 4.6). Other methods for separating overlapping partials are reviewed in section 4.7, and section 4.8 evaluates the comparative performance of filtering and sinusoidal modelling for separating partial content.

4.1 Spectral peak picking

An initial task in the harmonic tracking stage is to detect spectral peaks in each time frame of the discrete STFT that are likely to have been produced by partials. Once this has been performed, the parameters of these peaks, namely frequency, amplitude and phase, can be improved over the rough estimates obtained by directly sampling the DFT spectrum (section 4.2). For the moment we wish to find all prominent spectral peaks containing partials, whilst ideally minimising the number of detected peaks produced by noise or artifacts of the representation.

The peak picking method begins with an initial spectral thresholding stage and is followed by a detection of all maxima above the threshold. The DFT amplitude spectrum in an arbitrary time frame is $A[k] = |F[k]|$. It is closely related to the discrete STFT at a particular time index according to eqn. 2.7, and so the two terms are sometimes used interchangeably. The frequency dependent threshold will be denoted $\eta\ E[k]$, where $E[k]$ is the shape of the threshold and η is a frequency independent threshold height. The reason for a variable threshold is that partial amplitudes in musical spectra typically decay with increasing frequency. A constant threshold could cut off higher frequency partials, or alternatively result in too many spectral peaks being detected at lower frequencies. As upper harmonics are perceptually significant and useful for purposes such as pitch estimation, they should not be neglected.

$E[k]$ was determined as follows. The smoothed amplitude envelope $\tilde{A}[k]$ was calculated by convolving $A[k]$ in the frequency-domain with a normalised Hamming window of length $1 + N/64$ samples, where N is the DFT length. An odd numbered window length was chosen for symmetry reasons, i.e. the calculation of $\tilde{A}[k]$ involves a weighted sum of terms $A[j]$, at an equal number of bins on either side of bin k. Then we define $E[k] = (\tilde{A}[k])^c$ $\forall\ k$ up to the Nyquist limit, where a suitable range for c is $[0.5, 1]$, and $c = 0.7$ was used for results given here. Smaller values of c produce a flatter envelope, which helps to avoid spurious peaks being detected in regions of low spectral amplitude. To illustrate this, fig.

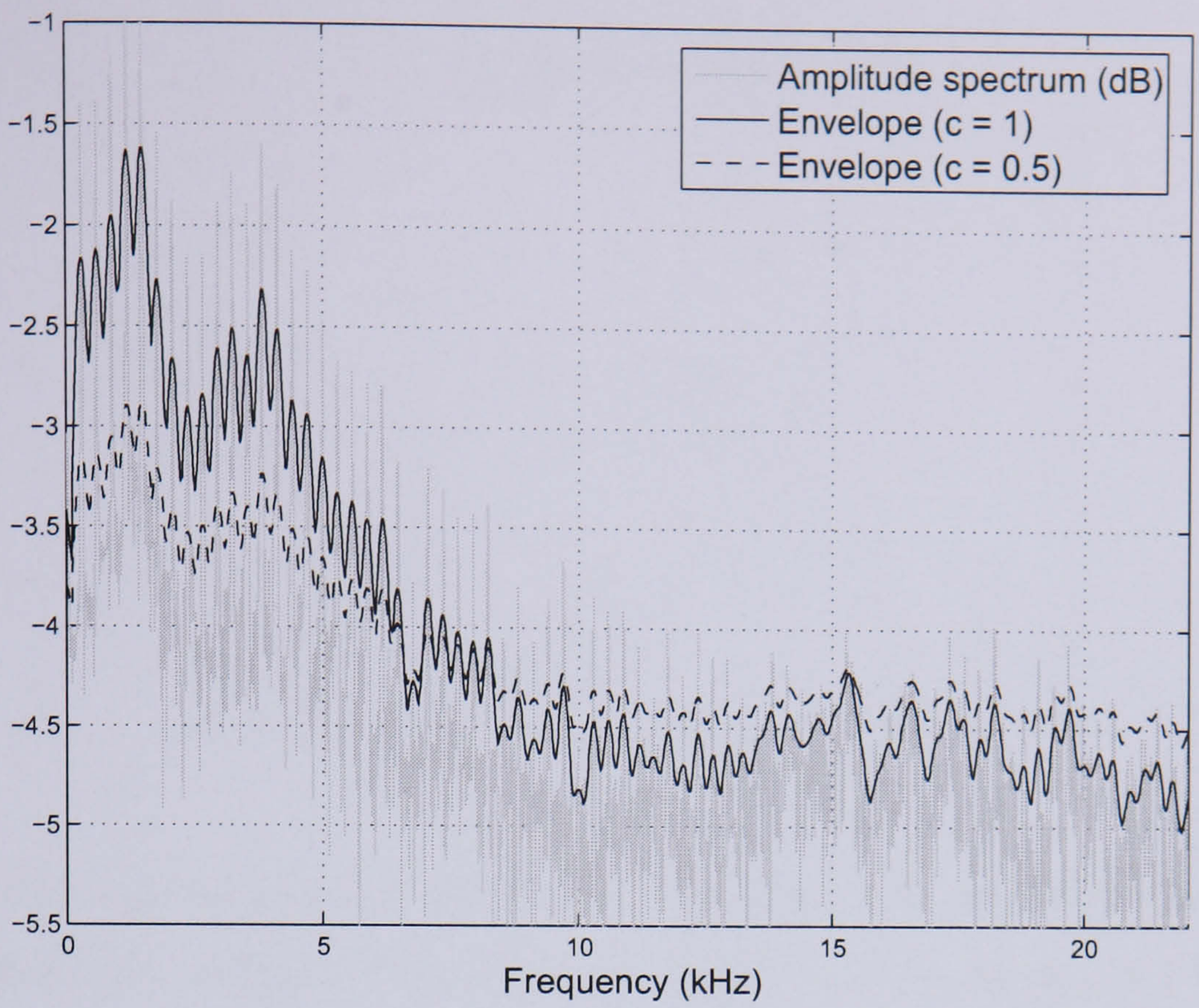

Figure 4.1: The amplitude spectrum of an oboe note ($f_0 = 293$ Hz), showing the estimated spectral envelope using both c = 0.5 and c = 1.

4.1 shows the estimated spectral envelopes for two values of c. To satisfy scaling invariance, i.e. if the amplitude spectrum is multiplied by a constant factor then the envelope should increase by the same factor, the threshold height should obey:

$$\eta \propto \overline{A}^{\,1-c} \tag{4.1}$$

where $\overline{A}$ is the mean spectral amplitude, RMS level, or other similar quantity measuring the average spectral amplitude. A number of alternative methods for spectral envelope estimation are described in detail in [111]. These include linear predictive coding (LPC), cepstrum and discrete cepstrum methods, and improvements on the discrete cepstrum method such as regularisation and smoothing. It was found that the above method was computationally efficient and suitable for our needs, but the interested reader is referred to [111] for a comparative review of other techniques.

The second step is to find all local maxima in $A[k]$ above the threshold. A frequency bin k was detected as a peak maximum if:

$$A[k] > b[|k-j|] \cdot A[j] \qquad \forall\, j \in \{k-d, \ldots, k-1, k+1, \ldots, k+d\} \tag{4.2}$$

where $b[|k-j|]$ is in the range $(0, 1]$, d is the length of vector $\mathbf{b}$, and it was empirically chosen that $\mathbf{b} = (\ b_1 \ \ b_2 \ \ b_3\ ) = (\ 1 \ \ 1 \ \ .5\ )$ when $N = 4096$ or 8192. This effectively implements a variable threshold on bins j around bin k that depends on the distance $|k-j|$, as illustrated in fig. 4.2. It encompasses the simplest case, $\mathbf{b} = (\ b_1\ ) = (\ 1\ )$, meaning that the amplitude in the peak bin must be larger than only its nearest neighbours, but can also be adapted to

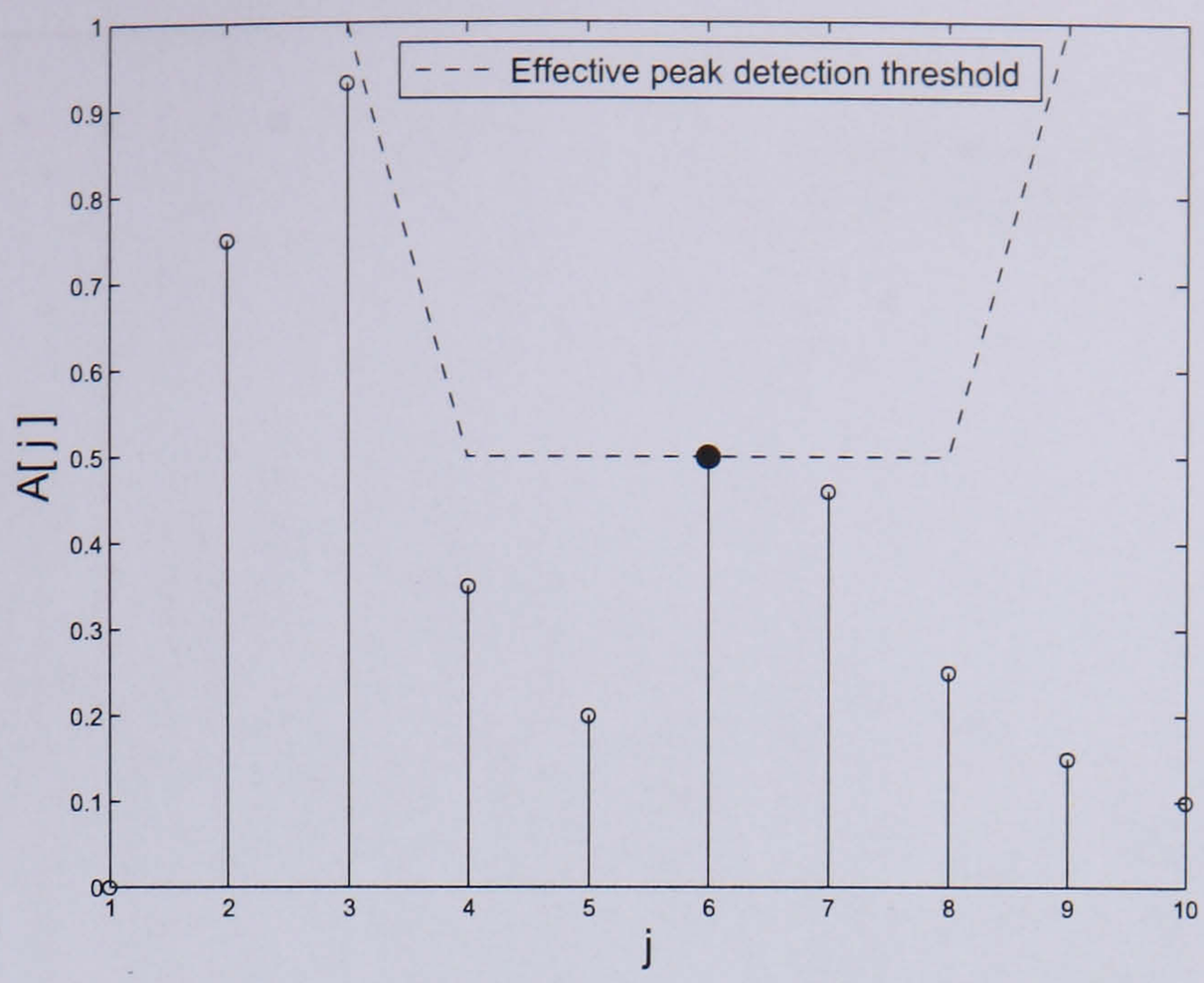

Figure 4.2: The peak picking algorithm in eqn. 4.2 effectively implements a threshold on the amplitudes $A[j]$ as shown above, where $k = 6$ and $\mathbf{b} = (\ b_1 \quad b_2 \quad b_3\ ) = (\ 1 \quad 1 \quad .5\ )$

a more noisy spectrum by using a longer vector for $\mathbf{b}$ that attenuates as $|k - j|$ increases. Finally, fig. 4.3 illustrates an amplitude spectrum and threshold for a mix of two violin notes with pitches of 880 Hz and 1319 Hz, and the detected spectral peaks using eqn. 4.2. Eqn. 4.2 is computationally inexpensive, easy to implement, and its behaviour is adequate for our purposes. However, a systematic comparison has not been made with more complex peak-picking methods such as [52], which describes how sinusoids can be detected by finding regions of similarity of the DFT spectrum with the spectral shape of the window function, and [59], which gives a sinusoidality measure based upon a linear sinusoidal frequency variation model. Whilst sinusoidality measures are useful to distinguish real partials from spurious peaks, we would not want to reject abnormally shaped peaks that could have been produced by non-stationary partials or multiple partials overlapping within the same frequency region, hence an unbiased peak-picking algorithm has been used. Partials up to the Nyquist frequency were easily detected using eqn. 4.2.

4.2 DFT peak estimators

In the previous section, rough estimates of peak frequencies and amplitudes were obtained by direct sampling of the DFT spectrum of each windowed signal segment. Let k_v denote the frequency bin in which the v^{th} peak occurs in the DFT amplitude spectrum. The rough peak amplitude and phase are obtained by simply sampling the DFT at this frequency bin, giving $A[k_v] = |F[k_v]|$ and $\varphi[k_v] = \angle\ F[k_v]$ respectively. As the act of multiplying the signal by the window function produces a convolution of their Fourier transforms, these rough parameter estimates are partly determined by the shape of the Fourier transform

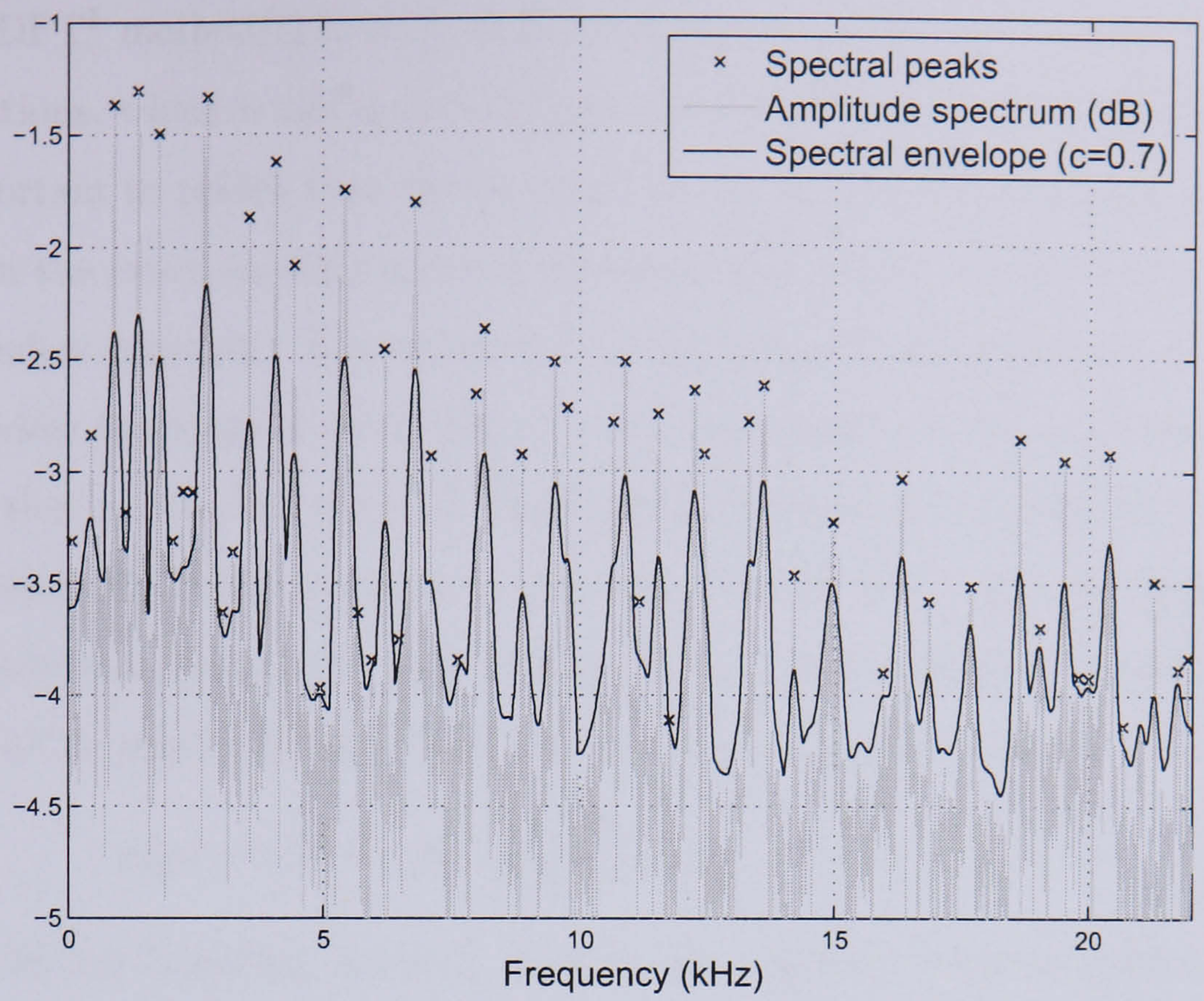

Figure 4.3: Thresholding and peak picking the amplitude spectrum $|F[k]|$ using a frequency dependent threshold $\eta\ E[k]$.

of the window function centred at the true partial frequency, which in general will not be situated exactly at the DFT maximum bin k_v. The estimates are also partly determined by any other overlapping content in this frequency region, and also the time-varying properties of the partial which modify its peak shape, but these effects are more difficult to determine. Thus, we continue by describing some methods for improving the peak parameter estimates based solely upon the shape of the window function.

The minimum resolution of the DFT is f_s/N, which for a 44.1 kHz sampling rate and a DFT length of 2048 samples (46 ms) gives a frequency spacing between bins of 21.5 Hz, or a maximum difference between a true sinusoidal frequency and the nearest frequency bin of $\simeq 10$ Hz. This represents a 0.1% relative frequency resolution for a 10 kHz sinusoid, which is probably more than accurate enough, but a 10% relative frequency resolution for a 100 Hz sinusoid, which is rather large. The question is whether the resolution can be improved without using longer window lengths which could discredit assumptions of signal stationarity. Zero-padding is one solution, and this involves lengthening the windowed signal with a string of zeros before computing the DFT of the padded segment. It has the effect of interpolating the DFT at additional frequency bins between those of the un-padded DFT, but is computationally expensive as a much larger DFT must be computed in each frame. An alternative is to estimate the parameters of the spectral peak by an interpolation using the adjacent bins to the peak maximum: $k_v - 1,\ k_v,\ k_v + 1$. Other solutions are to derive information from both the DFT of the signal and higher order signal derivatives,

such as the DFT[1] method[112. 113]. or to compute the DFT of the signal using multiple window functions, which is the approach taken in time-frequency reassignment[114, 115].

It is important to realise that the accuracy of the various spectral peak estimators are dependent on the exact analysis window function used. Thus, we begin by discussing the choice of window function. A quantitative comparison of the properties of a number of common window functions is provided in [116]. A common compromise when choosing a window function is that the effective bandwidth of the window, given by the equivalent noise bandwidth (ENBW), increases as the the ratio of the main lobe amplitude to the spectral sidelobe amplitudes increases. Whilst there are other windows with better spectral sidelobe roll-off or smaller ENBW[116], the (periodic) Hamming window:

$$w[n] = 0.54 - 0.46\ \cos\left(\frac{2\pi}{N}\ n\right) \quad ;\ n = 0, \ldots, N-1 \tag{4.3}$$

was chosen for the following reasons. Firstly, the highest side-lobe level at -43 dB is small enough given that musical signals invariably contain a noise component which would probably mask any sidelobes of this height or smaller anyway. Secondly, it has a moderately narrow ENBW of 1.36 bins which is desirable in minimising the spread of spectral lines, and thirdly, the synthesis window defined in eqn. 2.14 (a triangular window divided by the Hamming window) and illustrated in fig. 2.3, is attenuated at its endpoints which removes frame edge discontinuities during re-synthesis after transformations have been made to the STFT data. The Hamming window function is not necessarily the best window when it comes to DFT peak estimation. The DFT^1 method and Grandke's method work slightly better for Hanning windowed data, and Quinn's methods are designed for un-windowed (rectangular windowed) data. However, this shortcoming should be weighed against the other factors mentioned above.

A summary of spectral peak interpolation methods is given in [117] with a more detailed analysis provided in [112]. These include the parabolic (quadratic) method, barycentric method, Quinn's first and second estimator, Grandke's method and Jain's method. The DFT^1 method implemented in the software package InSpect[118] is described in [112, 113]. To find the best peak estimator for our choice of window function, the performance of these estimators was evaluated for Hamming windowed data. Their relative performance would, of course, be different using other window functions. Figs. 4.4 and 4.5 illustrate the results for a windowed unity-amplitude sinusoid in -10 and 10 dB white noise respectively. Only Grandke's method, the quadratic method and the DFT[1] method are displayed as the other estimators were found to perform substantially worse. Figs. 4.4a and 4.5a show the error in the estimated sinusoidal frequency in bins as a function of the sinusoidal frequency. This would be a maximum of 0.5 bins when the sinusoidal frequency is mid-way between bins if we were simply to use k_p as the sinusoidal frequency estimate. Figs. 4.4b and 4.5b show the

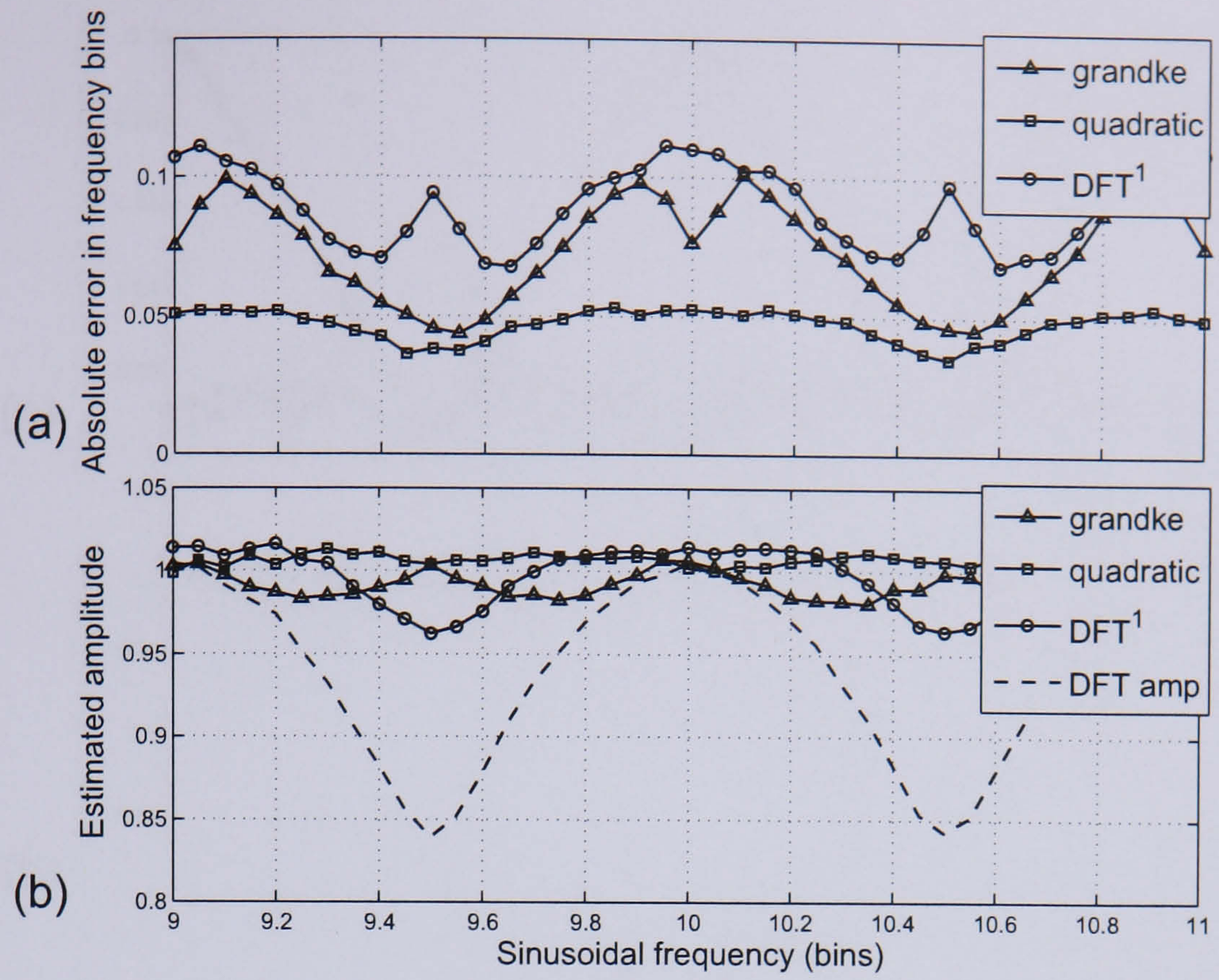

Figure 4.4: DFT peak estimation for a Hamming windowed sinusoid in noise using Grandke's method, the quadratic method and the DFT^1 method ($N = 4096$, $SNR = -10$ dB) as a function of the sinusoidal frequency. **(a)** The absolute error in the estimated sinusoidal frequency in bins, and **(b)** the estimated sinusoidal amplitude a_v.

estimated sinusoidal amplitude as a function of frequency, which was estimated using[113]:

$$a_v = \frac{A[k_v]}{W(|f_v - f(k_v)|)} \tag{4.4}$$

where $W(f)$ is the amplitude of the Fourier transform of the window function, $f(k_v)$ is the equivalent frequency of the peak frequency bin, and f_v is the improved estimate of the peak frequency. The measured DFT amplitude $A[k_v]$ is also shown in figs. 4.4b and 4.5b, and again we see that this differs most from the true sinusoidal amplitude when the sinusoidal frequency is mid-way between bins. One can conclude from these results that the quadratic estimator[65, 119] using the Brent method:

$$d = \frac{1}{2} \frac{\log_{10} A[k_v - 1] - \log_{10} A[k_v + 1]}{\log_{10} A[k_v - 1] - 2 \log_{10} A[k_v] + \log_{10} A[k_v + 1]} \tag{4.5}$$

where $f_v = (k_v + d)\frac{f_s}{N}$ is the improved frequency estimate, performs overall the best out of the evaluated estimators. However, music typically contains time-varying sinusoids, as opposed to stationary ones as assumed in the above interpolation methods. As the DFT^1 method was found to produce slightly better partial tracking results measured in terms of the overall evaluation of separation performance discussed in section 4.8.3, this method was used in further processing.

In some circumstances, such as when performing sinusoidal modelling, in addition to the frequency and amplitude, the phase of a partial is also required. If zero-phase windowing

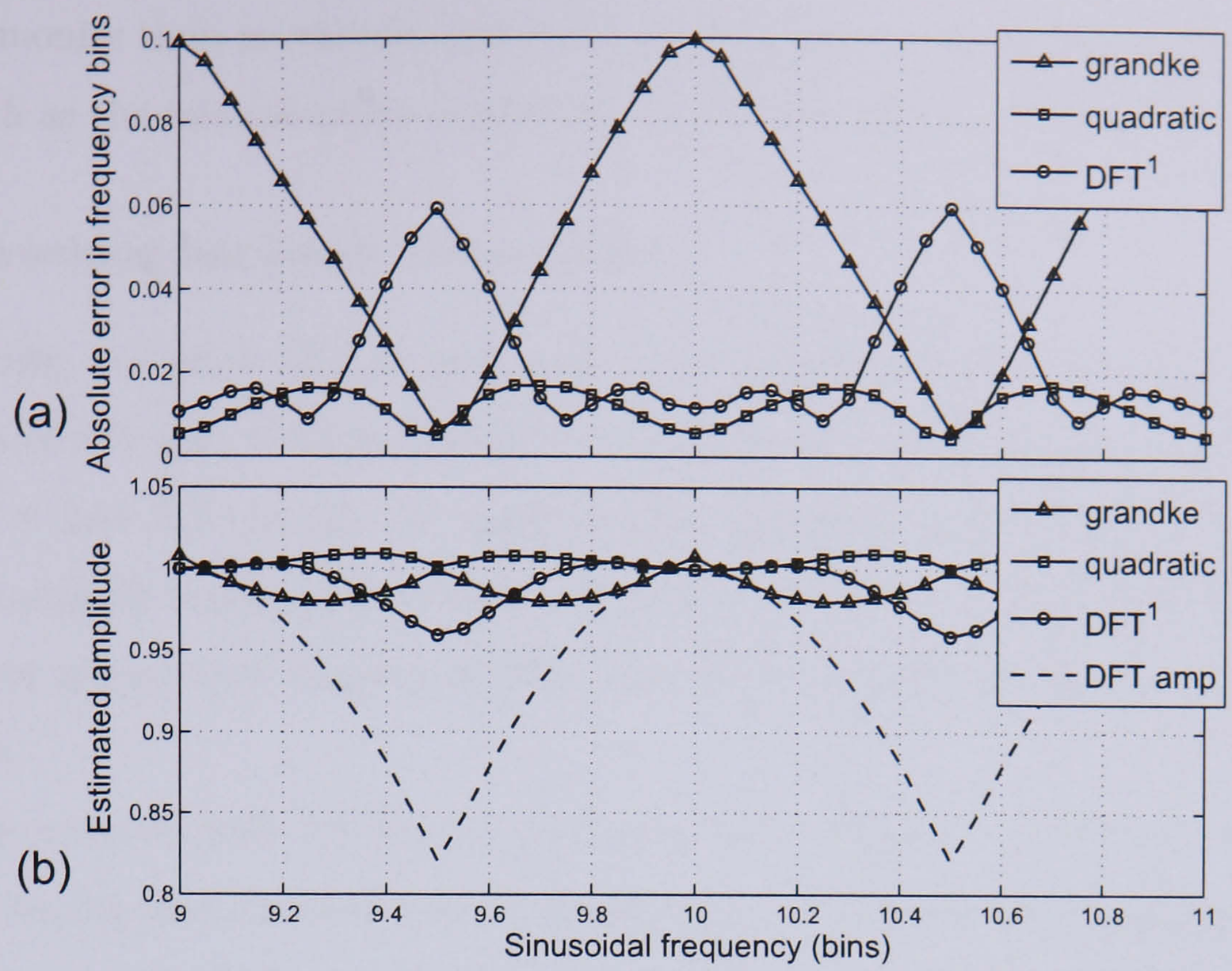

Figure 4.5: DFT peak estimation for a Hamming windowed sinusoid in noise using Grandke's method, the quadratic method and the DFT^1 method ($N = 4096$, $SNR = 10$ dB) as a function of the sinusoidal frequency. **(a)** The absolute error in the estimated sinusoidal frequency in bins, and **(b)** the estimated sinusoidal amplitude a_v.

is used[112], i.e. the windowed signal is rotated so that the centre of the analysis window now occurs at the first sample rather than the middle sample, then the linear phase trend induced by the analysis window is removed, and the phase across the main lobe is constant for a stationary sinusoid. A simple estimator of the spectral peak phase is then the phase of the maximum frequency bin:

$$\varphi_v \;=\; \angle\, F[k_v] \;=\; \arctan\left(\frac{\Im\{F[k_v]\}}{\Re\{F[k_v]\}}\right). \tag{4.6}$$

4.3 Tracking harmonics

We continue now with the task of tracking harmonics in the discrete STFT given the set of estimated peak frequencies $\{f_v\}$ and amplitudes $\{a_v\}$ in each time frame, where v is the peak index. The harmonic tracking problem differs slightly from most partial tracking problems, e.g. [30, 54, 55], in that the pitches of each note are already known from section 3.3, and hence it is required to track harmonics only near the predicted harmonic frequencies rather than at any location in the STFT representation. Since in music it is common for notes the are harmonically related to be played together, there is a frequent occurrence of overlapping harmonics from multiple instruments, which should be taken account of in the tracking process. The harmonic tracking algorithm is similar in structure to the pitch refinement algorithm described in section 3.3, although we are now interested in tracking

higher harmonics than previously. and it is also shown how parameters of the harmonicity model, such as the inharmonicity coefficient. can be estimated over multiple time-frames.

4.3.1 Tracking harmonic frequencies

To begin with, the pitch $f_0^p[r]$ of each note p over all time frames r between the start and end frames of the note is known from the alignment of the MIDI data to audio described in chapter 3. Let $\hat{f}_m^p[r]$ be the m^{th} predicted harmonic frequency of note p in frame r. A harmonic template matching procedure will now be described that attempts to match and align the set of predicted harmonics $\hat{f}_m^p[r]$ with the set of detected spectral peaks in each time frame.

$\hat{f}_m^p[r]$ is matched with the largest spectral peak v in frame r such that $|\hat{f}_m^p[r] - f_v| < \delta\ f_0^p[r]$, providing that the peak cannot be matched with a predicted harmonic of any other note $\hat{f}_n^q[r]$ using a similarly derived matching range. δ is a constant in the range $[0.01, 0.1]$, and it determines the maximum frequency difference between a harmonic and a peak for a matching to be possible. If this peak cannot be matched uniquely to $\hat{f}_m^p[r]$ a search is made for the second largest peak u within matching range of $\hat{f}_m^p[r]$. If such a peak exists, and if $\hat{f}_m^p[r] < \hat{f}_n^q[r]$, then $\hat{f}_m^p[r]$ is matched to the peak with the lower frequency, otherwise it is matched to the higher peak. If peak u does not exist, $\hat{f}_m^p[r]$ is matched with f_v only if $|\hat{f}_m^p[r] - f_v| < 0.5\ |\hat{f}_n^q[r] - f_v|$ and $|\hat{f}_m^p[r] - \hat{f}_n^q[r]| > \rho$ where ρ is the half bandwidth of the main lobe of the window function, resulting in $\rho = 2$ bins for the Hamming window. Finally, if the harmonic is matched to a peak v for argument's sake, then $f_m^p[r] = f_v$ is the corrected harmonic frequency, otherwise $f_m^p[r] = \hat{f}_m^p[r]$ is the uncorrected harmonic frequency.

The predictions of the harmonic frequencies are made using the function $h(f_0, m)$:

$$\hat{f}_m^p[r] \;=\; h(f_0^p[r], m) \;=\; m\ f_0^p[r]. \tag{4.7}$$

For the case of the piano, nonlinear effects due to string stiffness are more substantial than in most stringed Western instruments, and harmonics occur at noticeably stretched frequencies[110]. Thus, a different harmonic prediction function is used for piano notes:

$$\hat{f}_m^p[r] \;=\; h(f_0^p[r], m, B) \;=\; m\ f_0^p[r]\ \sqrt{1 + m^2 B}. \tag{4.8}$$

The equation for partial stretching arises from a physical consideration of the piano string stiffness in the equation of motion for transverse waves in a vibrating bar. B is the inharmonicity coefficient. and for a value of $B = 0.001$ in the middle register, the 13^{th} partial would be shifted to about the frequency of the 14^{th} partial had the note been purely harmonic ($B = 0$). Values of the inharmonicity coefficient vary from piano to piano, but

are typically between approximately 0.0001 for bass tones and 0.015 for treble tones[120]. Eqn. 4.7 is clearly a special case of eqn. 4.8 when $B = 0$. A system for polyphonic chord transcription incorporating inharmonicity was described in [121].

Each time $\hat{f}_m^p[r]$ is matched uniquely to a spectral peak v, then the corrected harmonic frequency $f_m^p[r]$ is used to refine the note's pitch by minimising the weighted LSE error fit to the harmonic frequencies:

$$\arg\min_{f_0^p[r]} \{ E(f_0^p[r]) \} = \arg\min_{f_0^p[r]} \left\{ \sum_{m \in M_r^p} a_m^p[r] \ |m f_0^p[r] - f_m^p[r]|^2 \right\} \tag{4.9}$$

where $a_m^p[r] = a_v$ is the detected amplitude of the m^{th} harmonic of note p matched to peak v, and M_r^p is the set of all currently matched harmonics of note p in frame r. It was also found beneficial to enforce some time continuity on the trajectory $f_0^p[r]$, by incorporating constraints relating to the maximum allowed pitch deviation between consecutive frames, and the maximum allowed pitch deviation from the mean pitch value. This helps to correct wrong pitch estimates in sections where there is, for example, a large interference or masking component. Ultimately though, it would perhaps be better to build a probabilistic model initialised with the estimates of $f_0^p[r]$, that is able to estimate the most likely continuous pitch trajectory from the beginning to the end of the note, and be robust to clearly incorrect pitch estimates in isolated frames. Given that the user-improvised MIDI data provides a rough estimate of the pitch in the current system, constraints can be imposed on the range of the estimated pitch in each time frame, and so the problem of estimating the pitch trajectory becomes far easier than had this prior information not been available.

A similar estimator can be derived for alternative models of harmonicity, for example, to estimate the pitch envelope and inharmonicity coefficient in the particular case of the piano. The inharmonicity coefficient is assumed here to be constant throughout the duration of a note, and so it makes sense to perform a multi-frame estimation of each inharmonicity coefficient B^p. As B^p and $f_0^p[r]$ are interdependent according to eqn. 4.8, ideally we would like to perform a joint multi-frame minimisation over B^p and $f_0^p[r]$ of the weighted LSE between the theoretical and corrected partial frequencies. However, the number of parameters in this optimisation makes this option computationally expensive. Instead, starting with the rough initial estimate of $f_0^p[r]$, B^p is globally estimated using:

$$\arg\min_{B^p} \{ E(B^p) \} = \arg\min_{B^p} \left\{ \sum_r \sum_{m \in M_r^p} a_m^p[r] \ |h(f_0^p[r], m, B^p) - f_m^p[r]|^2 \right\} \tag{4.10}$$

where $h(f_0, m, B)$ is defined in eqn. 4.8 and the sum over r extends from the first to the last frame of the note. It was mentioned in [122] that lower partials are more affected by the impedance of the soundboard which tends to alter their frequencies slightly, and hence it was advised to avoid using isolated lower partials for inharmonicity calculations. As

the estimation of B in eqn. 4.10 measures the model fit to all harmonics. this problem is mostly avoided. The pitch trajectory $f_0^p[r]$ is then refined on a frame-by-frame basis using the multi-frame estimate of B^p:

$$\arg\min_{f_0^p[r]} \{ E(f_0^p[r]) \} = \arg\min_{f_0^p[r]} \left\{ \sum_{m \in M_r^p} a_m^p[r] \ |h(f_0^p[r], m, B^p) - f_m^p[r]|^2 \right\}. \quad (4.11)$$

The above estimators (eqn. 4.9, and eqns. 4.10 and 4.11) for the two alternative models of harmonicity are performed repeatedly each time a harmonic of note p is matched with a spectral peak. In other words, the most up-to-date information is used to predict the location of the next harmonic, and the larger the number of harmonics that match the predictions, the more robust the model fit is to errors.

To match a harmonic to a spectral peak, it must be known whether this peak is within range of one or multiple predicted harmonics. This means that the harmonic template matching process must be performed concurrently for all notes, so that at any time, the predictions of the nearest harmonic frequencies of all notes to this peak will be up to date. The order of iteration is decided by choosing as the candidate note for the next iteration, the note corresponding to the minimum of the set $\{\overline{\hat{f}^p_{m(p)+1}} \ ; \ p = 1, \ldots, P\}$, where $\overline{\hat{f}^p_{m(p)+1}}$ is the average of $\hat{f}^p_{m(p)+1}[r]$ between its start and end frames. To clarify, $m(p)$ denotes the m^{th} harmonic of note p. To begin with $m(p) = 0 \ \forall \ p$, and $m(p)$ is incremented with each iteration of note p. Finally, some harmonic tracking results are shown in fig. 4.6 for a sum of two cello notes, in fig. 4.7 for a soprano singing with vibrato, and in fig. 4.8 for a piano note, for which the effects of inharmonicity are visible.

The multi-frame estimation of inharmonicity in piano notes was introduced here for achieving more robust and accurate tracking of piano partials. Although only this one specific instrument model has been tested, it demonstrates that the inclusion of source or instrument specific information can lead to better separation results. This must be balanced against the loss of generality resulting from the requirement that this prior information be known. The particular model of piano inharmonicity is computationally more expensive than the purely harmonic model and could result in a slight decrease in robustness. However, it does not result in any loss of generality, as the LSE minimisation in eqn. 4.10 would probably result in values of $B \simeq 0$ for most instruments, i.e. the same harmonic frequency predictions would result if using the purely harmonic model.

4.3.2 Interpolating harmonic amplitudes and phases

The template matching procedure in the previous section requires that when the m^{th} harmonic of note p is unambiguously matched to a spectral peak v. the predicted harmonic frequency is corrected: $f_m^p = f_v$. If the harmonic remains unmatched. eqns. 4.7 or 4.8 give

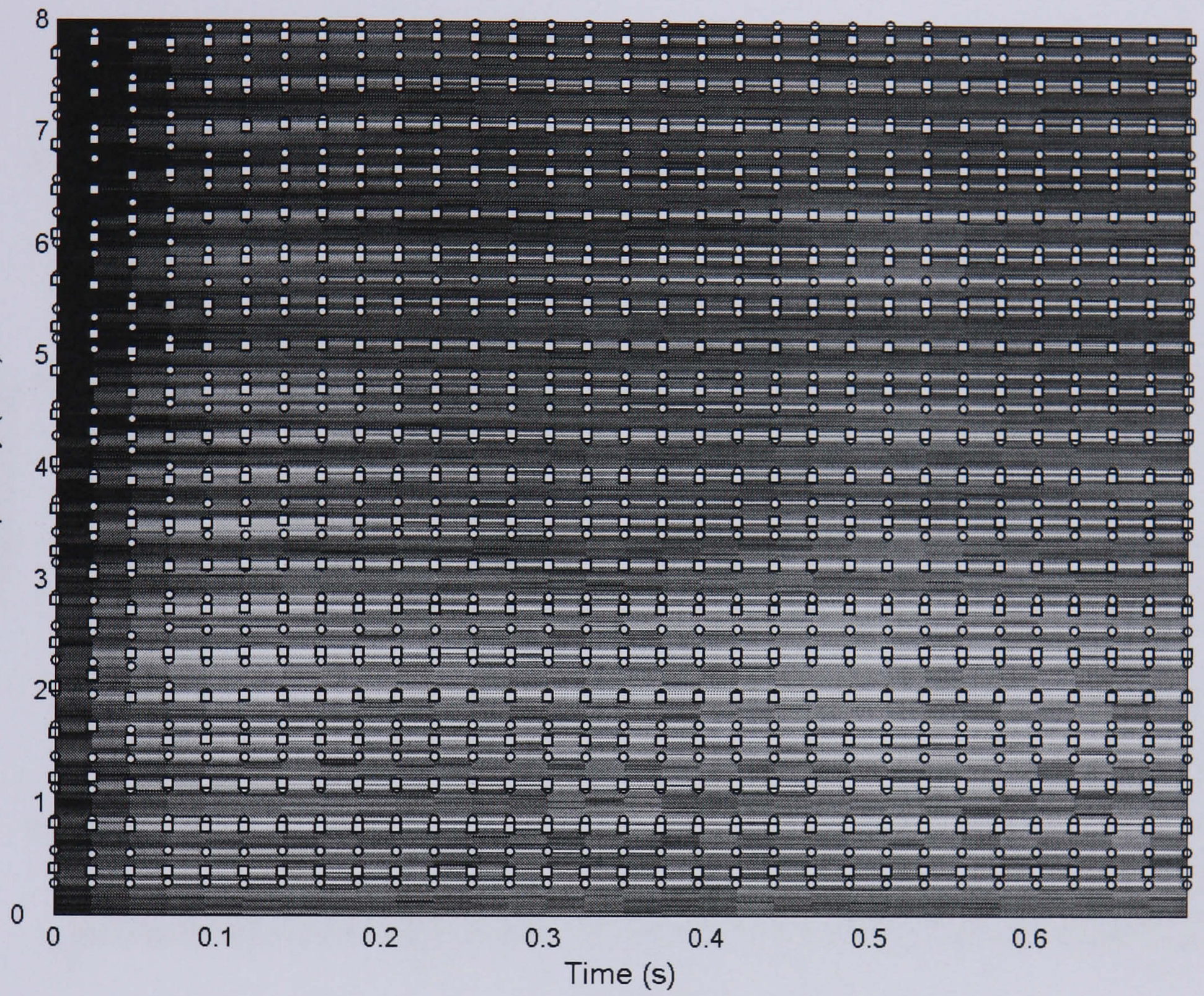

Figure 4.6: Spectrogram of two cello notes (D4 = 294 Hz and G4 = 392 Hz) with the estimated harmonic trajectories in each frame shown as circles/squares. Around 20-30 harmonics of both notes were tracked reliably, despite a large number of overlapping harmonics due to the notes being a fourth apart.

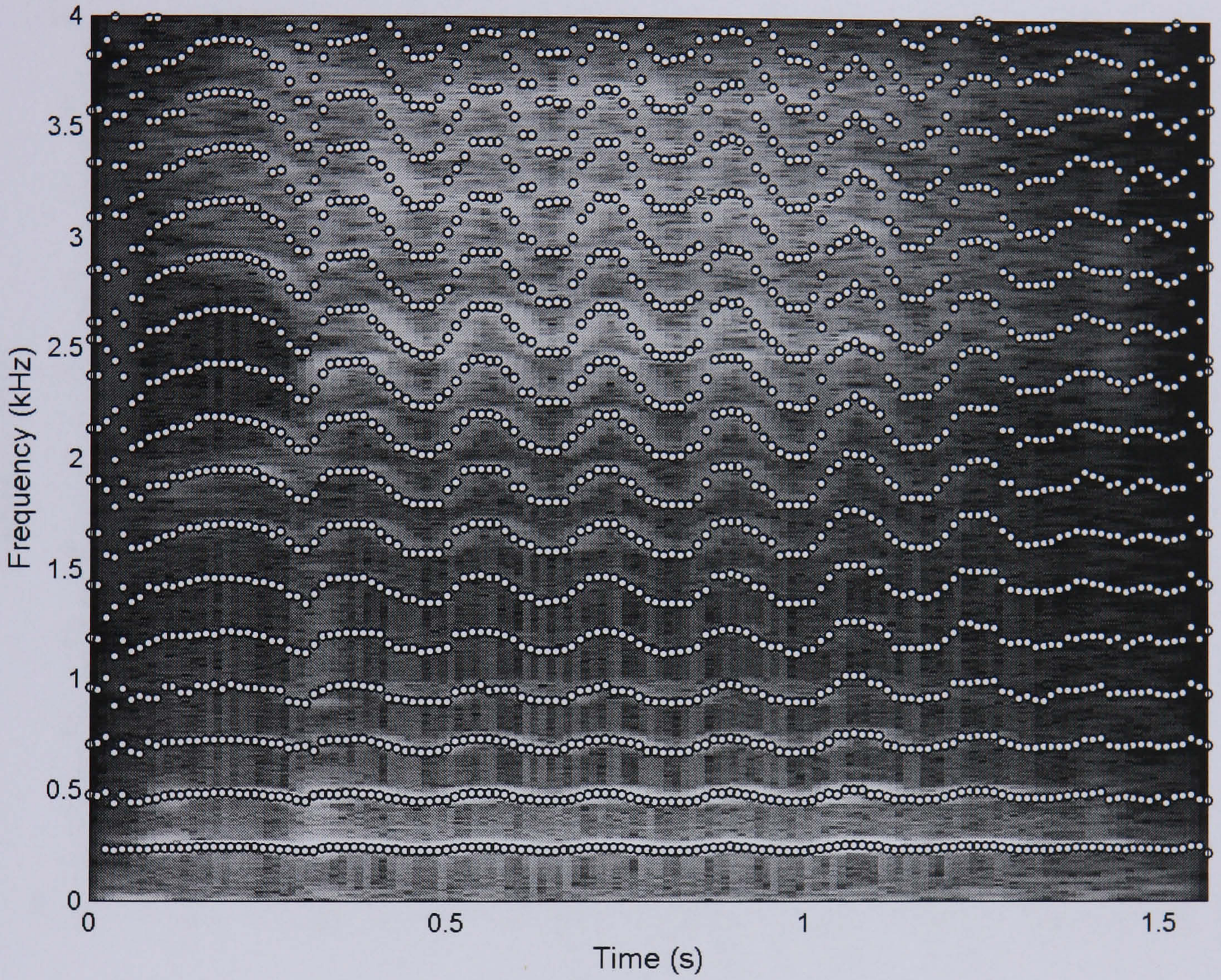

Figure 4.7: Spectrogram of a soprano singing with vibrato (mean pitch = 237 Hz) with the estimated harmonic trajectories in each frame shown as circles. The harmonic tracking algorithm shows a robustness to the time-varying nature of the vibrato.

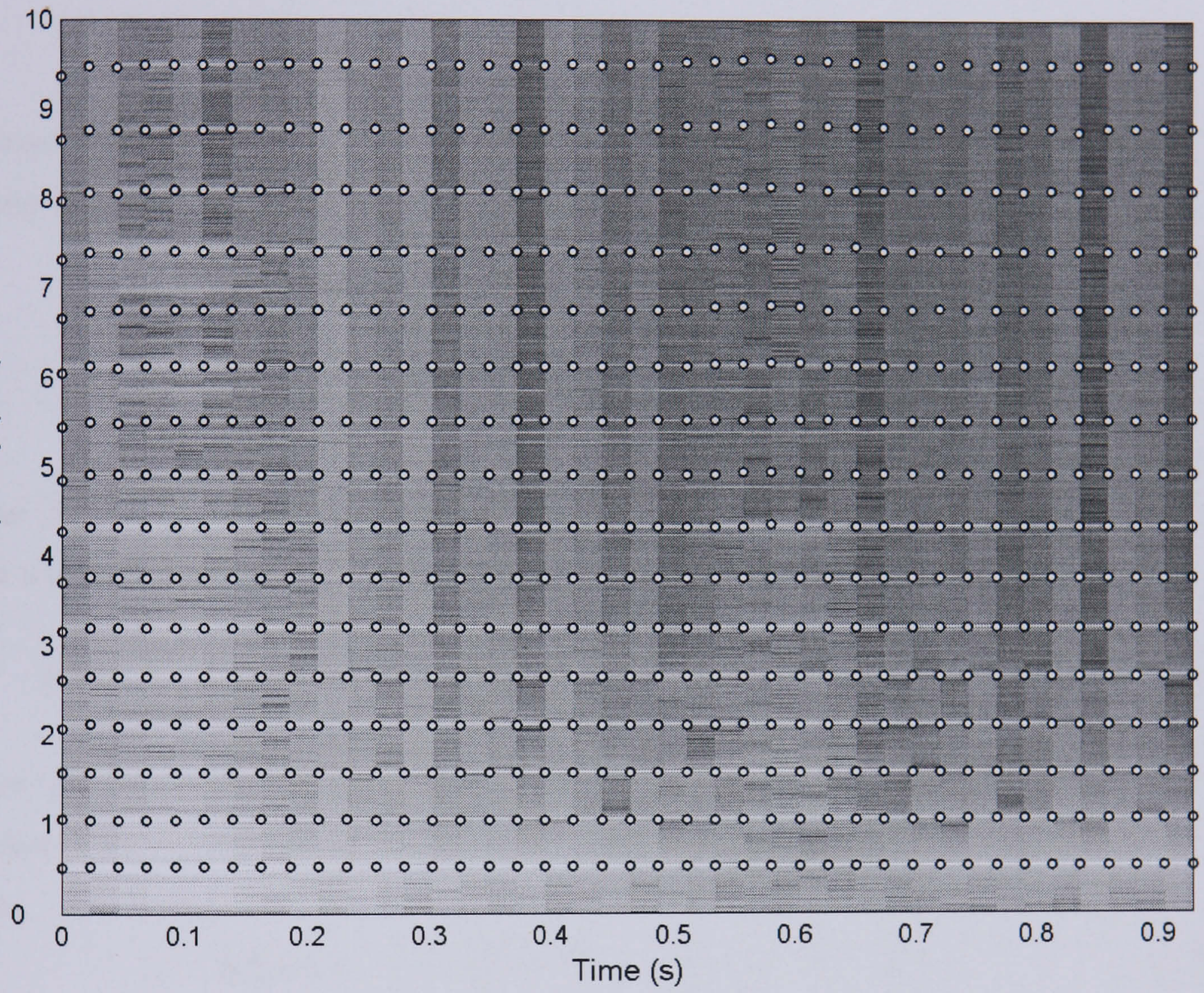

Figure 4.8: Spectrogram of a piano note (C5 = 523 Hz) with estimated harmonics in each frame shown as circles. Notice that the piano harmonics are stretched apart at higher frequencies.

its uncorrected harmonic frequency. Similarly, the harmonic amplitude and phase can be corrected when matched to a spectral peak using the peak parameters determined during peak-picking: $a_m^p = a_v$ and $\varphi_m^p = \varphi_v$. Therefore, what remains is to determine a_m^p and φ_m^p when the harmonic is not matched to a peak, which is likely to occur if this harmonic is overlapping with one or more harmonics from other instruments.

One option is to determine a_m^p by linearly interpolating between the nearest neighbouring harmonics $m_l < m$ and $m_r > m$ that are non-overlapping, and therefore whose amplitudes are known, i.e.:

$$a_m^p[r] \;=\; \frac{(m_r - m)\; a_{m_l}^p[r] + (m - m_l)\; a_{m_r}^p[r]}{m_r - m_l}. \tag{4.12}$$

Alternatively, a time-domain interpolation of $a_m^p[r]$ and $\varphi_m^p[r]$ can be performed between the nearest neighbouring frames $r_l < r$ and $r_r > r$ at which the harmonic is non-overlapping:

$$a_m^p[r] \;=\; \frac{(r_r - r)\; a_m^p[r_l] + (r - r_l)\; a_m^p[r_r]}{r_r - r_l} \tag{4.13}$$

$$\varphi_m^p[r] \;=\; \varphi_m^p[r_l] \;+\; 2\pi\Delta T \sum_{r'=r_l}^{r-1} (f_m^p[r'] + \Delta f_m^p[r']) \tag{4.14}$$

where $\Delta T = \frac{N_{hop}}{f_s}$ is the frame hop size in seconds, and $\Delta f_m^p[r']$ is a frequency deviation term which is chosen to satisfy phase matching at the boundary frames, i.e.:

$$2\pi\Delta T \sum_{r'=r_l}^{r_r-1} \Delta f_m^p[r'] \;=\; \text{princarg}\,\{\; \varphi_m^p[r_r] - \varphi_m^p[r_l] - 2\pi\Delta T \sum_{r'=r_l}^{r_r-1} f_m^p[r'] \;\} \tag{4.15}$$

where 'princarg' maps the phase difference to a range $[-\pi,\ \pi]$. $\Delta f_m^p[r]$ is chosen to be a quadratic function that tends to zero at $r_l - 1$ and r_r, and is a maximum mid-way between them:

$$\Delta f_m^p[r] \;\propto\; \left(\frac{r_r - r_l + 1}{2}\right)^2 \;-\; \left(r - \frac{r_r + r_l - 1}{2}\right)^2. \tag{4.16}$$

Consequently, the interpolated phase in eqn. 4.14 is a cubic or higher order function that satisfies boundary conditions, similarly to the phase interpolation function used in the MQ sinusoidal model[30]. The time-domain amplitude and phase interpolation was generally deemed to be more accurate than the spectral-domain amplitude interpolation, at least when the duration of the overlap is small. The latter was used only when it was unreliable or inpossible to apply the former. That is, when the time difference between the closest preceding and following frames to r where the harmonic was non-overlapping, exceeded 200 ms. Fig. 4.9 shows the interpolated harmonic trajectories for a mix of two violin notes using both the spectral and temporal amplitude and frequency interpolation methods described above.

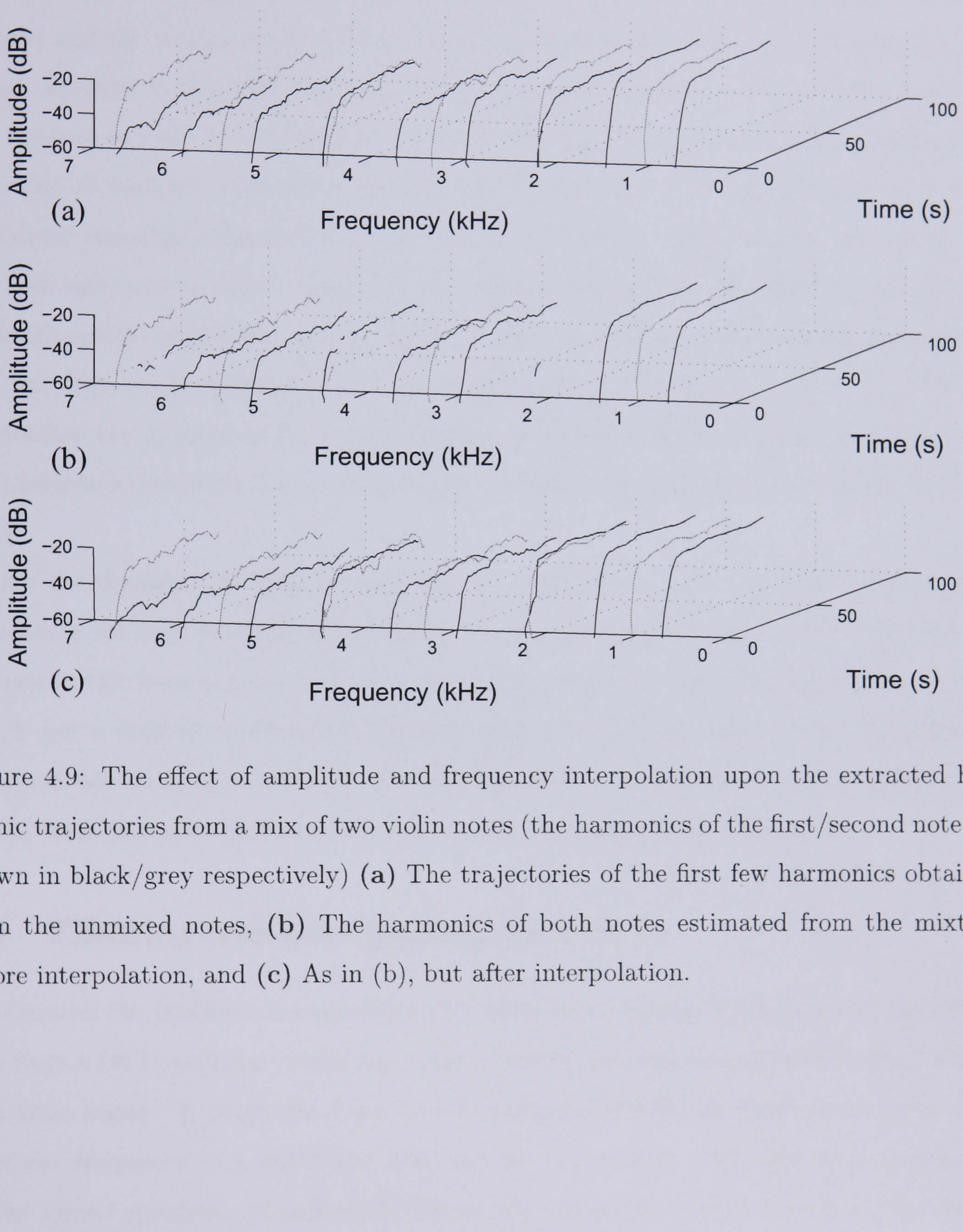

Figure 4.9: The effect of amplitude and frequency interpolation upon the extracted harmonic trajectories from a mix of two violin notes (the harmonics of the first/second note are shown in black/grey respectively) **(a)** The trajectories of the first few harmonics obtained from the unmixed notes, **(b)** The harmonics of both notes estimated from the mixture before interpolation, and **(c)** As in (b), but after interpolation.

4.4 Spectral filtering

Given the set of harmonic frequency and amplitude envelopes of each note, and phase envelopes if need be, there are two alternative approaches for separating a set of mixed notes that come to mind. The first is sinusoidal modelling, that is, the parameter trajectories are used to drive a set of time-varying oscillators, and the separated sounds are synthesised using additive synthesis. The second approach is to construct filters in the frequency-domain that selectively filter spectral content around the locations of the harmonics of each note. The two approaches are qualitatively different in that the first is an analysis-driven synthesis method and the second is an analysis-driven filtering operation. In the first approach, the broad objective is to achieve a clean or distortion free separated note, and the quality of the residual is more of a secondary concern. Whereas in the second, the objective is to filter out all content contributed by the note from the recording, ideally resulting in both a realistic sounding separated note, but also a residual free of any trace of the note. In the first approach we can be sure that the separated sound will be relatively free of noise although not necessarily free of other artifacts, whereas in the second, the filtering operation will also filter any broadband noise that is overlapping with harmonic frequencies. The two approaches are in some sense complementary, and their relative performance depends on the characteristics of the sound. They will be compared quantitatively in sections 4.8.2 and 4.8.3.

For the moment, the chapter continues by explaining the filtering approach in detail. Essentially we wish to construct in the spectral-domain a comb-like filter whose effect is to remove the harmonic content of a particular note from the mixed spectrum. Whilst this is not a difficult problem for isolated harmonics (section 4.5), due to the frequency of occurrence of harmonically related notes in music, some filter designs for separating overlapping harmonics from multiple instruments will be needed (section 4.6).

4.5 Filtering non-overlapping harmonics

We consider the problem of separating a set of non-overlapping harmonics of a particular note from a DFT spectrum containing a mix of notes, and assume quasi-stationarity within each time frame. A comb-like filter with a unity-amplitude band-pass response at each harmonic frequency of a particular note can be constructed, such that when multiplied by the mixed spectrum, it selectively filters the harmonics of that note from the mixed spectrum. Henceforth, we will refer to each unity-amplitude band-pass filter as a resonance of the comb-like filter. As the harmonics are of finite bandwidth due to the windowing process, the resonances should have bandwidths at least as large as the width of the main

spectral lobe of each harmonic. If the assumptions of stationarity are relaxed a little, sinusoids with time-varying frequency or amplitude trajectories will produce slightly broader spectral lines than normal. In fact, if the time-varying behaviour of these harmonics is too rapid to be measured for the chosen window parameters, there is no easy way to predict exactly how broad these spectral lines will be. Thus it makes sense to use an adaptive resonance bandwidth which spans the main spectral lobe, but is not so wide as to overlap too much nearby spectral content which is not produced by this harmonic.

Let $k_c = \text{round}(N f_m^p / f_s)$ be the index of the frequency bin closest to the corrected harmonic frequency f_m^p. Notice that the frame index has been dropped for convenience as this is a frame-by-frame filtering method. The width of the harmonic is determined by searching for the first minima in $A[k]$ at frequency bins $k_l \leq k_c - \upsilon$ and $k_r \geq k_c + \upsilon$ on opposite sides of the main lobe, where $2\upsilon + 1$ is at least the main lobe bandwidth for a stationary sinusoid, here chosen as $\upsilon = 2$. We denote a spectral filter or weighted mask $H^p[k]$ for note p, and set:

$$H^p[k] \;=\; 1 \quad \forall\, k \text{ s.t. } k_l \leq k \leq k_r. \tag{4.17}$$

Therefore, when $F[k]$ is multiplied by $H^p[k]$, the entire main lobe of the harmonic is extracted from the original mixed spectrum. The use of variable filter resonance bandwidths means that even if the harmonic is slowly time-varying, most of its spectral content will still be extracted from the mixed spectrum by increasing the resonance bandwidth. If only the main lobe of the harmonic is separated, it is possibly that sidelobes at a level of -43 dB or lower would remain in the residual spectrum. However, this seems to be an acceptable level of error for most audio applications, and could be remedied by using a window function with smaller sidelobes or using wider filter resonances.

Re-synthesis of the separated notes using the filters $H^p[k]$ is fairly simple. Denote the filtered spectrum of note p by:

$$F^p[k] \;=\; F[k]\, H^p[k] \quad ;\; k \;=\; 0, \ldots, N-1. \tag{4.18}$$

Note p is re-synthesised by performing the DFT^{-1} of $F^p[k]$ and then using an overlap-add process to smooth the resulting time segments across consecutive frames. Section 2.1.1 discusses the overlap-add procedure in more detail. The values of $H^p[k]$ for k greater than the Nyquist frequency $(N/2)$ can be determined from frequency bins below the Nyquist frequency. As the signal is real, the original DFT spectrum is complex conjugate symmetric, i.e. $F[k] = F[N-k]^*$ for $k = 1, \ldots, N-1$. Thus setting:

$$H^p[k] \;=\; H^p[N-k]^* \tag{4.19}$$

ensures that $F^p[k] = F^p[N-k]^*$, producing a real signal when the DFT^{-1} operates on

$F^p[k]$. For the moment $H^p[k]$ is actually real, but complex filters will be introduced in section 4.6.

An advantage of this filtering approach is that since the amplitude of each filter resonance is unity across the bandwidth of the main lobe of the harmonic, then the residual contains at most some traces of the harmonic due to sidelobes, which are not usually audible. This also holds for peaks containing overlapping harmonics due to a normalisation (eqn. 4.21) which will be discussed in the following section. On the other hand, if time-varying harmonics are modelled as well-behaved sinusoids, such as in [123, 124, 125, 49, 126], and the complexity of the sinusoidal model is not adequate to model the true harmonic evolution, then any residual calculated by subtracting the set of sinusoids from the original waveform is likely to contain some leakage of the harmonic content due to an imperfect sinusoidal subtraction. As the quality of the sinusoidal subtraction is fairly sensitive to errors in the sinusoidal frequency and phase estimates, it is fairly susceptible to producing this kind of artifact.

4.6 Filtering methods for overlapping harmonics

Section 4.5 described how spectral-domain filters of unity amplitude across the width of the non-overlapping harmonics of each note could be constructed easily from the corrected harmonic frequencies f_m^p. When a group of two or more harmonics from different notes are clustered about a spectral peak, the set of frequencies f_m^p remain uncorrected, i.e. they remain as the original predictions of the harmonic frequencies using eqns. 4.7 or 4.8. Let the set of notes contributing to a particular set of two or more overlapping harmonics be denoted Q. To separate the overlapping harmonics, the filters $H^q[k],\ q \in Q$ must be designed in a non-trivial way so that an appropriate division of the overlapping harmonic content is achieved using eqn. 4.18. In other words, if the number of overlapping harmonics is $|Q|$, then $|Q|$ overlapping filters are designed to split the overlapping harmonic content into $|Q|$ parts.

Three filter designs are proposed for the sharing of overlapping harmonic content. The first two were empirically derived but motivated by the following expectation: the amount of spectral content in $F[k]$ attributed to the m^{th} harmonic of note p decays as $|f_k - f_m^p|$ increases, and is also proportional to the amplitude a_m^p estimated in section 4.3.2. From this point of view, effectively we are trying to partition the energy in the spectral peak without any regard for phase information, hence these two methods will be referred to as energy-based filter designs. The third method is based upon the theoretical shape of the DFT spectrum for a sum of stationary harmonics, whose frequency and amplitude estimates are given by f_m^p and a_m^p.

The first un-normalised energy-based filter design is:

$$\hat{H}^p[k] \;=\; a_m^p \;\exp\left(-\frac{|f(k)-f_m^p|}{\sigma}\right) \quad ; \quad \begin{matrix}\forall\, p \in Q \\ k_l \le k \le k_r\end{matrix} \tag{4.20}$$

where $f(k) = kf_s/N$ is the equivalent frequency in Hz of bin k. Similarly to the non-overlapping case, $k_l \le \min_{q\in Q}\{k_c^q\} - \upsilon$ and $k_r \ge \max_{q\in Q}\{k_c^q\} + \upsilon$, where $k_c^q = \text{round}(f_m^q\, N/f_s)$. In other words, k_l and k_r are the first minima in $A[k]$ above and below the set of predicted overlapping harmonic frequencies. A suitable value for σ is 0.25 f_s/N. Again it is implicit that $m \equiv m(p)$, i.e. m is the harmonic number of pitch p that forms a part of the overlapping peak. $\hat{H}^p[k]$ in eqn. 4.20 is then followed by a normalisation to obtain $H^p[k]$:

$$H^p[k] \;=\; \frac{\hat{H}^p[k]}{\sum_{q\in Q}\hat{H}^q[k]} \quad ;\; k_l \le k \le k_r. \tag{4.21}$$

The purpose of the normalisation is to ensure that the overlapping filters sum to unity across the peak, this way the entire peak is filtered out of the mixed spectrum leaving no residual. Thus, we have:

$$\sum_{q\in Q} H^q[k] \;=\; 1 \quad ;\; k_l \le k \le k_r. \tag{4.22}$$

One can see from eqn. 4.20 that the region of influence of $\hat{H}^p[k]$ decreases as $|f(k) - f_m^p|$ increases, and is proportional to a_m^p in accordance with the initial expectation. The choice of exponential decay and the rate of decay σ were found empirically, by a combination of informal listening and maximisation of the mean signal-to-residual ratio (MSRR) (eqn. 4.49) for various random mixes of instruments.

The second filter design postulates that the region of influence of $\hat{H}^p[k]$ should decay according to the shape of the window function, and hence introduces a dependency on the Fourier transform of the window function $W(f)$:

$$\hat{H}^p[k] \;=\; a_m^p \;|W(|f(k) - f_m^p|)| \quad ;\; \forall\; p \in Q,\; k_l \le k \le k_r. \tag{4.23}$$

Again, this is normalised using eqn. 4.21 to obtain $H^p[k]$. In practice, the continuous function $W(f)$ is approximated by the DFT of the zero-padded window function (a zero-padding factor of 64 has been used) and $|f(k) - f_m^p|$ is rounded to the nearest equivalent frequency bin. $|W(f)|$ is the amplitude of the window function in the frequency-domain, and is normalised to $|W(0)| = 1$.

The overlapping filters of the second method are illustrated in fig. 4.10a for a sum of two harmonics producing an overlapping spectral peak in $|F[k]|$. Fig. 4.10b also shows the filtered amplitude spectra $|H^p[k]\;F[k]|$ for each source in comparison to the original amplitude spectra of the un-mixed harmonics $|F^p[k]|$. In fig. 4.11 the filters $|H^p[k]|$ have been constructed for a mix of a flute and violin note, and the first energy-based filter design

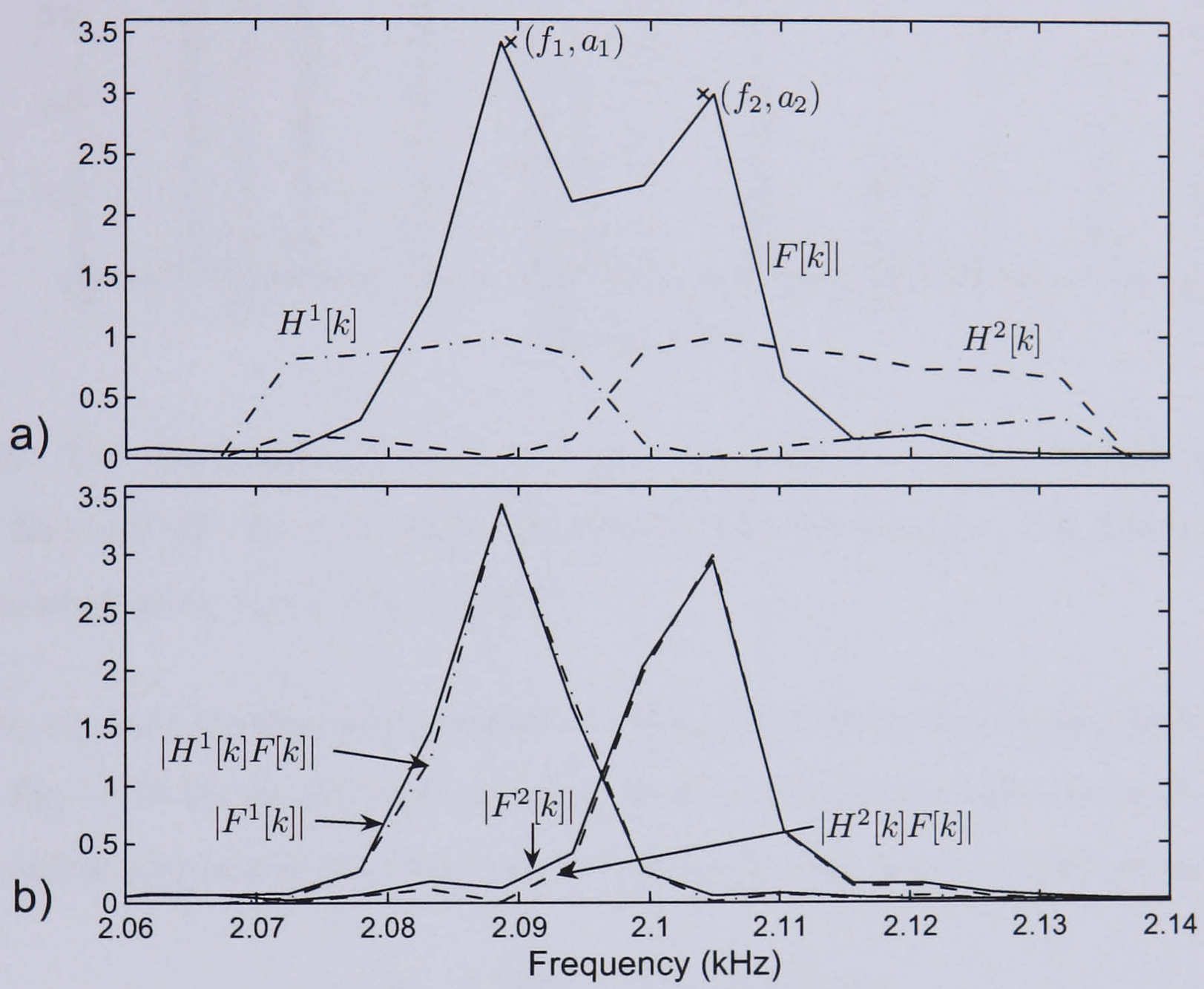

Figure 4.10: Filtering of a spectral peak in the DFT spectrum $F[k]$ arising from two overlapping harmonics. **(a)** The overlapping filters $H^p(k)$ defined in eqn. 4.23 and 4.21 and estimated harmonic frequencies and amplitudes f_p and a_p are shown. **(b)** Comparison of the filtered amplitude spectra, $|F[k]H^p[k]|$, with the original amplitude spectra, $|F^p[k]|$, of the individual harmonics.

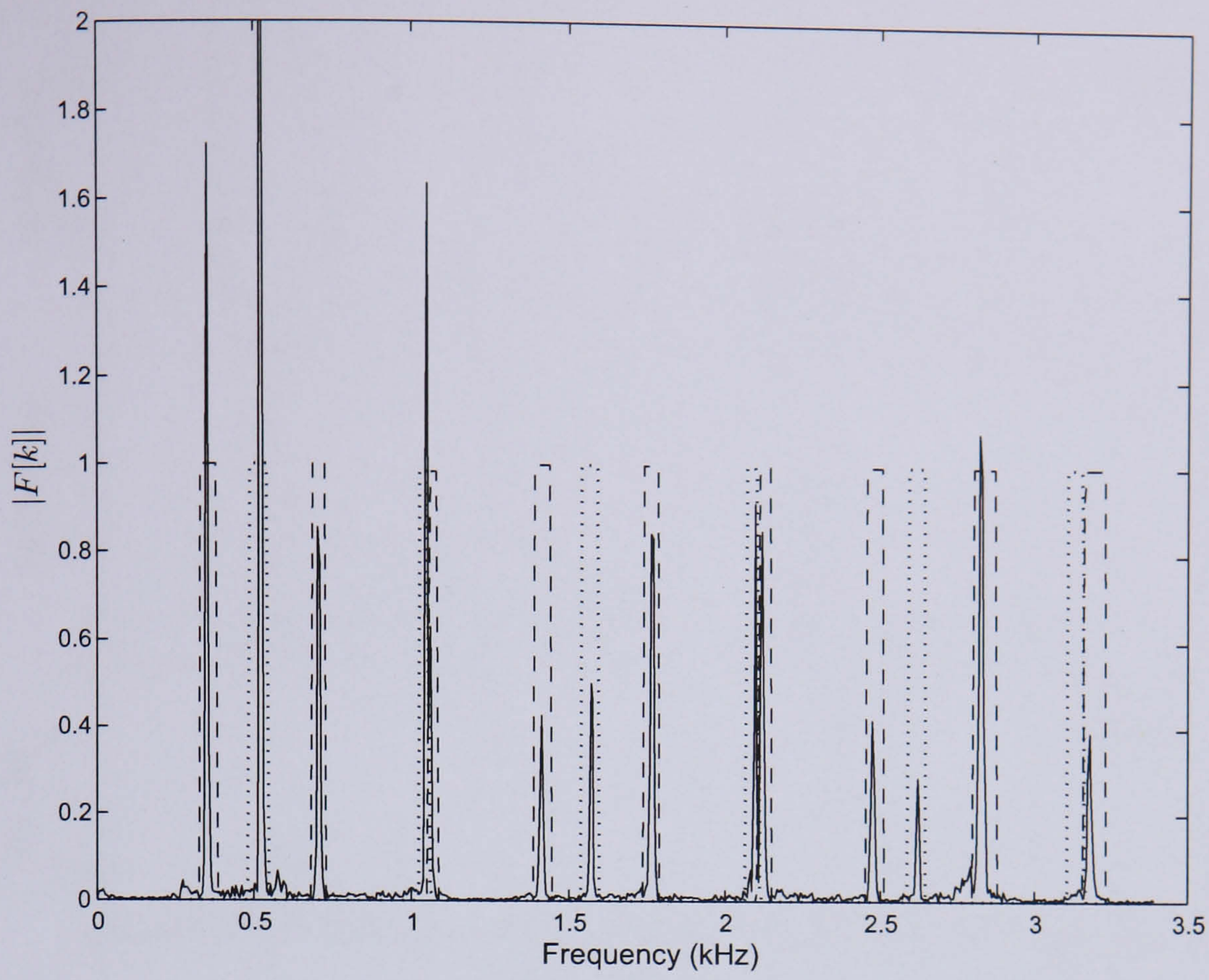

Figure 4.11: The amplitude spectrum of a mix of a violin and flute note with pitches F4 ($f_0 = 349$ Hz) and C5 ($f_0 = 523$ Hz) respectively (without vibrato), and filters $H^1[k]$ and $H^2[k]$ calculated using eqns. 4.20 and 4.21.

was used to separate overlapping harmonics located around the frequencies 1040, 2080, and 3120 Hz. Fig. 4.12 shows the corresponding filtered harmonic spectra for each note, and also the residual amplitude spectrum $|F_{res}[k]|$ after filtering, which is defined as:

$$F_{res}[k] \;=\; F[k] - \sum_{p=1}^{P} H^p[k]\; F[k]. \tag{4.24}$$

The above two energy-based filter designs were proposed simply as a way of splitting the energy in an overlapping peak into $|Q|$ parts in a way that reflects the predictions of the amplitudes and frequencies of the constituent harmonics, whilst ignoring phase information. If the predictions f_m^p, a_m^p and φ_m^p are accurate, it is possible to separate the overlapping peak almost exactly into its constituent parts using complex filters. This approach will now be elaborated upon.

Let the following model describe a cluster of frequency components $f_1 < f_2 < \ldots < f_M$ whose spectral content is overlapping once windowed and operated on by the DFT:

$$x(t) \;=\; \sum_{m=1}^{M} a_m \cos(2\pi f_m t + \phi_m). \tag{4.25}$$

The signal is assumed to be stationary and continuous in the range $t = (-\infty, \infty)$. Therefore, the Fourier transform of $x(t)$ multiplied by the window function $h(t)$, is a convolution of

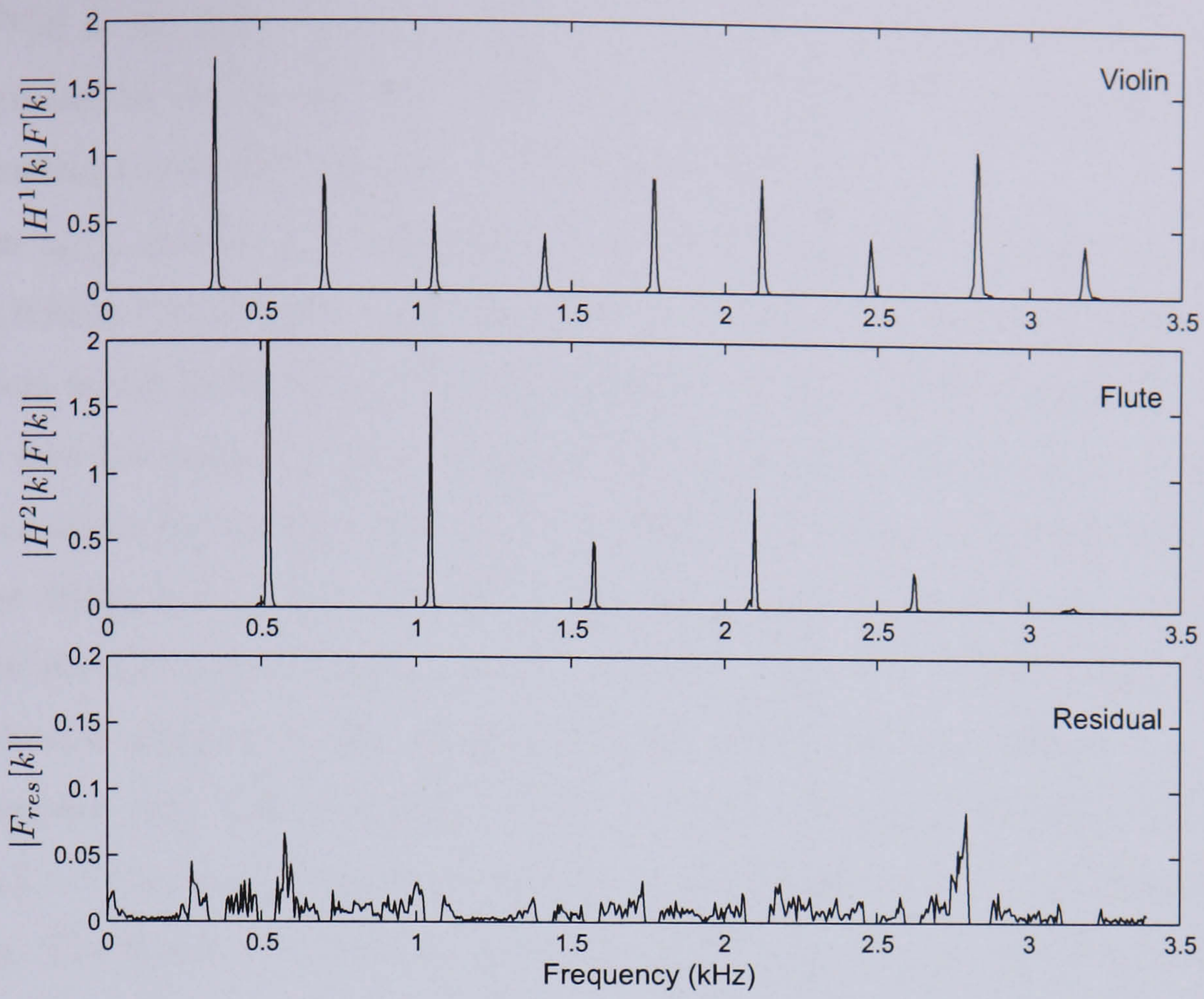

Figure 4.12: Filtering of the spectrum in fig. 4.11 into two harmonic spectra and a residual spectrum (*note different amplitude scales*).

the individual Fourier transforms, producing:

$$\mathcal{F}_{xh}(f) \;=\; \sum_{m=1}^{M} \frac{a_m}{2} \left[\, e^{i\phi_m}\; \mathcal{F}_h(f - f_m) + e^{-i\phi_m}\; \mathcal{F}_h(f + f_m) \,\right] \tag{4.26}$$

where $\mathcal{F}_h(f)$ and $\mathcal{F}_{xh}(f)$ are the Fourier transforms of the window function and windowed signal respectively. To derive the discrete signal case it can be shown, by choosing the window function to be zero outside of the analysis frame, that apart from an arbitrary constant, the DFT of the theoretical sampled and windowed signal, $\hat{F}[k]$, is approximately equal to $\mathcal{F}_{xh}(f(k))$. Also, when dealing with positive frequency components, the second term in brackets in eqn. 4.26 has little effect. Therefore,

$$\hat{F}[k] \;\simeq\; \mathcal{F}_{xh}(f(k)) \;\simeq\; \sum_{m=1}^{M} \frac{a_m}{2}\; e^{i\phi_m}\; \mathcal{F}_h(f(k) - f_m). \tag{4.27}$$

Now suppose that a_m, f_m and ϕ_m are accurate. Then it is possible to design a filter $H^p[k]$:

$$H^p[k] \;=\; \frac{a_p\; e^{i\phi_p}\; \mathcal{F}_h(f(k) - f_p)}{\sum_{m=1}^{M} a_m\; e^{i\phi_m}\; \mathcal{F}_h(f(k) - f_m)} \tag{4.28}$$

that when multiplied by $\hat{F}[k]$ results in approximately the DFT of the windowed sinusoid p. In other words:

$$\begin{aligned} H^p[k]\;\hat{F}[k] \;&=\; \frac{a_p\; e^{i\phi_p}\; \mathcal{F}_h(f(k) - f_p)}{\sum_{m=1}^{M} a_m\; e^{i\phi_m}\; \mathcal{F}_h(f(k) - f_m)}\; \hat{F}[k] \\ &\simeq\; \frac{a_p}{2}\; e^{i\phi_p}\; \mathcal{F}_h(f(k) - f_p) \\ &\simeq\; \hat{F}^p[k] \end{aligned} \tag{4.29}$$

where $\hat{F}^p[k]$ is the DFT of the isolated p^{th} sinusoid. In non-ideal practical situations, $H^p[k]$ can still be calculated and applied to the measured DFT spectrum $F[k]$ giving an approximation to the DFT of the p^{th} overlapping harmonic.

Given a_p, f_p and ϕ_p, it would also be possible to compute $\hat{F}^p[k]$ directly and simply subtract it from $F[k]$ to yield a residual spectrum. However, any slight error in the sinusoidal parameters would result in an imperfect subtraction, and a potential leakage of the p^{th} sinusoid into the residual. Artifacts would also appear if the original signal model was at all inaccurate, for instance due to the sinusoidal parameters being slowly time-varying. The filter design in eqn. 4.28 performs a reasonably good separation when the parameter estimates are only approximately accurate, and like the first two filter designs, eqn. 4.22 holds, which avoids any audible leakage of the harmonics into the residual.

In practice, eqn. 4.28 is applied to the set of $|Q|$ overlapping harmonics by mapping $\{f_m^p,\ a_m^p\} :\rightarrow \{f_m,\ a_m\}$. There are at least two possibilities for acquiring the phase estimates ϕ_p. The first is to simply use the predicted harmonic phases from section 4.3.2, i.e. $\{\varphi_m^p\} :\rightarrow \{\phi_p\}$. However, it is not always possible to perform a time-domain interpolation between non-overlapping frames in order to find $\{\varphi_m^p\}$. The second option is to estimate the phases directly from the shape of the measured spectrum $F[k]$. Suppose $F[k]$ is sampled at M different frequency bins: $k_1, \ldots, k_M$, with $f(k_m)$ chosen to be as close to f_m as possible under the condition that $k_1 < k_2 < \ldots < k_M$. Then a set of equations can be found using eqn. 4.27 that are solved by the inversion of a non-singular matrix U, yielding $\phi_1, \ldots, \phi_M$:

$$\hat{F} \ = \ U \cdot \Phi \tag{4.30}$$

where

$$\hat{F} \ = \ \begin{pmatrix} F[k_1] \\ \vdots \\ F[k_M] \end{pmatrix}, \quad \Phi \ = \ \begin{pmatrix} e^{i\phi_1} \\ \vdots \\ e^{i\phi_M} \end{pmatrix},$$

$$U \ = \ \begin{pmatrix} a_1\mathcal{F}_h(f(k_1) - f_1) & \cdots & a_M\mathcal{F}_h(f(k_1) - f_M) \\ \vdots & & \vdots \\ a_1\mathcal{F}_h(f(k_M) - f_1) & \cdots & a_M\mathcal{F}_h(f(k_M) - f_M) \end{pmatrix}$$

Therefore,

$$\begin{aligned} \Phi \ &= \ U^{-1} \cdot \hat{F} \\ \phi_m \ &= \ \angle\Phi[m] \end{aligned} \tag{4.31}$$

In fact, a third way to calculate the set $\{\phi_m\}$ would be to perform a direct LSE minimisation between the measured and theoretical DFT spectra of the form:

$$\arg\min_{\{\phi_m\}} \sum_{k=k_l}^{k_r} \left| F[k] - \hat{F}[k] \right|^2 \tag{4.32}$$

where $\hat{F}[k]$ is computed as in eqn. 4.27. This is more computationally expensive than the other methods as it involves an optimisation over M parameters, $\{\phi_m\}$, and as it only converges to zero in ideal sinusoidal conditions and when $\{f_m,\ a_m\}$ are exact, the method was not taken any further. It does, however, have some similarity with the nonlinear least squares (NLS) method[125] mentioned in section 4.7.5 for estimating sinusoidal parameters in white noise, although the NLS method is actually a time-domain approach.

A quantitative comparison of the three filtering methods to each other and to other methods for separating overlapping harmonics will be given in section 4.8. For the moment, a test case will be used to illustrate some of the characteristics of these filters. The test case reflects a 'best case scenario' in which the estimates $\{a_m, f_m\}$ are exact and the overlapping harmonics are stationary sinusoids. Figs. 4.13–4.15 show the results of filtering the spectral peak arising from two sinusoids which are separated by 12 Hz (slightly more than 2 frequency bins for $N = 8192$) with -20 dB added white noise. One notices for the first filter design (fig. 4.13) that the shape of each filter is roughly exponentially decaying mid-way between the two harmonic frequencies. For the second filter design (fig. 4.14), each filter is roughly balanced around the predicted harmonic frequency, but is not monotonically decreasing in the region of overlap, which is due to sidelobes in the window's Fourier transform $W(f)$.

For the last complex filter design (fig. 4.15), both filters are very unpredictable and at times the filter amplitude is greater than one, although a nearly perfect reconstruction of the un-mixed sinusoidal spectra is obtained. The fact that it is possible for $|H^p[k]|$ to be greater than one is, however, problematic when any of the estimates $\{a_m, f_m, \phi_m\}$ are inaccurate, and can give rise to very large peaks in the filtered spectra. In this situation it would be better to separate the partials by spectral subtraction rather than filtering. In [82] a linear equations solution was used to find the amplitudes of two colliding partials, and some reasons were given to explain the deficiencies in this approach, which are also exhibited by this last filtering algorithm: Firstly, the partial frequencies and shape of the spectral peak can usually not be known accurately, secondly, the set of linear equations becomes singular as the partial frequencies approach each other (this was avoided in eqn. 4.31 by requiring $k_1 < k_2 < \ldots < k_M$), and thirdly, overlapping partials of very different amplitudes exaggerate parameter errors. Overall, results given in section 4.8.3 indicate that the first two filter designs, although not optimal in ideal sinusoidal conditions, are more reliable than the complex filter when applied to real audio.

Whilst the previous test case identified some general characteristics of the filters, a more rigorous experiment was necessary to measure their average performances in test conditions, and was also given in [127]. Again we consider the problem of separating two sinusoids where the sinusoidal parameters are estimated from the spectrum. An experiment was

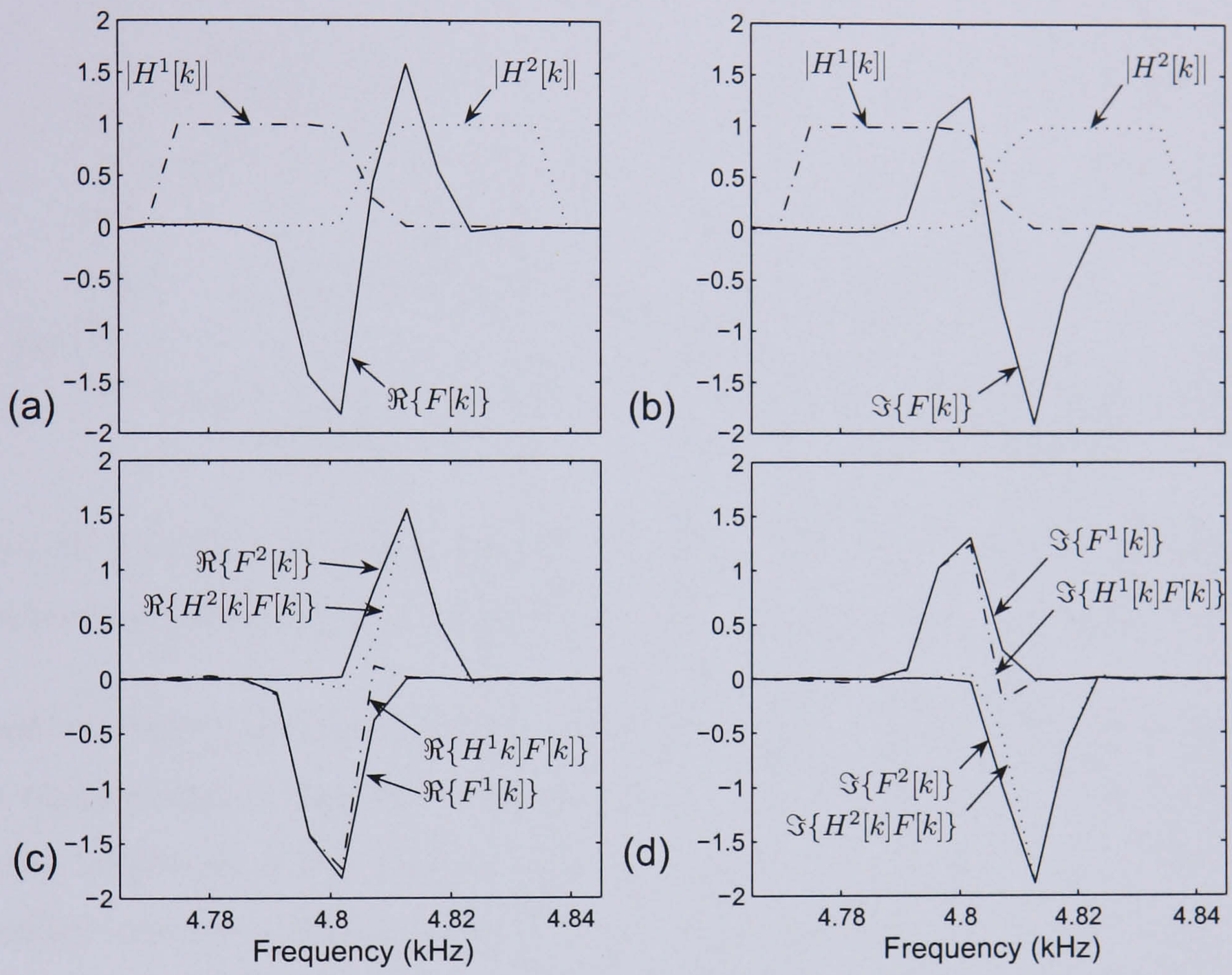

Figure 4.13: Filtering of a spectral peak in the DFT spectrum arising from two overlapping sinusoids of frequencies 4800 and 4812 Hz using eqns. 4.20 and 4.21. **(a)** Real value of the DFT, $\Re\{F[k]\}$, and filter amplitudes, $|H^p[k]|$ **(b)** Imaginary value of the DFT and filter amplitudes **(c)** Comparison of the real DFT of the un-mixed sinusoids, $\Re\{F^p[k]\}$, with the real filtered spectra, $\Re\{H^p[k]F[k]\}$ **(d)** Comparison of the imaginary DFT spectra of the un-mixed sinusoids with the imaginary filtered spectra.

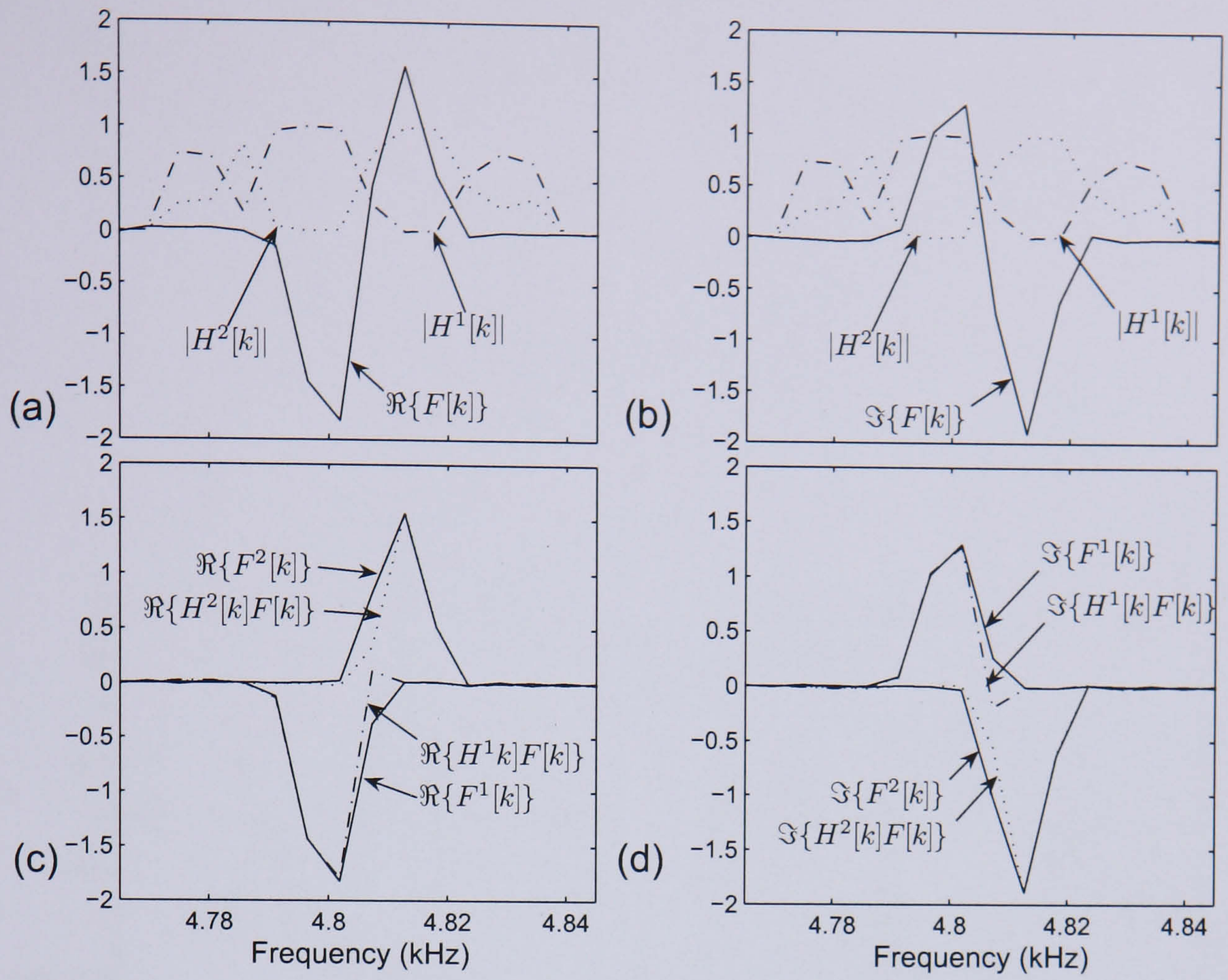

Figure 4.14: Filtering of a spectral peak in the DFT spectrum arising from two overlapping sinusoids of frequencies 4800 and 4812 Hz using eqns. 4.23 and 4.21. **(a)-(d)** see fig. 4.13.

designed to obtain a measure of average separation performance as follows. Two sinusoids with a random relative frequency difference in the range $[0, 4]$ frequency bins, each having a random amplitude in the range $[0, 1]$, and a random phase offset in the range $[-\pi, \pi]$, were added together to simulate a random pair of overlapping sinusoids. The robustness of the three filter designs as a function of the error which would be incurred when estimating the sinusoidal frequencies and amplitudes was then evaluated, by using either the correct amplitude of both sinusoids and a rough estimate of their frequencies, or vice versa. In the former case, the rough estimates $\hat{f}_m$ were produced by adding a random frequency in the range $[-r, r] \times f_s/N$ to each known sinusoidal frequency f_m. In the latter case, the rough estimates $\hat{a}_m$ were produced by multiplying each known sinusoidal amplitude a_m by a random number in the range $[1 - r, 1 + r]$. A measure of the error between the original and separated DFTs of the two sinusoids was defined as:

$$R(r) = \frac{1}{2} \sum_{p=1}^{2} \frac{\sum_{k=0}^{N/2} |F^p[k] - H^p[k]\ F[k]|^2}{\sum_{k=0}^{N/2} |F^p[k]|^2}. \tag{4.33}$$

Fig. 4.16 shows the average value of $R(r)$ over 10^5 iterations for each value of r as it is varied in the range $[0, 1]$. Unsurprisingly, the separation performances of all filters decrease when the frequency and amplitude estimates decrease in accuracy, i.e. r increases. The results reveal that when the frequency and amplitude estimates are accurate, the complex filter

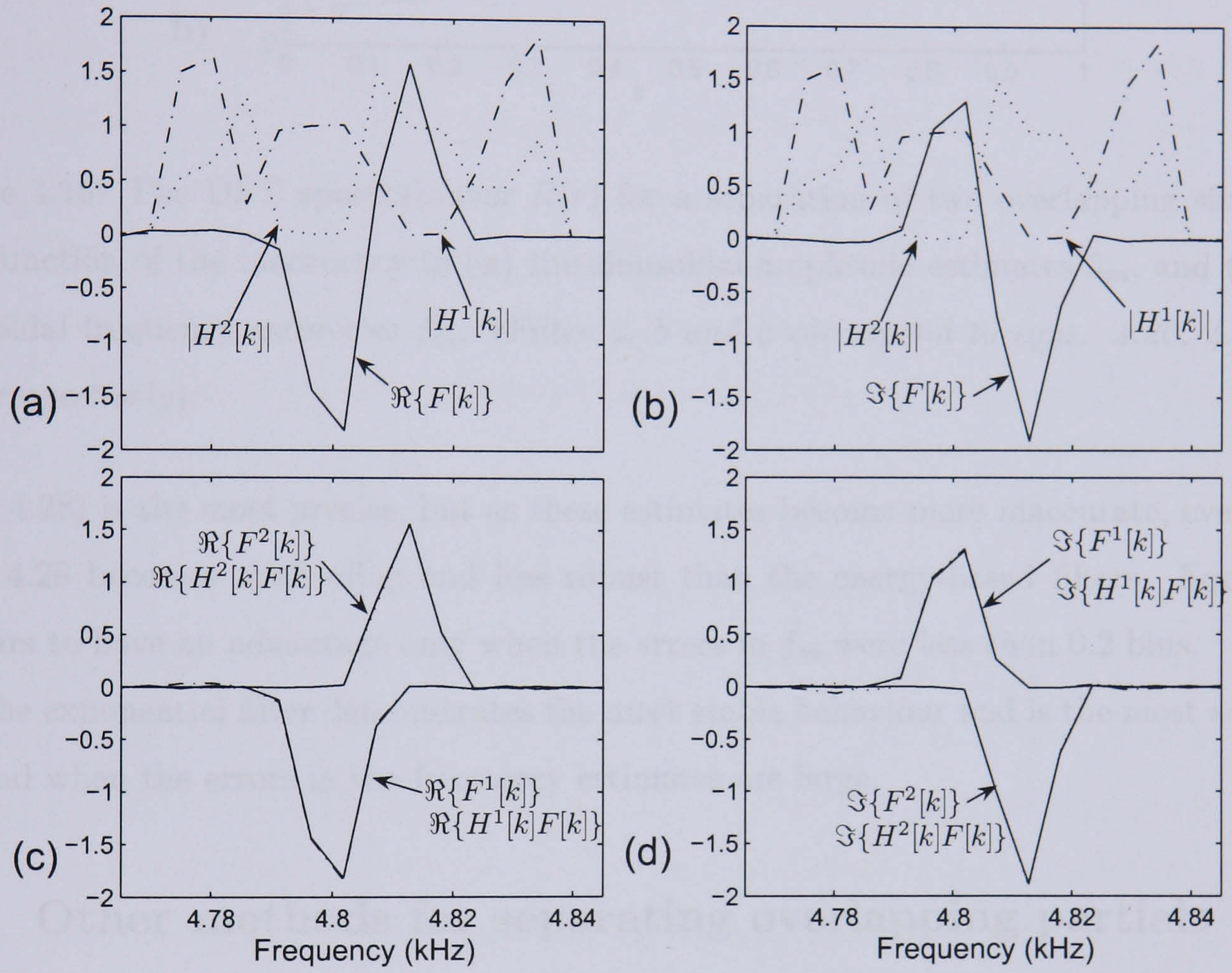

Figure 4.15: Filtering of a spectral peak in the DFT spectrum arising from two overlapping sinusoids of frequencies 4800 and 4812 Hz using eqn. 4.28. **(a)-(d)** see fig. 4.13. The original un-mixed spectra are almost indistinguishable from the filtered spectra in (c) and (d).

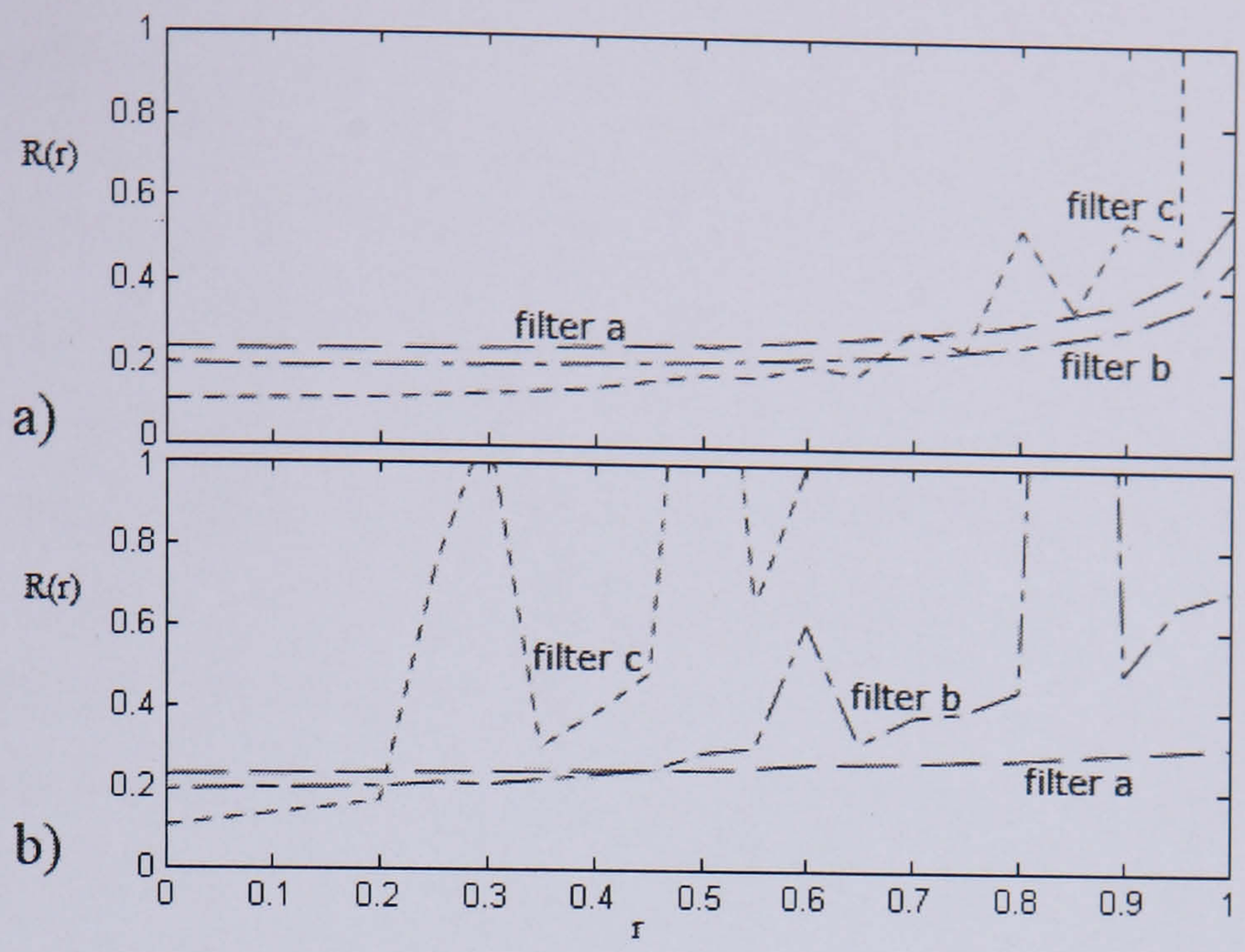

Figure 4.16: The DFT spectral error $R(r)$ for a separation of two overlapping sinusoids, as a function of the inaccuracy in **(a)** the sinusoidal amplitude estimates $\hat{a}_m$, and **(b)** the sinusoidal frequency estimates $\hat{f}_m$. (*Filter a, b and c correspond to eqns. 4.20, 4.23 and 4.28 respectively*).

(eqn. 4.28) is the most precise, but as these estimates become more inaccurate, eventually eqn. 4.28 becomes misleading and less robust than the energy-based filters. Eqn. 4.28 appears to have an advantage only when the errors in $\hat{f}_m$ were less than 0.2 bins.

The exponential filter demonstrates the most stable behaviour and is the most accurate method when the errors in the frequency estimates are large.

4.7 Other methods for separating overlapping partials

In the previous section three filtering methods were introduced with the aim of separating a set of overlapping harmonics of different notes from the mixed DFT spectrum. Here, a number of other approaches to separating overlapping harmonics will be reviewed, and in section 4.8.3 a quantitative comparison will be made between some of these approaches and the filtering methods described above.

4.7.1 Parsons' method/partial amplitude interpolation

An early attempt at separating overlapping harmonics was described in [128] in the context of separation of vocalic speech of two competing talkers. Spectral peaks arising from overlapping harmonics were identified using three criteria: the distance from neighbouring spectral peaks, the symmetry of the peak, and the phase behaviour across the spectral peak. Peaks further apart than one half the 20 dB ideal peak bandwidth were not considered to be overlapping. The symmetry test indicated whether the asymmetry of the peak shape was

any larger than the maximum that could result solely from asymmetric sampling. Lastly, the phase test detected any rapid change in phase occurring across the peak that could potentially have arisen from overlapping partials with different phases.

Overlapping partials with an appreciable degree of separation were separated by subtracting the ideal peak shape of the largest partial from the mixed spectrum, computed from its estimated frequency, amplitude, phase, FM rate and the window function, to reveal smaller partials. Peaks that were irresolvable, referred to as 'shared peaks', were identified by post analysis of the harmonic series of the two speakers. Crosstalk arising from shared peaks was identified as a major cause of degradation in the separated waveforms. It was stated in [128] that these peaks were simply shared between the two speakers by performing an amplitude interpolation between adjacent harmonics. We will refer to this technique as Parsons' method, bearing in mind that the amplitude interpolation was used in [128] only for irresolvable peaks. A method for amplitude interpolation between the first non-overlapping adjacent harmonics of each overlapping harmonic is already given in eqn. 4.12. This amplitude interpolation is used in section 4.8.3 when referring to Parsons' method.

4.7.2 Spectral models of neighbouring harmonics

It was already mentioned in section 4.7.1 that a simple amplitude interpolation between adjacent resolved harmonics could be used to estimate the amplitudes of unresolved/-overlapping harmonics. Perhaps this idea can be extended to incorporate more elaborate spectral models or correlations between neighbouring harmonics trajectories.

In [82] a spectral template matching method was used to determine the missing harmonic amplitudes in a harmonic series in which most of the harmonics were resolvable. A set of spectral templates, each consisting of a series of harmonic amplitudes, was pre-determined by analysing clean recordings of the voice. In the mixture, the set of resolved harmonics of a particular voice was matched to the closest template using a LSE error fit over the harmonic amplitudes. The unresolved harmonics of this voice were then determined by using the template as a look-up table. This technique was claimed to be unreliable in [82], in part due to the inadequacy of representing the enormous range of spectral variation possible in a single voice/instrument due to both expressive qualities and the recording environment by a finite set of spectral templates.

In [129, 93] the notion of 'spectral smoothness' was used in estimating harmonic amplitudes in an iterative subtractive system for polyphonic pitch estimation. The iterative multi-pitch estimator operates by determining the most dominant pitch in the mix, then subtracts the harmonics of this pitch from the mixed spectrum, and then applies the pitch

estimator to the residual to find the next most significant pitch. The idea behind spectral smoothness is that the human auditory system tends to perceive a series of harmonic amplitudes showing a smooth decline with frequency as arising from a single source. Single harmonics that are of higher amplitude than the smoothed envelope tend to be perceived as independent sounds. The system in [129, 93] is based upon the assumption that when multiple harmonics overlap, overlapping spectral peaks stick out from the smoothed envelope of each harmonic series. This is particularly relevant to harmonic intervals such as octaves where all of the harmonics of the upper pitch would be overlapping with harmonics of the lower pitch, and so could not be measured directly. Therefore, when the harmonics of the dominant pitch are subtracted from the spectrum, care must be taken to adjust its estimated harmonic amplitudes to be no higher than the height of the smoothed spectral envelope, so that any overlapping peaks are only partially removed in the process. This was claimed to approximately halve multi-pitch estimation note error rates.

Another method was developed in [130, 63] for resolving overlapping partials across multiple time frames in multi-channel mixtures, which combined spatial de-mixing techniques with inference based on the fact that neighbouring harmonics of a single note usually have similar AM and FM characteristics over time. The method begins by segmenting the spectrum into frequency regions containing one or more partials. Then, a spatial de-mixing matrix is estimated for each spectral region containing more than one unresolved harmonic, that produces separated partials whose amplitude or frequency envelope variations closely match those of adjacent resolved partials in regions containing only one partial. A similarity measure between two amplitude envelopes, where $E_1[n]$ and $E_2[n]$ are square roots of the corresponding energies in both partials as a function of time, was given as:

$$\beta = \frac{\sum_n E_1[n]\, E_2[n]}{\sqrt{\sum_n |E_1[n]|^2}\,\sqrt{\sum_n |E_2[n]|^2}}. \tag{4.34}$$

This was found to be close to one for neighbouring partials, and on average a slowly decreasing function of the harmonic number when measured relative to the first partial, as shown in fig. 4.17 for a flute note played with vibrato (the flute vibrato is predominantly a result of amplitude modulation rather than frequency modulation). Although the above equation measures the similarity of two harmonic amplitude envelopes, it was mentioned in [63] that the similarity between frequency envelopes could also be useful for sources in which the FM characteristics of all partials are synchronous, especially as strong amplitude modulations of partials due to vibrato can sometimes be out of phase, leading to low similarities of the amplitude envelopes. The technique applies to additive mixing models in which M microphones in a room record M different mixtures of the N sources, where in general $M \geq N$. Although this technique is of limited use here as we are dealing with mono signals, it has demonstrated that inference based upon correlations of adjacent harmonics

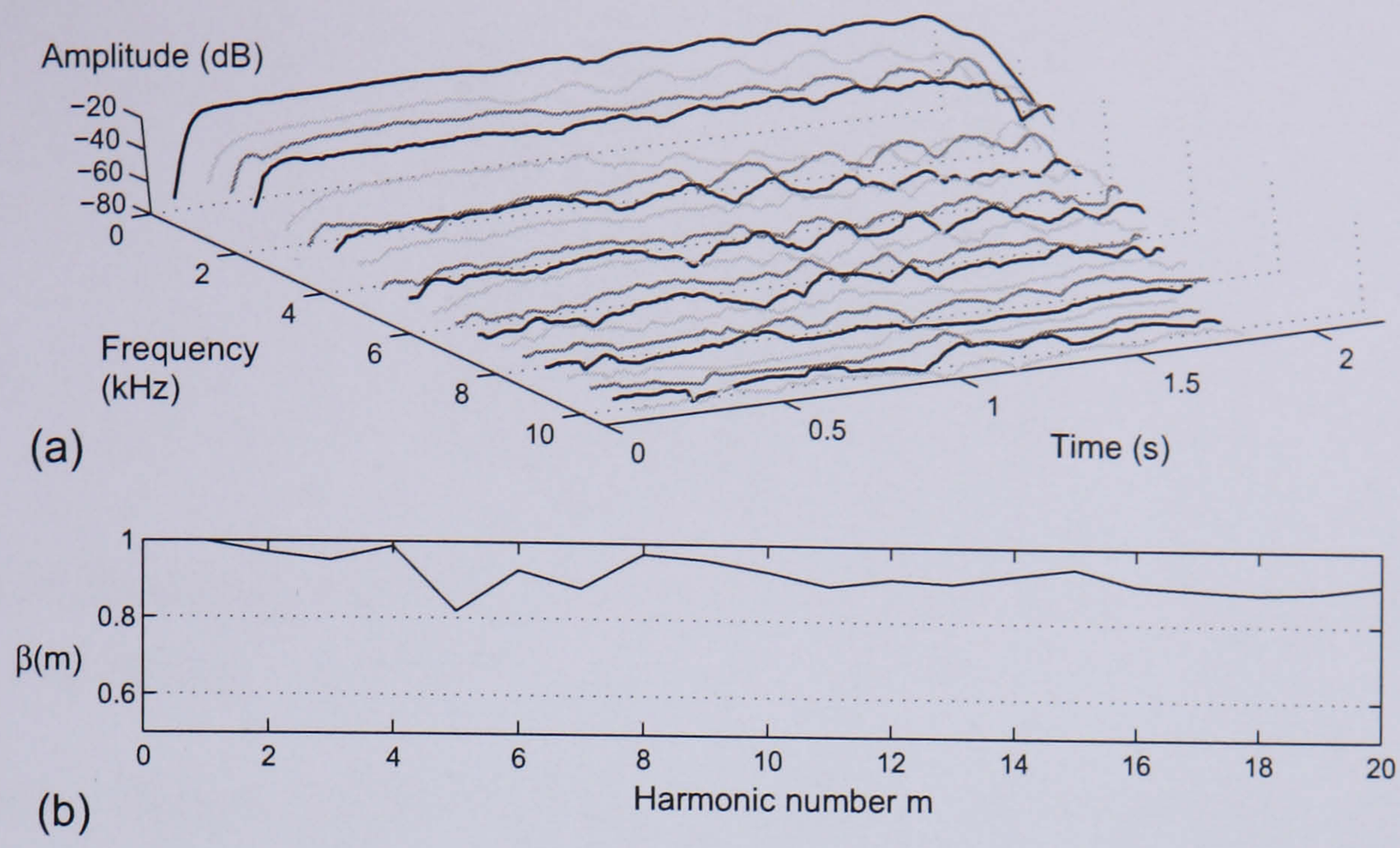

Figure 4.17: **(a)** The first 20 harmonics of a flute note of pitch 523 Hz played with vibrato (adjacent harmonics have been given different shades of grey for ease of viewing), **(b)** The similarity measure defined in eqn. 4.34 for the first 20 harmonics relative to the first harmonic (i.e. $E_1[n]$ in eqn. 4.34 is the amplitude envelope of the first harmonic and $E_2[n]$ is the amplitude envelope of each higher harmonic)

to overlapping harmonics can be useful for separation.

4.7.3 Exploitation of beating

It is well known that a sum of sinusoids of angular frequencies ω_1 and ω_2 can be written as the product of a sinusoid of frequency $(\omega_1 + \omega_2)/2$ with a slower amplitude modulation at the difference frequency $|\omega_2 - \omega_1|/2$. In general, when the two sinusoids have different amplitudes and phases, we have:

$$\begin{aligned} a_1 \cos(\omega_1 t + \Delta\varphi) \;+\; a_2 \cos(\omega_2 t) \;=\; (a_2 - a_1)\cos(\omega_2 t) \;+\; \ldots \\ 2a_1 \cos\left(\frac{\omega_1 + \omega_2}{2} t + \frac{\Delta\varphi}{2}\right) \cos\left(\frac{\omega_1 - \omega_2}{2} t + \frac{\Delta\varphi}{2}\right) \end{aligned} \tag{4.35}$$

When ω_1 and ω_2 are sufficiently close to one another, the difference frequency $\frac{\omega_1 - \omega_2}{2}$ is quite small and so $\cos(\frac{\omega_1 - \omega_2}{2} t + \frac{\Delta\varphi}{2})$ effectively produces a slow amplitude modulation of the higher frequency sinusoidal component of frequency $\frac{\omega_1 + \omega_2}{2}$, giving rise to an effect known as 'beating', which can be seen in fig. 4.18. The beat frequency is actually $|\omega_1 - \omega_2|$ rather than $|\frac{\omega_1 - \omega_2}{2}|$ as the last cosine term in eqn. 4.35 is of unity magnitude twice during each period. The maximum of the beat envelope is $|a_1| + |a_2|$ and the minimum is $||a_1| - |a_2||$.

In [123] a method is proposed that exploits beating to obtain the amplitude envelopes of individual partials in a mix of two slowly time-varying partials. The amplitude envelope of the larger amplitude partial can be obtained by low-pass filtering the envelope of the mixed

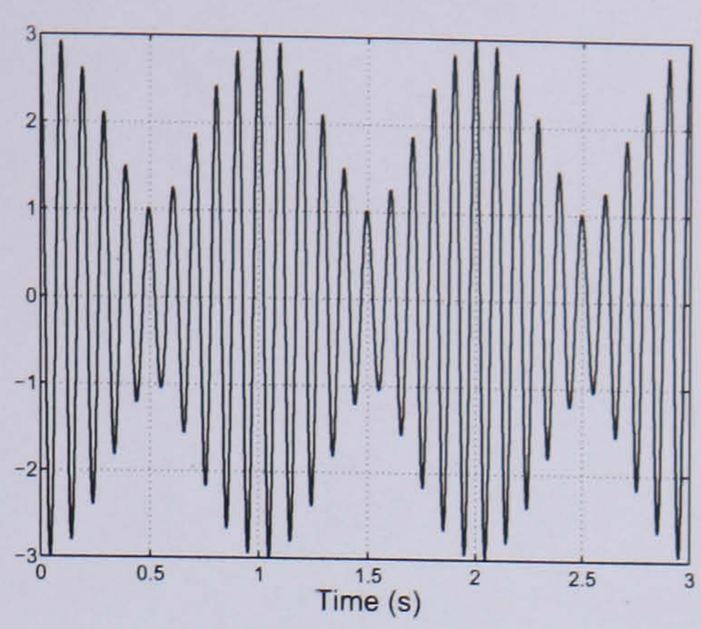

Figure 4.18: Beating of a sum of two sinusoids with frequencies 10 and 11 Hz and amplitudes 2 and 1 respectively.

components. Then, the amplitude envelope of the second partial is obtained by subtracting the first envelope from the original, half-wave rectifying and low-pass filtering the result. The extracted amplitude trajectories were used in a system for source separation based upon additive synthesis. As the method only applies to sums of two partials, [123] used an amplitude interpolation between resolved partials when more than two partials were overlapping.

Beating was also employed in the multi-strategy approach of [82] for separating overlapping harmonics in duet signals when the harmonics were separated by less than 25 Hz and the duration of the overlap was longer than two beat periods. It was described how to determine which of two colliding sinusoids is of larger amplitude. The two sinusoids produce a single spectral peak which exhibits amplitude and frequency modulation. If the amplitude minimum occurs at the same time as the frequency minimum, then the amplitude of the lower frequency sinusoid is the larger of the two amplitudes a_1 and a_2. Otherwise, the lower frequency sinusoid has the smaller of the two amplitudes.

A number of problems in using beating to determine partial trajectories have been identified. Firstly, solutions have only been presented for two overlapping partials, as it would be rather more difficult to estimate the amplitude trajectories of more than two overlapping sinusoids from the beat characteristics. Secondly, it relies on the fact the amplitude and frequency trajectories of both sinusoids are relatively constant over the duration of the beat analysis. Otherwise, the beat characteristics would change too rapidly to allow reliable estimates of the beat parameters. Thus, this excludes notes played with vibrato or showing moderate AM or FM behaviour. Thirdly, the duration of the beating must be long enough to allow accurate measurements of the beat period, maxima and minima. In [82] beating was only used when the duration of overlap was at least two beat periods.

4.7.4 Linear equations solutions

Linear equation solutions, similar to that used in eqn. 4.31 for estimating the phases of overlapping partials, have been used on a number of occasions for estimating partial parameters in a LSE sense in the spectral domain[49, 82, 131, 124, 126, 132]. We start by giving the basic LSE solution in [131] for estimating partial amplitudes and phases, and then its extension to partial frequency estimation by linearising the window function.

The sinusoidal model of the windowed spectrum:

$$\hat{S}(f) \;=\; \sum_{k=1}^{K} \frac{a_k}{2} \left(e^{i\phi_k} W(f - f_k) + e^{-i\phi_k} W(f + f_k) \right) \tag{4.36}$$

which is linear in terms of the amplitudes a_k and $e^{i\phi_k}$, but nonlinear in terms of frequency, can be rewritten as:

$$\hat{S}(f) \;=\; \sum_{k=1}^{2K} p_k \; R_k(f) \tag{4.37}$$

where for $k = 1, \ldots, K$:

$$\begin{aligned} p_k &= \tfrac{a_k}{2} \cos(\phi_k) \\ p_{k+K} &= \tfrac{a_k}{2} \sin(\phi_k) \end{aligned} \tag{4.38}$$

and

$$\begin{aligned} R_k(f) &= W(f - f_k) + W(f + f_k) \\ R_{k+K}(f) &= iW(f - f_k) - iW(f + f_k) \end{aligned} \;. \tag{4.39}$$

Defining $\mathcal{R}_{j,k} = R_k(F_j)$, eqn. 4.37 can be written in matrix notation:

$$\hat{\mathbf{S}} \;=\; \mathcal{R}\, \mathbf{p} \tag{4.40}$$

where $\hat{\mathbf{S}}$ is a vector of samples of the ideal spectrum at frequencies F_j and of size $J \times 1$, $\mathbf{p}$ is the vector of unknown amplitude and phase parameters of size $2K \times 1$ and $\mathcal{R}$ is of size $J \times 2K$. The LSE error solution to the above, i.e. which minimises the error function $||\mathbf{S} - \hat{\mathbf{S}}||$ where $\mathbf{S}$ is the measured spectrum, and from which the partial amplitudes and phases can be determined, is:

$$\mathbf{p} \;=\; (\mathcal{R}^H \mathcal{R})^{-1}\, \mathcal{R}^H\, \mathbf{S}. \tag{4.41}$$

When partial frequencies are very close together, $\mathcal{R}$ in eqn. 4.41 can become singular, leading to numerical instability. This was solved in [126] by using a linear model for the series of harmonic amplitudes of each sound. The vector of harmonic amplitudes $\mathbf{a}$, was modelled using a linear model: $\mathbf{a} = X\mathbf{y}$, in terms of the lower-order parameter vector $\mathbf{y}$, and transform matrix X. $\mathbf{y}$ was solved for in a least-squares sense similarly to eqn. 4.41, from which $\mathbf{a}$ could easily be calculated by multiplying by X. This technique was motivated by the idea of 'spectral smoothness' discussed in section 4.7.2, and the observation that the amplitudes of adjacent harmonics are correlated.

Given the amplitudes and phases of a set of partials, their frequencies $\mathbf{f} = [f_1, \ldots, f_K]$ were estimated in [131] by linearising the window function using a first order Taylor expansion about the rough estimates of the partial frequencies, $\hat{\mathbf{f}}$:

$$W(f \mp f_k) \;=\; W(f \mp \hat{f}_k) \;\mp\; W'(f \mp \hat{f}_k)\;\Delta_k \;+\; O(\Delta_k^2) \tag{4.42}$$

where $\Delta_k \;=\; f_k - \hat{f}_k$, and $W'(f)$ is the derivative of $W(f)$. Defining the matrix Ω by:

$$\Omega_{j,k} \;=\; \frac{a_k}{2}\;\left(-e^{i\phi_k}W'(F_j - \hat{f}_k) + e^{-i\phi_k}W'(F_j + \hat{f}_k)\right) \tag{4.43}$$

the LSE solution for the partial frequencies is:

$$\mathbf{f} \;=\; \hat{\mathbf{f}} + (\Omega^H\Omega)^{-1}\;\Omega^H\;(\mathbf{S} - \hat{\mathbf{S}}). \tag{4.44}$$

The joint estimation of the partial parameters was performed iteratively, alternately computing frequency and then amplitude and phase estimates, starting from the initial frequencies determined using a spectral peak-picking algorithm. In [82, 49, 124, 126] rough estimates of the partial frequencies were known *a priori* or from an initial pitch estimation.

An extension of this technique to harmonic sounds was introduced in [124]. This maintains that the ratio of the k^{th} harmonic frequency to the fundamental frequency, $r_k \simeq k$, remains constant over time. Hence, it was shown that eqn. 4.44 can be rewritten in a form that retains the harmonic structure of the sound. The multi-strategy approach to separating overlapping harmonics in [82] also employed a linear equations solution when the partials were further apart than 25 Hz but less than 50 Hz (a Kaiser window with a 6 dB bandwidth of 40 Hz was used for the STFT).

To conclude, the LSE approach produces an optimal solution to the harmonic amplitudes, frequencies and phases as long as the signal is stationary over the duration of the analysis frame. The basis of the method is that each spectral peak arising from one or more partial is made up of a sum of scaled and translated Fourier transforms of the window function. However, in non-stationary conditions, the shape of the spectral peak also depends on any amplitude and frequency modulation of the underlying partial/s. It is also possible that iterative solutions can converge to window sidelobes, although this was avoided in [131] by using windows without sidelobes. Lastly, the method is liable to numerical instability when the partials are closely spaced together in frequency, although [126] has suggested one way of overcoming this problem.

4.7.5 Nonlinear least squares method

The nonlinear least squares (NLS) method[133, 125] for separating colliding sinusoids provides a time-domain LSE fit of a sinusoidal model to the signal. The sinusoidal parameter

estimates are also equal to the maximum likelihood estimates in the situation that the sinusoidal model differs from the signal by a white noise component. The method attempts to find the sinusoidal parameters that minimise the cost function:

$$E(\mathbf{f}, \mathbf{a}, \phi) = \sum_{n=0}^{N-1} \left| x[n] - \sum_{k=1}^{K} a_k \, e^{i(2\pi f_k n + \phi_k)} \right|^2 \tag{4.45}$$

where rough estimates of f_k are pre-determined. In [125] the method is formulated for a sum of two sinusoids, i.e. $K = 2$, but it is straightforward to extend the technique to more than two sinusoids. The cost function is minimised using the following separated equations:

$$\begin{aligned} \hat{\mathbf{f}} &= \arg\max_{\hat{\mathbf{f}}} \{\mathbf{x}^H B (B^H B)^{-1} B^H \mathbf{x}\} \\ \hat{\beta} &= (B^H B)^{-1} B^H \mathbf{x}|_{\mathbf{f}=\hat{\mathbf{f}}} \end{aligned} \tag{4.46}$$

where

$$\begin{aligned} \beta_k &= a_k \, e^{i\phi_k} \\ \beta &= [\beta_1 \ \beta_2]^T \\ B &= \begin{pmatrix} 1 & 1 \\ e^{i2\pi f_1} & e^{i2\pi f_2} \\ \vdots & \vdots \\ e^{i2\pi(N-1)f_1} & e^{i2\pi(N-1)f_2} \end{pmatrix}. \end{aligned} \tag{4.47}$$

Starting with the initial frequency estimates, an iterative optimisation method was used to find the frequencies $\hat{\mathbf{f}}$ in eqn. 4.46, and this was followed by the estimation of the joint phase and amplitudes $\hat{\beta}$ given the estimated frequencies. Alternative forms of the cost function in eqn. 4.45 were given in [134] for estimating the fundamental frequency of purely harmonic sources, and for the detection of inharmonicity in sources like the piano where partial spacings are stretched according to eqn. 3.16. Although a quantitative evaluation of the NLS method for estimating the parameters of overlapping sinusoids was not given in [125, 134], we might expect that the technique is susceptible to some of the same problems as the spectral-domain LSE methods given in section 4.7.4, in part due to the partial trajectories again being assumed to be stationary. The NLS method will be compared in section 4.8.3 with the other filtering approaches.

4.8 Comparative evaluation of partial filtering with other methods

We see in the previous discussions of sections 4.3 and 4.7 a very strong emphasis on the sinusoidal model for representing partials. Whilst, in section 4.3 importance was placed on tracking time-varying sinusoidal parameters, ironically in section 4.7, many of the methods reviewed for separating overlapping partials were based upon stationary sinusoidal models.

It is therefore of interest to quantitatively compare some of these methods for separating overlapping partials with the filter designs described in section 4.6, which were designed specifically to permit slight inaccuracies in partial parameters. Before we attempt this in section 4.8.3, it is worthwhile to compare the two approaches for separating non-overlapping partials in test conditions: the first approach being to filter spectral peaks using the unity amplitude filters $H^p[k]$, which was described in section 4.5, and the second, to subtract sinusoids from the mix in the time or spectral-domain given their parameter estimates. The results of this experiment are provided in section 4.8.2.

4.8.1 Quantitative measures of separation performance

It is unfortunately very difficult to define a measure of separation performance. Whilst, in theory, a perceptual measure of the degree of separation would be ideal, this is prone to controversy as to whether the measure is actually perceptually accurate. Furthermore, an accurate perceptual measure is unlikely to be easily computable and free of modelling parameters. Although another option is to perform listening tests, this does not lend itself to quick evaluation of algorithms during development, and is impractical if the algorithm contains many user-defined parameters which would have to be determined optimally in a perceptual sense. For these reasons two simple quantitative measures of separation performance were used: the signal-to-residual ratio (SRR), and the average increase in the sum of SRRs, denoted χ/M. The measures do not directly measure the audibility of distortions. Hence, particularly for real recordings, it is important to make available the actual processed waveforms for informal or formal performance evaluation by the listener or set of listeners. Some audio results are available to listen to on the internet[135, 136, 137].

The SRR evaluates the similarity between the waveform of the m^{th} separated source, $x'_m[n]$, and its corresponding unmixed original $x_m[n]$:

$$SRR_{\mathbf{x_m}}(\mathbf{x'_m})\ [dB] \ = \ 10 \log \frac{\sum_n x_m[n]^2}{\sum_n (x_m[n] - x'_m[n])^2}. \tag{4.48}$$

If the order of the separated sources is different to that of the originals, the correct matching of the original to separated sources can be achieved by swapping the order of the separated sources until the maximum of $SRR_{\mathbf{x_m}}(\mathbf{x'_m})$ is achieved for each value of m. It is also useful to define the mean signal-to-residual ratio (MSRR) for a mix of M sources as:

$$MSRR \ = \ \frac{1}{M} \sum_{m=1}^{M} SRR_{\mathbf{x_m}}(\mathbf{x'_m}). \tag{4.49}$$

The second quantitative measure which is related to the SRR is the average increase in the

sum of SRRs:

$$\frac{1}{M}\,\chi(\mathbf{x_1}\ldots\mathbf{x_m},\mathbf{x'_1}\ldots\mathbf{x'_m},\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}(SRR_{\mathbf{x_m}}(\mathbf{x'_m}) - SRR_{\mathbf{x_m}}(\mathbf{x})) = \frac{1}{M}\sum_{m=1}^{M}10\log\frac{\sum_n(x_m[n]-x[n])^2}{\sum_n(x_m[n]-x'_m[n])^2} \quad (4.50)$$

where $x[n] = \sum_{m=1}^{M} x_m[n]$ is the mixed original signal. The MSRR and χ/M provide an overall measure of how well the mix has been separated into its original components. with larger values indicating better separation performance. No attempt is made until chapter 5 to split the non-harmonic residual waveform any further, and so for mixes of notes containing large transient or noise components, the MSRR and χ/M decrease on average.

It is clear that these measures all rely on the availability of the original un-mixed components $x_m[n]$. Whilst this is unproblematic for test evaluations in which the mix is manually summed from individual un-mixed sources, it is impossible to apply these methods to real recordings apart from in the fortunate situation that original master tapes of the recording are available, and the sources were recorded with a large degree of separation. When using quantitative measures of separation performance such as these, one must be careful not to assume that algorithms evaluated in test conditions are truly reflective of their performance on real music, which is quite often of higher complexity than test conditions. Hence, this work has been developed alongside continual informal evaluations on real recordings.

4.8.2 Comparison between sinusoidal extraction and partial filtering for a sinusoid in noise

As one of the main aims of this chapter is to propose and evaluate spectral filtering methods as an alternative to sinusoidal subtraction or additive synthesis, it is useful to begin by comparing the performances of the two approaches in test conditions. For the moment, non-overlapping partials in noise will be treated, and section 4.8.3 gives some results for overlapping partials.

Three test signals have been analysed: firstly, a pure 400 Hz sinusoid, secondly, a chirp signal varying linearly from 400 to 500 Hz over 2 seconds, and thirdly, a 400 Hz sinusoid with a FM vibrato of frequency 5 Hz and amplitude 6 Hz. The first test sample is indicative of the separation of a perfectly stationary partial from noise, and the second and third attempt to find some clues relating to the performance of the methods on time-varying partials. The separation of each of these signals from a variable level white noise component was evaluated using the SRR, for different settings of the window parameters N and N_{hop}, so that the effects of the noise level and window properties could be determined. The signal-

to-noise ratio (SNR) was varied within the range $[-15, 25]$ dB, and all three samples were two seconds long and sampled at 44.1 kHz.

Although the variable noise level is measured using the SNR, this does not directly provide a measure of how large the sinusoidal spectral peak is in relation to the spectral noise level, as this also depends on the DFT length. As the white noise component is distributed over all frequency bins, the larger the DFT length, the lower the relative spectral noise level for a given SNR. However, a rough empirical relationship between the SNR and the ratio in dB, called λ, of the peak amplitude to the root-mean-square (RMS) spectral noise level was found to be $\lambda \simeq \text{SNR(dB)} - 4 + 3\log_2(N)$. Thus, a SNR of -15 dB is equivalent to the sinusoidal peak being roughly 14 dB or 5 times higher than the RMS spectral noise level for $N = 2048$, and similarly 20 dB or 10 times higher than the RMS spectral noise level for the longer window $N = 8192$.

For the filtering method, the bandwidth of the filter $H[k]$ was fixed at 4 frequency bins, which is the main lobe bandwidth of the Hamming window used in the analysis. In other words, the filtering algorithm simply finds the maximum of the absolute spectrum $|F[k_c]|$, then multiplies $|F[k]|$ by a filter of unity amplitude across the 4 bins containing the main lobe of the sinusoid. A fixed resonance bandwidth was used only for evaluation purposes, in practice it would adapt to the actual shape of the peak leading to better separation performance, but would have to be limited to avoid interfering with adjacent partials. The DFT^{-1} of the filtered spectrum was then computed, and the overlap-add method described in section 2.1.1 was used to re-construct the filtered signal. For the sinusoidal model, the sinusoid was synthesised in the time-domain using the McAulay-Quatieri (MQ) algorithm[30] with cubic phase and linear amplitude interpolation (using code available at [138]). The sinusoid's frequency was determined using the quadratic frequency estimator (eqn. 4.5), and its amplitude was estimated using the shape of the window function (eqn. 4.4). The performance of the sinusoidal model is, of course, highly dependent on these parameter estimates, which are affected by the choice of window function, and frequency and amplitude estimators. However, the Hamming window and quadratic estimator are both popular methods for sinusoidal analysis.

Figs. 4.19, 4.20 and 4.21 show the results for the pure sinusoid, chirp and vibrato signal respectively. Some general observations and conclusions can be made from these figures:

1. For the non-stationary samples, the sinusoidal model consistently performs better than filtering when the DFT length is small. This is not surprising as smaller window lengths provide more accurate estimates of time-varying sinusoidal parameters.

2. The filtering process favours smaller hop sizes even for the stationary sinusoid. This could indicate that the accuracy of the overlap-add procedure is better for smaller

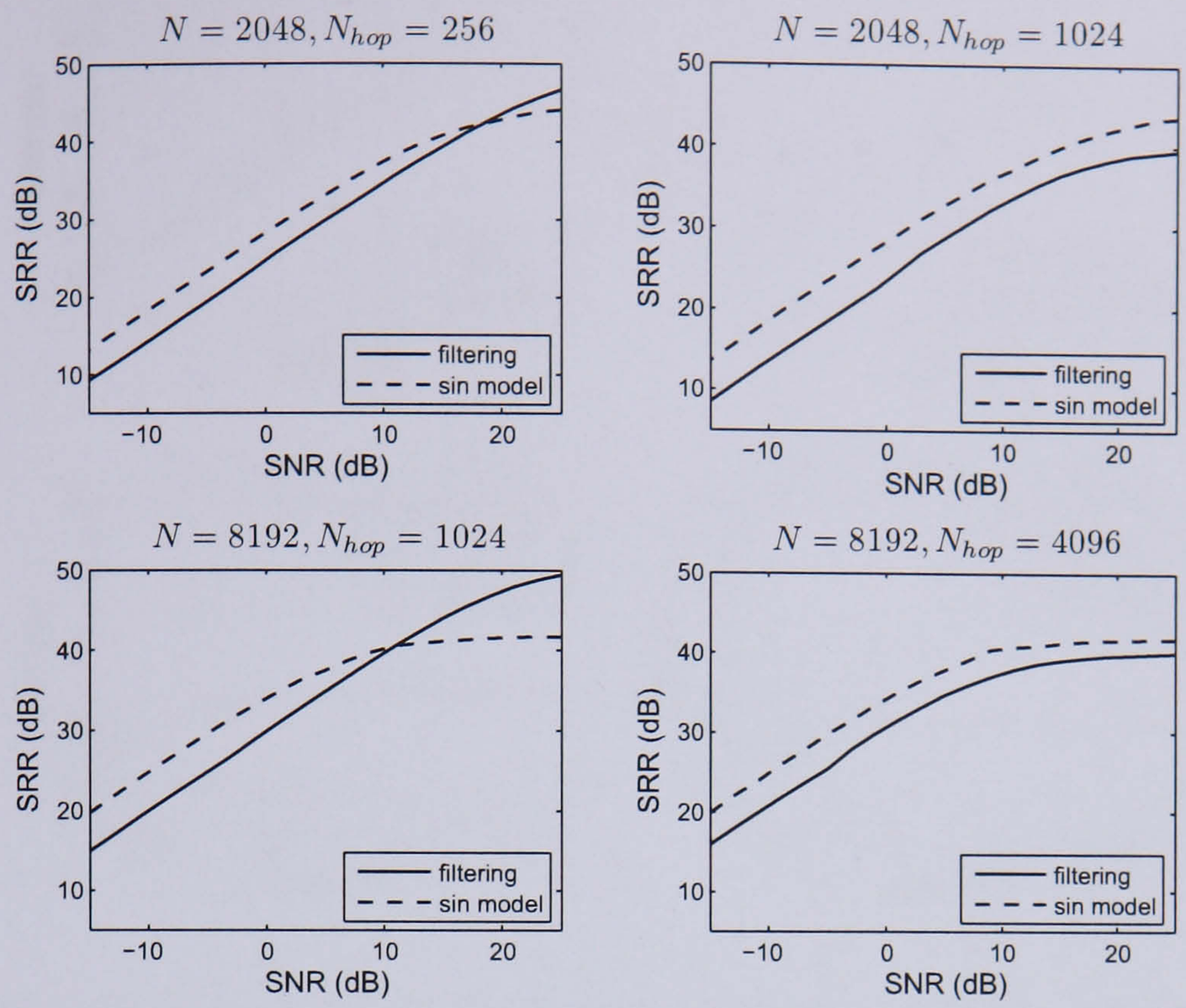

Figure 4.19: Comparison of the SRR when separating a 400 Hz sinusoid from white noise using sinusoidal modelling and by spectral peak filtering, as a function of SNR, DFT length and hop size in samples (f_s is always 44.1 kHz).

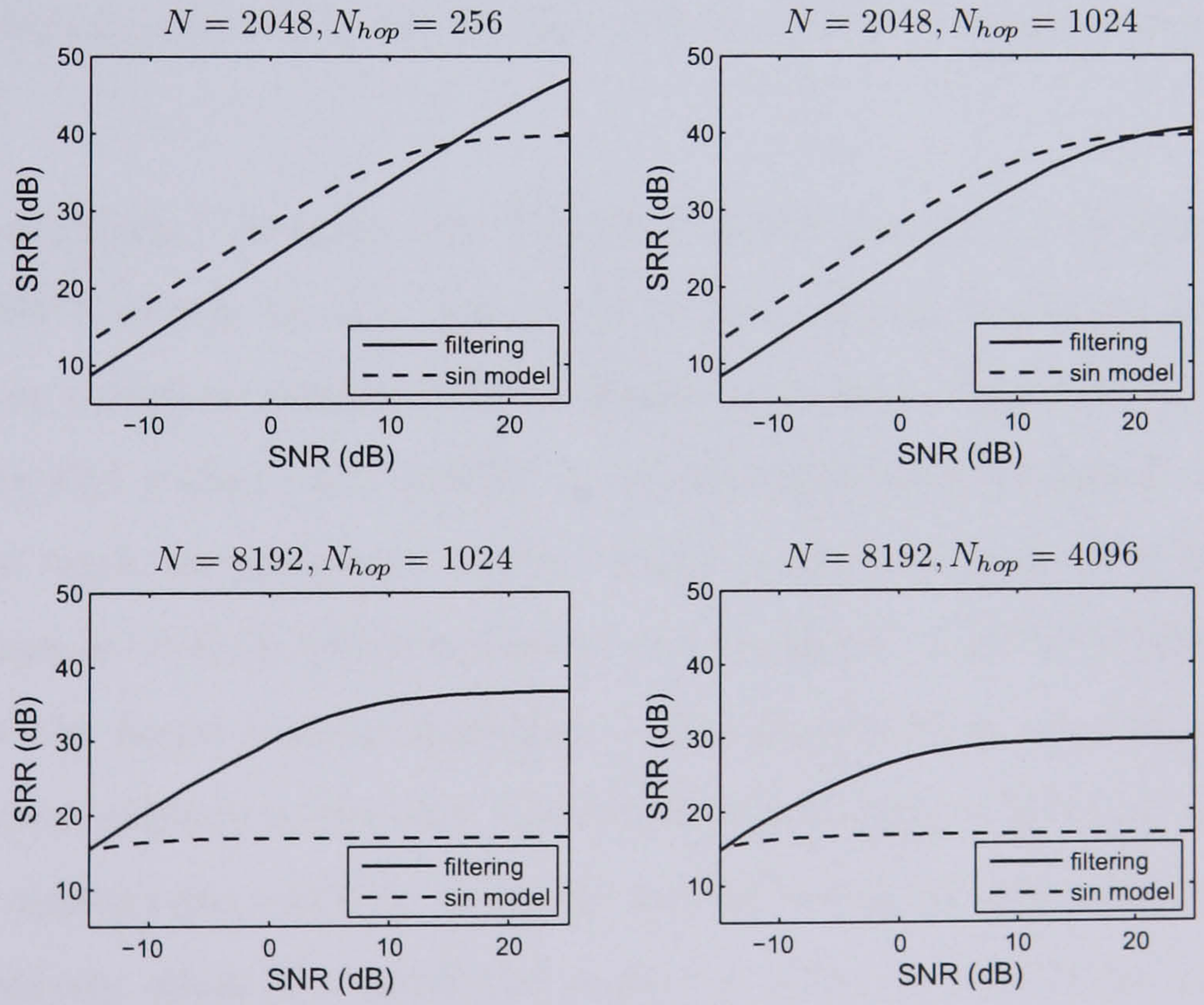

Figure 4.20: Comparison of the SRR when separating a linear chirp (f_{start} = 400 Hz, f_{end} = 500 Hz, duration = 2s) from white noise using sinusoidal modelling and by spectral peak filtering, as a function of SNR, DFT length and hop size in samples.

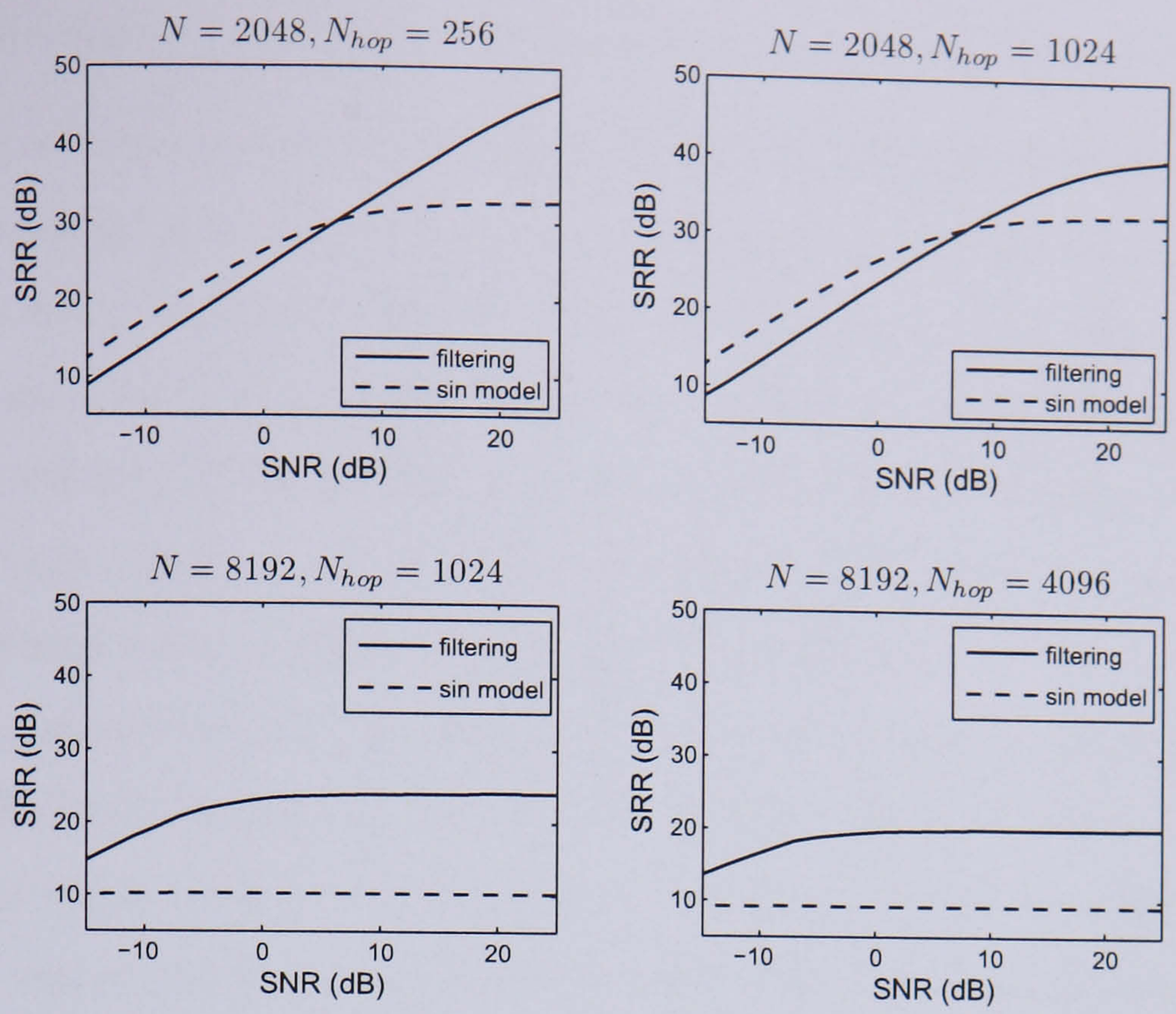

Figure 4.21: Comparison of the SRR when separating a 400 Hz sinusoid with vibrato (vibrato frequency/amplitude = 5/6 Hz respectively) from white noise using sinusoidal modelling and by spectral peak filtering, as a function of SNR, DFT length and hop size in samples.

values of N_{hop}. The hop size does not seem to have much effect on the sinusoidal model, although it would be expected that the choice of N_{hop} would be more critical for higher frequency sinusoids, as the phase difference between frames would be relatively larger.

3. For non-stationary samples, the filtering method generally performs better for the smaller DFT length. In the case of the vibrato signal, the larger window length is almost one complete vibrato cycle in length (f_s/vibrato frequency = 8820 samples). Thus, the FM within each window is almost completely averaged out, and we are unable to track the frequency of the vibrato signal effectively. For the chirp signal, the FM rate is 50 Hz/s which works out at a variation of almost 10 Hz or 2 frequency bins over the larger window duration. This causes a large increase in width of the spectral peak relative to the filter bandwidth of 4 frequency bins, and hence a decrease in performance (this could be partially avoided using an adaptive filter bandwidth). Comparatively, when $N = 2048$, the frequency varies by only about 0.1 frequency bin over the window duration.

4. At low SNRs and sufficiently small values of N and N_{hop} to account for the non-stationarity of the signal, the sinusoidal model performs better than filtering.

5. At higher SNRs, or in situations where the signal's non-stationarity is not tracked

effectively using the sinusoidal model, the filtering method performs better.

Observation 3 indicates that the filtering method performs best when the filter resonance fully covers the main lobe of the spectral peak. Naturally, a compromise must be reached, as using larger filter widths provides better removal of non-overlapping partials, but it also makes it more likely that adjacent partials are treated as overlapping. The lower performance of filtering at low SNRs reiterates the fact that the filtering method is not a 'noise-removal' technique, i.e. any noise located within the frequency bins around the peak maximum is filtered out in addition to the sinusoidal content.

The performance of the sinusoidal method is generally better in favourable conditions, that is, when the DFT size and hop length are sufficiently small to track the non-stationary behaviour of the signal. However, a decrease in performance of the sinusoidal model on the non-stationary signals occurred even for the smaller of the two window lengths $N = 2048$ (46 ms), especially for the vibrato signal. This would suggest using an even smaller window length to improve tracking, and tests using $N = 1024$ did indeed result in a better performance for the sinusoidal model. However, using short windows reduces the frequency resolution of the DFT spectrum (for $N = 1024$, the difference in frequency between adjacent bins is 43 Hz with $f_s = 44.1$ kHz), and though it may lead to better tracking of a single sinusoid in noise, it would not necessarily have an equally good performance on a multi-component signal. Closely spaced sinuoids would be relatively more overlapping at shorter window lengths, and this would decrease the accuracy of the sinusoidal parameter estimation. Given that the musical spectra that we are dealing with typically contain a large number of closely spaced partials, it is advisable to work with windows of at least 2048 samples in length.

To summarise, for the sole purpose of separating a set of partials from audio, i.e. without reference to the superiority of the sinusoidal model for flexible parametric sound transformation, it is advisable to use spectral filtering when the signal cannot be considered quasi-stationary for the chosen window parameters, and the noise level is sufficiently low that any filtered noise at the partial locations is not deemed to be disturbing.

4.8.3 Comparison between partial filtering and other separation methods for real signals

Monophonic signals

We have found that in test cases where the signal is non-stationary and moderately clean, spectral filtering seems to perform better than sinusoidal extraction for extracting partials from the mixed spectrum, although overlapping partials have not yet been dealt with. Before dealing with polyphonic samples, it should be confirmed whether this statement

Table 4.1: Average SRR for the extraction of random pitched notes from $0/-20$ dB white noise using spectral filtering (eqns. 4.20 and 4.21) and sinusoidal subtraction. Standard deviations of the average SRRs are given in brackets (standard deviation of the average = standard deviation of the samples/$\sqrt{\text{number of samples}}$).

	SNR = 0 dB $N = 2048$ $N_{hop} = 256$	SNR = 0 dB $N = 8192$ $N_{hop} = 1024$	SNR = 20 dB $N = 2048$ $N_{hop} = 256$	SNR = 20 dB $N = 8192$ $N_{hop} = 1024$
Filtering	10.6 (0.4)	15.7 (0.3)	25.4 (0.7)	24.3 (0.8)
Sin Model	8.0 (0.4)	12.6 (0.4)	19.5 (0.7)	17.3 (0.5)

holds for real sounds. We now compare SRRs using the first energy-based filtering design (with the filter resonance width again fixed as the window main lobe width) with the sinusoidal extraction method described above, for the separation of the harmonic content of single instrument notes from variable level white noise. The experiment was conducted at 0 dB and -20 dB white noise levels, and at two settings of the window parameters: $N = 2048$, $N_{hop} = 256$ and $N = 8192$, $N_{hop} = 1024$. Table 4.1 shows the average SRRs for the two methods obtained by averaging over at least 50 randomly chosen notes for each parameter setting. The notes were selected by first choosing a random instrument from amongst 11 different instrument types (bassoon, B♭ clarinet, cello, E♭ clarinet, flute, French horn, oboe, piano, soprano saxophone, tenor trombone and violin), and then choosing a random note from within the instrument's complete pitch range.

The results in fig. 4.1 demonstrate consistently better performance for the filtering approach over sinusoidal modelling. The filter resonance bandwidths were fixed in this experiment to the window's main lobe bandwidth, and significantly better results are obtained for the filtering method when these bandwidths are adaptable. The difference in performance between the two approaches is slightly larger at the 20 dB SNR, which confirms the conclusion of the previous experiment on synthetic samples that filtering has a more marked advantage on relatively clean spectra. The effect of the window parameters can be explained by a combination of competing factors: firstly, for the smaller window size, the spectral noise level is higher as the noise is spread over fewer frequency bins, making parameter estimates less reliable. However, at longer window lengths the tracking of the non-stationary partial behaviour is worse, leading to lower performance particularly for the sinusoidal model. In the filtering method the resonance bandwidth is equivalent to 86/21.5 Hz for $N = 2048/8192$, meaning that at the shorter window length a larger bandwidth noise component is being filtered out along with each harmonic, which is obviously more critical at the 0 dB SNR.

Polyphonic signals

It now remains to evaluate the filtering methods described in section 4.6 on polyphonic samples. Of the methods reviewed in section 4.7, three methods have been chosen to compare the filtering algorithms with. These will now be explained in more detail.

The first method performs amplitude interpolation between adjacent harmonics, and this was used in [128] to separate shared spectral peaks. This will be referred to as Parsons' method, bearing in mind though that he used another spectral subtraction method when the partials were reasonably well separated. Eqn. 4.12 was used for interpolating the missing harmonic spectral amplitude between the nearest non-overlapping harmonics of each note on opposite sides. The overlapping spectral peak was then shared accordingly between each of the sources by setting: $H^p[k] = a_m^p / \sum_{q \in Q} a_m^q \quad ; \; k = k_l, \ldots, k_r$. One notices a lack of frequency dependence in $H^p[k]$ in contrast to the three frequency dependent filter designs proposed in section 4.6.

The second method tested was the nonlinear least squares (NLS) method[125, 134]. It is possible to apply this method to the entire original signal containing both overlapping and non-overlapping harmonics, however, this is computationally expensive. Instead, the method was applied separately to the waveform of each set of overlapping harmonics. This was obtained by multiplying $F[k]$ with a spectral mask across the width of each set of overlapping partials, then performing a DFT^{-1} of the masked spectrum, and then dividing by the analysis window function.

The third method, referred to as sinusoidal modelling, is an additive synthesis approach. Overlapping harmonics are separated by interpolating the amplitudes and phases of each harmonic between boundary frames at which the harmonics are resolvable and these parameters can be measured directly. The amplitude and phase interpolation algorithms in eqns. 4.13 and 4.14 were used for this purpose. All separated harmonics, both overlapping and non-overlapping, were generated by additive synthesis from the estimated harmonic parameters.

Non-overlapping partials were separated identically for the filtering methods, Parsons' and the NLS method by using unit amplitude filters across each harmonic (eqn. 4.17). Results will also be shown for which no treatment of overlapping harmonics has been provided, i.e. non-overlapping harmonics are separated using unity-amplitude band-pass filters, but overlapping harmonics are simply left in the mixed spectrum. This is included to show that the treatment of overlapping harmonics has a significant effect on the overall separation performance. The linear equations solution in section 4.7.4 was also experimented with, but the matrix inverse in eqn. 4.41 was found to be numerically unstable when partials were closely spaced, and so the method has not been included in these results.

Average results for each of the methods were measured when separating random mixes of notes, with the degree of polyphony varying from $P = 2$ to 5. The notes were obtained from the University of Iowa, Musical Instrument Samples Database[139], and were all, apart from the piano, recorded in an anechoic chamber and sampled at 44.1 kHz. The set of individual notes contained a total of 479 samples extending in pitch from A0 (27.5 Hz) to C8 (4186 Hz), and covering 11 different orchestral instruments (bassoon, cello, B♭ clarinet, E♭ clarinet, flute, French horn, oboe, piano, saxophone, trombone and violin). Random mixes were constructed by firstly selecting a random but unique set of P instruments, and then selecting a random note for each instrument from within its complete pitch range. The note waveforms were summed with equal RMS values, with onsets starting at roughly the same time, and the rough pitches of each note were refined as described in section 3.3. The average MSRR and average χ/M were then measured over 100 random sample mixes for each degree of polyphony, and are given in table 4.2. As the experiment was conducted in somewhat ideal conditions, i.e. equal RMS amplitudes of each note, fairly static pitches, anechoically recorded samples, similar durations of each note, etc., we can expect that the results are optimistic in comparison to those on real music. This evaluation approach was adopted, firstly, for consistency with other experiments such as [126], and secondly, because it is difficult to define a test set of music samples that is both representative and of a manageable size.

Table 4.2 shows a marked improvement using the frequency dependent filters in comparison to any of the other methods evaluated for separating overlapping harmonics. Improvements of around 2 dB were demonstrated over the frequency-independent filter design (Parsons' method) by using frequency dependent filters, which in turn was about another $3-4$ dB higher than not separating overlapping harmonics at all. The sinusoidal modelling approach employing time-domain interpolation of the harmonic amplitudes and phases does not seem to have provided an adequate treatment of overlapping harmonics. This may be due to the fact that the notes are all relatively stationary, so the duration of the collisions between harmonics is normally large, meaning that the time-domain amplitude and phase interpolations had to be made over many times frames and are hence probably fairly inaccurate. The NLS method shows the lowest performance, which seems to be a result of the method being numerically unstable due to the matrix $B^H B$ in eqn. 4.46 often being close to singular. It was observed that on occasion the NLS method grossly overestimated the harmonic amplitudes, interpreting a sum of two low amplitude harmonics as the addition of two sinusoids of very large amplitude but nearly opposite phase which destructively interfere. Given that the spectrum is unlikely to consist of pure sinusoids, and hence that overlapping spectral peaks are often misshapen from pure sinusoidality, it is possible that

Table 4.2: Average MSRR and average χ/M for polyphonies of 2-5 instruments and various harmonic separation methods. The standard deviation of the average MSRR is given in brackets, and this was virtually identical for the average χ/M values.

Polyphony		*None*	*Pars*	*NLS*	*Sin*	*Filt a*	*Filt b*	*Filt c*
2	**MSRR:**	**16.1 (1.0)**	**18.9 (1.0)**	**10.9 (1.2)**	**14.3 (0.6)**	**21.3 (1.0)**	**19.2 (1.0)**	**21.0 (0.8)**
	χ/M:	16.1	18.9	10.9	14.3	21.3	19.2	21.0
3	**MSRR:**	**9.9 (0.6)**	**14.9 (0.8)**	**4.5 (0.9)**	**11.0 (0.5)**	**17.9 (0.8)**	**16.3 (0.7)**	**16.0 (0.9)**
	χ/M:	13.1	18.2	7.8	14.2	21.1	19.6	19.2
4	**MSRR:**	**7.4 (0.5)**	**11.7 (0.6)**	**1.9 (0.5)**	**8.1 (0.5)**	**13.8 (0.6)**	**13.3 (0.6)**	**12.4 (0.7)**
	χ/M:	12.4	16.7	7.0	13.1	18.9	18.4	17.4
5	**MSRR:**	**5.7 (0.5)**	**9.0 (0.4)**	**0.9 (0.8)**	**7.0 (0.4)**	**10.8 (0.5)**	**13.2 (0.6)**	**10.3 (0.5)**
	χ/M:	12.1	15.4	7.3	13.3	17.2	19.5	16.7

Processing schemes:

None - Overlapping harmonics are removed from separated sounds.

Pars - Parsons' amplitude interpolation method for splitting shared peaks[128].

NLS - Time-domain nonlinear least-squares method[125].

Sin - Sinusoidal model with amplitude and phase interpolation.

Filt a - Eqns. 4.20 and 4.21 used to separate overlapping harmonics.

Filt b - Eqns. 4.23 and 4.21 used to separate overlapping harmonics.

Filt c - Eqn. 4.28 used to separate overlapping harmonics.

very large amplitude sinusoids of nearly opposite phase might destructively interfere and lead to a LSE solution, although this may not actually be a likely solution physically. Apart from this concern, the method involves an optimisation (eqn. 4.46) that makes it computationally slow, and there is no guarantee that the optimisation converges to the global maximum.

Of the three frequency dependent filter designs, the two empirical energy-based filter designs, eqns. 4.20 and 4.23, achieved the highest SRRs. The third filter design, eqn. 4.28, is not as robust and can sometimes produce unrealistically large values of $H^p[k]$. Along with the deterioration of its performance in fig. 4.16 as the parameter error r increases, it seems that this filter design is not robust enough to have any advantage over the other two filtering methods for non-stationary sounds where harmonic frequency and amplitude estimates are normally slightly inaccurate. Overall, not only is the exponential filter design the most predictable with respect to errors in harmonic amplitude and frequency estimates as shown in fig. 4.16, but table 4.2 shows that it also has the best performance of all three filter designs.

The SRRs in table 4.2 for the first filter design are about 7 dB higher than the best average separation results reported in [126]. In [126] a larger selection of 26 different instruments was used although pitches were restricted between 65 and 2100 Hz, and the results reported were the average over clean mixes and mixes with additive -10 dB pink noise. The cases in which the multi-pitch estimator failed were not accounted for in the average separation results in [126]. This may have led to slightly higher results than expected, since note mixes for which the multi-pitch estimation failed are more likely to be harmonically related, and it is for these mixes that there is a higher incidence of overlapping harmonics, which usually leads to lower SRRs. The percentage of samples removed due to errors in multi-pitch estimation was 20%, 35%, 68% and 80% for polyphonies of 2 to 5 respectively.

To complete the results section, an example separation of a real recording is shown in fig. 4.22. The figure shows the spectrogram of a few seconds of a jazz duet recording with trumpet and saxophone parts, and the spectrograms of the separated parts from the mix. The harmonic structures of each note have evidently been disentangled, and one also notices that the separated parts do not contain much of the broadband noise component of the original recording. This component will be dealt with in the following chapter concerning the separation of transient and noise content from polyphonic mixes.

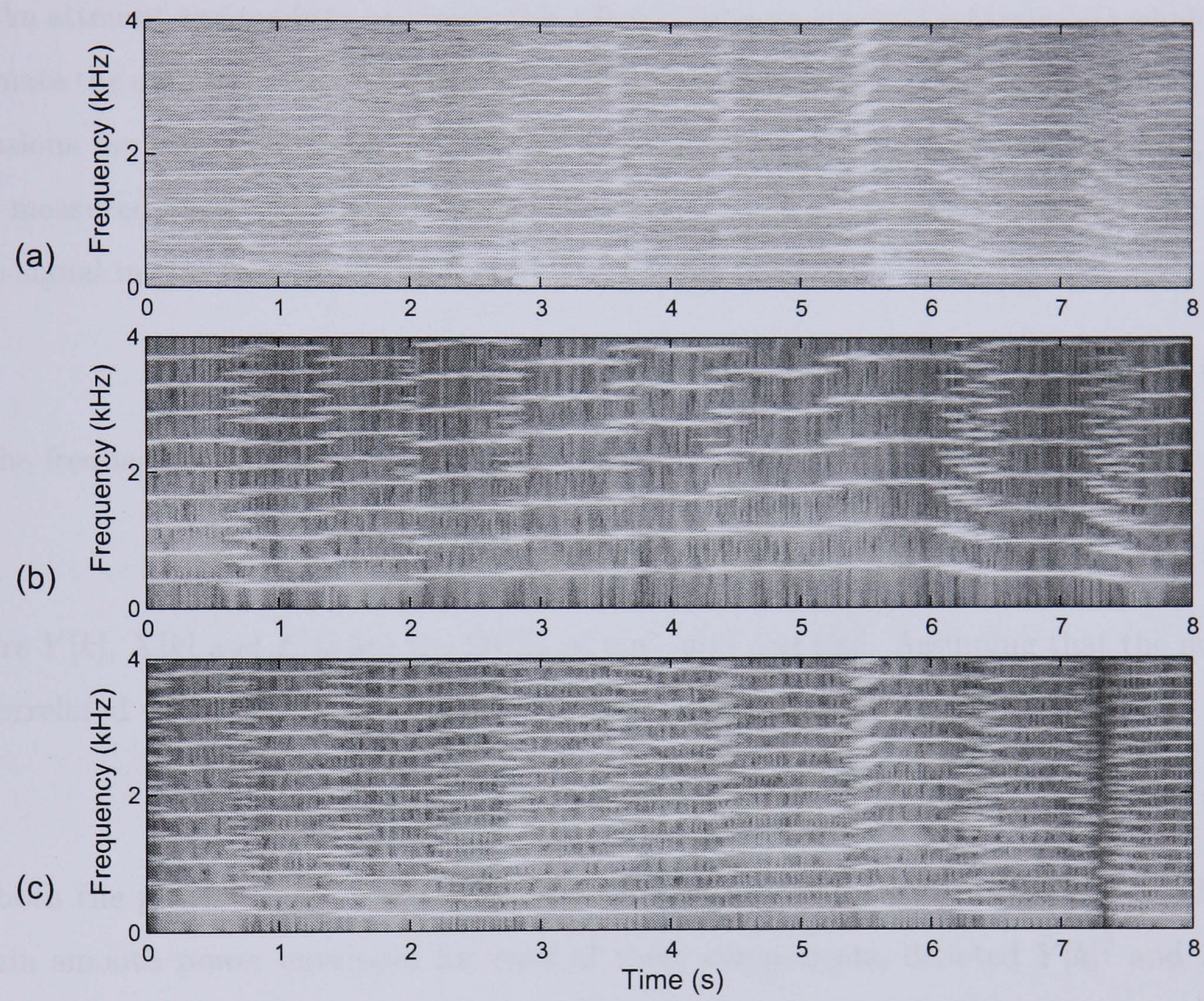

Figure 4.22: The spectrograms of **(a)** an original duet recording (trumpet and sax), **(b)** the separated trumpet, and **(c)** the separated saxophone.

4.9 A spectral subtractive method for suppressing filtered noise

One of the shortcomings of the filtering approach is that any broadband noise overlapping with the harmonic frequencies is filtered out along with the harmonic content. The wider the filter resonances, the more noticeable this effect becomes. Whilst this is not too problematic if the recording contains only pitched instruments, percussive or impulsive sounds characterised by noisy onsets which are spread over a large bandwidth have a tendency to leak into the separated pitch notes.

An attempt was made to minimise this effect by using a spectral subtractive technique to estimate the desired partials in noise. Spectral subtractive methods have been used on many occasions for suppression of uncorrelated noise in speech signals[140, 141, 142, 143, 144]. The measured noisy signal $y[n]$ is considered to arise from a desired signal $x[n]$ which is the note signal in this case, plus a noise component $e[n]$:

$$y[n] = x[n] + e[n]. \tag{4.51}$$

In the frequency-domain we have:

$$Y[k] = X[k] + E[k] \tag{4.52}$$

where $Y[k]$, $X[k]$ and $E[k]$ are the DFTs of $y[n]$, $x[n]$ and $e[n]$. Assuming that the noise is uncorrelated with the desired signal, the powers should be additive, i.e.:

$$|Y[k]|^2 = |X[k]|^2 + |E[k]|^2. \tag{4.53}$$

As both the power envelope of the noise term, and hence the mix, are noisy, we will first obtain smooth power envelopes for each of these components, denoted $\tilde{Y}[k]^2$ and $\tilde{E}[k]^2$. Thus, an estimate of the power of the desired signal can be defined as:

$$\hat{P}_x[k] = \max\left\{0,\ \frac{\tilde{Y}[k]^2 - \tilde{E}[k]^2}{\tilde{Y}[k]^2}\ |Y[k]|^2\right\}. \tag{4.54}$$

Hence an estimate of the desired signal is obtained using:

$$\hat{X}[k] = \sqrt{\hat{P}_x[k]}\ \angle Y[k] \tag{4.55}$$

where $\angle Y[k]$ is the phase spectrum of $Y[k]$. We notice therefore that $\hat{X}[k]$ is in phase with $Y[k]$ and also that $|\hat{X}[k]| \leq |Y[k]|$.

Whilst the smoothed DFT spectrum $\tilde{Y}[k]$ can be calculated directly from $Y[k]$, some way of estimating the smoothed noise spectrum $\tilde{E}[k]$ is required. In speech processing, estimates of the noise power envelope are often obtained from segments of audio in which the speech is absent. However, the noise envelope should not be assumed to be stationary.

especially in musical contexts in which the noise could be a background of impulsive or percussive events, so it is advisable to estimate the noise envelope in every time frame. This actually fits in rather well with the filtering approach previously described, as we can directly measure the noise component at locations between filter resonances where any partial content is sufficiently attenuated. This does assume though that we have identified all possible sources of partial content within the mixed spectrum. $\tilde{E}[k]$ is therefore estimated by smoothing the amplitude spectrum $|Y[k]|$ across frequency bins which do not contain any partial content, i.e. where $H^p[k] = 0 \; \forall \; p$, by calculating the mean in a sliding window centred at k. The result is then linearly interpolated across any regions containing harmonic content, i.e. where $\sum_p H^p[k] > 0$. Moving on to the estimation of $\tilde{Y}[k]$, it was found that if $|Y[k]|$ was smoothed using the mean in a sliding window or by convoluting with a window function, the partial amplitudes in $\tilde{Y}[k]$ were significantly reduced. Instead, $|Y[k]|$ was adaptively smoothed to preserve the shape of the spectrum whilst suppressing noise:

$$\tilde{Y}[k] \;=\; \frac{\sum_{j=k-d}^{k+d} \rho_{j,k} \; |Y[j]|}{\sum_{j=k-d}^{k+d} \rho_{j,k}} \tag{4.56}$$

where

$$\rho_{j,k} \;=\; \exp\left\{-\frac{(|Y[j]| - |Y[k]|)^2}{2 \; \sigma \; \overline{Y}^{\,2}}\right\} \tag{4.57}$$

where $\overline{Y}$ is the mean spectral amplitude introduced to satisfy scaling invariance. σ must be chosen large enough that small variations in $|Y[k]|$ due to noise are characterised by values of $\rho_{j,k}$ close to unity, and small enough that large variations in $|Y[k]|$ at partial locations are characterised by smaller values of $\rho_{j,k}$, leading to less suppression of partial amplitudes. $\sigma = 0.5$ was chosen here. Fig. 4.23 illustrates the spectral subtraction of a flute harmonic in added white noise at a SNR of 0 dB.

Eqn. 4.54 is often modified[145] to include a small noise component according to listening preference, and to reduce an effect known as 'musical noise', which manifests in the subtracted signal as short tonal components at random frequencies and time intervals:

$$\hat{P}_x[k] \;=\; \max\left\{\xi \; |Y[k]|^2, \; \frac{\tilde{Y}[k]^2 - \lambda \; \tilde{E}[k]^2}{\tilde{Y}[k]^2} \; |Y[k]|^2\right\} \tag{4.58}$$

where typical choices of the constants are $\xi = 0.02$, $\lambda = 1.5$.

Some difficulty has been encountered in suppressing noise when $e[n]$ is due to an interfering percussive or impulsive source in the music signal. This is different to most speech applications in which the noise component is usually more slowly time-varying, and is considered to be stationary within a particular time frame. However, if we simply reduce the window size to account for this non-stationarity, the accuracy of the harmonic tracking algorithm and filtering methods decrease. On the whole, the spectral subtraction method seems to be useful for suppressing quasi-stationary background noise being filtered out at

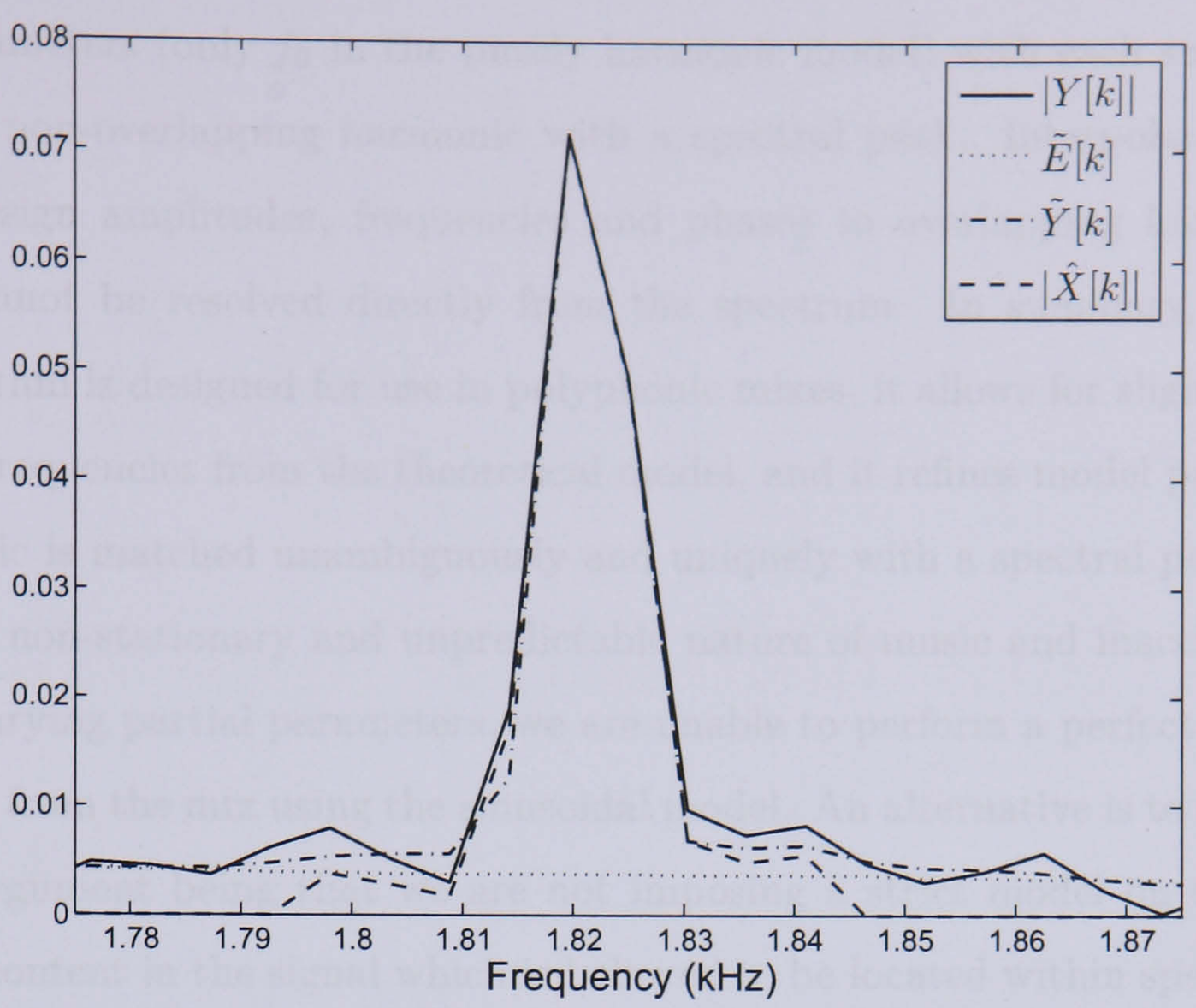

Figure 4.23: Spectral subtraction of a flute harmonic from 0 dB added white noise

the harmonic locations, but has not completely solved the problem of percussive sounds leaking into filtered notes.

4.10 Conclusions

The chapter has focused on separating the harmonic content of each individual pitched note from a note mixture or recording. The main issues have been in designing adaptive methods for tracking harmonic frequencies over time, and treating overlapping partial content from multiple sources. Separation was performed on a frame-by-frame basis by adaptive filtering in the spectral-domain, after which each separated note waveform was obtained by performing a DFT^{-1} of the filtered spectrum, and using an overlap-add procedure to smooth time segments across frame boundaries. The adaptive filters are comb-like filters that align themselves with the harmonic frequencies of each note. The filter resonances are of unity amplitude across the bandwidth of each harmonic, except when harmonics from multiple notes are overlapping. In the overlapping case, the filter resonances are designed according to energy considerations and window characteristics to share the overlapping spectral peak between the constituent notes.

Given the pitch trajectory of each note from section 3.3, its harmonics are tracked over time by matching a harmonic template to a set of spectral peaks in each frame. The matching procedure is designed in such a way that alternative models of partial spacings, such as a model incorporating string stiffness often associated with the piano (eqn. 4.8), can be substituted for the purely harmonic template. The template matching procedure updates

the model parameters (only f_0 in the purely harmonic model) with each successful match of a predicted non-overlapping harmonic with a spectral peak. Interpolation algorithms are used to assign amplitudes, frequencies and phases to overlapping harmonics whose parameters cannot be resolved directly from the spectrum. In summary, the harmonic tracking algorithm is designed for use in polyphonic mixes, it allows for slight deviations of the harmonic frequencies from the theoretical model, and it refines model parameters each time a harmonic is matched unambiguously and uniquely with a spectral peak.

Due to the non-stationary and unpredictable nature of music and inaccuracies in estimating time-varying partial parameters, we are unable to perform a perfect subtraction of partial content from the mix using the sinusoidal model. An alternative is to filter harmonic content, the argument being that we are not imposing a strict model on the signal, but only isolating content in the signal which is believed to be located within specific frequency regions. This does have a down side, however, that any other content also located within this frequency region is also filtered out in the process. An attempt was made to suppress filtered noise using spectral subtraction methods. Measurements on non-overlapping synthetic and real samples showed the relative benefits and ideal operating conditions of the sinusoidal and filtering approaches.

A difficult task in sinusoidal modelling is to separate overlapping partials. Methods such as the NLS method[125] and linear equation solutions[131] attempt to do this, but do not allow for time-varying partial behaviour, which is a crucial element of sinusoidal modelling and integral to the MQ algorithm. Furthermore, these methods are sometimes numerically unstable when partials are closely spaced, and can produce results that whilst being mathematically accurate, are physically unrealistic. The initial filtering solution to this problem proposed sharing the energy in an overlapping spectral peak in a way that reflected the predicted frequencies and amplitudes of each harmonic contained within it. We then continued by incorporating the shape of the window function into the energy distribution task. However, the results have consistently shown that the empirical approach of simply assigning a share of the spectral peak to each harmonic that is proportional to its predicted amplitude, and that decays away from the predicted frequency, is the most accurate.

For the reason that it is difficult to quantify separation performance on real recordings in which the un-mixed instrumental parts are usually not accessible, results have been given for test samples summed manually from real note waveforms. When applied to real recordings in practice, these harmonic extraction methods are synchronised with the refined MIDI data from chapter 3. The end result is a system that extracts the harmonic content associated with each note in the MIDI data from a polyphonic recording.

Chapter 5

Separation of Transient and Noise

Content

> "If you develop an ear for sounds that are musical it is like developing an ego. You begin to refuse sounds that are not musical and that way cut yourself off from a great deal of experience."
>
> **- John Cage (1912–1992)**

An audio signal can be described in general terms as consisting of three additive components: partial content, noise and transient content. As the prime interest has been to separate pitched notes from a polyphonic recording, we equated partial content with a set of harmonics, allowing slight de-tuning from pure harmonicity, and a specific model of inharmonicity in the case of the piano. There is a small discrepancy in that partials that are completely inharmonic, i.e. not at all at integer ratios of the fundamental frequency, can sometimes exist, arising from longitudinal as opposed to transversal vibrations of a string for example, and these would be unaccounted for. The energy in completely inharmonic partials is, however, usually relatively small for the majority of pitched instruments encountered in music. So, that aside, the focus of this chapter is the processing of the transient and noise components.

The developments leading to the amalgamation of transient and noise content with sinusoidal modelling of audio are well known. As discussed in section 2.2.2, the noise model was originally amalgamated with the sinusoidal model[65, 66] to represent non-partial audio content, as it is highly inefficient to model non-partial content as a sum of time-varying sinusoids. Later, various models of transient content were introduced (section 2.2.3) to improve the modelling of sharp attacks, as there is a loss of clarity and naturalness in modelling impulsive content as filtered noise.

Whilst there have been many accounts of applying sinusoidal modelling or harmonic

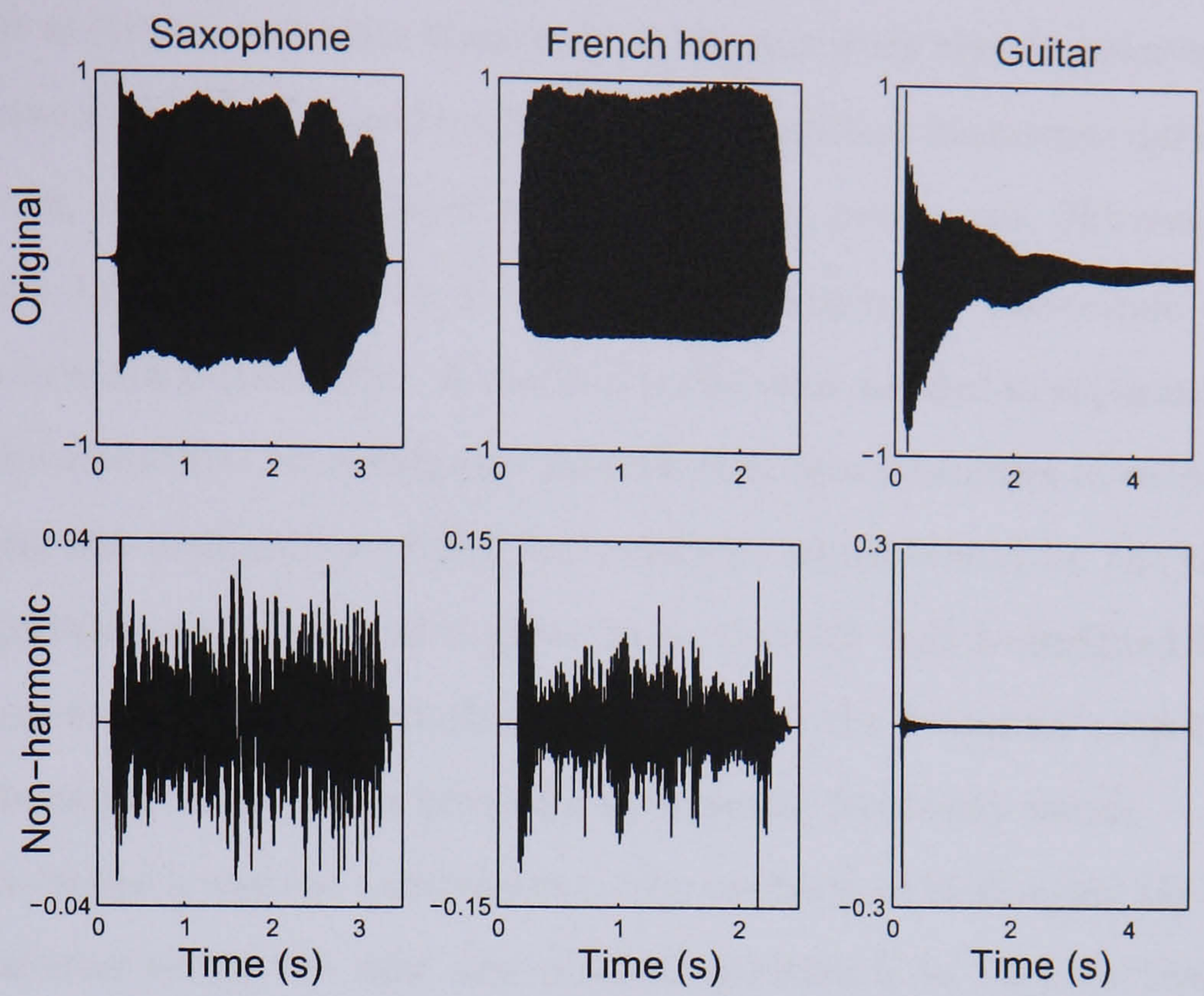

Figure 5.1: The original waveforms and non-harmonic components of a saxophone, French horn and acoustic guitar note. The guitar transient is very sharp, whereas in the saxophone it is hardly noticeable in comparison to its noise component. The French horn note contains a transient component that is easily noticeable but not as impulsive as that of the guitar.

filtering to separating the harmonic content of overlapping sources from polyphonic mixes, as discussed in chapter 4, the problem of separating overlapping transient or noise content has remained relatively unexplored. In [81] a statistical-based approach to transcription and separation of a limited set of percussion instruments from monophonic percussive mixtures was described. This combined the separation capabilities of independent subspace analysis with binary time-frequency masking (section 2.2.5). A difficulty in separating overlapping transient content is that it is highly non-stationary and specific to instrument type, as seen in the example notes in fig. 5.1. At the start of an excitation an instrument can exhibit highly nonlinear behaviour before stable vibrational modes are established. Whereas simple 1-dimensional physical equations provide a rough approximation to the harmonic series in the stationary region of most pitched instrument sounds, the propagation of excitation energy through the instrument body and 3-dimensional physical nature of the source is fully embodied during a transient excitation. Noise content also arises from complex nonlinear processes such as turbulent air flow and friction.

Another difficulty in separating overlapping noise content is the fact that, while one can rely on the principle of common harmonicity or Gestalt grouping cues to identify the partial content of a particular source, there does not seem to be an obvious way of associating a band of noise, for example, with a particular source. The transient component, however, does normally occur at the note onset. Thus, if the note onsets in a recording are reasonably

well separated, it is easy to associate transients in the mix with their respective sources. An approach is therefore described in section 5.1 whereby transient events are initially separated from the recording, and then associated with the nearest note onset. However, if the notes are arranged such that the transient attacks are overlapping, a one-to-one association of transient events to notes is infeasible. A method is therefore needed to separate the transient events, and in this situation, knowledge of the transient characteristics of each source would be useful. Failing the availability of this information, which would be the case in all but a few specific applications, a method is given in section 5.2 that is designed for separating overlapping transient events. The method assumes that the transient attack of each note is a uniformly decaying noise power envelope in separate frequency bands.

Moving on from the transient component to the problem of identifying the noise content of a particular source within the mix, one possible solution is to consider that the partials and noise component of a note are correlated in some way. This is not entirely unrealistic, as if the frequency or amplitude of a partial has a small random variation over time, a correlated noise component is produced. This is demonstrated in fig. 5.2, which shows the DFT amplitude spectrum of a clean 10 kHz sinusoid, $|F[k]|$, and the noise spectrum, $|F_n[k]-F[k]|$, produced by adding a small random variation to the frequency of the sinusoid, where $F_n[k]$ is the amplitude spectrum of the time-varying sinusoid. A similar correlation between the noise and sinusoid exists when the amplitude is fluctuating and the frequency is constant. Without going into detail concerning the form of this correlation yet, as the partial content of a note is usually capable of being isolated from the mix, it is possible that this correlation could aid the separation of the overlapping noise content of each note within the mix. A simple implementation of this idea using a linear correlation was given in [146], and this will be extended in section 5.3 using spectral envelope estimates related to linear predictive coding (LPC).

5.1 Separating transient content

Providing that the note onsets are reasonably well separated (say $>$ 100 ms apart), and assuming that the transient component of each note is located at the note attack, a simply one-to-one association between transients and notes can be made. The term 'transient event' will be used to refer to a time-localised transient component attributable to one or more sources. What is therefore needed is a method for extracting transient events from the recording, and if an event arises from a single source, the subsequent association of each transient event to a note is made if the two entities have similar onset times. The note onset times are already available from the MIDI-to-audio aligned note timing information (section 3.2).

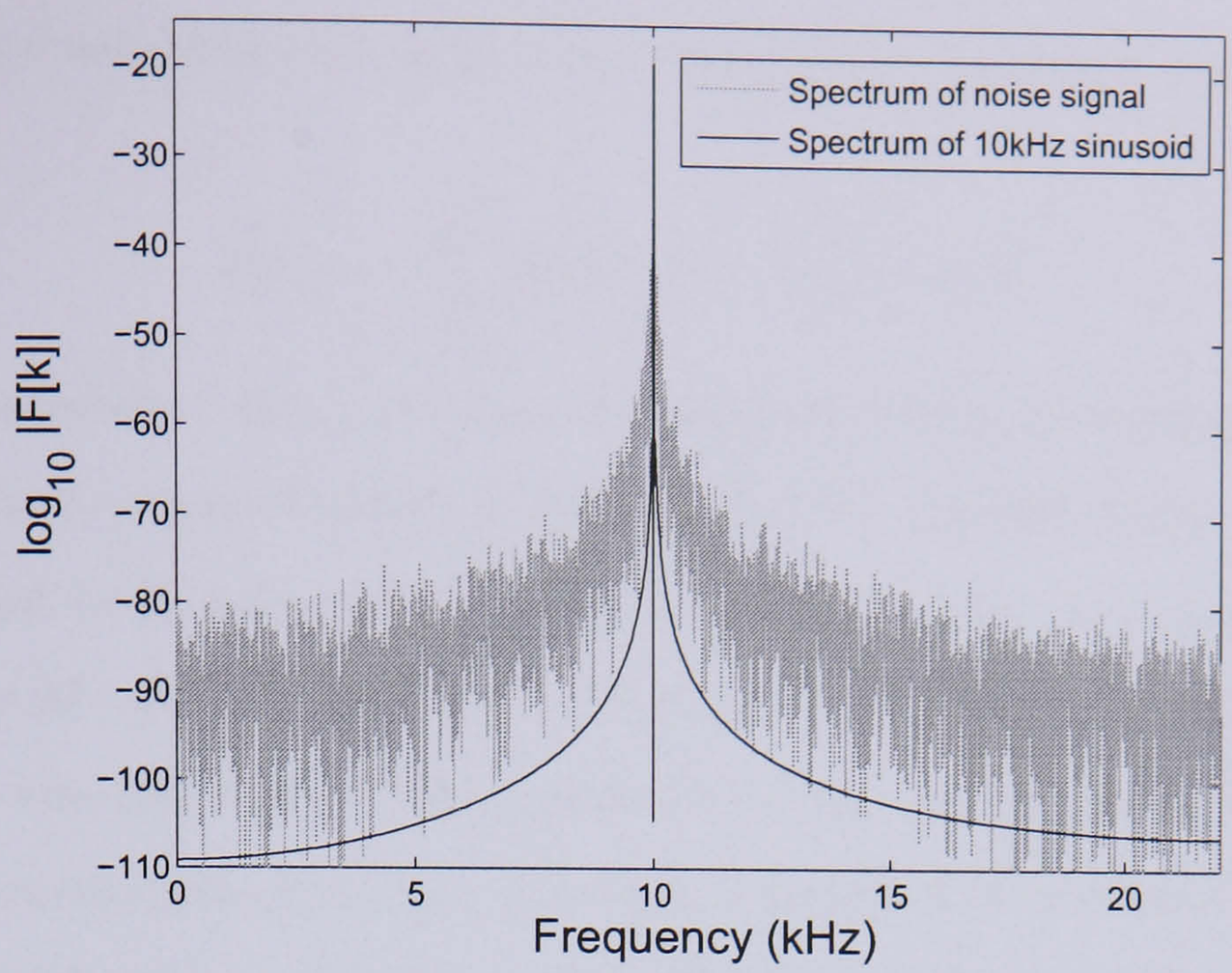

Figure 5.2: DFT amplitude spectrum of a 10 kHz sinusoid, and the DFT amplitude spectrum of the noise signal, which is the difference between the spectrum of the stationary sinusoid and that of a 10 kHz sinusoid with randomly fluctuating frequency. The figure clearly demonstrates that the noise signal is concentrated in the same frequency region as the sinusoid.

The specification that the transient content in the recording consists of time-localised transient events means that a segmentation of the signal into sections containing transient events and sections without transient events must be made. Various transient extraction methods such as [29, 68, 67, 71, 69] provide a segmentation of the signal[68], are able to incorporate information regarding the location of transient events[67], or model the signal as a set of atoms[68, 69] or time-frequency cells[71] at a specific time location. Other methods are more suited to extracting transient[70] or non-steady state[72] content that is spread out in time, i.e. without clearly delineated boundaries. As transient events are short and time-localised, and correspondingly spread out in the spectral-domain, it was decided to use a time-domain transient extraction method. The method described in section 5.1.1 is based upon an autoregressive (AR) model of the time-domain signal, and begins by determining the transient event onset and offset times.

5.1.1 Autoregressive model-based method

A transient event is characterised by a rapid change in dynamics, non-stationary and unpredictable time-domain behaviour. It is neither an ideal impulse in the time-domain, nor does it have any visible partial content. The fact that its unpredictability results in a bad fit to a time-domain model is precisely the basis of the following transient extraction method.

We begin by fitting an autoregressive (AR) model to the signal $x[n]$ within short time frames overlapping by 50%. An AR process is one in which the current data point can

be predicted from a weighted sum of previous data points, incurring a small random error term $e[n]$:

$$x[n] \;=\; \sum_{m=1}^{M} a[m]\; x[n-m] \;+\; e[n]. \tag{5.1}$$

The set of AR coefficients, $a[m]$, can be calculated efficiently by solving the Yule-Walker equations using the Levinson-Durbin recursion[147, 148]. An AR model order of $M = 128$ or larger was found to be adequate, which is much larger than commonly used in speech analysis (typically M=10–20). However, in speech analysis the AR model is usually used for modelling the smooth shape of the spectrum or formant structure, whereas here it is being used as a time-domain prediction function. The width of each time frame was chosen to be twice the model order ($\simeq$ 6 ms for a 44.1 kHz sampling rate). It is easy to see that the error signal $e[n]$ is the deviation between the measured data $x[n]$ and the forward predicted data $\hat{x}[n]$:

$$\hat{x}(n) \;=\; \sum_{m=1}^{M} a(m)\; x(n-m). \tag{5.2}$$

Suppose we denote the variance of $e[n]$ within a window of length L as $s[n]$:

$$\begin{aligned} s[n] \;&\simeq\; \frac{1}{L-1} \sum_{m=0}^{L-1} e[n-m]^2 \\ &=\; \frac{1}{L-1} \sum_{m=0}^{L-1} \left(x[n-m] - \hat{x}[n-m]\right)^2. \end{aligned} \tag{5.3}$$

When the data is not well modelled as an AR process, the error variance is obviously large. At the onset of a transient event, which it has been argued is unpredictable in the time-domain, we therefore expect a sharp increase in the error variance. Hence, it makes sense to use the error variance as the basis of a transient event detection function. It would not be wise to simply apply a constant threshold to the error variance as a means of detecting transients. Due to the fact that the AR model provides a poor fit to noise, this would result in audio segments containing large stationary noise components being detected as transients, when they should ideally be treated as noise. Instead, an adaptive threshold, $\alpha\; s_\lambda[n]$, is used as a transient event detection function. $s_\lambda[n]$ is the output of a median filter of length λ with input $s[n]$, where $s_\lambda[n]$ is centred on $s[n]$, α is a constant threshold height, and λ has been chosen to be equivalent to 1 s. As $s[n]$ is measured only every M samples since the windows are of length $2M$ and are overlapping by 50%, then $\lambda = f_s/\text{sampling frequency of } s[n] = f_s/M$. The adaptive threshold can be seen in fig. 5.3.

Transient events are detected by performing peak-picking using eqn. 4.2 on the error variance envelope $\tilde{s}[n]$ in all regions where $\tilde{s}[n] > \alpha\; s_\lambda[n]$. Using the envelope of the error variance helps to avoid spurious detections, and is obtained by filtering $s[n]$ with a second

order IIR low-pass filter[149, pp. 43]:

$$\tilde{s}[n] \;=\; b_0 s[n] + b_1 s[n-1] + b_2 s[n-2] - a_1 \tilde{s}[n-1] - a_2 \tilde{s}[n-2]. \tag{5.4}$$

The cutoff frequency $f_c = 20$ Hz, $\epsilon = \tan(\pi f_c / f_s)$, and the filter coefficients are given by:

$$\begin{aligned} b_0 &= \frac{\epsilon^2}{1+\sqrt{2}\epsilon+\epsilon^2} \\ b_1 &= 2b_0 \\ b_2 &= b_0 \\ a_1 &= \frac{2(\epsilon^2-1)}{1+\sqrt{2}\epsilon+\epsilon^2} \\ a_2 &= \frac{1-\sqrt{2}\epsilon+\epsilon^2}{1+\sqrt{2}\epsilon+\epsilon^2} \end{aligned} \quad . \tag{5.5}$$

During peak-picking, any peak above the threshold that is within 50 ms of a larger peak is rejected, thereby enforcing a minimum distance between transient events. The reason for this choice is that the method requires a reasonably stationary region of around this length between transients to form a time-domain AR interpolation across the transient events. The transient onset time, n_p^i, where p is the transient event index, is then set as the first minimum in $\tilde{s}[n]$ immediately preceding the peak location or the point where the threshold is exceeded, whichever is later. The method is fairly sensitive to the value determined for n_p^i, and performs best when n_p^i occurs < 10 ms before the transient attack. The end of each transient event n_p^f is set to whichever is the earlier of (i) the first minimum immediately following the peak, (ii) the first location at which $\tilde{s}[n] < \alpha\ s_\lambda[n]$, and (iii) the location at which a preset maximum transient event duration of 100 ms is exceeded. Fig. 5.3 shows $\tilde{s}(n)$ and the transient detection function $\alpha\ s_\lambda(n)$ for a percussive audio sample, with the detected transient onset times also shown.

Once the boundaries of each transient event n^i and n^f have been estimated, where p has been dropped for convenience, the non-transient component between the boundaries can be estimated as a weighted sum of the forward and backward AR predictions from either side of the transient event. We define the forward and backward predictions for $n^i \leq n \leq n^f$ as:

$$\hat{x}_f[n] \;=\; \sum_{m=1}^{\min(M,\ n-n^i)} a_f[m]\ \hat{x}_f[n-m] + \sum_{m=n-n^i+1}^{M} a_f[m]\ x[n-m] \tag{5.6}$$

$$\hat{x}_b[n] \;=\; \sum_{m=1}^{\min(M,\ n^f-n)} a_b[m]\ \hat{x}_b[n+m] + \sum_{m=n^f-n+1}^{M} a_b[m]\ x[n+m] \tag{5.7}$$

where $a_f[m]$ are the AR coefficients calculated over a segment of the signal immediately preceding the event boundary: $n^i - L \leq n \leq n^i - 1$, and $a_b[m]$ are the AR coefficients computed from the time reversed signal immediately following the event boundary: $n^f + 1 \leq n \leq n^f + L$. The Burg method[150, Chapter 7] was used to calculate the AR coefficients as, although it is computationally slower than the Yule-Walker method, it is more accurate at extrapolating the signal outside of the analysis window.

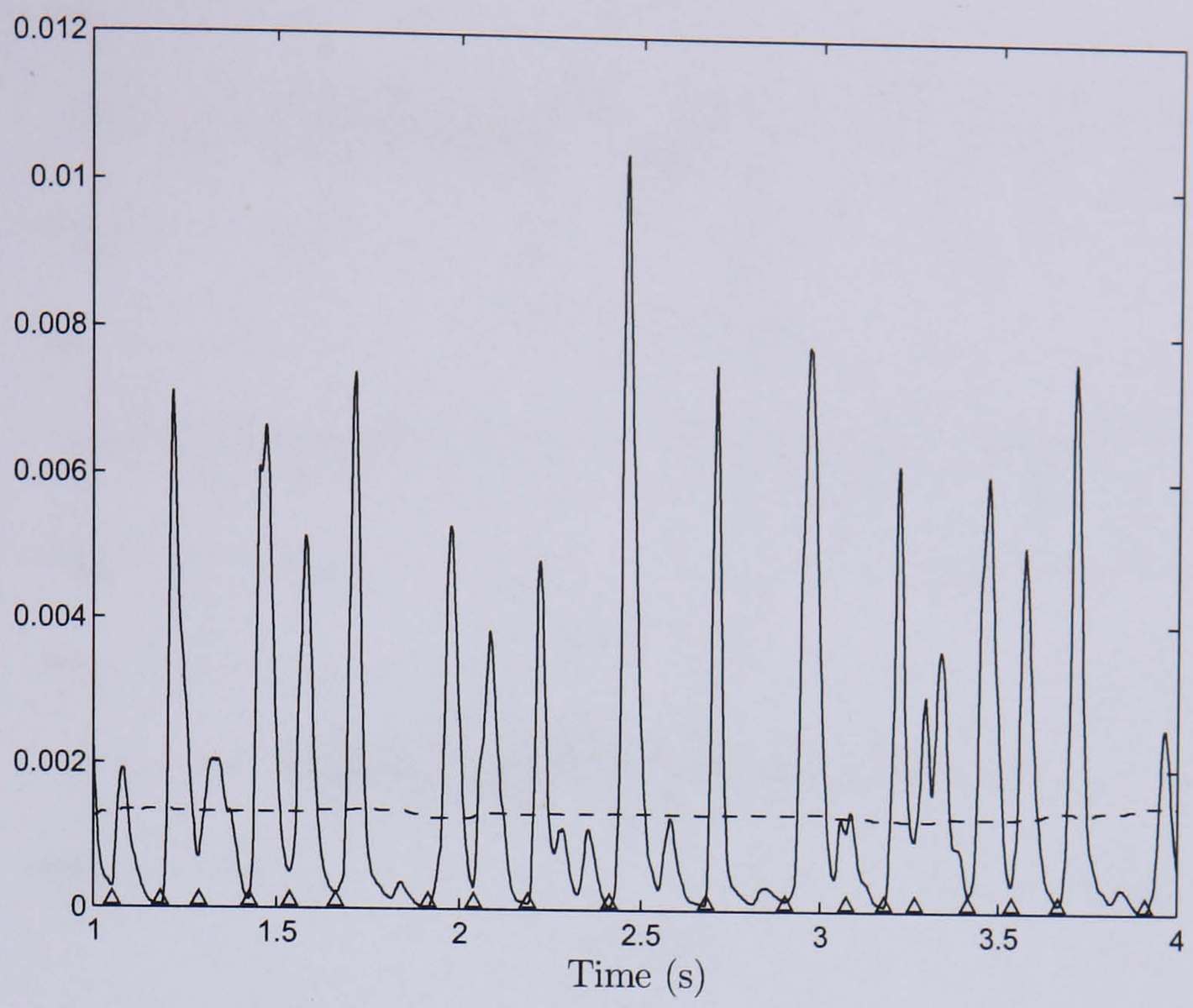

Figure 5.3: The error variance envelope $\tilde{s}[n]$ (solid), transient detection function $\alpha\ s_\lambda(n)$ (dashed), and transient event onset times n_p^i (triangles) for a short sample from a percussive recording ($\alpha = 2$).

Although a smaller AR model order was used for calculating the original detection function, it was found that using a AR model of $M = 1024$ improves the extrapolations of the AR models across the transient event. In fact, the model order is chosen as: $M = \min(1024, \Delta T/2)$, where $\Delta T = n_p^i - n_{p-1}^f$ for the forward prediction and $\Delta T = n_{p+1}^i - n_p^f$ for the backward prediction. As the window size is still chosen as $L = 2M$, this ensures that the AR coefficients for both extrapolations are calculated over relatively stationary regions between transient events. In practice, as a minimum distance between peak locations was set to 50 ms, this is large enough that usually $L = 2M = 2048$ (46 ms).

The weighted sum of the forward and backward predictions gives the non-transient estimate between the transient event boundaries:

$$x_{nt}[n] \;=\; \frac{\beta^{-1}(n^f - n)\ \hat{x}_f[n] \;+\; \beta(n - n^i)\ \hat{x}_b[n]}{\beta^{-1}(n^f - n) \;+\; \beta(n - n^i)}. \tag{5.8}$$

$\beta > 1$ favours the backwards prediction over the forward prediction, and vice versa for $\beta < 1$. Eqn. 5.8 is a weighted sum of $\hat{x}_f[n]$ and $\hat{x}_b[n]$, with a controllable fade in/out. Since transient events typically have a sharp attack followed by a slower decay, the backwards AR prediction is generally more accurate, and so $\beta = 2$ has been used. Finally, the transient event $x_t[n]$ is obtained by subtracting the non-transient prediction $x_{nt}[n]$ from $x[n]$ for $n^i \leq n \leq n^f$.

Fig. 5.4 shows the separated transient and non-transient components from a short segment of audio during which a single transient event occurs. Fig. 5.5 similarly shows the

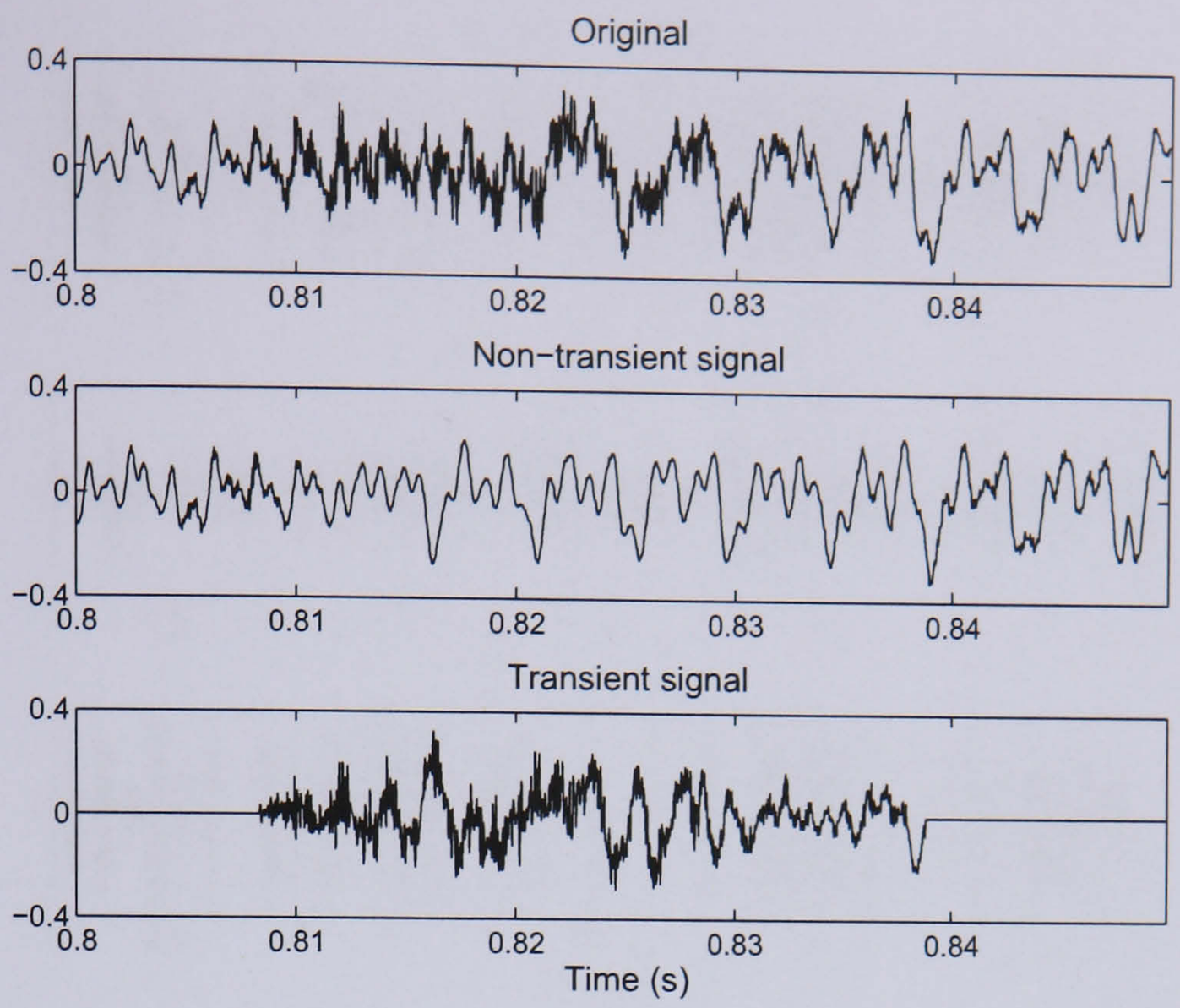

Figure 5.4: The separation of a transient event from a short segment of audio.

non-transient and transient components for a short section of a recording containing various percussive instruments. It was found that the quality of the transient event separation method was better for percussive or impulsive onsets than tonal onsets characterised by a sudden change in timbre. Without actually conducting formal listening tests, our impression was that for percussive-like attacks, such as produced by the piano, the separated transients retained the perceptual characteristics of the note attacks of the original recording. A simple musical effect controlling the level of 'staccato' in a recording can be implemented by adding a scaled transient component to the non-transient component.

As mentioned previously, each transient event is potentially matched with a nearby known note onset time. The transient event is then simply added to the extracted harmonic and/or noise content of the note with which the match has occurred. The situation in which the transient event is actually the result of several notes whose transient attacks overlap in time will be discussed in the following section.

5.2 Bandwise power envelope interpolation

The task of separating overlapping transients is hardly straightforward, and especially difficult when no detailed physical or prior information is available that would facilitate a specific model of each transient event. It is conceivable that a transient model could be constructed from other non-overlapping notes for each instrument in the recording, that is, assuming that the transients note attacks from a particular source all have a similar form. We leave this as a potential further study, and instead develop a generic method

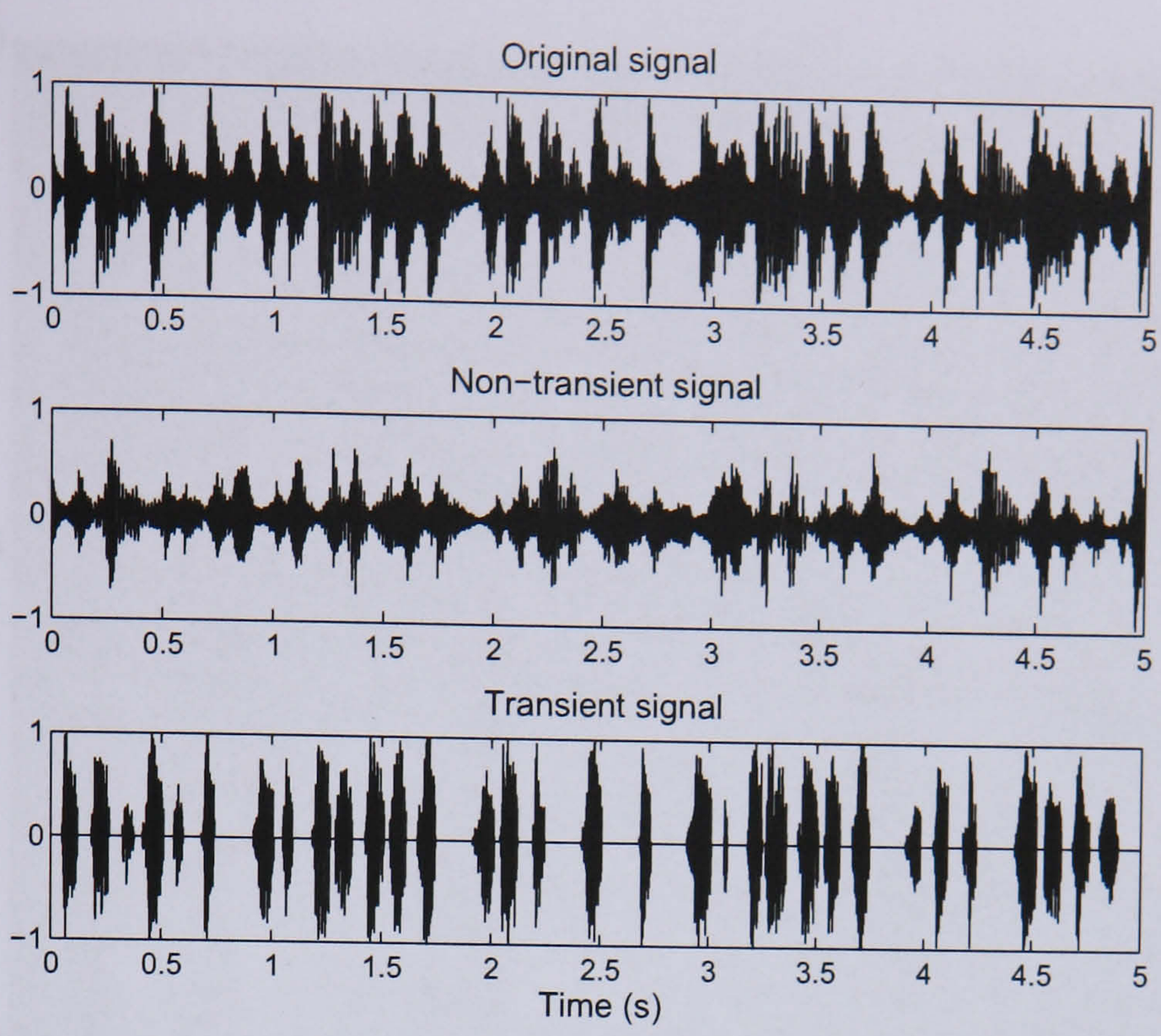

Figure 5.5: A short section of a recording containing percussive sounds, and its separation into non-transient and transient components.

capable of producing a rough separation of overlapping transients, that requires little prior information.

Each transient event is modelled as a set of band-passed noise power envelopes over time. This is similar to the residual classification used in [151, 13], which used a set of time-varying energy envelopes with a frequency distribution in terms of equivalent rectangular bandwidths (ERBs) of the human auditory system. The basis of the method is fairly simple: we assume that in every frequency band, the power of each transient event is in a state of uniform decay a short time after the initial excitation and before the onset of the following transient. Thus, we have made it a requirement that a short delay exists between onsets. It is now possible to interpolate the power envelope of each event across the duration of the overlap with subsequent events. Using these interpolated envelopes, one can share the noise in any band between all overlapping notes based upon energy considerations. The method will now be described in more detail, and is also found in [7].

5.2.1 Distribution of frequency bands

It can be observed in the spectrograms of figs. 5.6 and 5.7 that transient events are characterised by a sudden increase in broadband energy. This is perhaps a more accurate description of the percussive attacks in fig. 5.7 than the piano onsets in fig. 5.6, but this general behaviour is observed in both types of onsets. The broadband energy is evidently frequency dependent, but not highly concentrated in frequency, although there does seem to be a higher density of energy around the partial locations in the piano onsets. Another

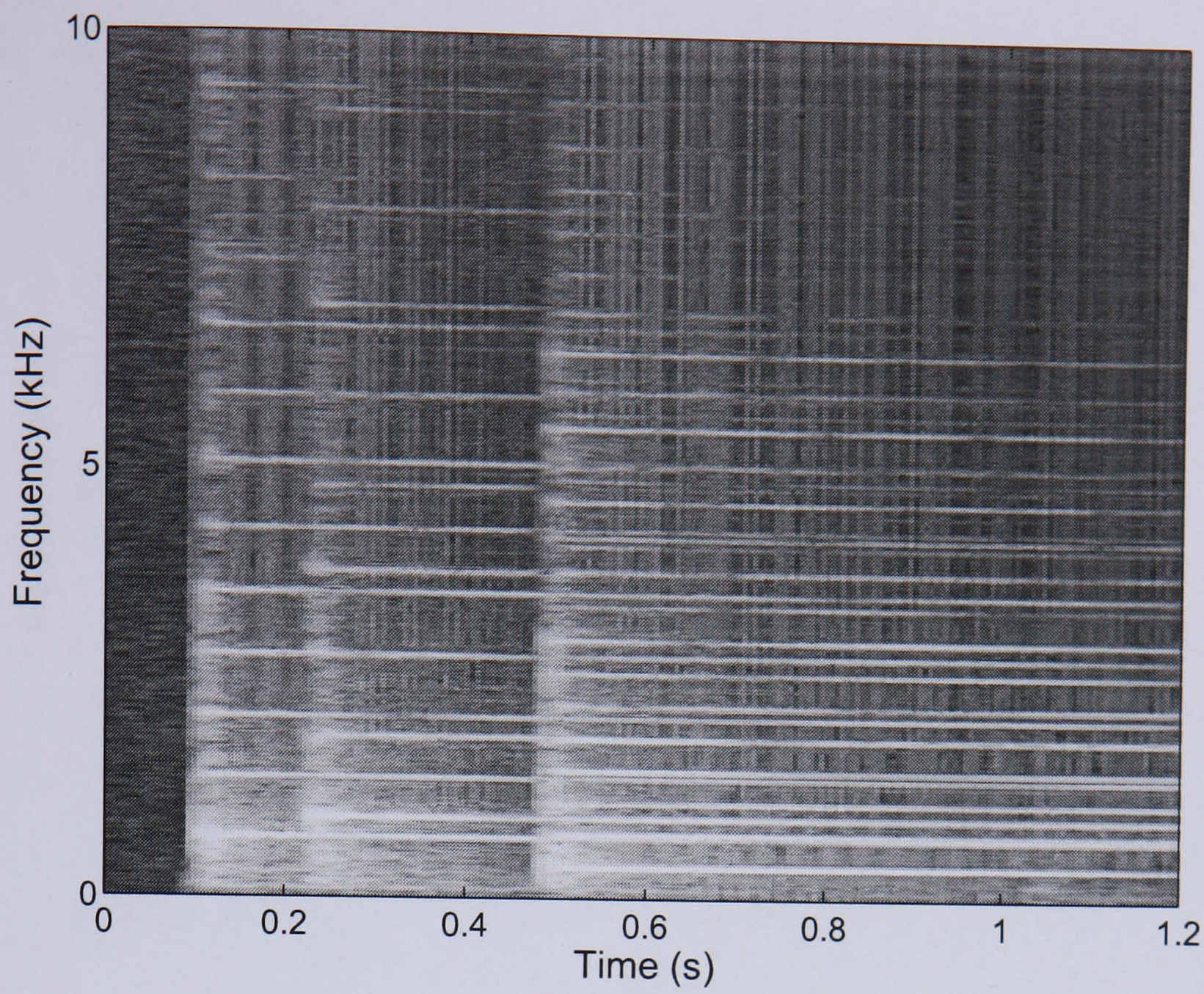

Figure 5.6: The spectrogram of a mix of three piano notes (N=2048)

thing to notice is that, especially for the drum solo, the rate of energy decay after the onset varies with frequency. As this can be observed in many different instruments, there is sufficient justification for bandwise processing of the signal, where the power envelopes in separate bands have different rates of decay.

We recall that the energy in a sum of random zero-mean distributed noise sources is equal to the sum of the energies of the unmixed sources. Thus, within a particular frequency band, and under the assumption that each transient is a random process, the sum of the noise power envelopes of a set of unmixed transient events should sum to the measured noise power envelope of the mix of these sounds. Thus, if the power envelope of the first transient is interpolated across the duration of the overlap with the second transient, then the envelope of the second transient can be estimated by simply subtracting the first envelope from the mixed envelope. Fig. 5.8 depicts the noise power envelope in a particular band b, denoted $E_b[r]$, for a sum of two transient events with a short delay between the events. The onset times r^1_{on} and r^2_{on}, and offset times r^1_{off} and r^2_{off} are shown.

The precise distribution of the frequency bands must be decided. As the power envelope measures the total power within a particular band, if the bandwidth is too small, the envelope is more random or noisy. This affects the accuracy of the envelope interpolation illustrated in fig. 5.8, which depends on having accurate starting and end points. However, if the bandwidth is too large, a blurring of the frequency dependent nature of the sound occurs. Furthermore, if the signal is processed in time frames rather than sample by sample, the time

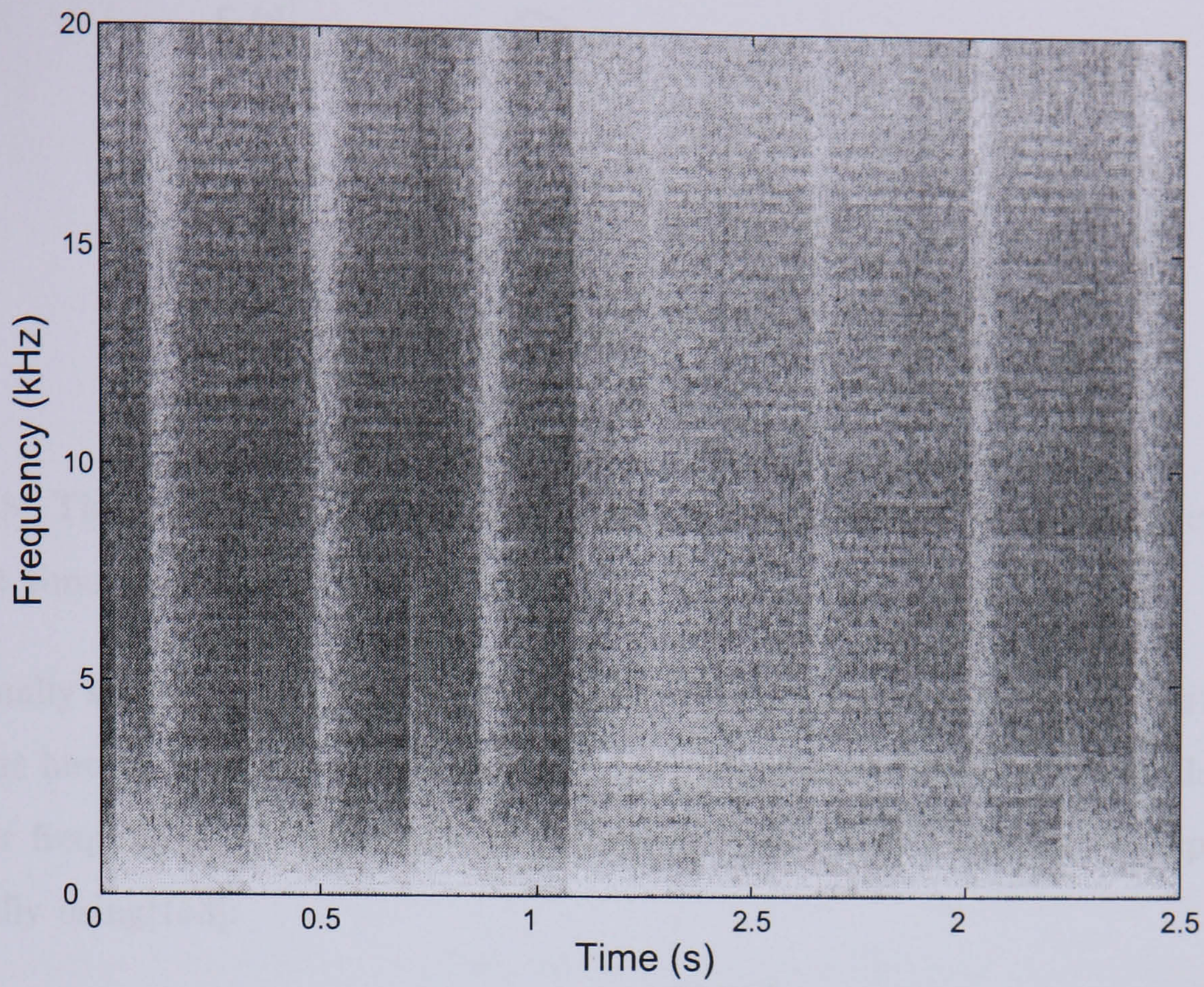

Figure 5.7: The spectrogram of a short excerpt from a jazz drum solo (N=512)

resolution should be sufficient so as not to miss rapid changes in the envelope occurring at attacks. From the perspective of maximising perceptual relevance of the division of the frequency axis, given that critical bands in the human hearing system are roughly equally spaced on a logarithmic scale, it may be advantageous to use a constant-Q filter bank. This provides good time resolution and large bandwidths at higher frequencies, and better frequency resolution at lower frequencies. Alternatively, a frequency division in terms of ERBs[151, 13] of the human auditory system could be used. It was decided to compare three different representations: the STFT followed by processing in 24 Bark bands, the discrete wavelet transform (DWT), and the wavelet packet transform (WPT). It was expected that by using these three complementary analysis methods, it would be possible to ascertain whether a Fourier or wavelet-based representation is more appropriate for the separation task, and whether an adaptive resolution method (WPT) would lead to a closer correspondence of the division of the frequency axis with the actual spectral shape of the noise, and ultimately lead to a better separation of the overlapping noise envelopes from the different sources.

Analysis method (i): Processing in Bark bands

We begin by calculating the discrete STFT (eqn. 2.5) of the mix, where a suitable value of the window length is 512 samples (11.6 ms) with a hop size of 128 samples (2.9 ms) between frames. The positive frequency axis was then split into $B = 24$ non-overlapping

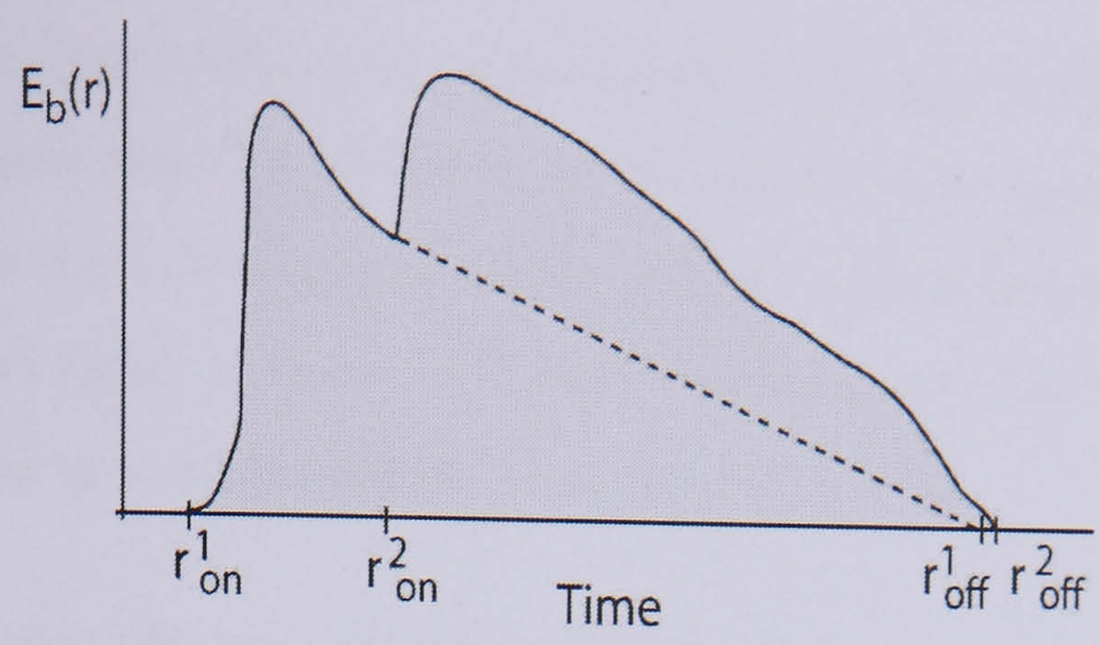

Figure 5.8: The noise power envelope of a sum of two impulsive sounds with event onset and offset times indicated.

bands equally spaced on the Bark scale[152, 153]. The Bark scale measures the critical band rate in the human auditory system, which is roughly linear below 500 Hz and logarithmic for higher frequencies. A conversion from frequency in Hz to Bark can be approximated analytically using[153]:

$$\begin{aligned} z &= \frac{26.81 f}{1960 + f} - 0.53 \\ \text{if } (z < 2): \; z &= z + 0.15\,(2 - z) \\ \text{if } (z > 20.1): \; z &= z + 0.22\,(z - 20.1) \end{aligned} \tag{5.9}$$

where f and z are the frequencies in Hz and Bark respectively.

Let $S(k, r)$ be the complex value of the k^{th} frequency bin of the STFT computed in the r^{th} time frame. By summing the STFT energy at all bins within each band, the power envelope in the b^{th} Bark band becomes:

$$E_b[r] = \sum_{k=k^b_{min}}^{k^b_{max}} |S(k, r)|^2 \tag{5.10}$$

where k^b_{min}, k^b_{max} are the lower and upper frequency bins for Bark band b. The envelope $E_b[r]$ can still be rather noisy, and so it was smoothed by convoluting with a Hamming window of length 10 ms, resulting in the smoothed envelope $\tilde{E}_b[r]$.

Analysis method (ii): Discrete Wavelet Transform

The discrete wavelet transform (DWT) was described in detail in section 2.1.4. It is calculated here using the pyramidal filter bank structure shown in fig. 2.6, leading to a dyadic sampling of the time-frequency plane, as illustrated in fig. 2.4b. A maximum depth of $d = 6$, this being the number of low-pass filtering plus down-sampling operations needed to go from the signal to the deepest level of the tree structure, was used with the Daubechies-6 ('db–6') wavelet (note that the number '6' in 'db–6' is the order of the wavelet, which is related to its regularity and it is unrelated to d). We recall that the DWT calculates a

single set of approximation coefficients which encode the largest scale signal features, and d sets of detail coefficients which encode successively smaller scale signal features. The signal is effectively split into $d+1$ overlapping bands, with d of these bands being band-pass and logarithmically spaced apart in terms of centre frequency, and the remaining band is the low-pass filtered signal or approximation.

Analysis method (iii): Wavelet Packet Transform

Wavelet packet analysis was discussed in section 2.1.5. The wavelet packet transform (WPT) is a decomposition into a signal-dependent best basis, leading to an irregular sampling of the time-frequency plane. It was implemented using a two-channel sub-band filter bank structure as shown in fig. 2.9. As the full WPT is followed by pruning to remove tree nodes that are suboptimal in terms of encoding the signal (according to the Shannon entropy criterion in eqns. 2.42 and 2.43), the resulting number of frequency bands, B, is signal dependent. Again the 'db–6' wavelet was used, and the depth of the full WPT tree was $d = 6$.

A formulation of the noise power envelope similar to eqn. 5.10 is now required for the DWT and WPT. In both methods the original signal can be considered as a sum of approximation and detail signals at various scales. This argument was developed in the continuous time case leading to eqn. 2.33, and also follows naturally from the quadrature mirror filter bank interpretations of the DWT and WPT. The nodes in the filter bank tree can be labelled with increasing centre frequency corresponding to bands $b = 1, \ldots, B$. Then the power in the frequency band $E_b[r]$ as a function of the translation index, r, is simply obtained from the approximation or detail coefficients within this band, denoted $c_b[r]$:

$$E_b[r] \;=\; |c_b[r]|^2. \tag{5.11}$$

As both analysis methods involve a sequence of j down-sampling operations by a factor 2, where j is the depth within the tree structure ($j = 0$ identifies the original signal), the coefficients $c_b[r]$ actually correspond to time translations of $r\ 2^j/f_s$. Thus, each band has a different time resolution, and the location of r^p_{on} at scale j becomes $2^{-j}\ r^p_{on}$. As in the Bark method, to obtain a smoother envelope for interpolation, $E_b[r]$ was convolved with a Hamming window of length 23 ms ($0.023 * f_s/2^j$ samples), resulting in the smoothed power envelope $\tilde{E}_b[r]$.

5.2.2 Envelope Interpolation

The method assumes that the onset times of all transient events are known beforehand, either from a prior onset detection stage, or from the aligned MIDI transcription of the

recording. Whilst it is advantageous to use predetermined offset times, these are not essential for the interpolation method. If this information is not available, the offset times of all events can be set to the time at which the mixed envelope falls beneath some predefined threshold. The starting point for interpolating the power envelope across a region of overlap, as shown in fig. 5.8, is chosen to immediately precede the onset of the subsequent event. The ending point of the interpolation is either the predetermined offset time or location where the envelope falls below the minimum threshold.

Let us first consider the case of two overlapping events. The power envelope of the first event, which must be interpolated between $(r^1_{on},\ \tilde{E}_b[r^1_{on}])$ and $(r^1_{off},\ \tilde{E}_b[r^1_{off}])$, will be denoted $E^1_b[r]$. There are a number of options for the particular interpolation method, and it was found that on the whole, linear interpolation of the logarithm of the envelope performed fairly well. This is consistent with findings in [29] that transient attacks are, after an initial edge extraction, succinctly encoded in frequency bands using exponentially decaying noise envelopes. In [29] the frequency bands were distributed according to a wavelet packet tree structure based upon the critical band structure of the human hearing system. It must be ensured that when using logarithmic interpolation, the amplitude of the end point of the interpolation, $\tilde{E}_b[r^p_{off}]$, is greater than zero.

Since the power envelopes of the events are assumed to be additive in each band, $E^1_b[r]$ should not exceed the power envelope of the mix, and is thus limited according to:

$$E^1_b[r] \ = \ \min(\tilde{E}_b[r], E^1_b[r]). \tag{5.12}$$

It then follows that the estimate of the remaining power in band b, $E^{rem}_b[r]$, which is attributed to the second event, is:

$$E^2_b[r] \ = \ E^{rem}_b[r] \ = \ \tilde{E}_b[r] - E^1_b[r]. \tag{5.13}$$

Fig. 5.9 shows the interpolated power envelopes for a mix of two transient events from a piano and guitar note whose onsets are separated by 50 ms. We now define two weighting functions:

$$w^1_b[r] \ = \ \sqrt{E^1_b[r]/\tilde{E}_b[r]} \quad \text{and} \quad w^2_b[r] \ = \ \sqrt{E^2_b[r]/\tilde{E}_b[r]} \tag{5.14}$$

such that $(w^1_b[r])^2$ and $(w^2_b[r])^2$, estimate the proportion of energy in band b contributed by the respective events as a function of time. Due to eqn. 5.12, both weighting functions have a range $[0, 1]$.

In a mix of several transient events, it is always possible to find at least one event whose power envelope can be interpolated across an overlapping section, unless two or more onsets occur at exactly the same time. Let $E^1_b[r]$ be this interpolated event, which is subtracted from $\tilde{E}_b[r]$ to yield $E^{rem}_b[r] = \tilde{E}_b[r] - E^1_b[r]$. The next event envelope, $E^2_b[r]$,

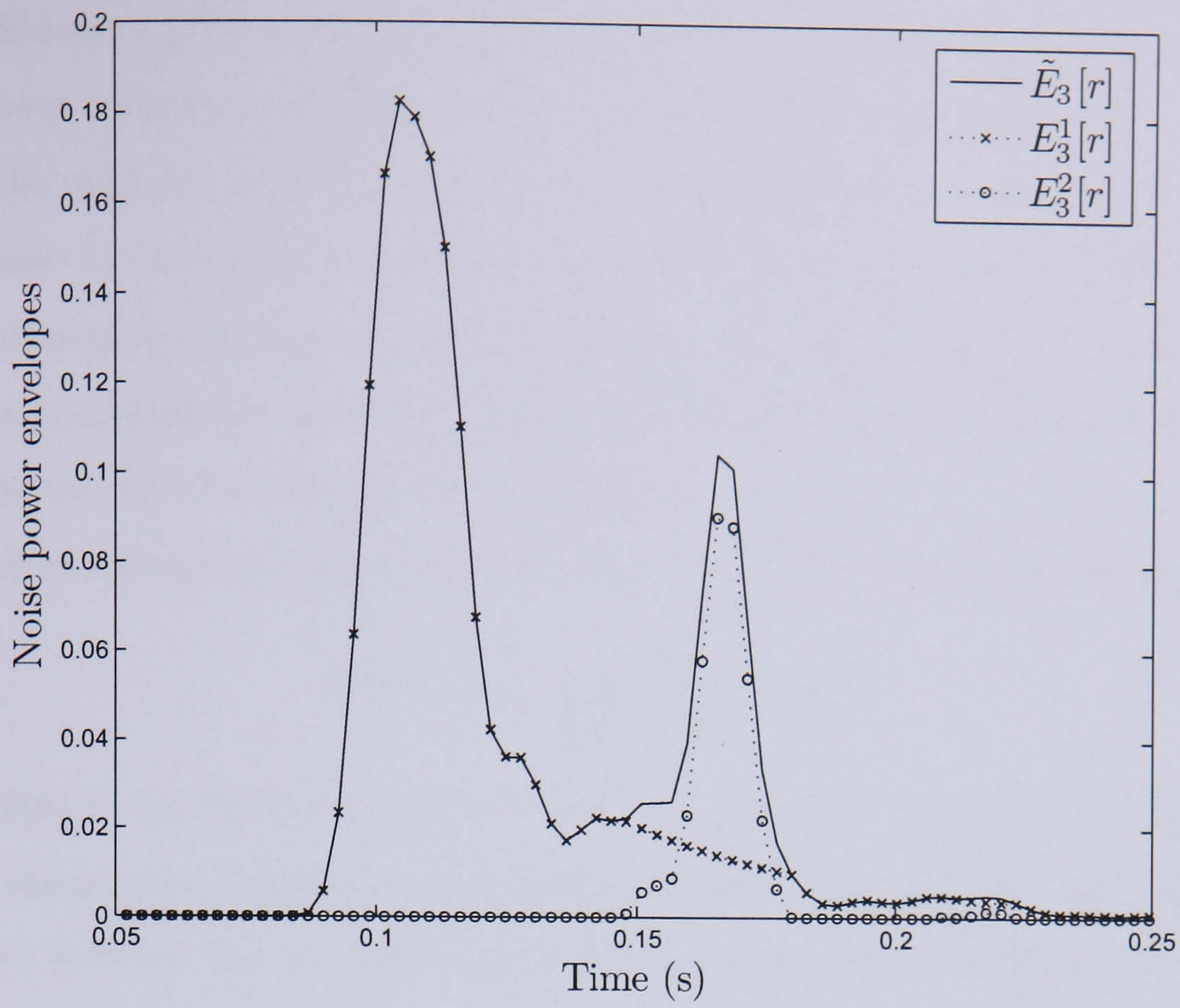

Figure 5.9: The smoothed power envelope in the 3^{rd} Bark band, $\tilde{E}_3[r]$, of a mix of the transients of a piano and guitar note whose onsets are separated by 50 ms. $E_3^1[r]$ and $E_3^2[r]$ are the interpolated power envelopes of the individual attacks using eqns. 5.12 and 5.13.

is then interpolated and subtracted from the residual envelope $E_b^{rem}[r]$. Likewise, all other event envelopes can be obtained in an iterative subtractive process from the residual of the previous iteration. The final residual is assigned to the last event, $E_b^P[r]$. A shortcoming of the method is that the envelopes determined for each event are affected by the order in which successive events are extracted from the mix. This effect can be substantial if the shape of $\tilde{E}_b[r]$ is irregular, and the method could perhaps be improved by using a joint, rather than iterative, estimation of the power envelopes. A joint estimation method is, however, not straightforward given that in general the regions for interpolation differ from note to note.

Once the envelopes of all events have been interpolated, the weighting function for the p^{th} event is defined as:

$$w_b^p[r] = \sqrt{\frac{E_b^p[r]}{E_b[r]}} \tag{5.15}$$

where $(w_b^p[r])^2$ estimates the proportion of energy in band b contributed by the p^{th} event.

5.2.3 Re-synthesis

The calculation of the energy envelope in eqn. 5.10 for the Bark method, and eqn. 5.11 for the two wavelet analyses, involves in both cases a modulus squaring operation. This results in a loss of both the STFT phase information for the Bark method, and the sign

of the coefficients $c_b[r]$ for the wavelet methods. As we actually wish to subtract each transient event from the mix rather than simply synthesising separate events that sound similar to the originals, it is essential to use the original phase or sign information. This is the purpose for which the weighting functions in eqn. 5.15 were constructed. Rather than re-synthesising the separate sources directly from the interpolated power envelopes, the weighting functions are multiplied by the original STFT or wavelet coefficients, yielding a phase coherent set of coefficients for re-synthesis.

For analysis method (i), the extracted STFT coefficients of event p are given as:

$$S^p(k,r) \;=\; \frac{w_b^p[r]}{\sum_{q=1}^{P} w_b^q[r]}\, S(k,r) \tag{5.16}$$

where $b \equiv b(k)$ is the Bark frequency band in which bin k is situated. $S^p(k,r)$ is thus in phase with the original complex spectrum $S(k,r)$. The set of weighting functions effectively share $S(k,r)$ between the P sources, in a way that reflects the estimated distribution of energy amongst the individual events. The normalisation also ensures that the STFT coefficients are split completely between all transient events without any residual, i.e.:

$$\sum_{p=1}^{P} S^p(k,r) \;=\; S(k,r). \tag{5.17}$$

The method would therefore have to be adapted slightly if the mix contained other content apart from the P transient events, such as background noise. The p^{th} transient event is then easily re-synthesised using an overlap-add method as discussed in section 2.1.1. For the wavelet analysis methods, the estimated wavelet coefficients for source p are similarly:

$$c_b^p[r] \;=\; \frac{w_b^p[r]}{\sum_{q=1}^{P} w_b^q[r]}\, c_b[r] \tag{5.18}$$

where again, the normalisation ensures that:

$$\sum_{p=1}^{P} c_b^p[r] \;=\; c_b[r]. \tag{5.19}$$

The p^{th} transient event is re-synthesised from $c_b^p[r]$ using either the inverse discrete wavelet transform (DWT^{-1}) illustrated in fig. 2.7 for analysis method (ii), or the inverse wavelet packet transform (WPT^{-1}) shown in fig. 2.10 for analysis method (iii). These are both easily implemented as a sequence of up-sampling and low-pass/high-pass filtering operations using quadrature mirror filters.

5.2.4 Results

The transient event separation method was firstly tested on three pairs and one mix of three percussive sounds. The audio results for this test are available on the internet[137].

Table 5.1: Mean signal-to-residual ratios (MSRRs) for 4 percussive sample mixes as a function of the analysis method and time between consecutive onsets (δT)

mix	**δT (ms)**	**MSRR**		
		(i) Bark	(ii) DWT	(iii) WPT
1	50	3.1	3.7	3.7
	100	17.4	19.8	19.8
	200	25.6	24.8	24.8
2	50	11.7	11.2	12.6
	100	17.2	23.5	23.8
	200	41.2	43.0	40.7
3	50	3.2	10.9	9.0
	100	12.8	10.9	10.3
	300	21.9	22.2	22.8
4	50	6.8	9.2	9.0
	100	11.1	10.8	10.7
	200	14.5	14.6	14.9

As the method is designed for sounds without any partial/sinusoidal content, any stable partial content in the original percussive sounds was removed beforehand using a sinusoidal extraction method[66]. The MSRRs (eqn. 4.49) for these sample mixes are given in table 5.1 using each of the three analysis methods, where the delay between consecutive event onsets was varied between 50 ms and 300 ms.

A similar comparison was made for overlapping transient onsets from pitched notes. Four individual notes having a perceivable transient attack, from the piano, cello, guitar and violin respectively, were analysed in this test. The harmonic content of each note was first removed by spectral filtering, with the result that the residual contained a strong transient component at the note attack, typical of each instrument's excitation characteristics. Each transient attack was then delayed and added to itself, and then the transient separation method was applied to the resulting mix. The onset times of both events were provided, the duration of the first event was given as 0.5 s, and the duration of the second event was unspecified. The MSRR was again evaluated for the three analysis methods with delays of 50 and 100 ms between the onsets. For comparison, the MSRR was also evaluated in table 5.2 for the case called 'no interpolation': this means that the original mix was simply partitioned immediately preceding the onset of the second event into two regions, which were assigned accordingly to the first and second separated transient event.

Tables 5.1 and 5.2 indicate similar performance for all three analysis methods. Phase

Table 5.2: Mean signal-to-residual ratios (MSRRs) for pairs of transient events extracted from pitched notes as a function of the analysis method and time between consecutive onsets (δT)

mix	δ**T (ms)**	**MSRR**			
		no interp.	(i) Bark	(ii) DWT	(iii) WPT
piano	50	10.0	9.0	8.5	8.7
C5	100	13.7	13.5	13.3	13.3
cello	50	2.9	3.8	4.0	4.0
C5	100	11.2	10.2	10.2	10.2
guitar	50	3.9	5.4	5.4	5.1
E4	100	7.3	8.2	8.1	8.0
violin	50	0.9	2.2	2.4	2.2
D♭4	100	5.1	7.3	7.8	7.4

artifacts typical of STFT processing can be heard in the separated sources when using the first analysis method, but on the other hand, in analysis methods (ii) and (iii) some 'granular' artifacts typical of processing with wavelets can be heard. The choice between the analysis methods therefore becomes a matter of personal preference.

As expected, the MSRR results decrease as the time between onsets decreases. This is partly due to the fact that relatively more content becomes overlapping, but also the assumption that each event is in a state of uniform decay across the region of interpolation becomes less likely as the notes become closer together. If an event has not yet reached a decay state before the onset of the next event, then the interpolated amplitude of its envelope across the overlapping region could be unrealistically small. For this reason, and as the transient attacks of the guitar and piano notes are relatively short in comparison to the violin and cello notes, the MSRRs are slightly better for piano and guitar at smaller values of δT in table 5.2.

On the whole, when attacks are separated by a relatively short time interval (say $50-100$ ms) resulting in a large proportion of energy in the overlapping region, MSRRs of between 2 and 24 dB were achieved. This is only a minor improvement in MSRR, if at all, in comparison to the examples in which no interpolation was performed in table 5.2. Whilst this is somewhat of a discouraging result, it is quite clearly not an accurate reflection of the perceptual quality of the separation judged by informal listening. The separated transients achieved by a crude partition of the original mix sound very abrupt and unnatural due to the discontinuity at the start of the second event. We find that the interpolated separated events have a slower and more realistic energy decay. Furthermore, the SRR is only a

suitable measure of separation performance when the original and separated waveform for each source are in phase. Eqns. 5.16 and 5.18 force the separated waveforms to be in phase with the original mix, but not necessarily with their corresponding original unmixed components.

5.3 Connecting noise envelopes to harmonic spectra

Up until now, the chapter has focused mainly on separating transient content from polyphonic mixes. In this section, we attempt to separate the noise component of a note from a mix of notes. The task faces several difficulties: firstly, it is not possible to make direct measurements of the noise content of a particular note, as it can only be observed mixed with the noise components of other notes. Whilst this is also true with regards to harmonic and transient content, at least the harmonic series of a note can be distinguished from other harmonic series in the mixed spectrum, and the transient events of different notes may be separated in time, meaning that measurements can be made on the mix that are, for all intents and purposes, direct measurements of the individual sources. However, the noise content of a note is neither well-localised in time nor in frequency. Another issue is that when subtracting one noise signal from another, the result is in general a larger noise signal, not a smaller one. Thus, when subtracting the noise component of a note from the noise component of the mix, the two signals must be in phase. This can be achieved similarly to the way in which weighting functions were used in section 5.2.3 to force phase coherence when sharing the transient component of the mix between overlapping notes. A third difficulty is that the noise component of a note is a non-stationary stochastic process, thus, the noise component at the start of the note is unlikely to be of the same loudness or spectral shape as at the end of the note. Whilst it may be possible in some circumstances to construct a prior noise model for a particular instrument from isolated notes, and later apply it to separating the instrument from a mixture, we have deliberately taken an approach to the separation problem that is not reliant on instrument-specific models. Even so, it might still be impractical to construct a noise model for a single instrument, as the noise component may be dependent on the exact note played and gesture used to play it.

This section investigates whether it might be useful to introduce a dependency between the harmonic and noise components of a note as a means to separating its noise component from a mixture of overlapping notes. We assume that the harmonic component of each note has already been separated using the methods described in chapter 4. If the notes contain sharp attacks, it is also beneficial to separate these attacks from the residual using either of the two transient separation methods. First, we list some reasons why a correlation between the harmonic and noise components could exist:

- In the source-filter instrument model[154], which is often used to describe the human voice, the sound is considered to be the output of a time-varying filter (e.g. vocal tract and lips) with an input that is the excitation signal (e.g. glottal impulse train for vowels, or noise source for fricatives and plosives). Assuming the source-filter model is an accurate description of the sound production mechanism, as both the harmonic and noise components of an excitation are subjected to the same time-varying filtering, there should be a correlated frequency shaping of the two components over time. We might also expect to see resonances in both the harmonic and noise spectra at the resonances or poles of the filter, which is evident in figs. 5.10 and 5.11 with respect to the human voice, although the harmonic and noise spectra in the initial excitation signal are not necessarily similar in shape to start with. In fig. 5.10, we can make out that the louder harmonics are numbers $1-3$, 6, $8-9$, $11-14$, which occur at frequencies where the residual/noise component is also loud, shown in fig. 5.11. For the general model of many acoustic instruments known as an exciter-resonator system[154], where the transfer of energy between the two blocks can be in both directions, the sound production mechanism is more complex. However, we can still expect the harmonic and noise components to be simultaneously shaped by the resonances of the sound radiating mechanism, which may or may not be time-varying.

- If the harmonics exhibit random frequency or amplitude modulations, which will collectively be referred to as 'microfluctuations', a noise component is introduced that is correlated with each harmonic. This was demonstrated in fig. 5.2, and was also observed in [155], where the noise component was modelled as a $1/f$-like process centred on the harmonic frequencies.

- If the sinusoidal subtraction or filtering of harmonics is imperfect, a difference signal will remain in the residual near the harmonic locations, which will be interpreted as noise. This component of the noise signal is definitely correlated with the harmonics.

- General observations of the spectrograms of various acoustic instruments confirm that noise and harmonic content tend to be concentrated in roughly the same frequency regions, and that individual harmonics often have spectra that are more spread out and noisy than could have arisen from a stationary sinusoid.

What is now required is to find a general model specifying how the harmonic and noise components should be correlated. One option is to follow a similar reasoning to [155] and assume that the noise decays at a rate $1/|f - f_m|$ in the power spectrum surrounding the harmonic frequency f_m. Whilst it would be worthwhile to evaluate this method for separating noise in the immediate vicinity of harmonics, this model is perhaps non-ideal

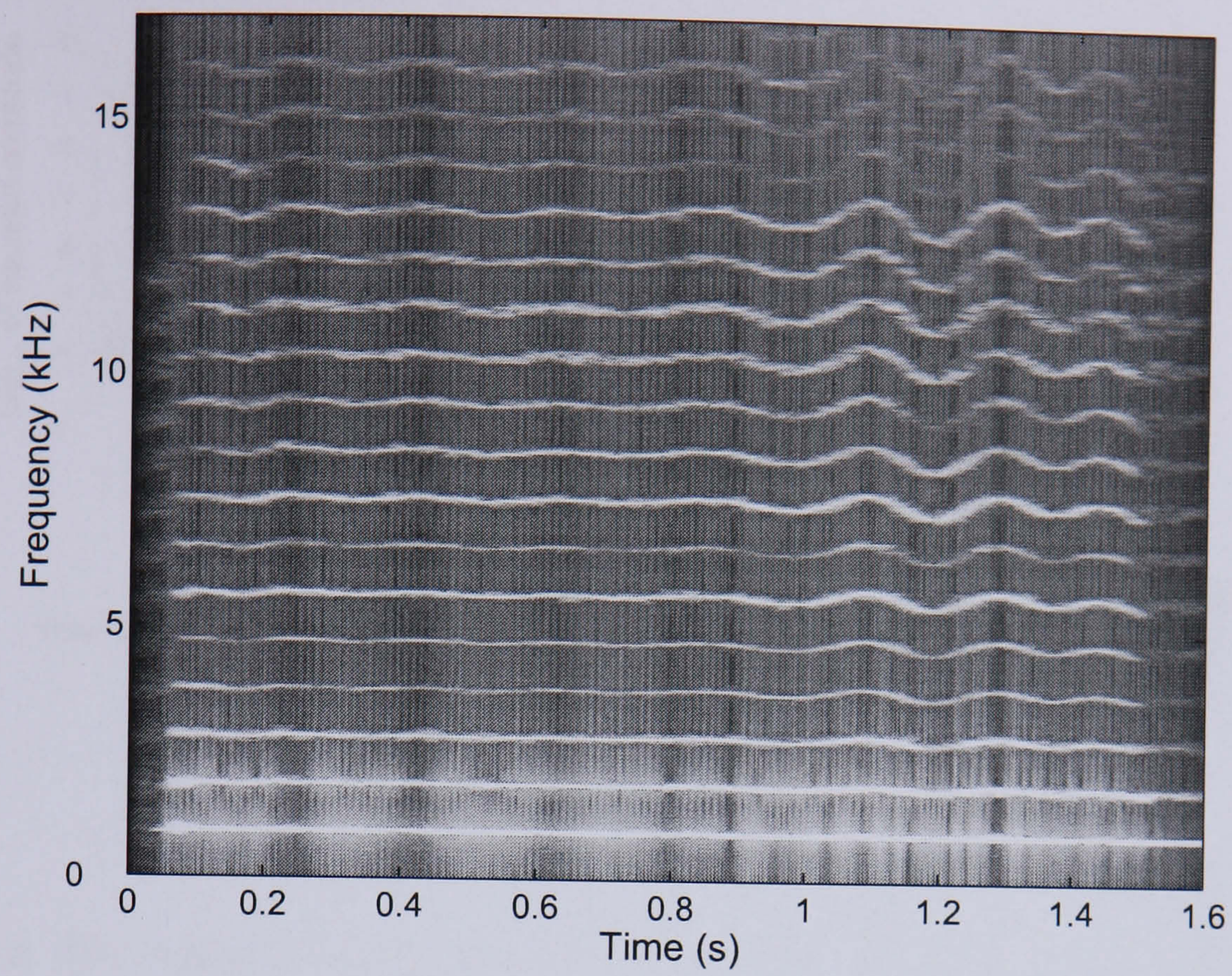

Figure 5.10: Spectrogram of the harmonic component of a female voice singing 'laa'.

Figure 5.11: Spectrogram of the residual component of the female voice in fig. 5.10.

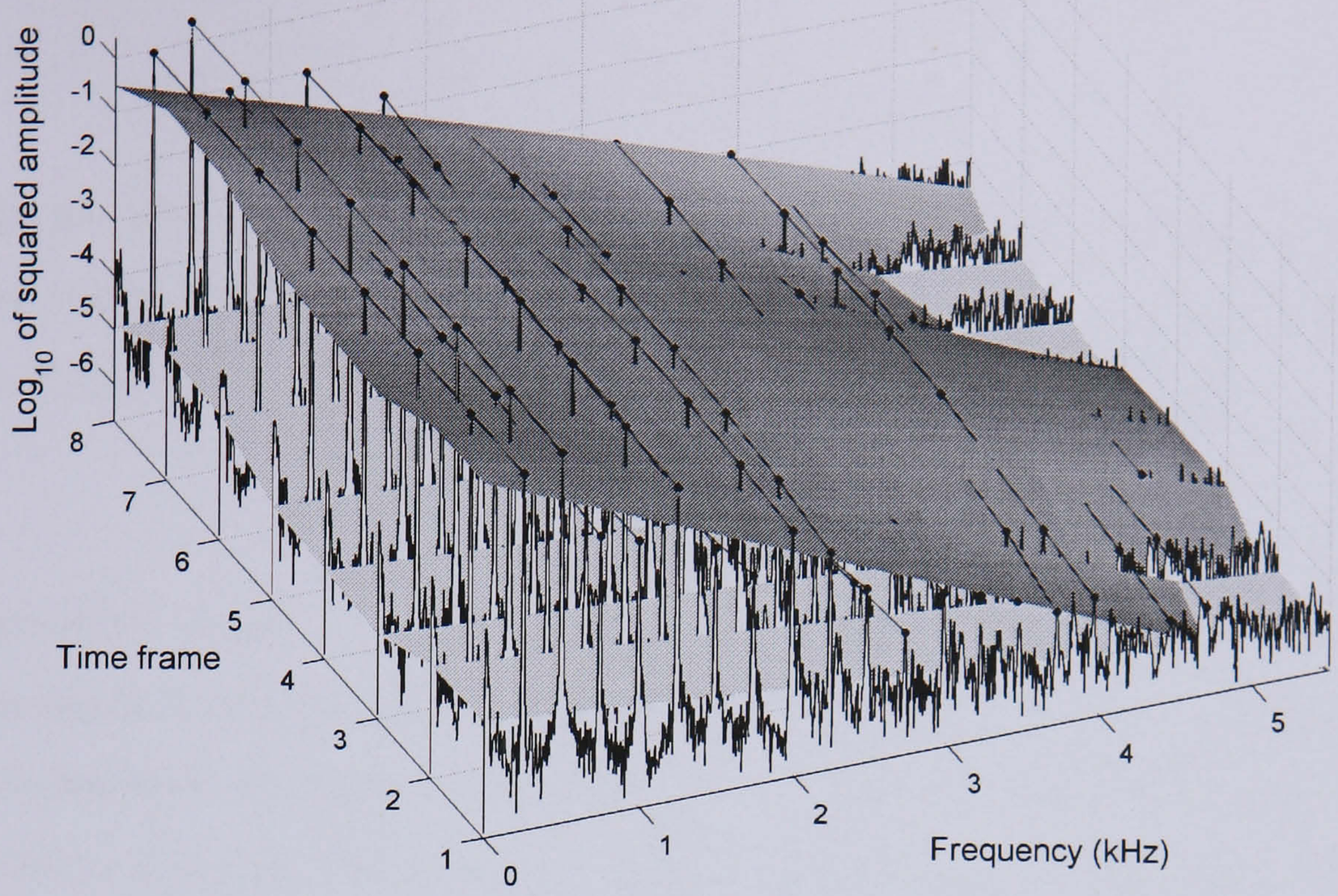

Figure 5.12: The logarithm of the squared STFT amplitude of a single note in a sequence of time frames, with harmonic trajectories also shown. The upper plane is a straight line fit to the harmonic amplitudes, and the lower plane is a straight line fit to the noise power envelope.

for separating frequency regions of the noise which are far from any detected harmonics. In [156] residual noise content was incorporated into the framework of the sinusoidal model by using 'bandwidth-enhanced oscillators', that is, partials with adjustable bandwidths allowing a control of the balance between sinusoidal and noise energy. It is interesting to note that in both the above systems the noise is a direct result of the non-stationary partial behaviour, implying a high degree of correlation between the partial and noise content.

Initial experiments in [146] used the correlation between a simple linear fit to the logarithm of the harmonic amplitudes and a linear fit to the log-noise spectrum as a means of sharing the noise component between several overlapping notes. The straight line fits are illustrated in fig. 5.12 for a single note over a sequence of time frames. The method was based on the general observation of a $1/f$ decay in the spectra of music signals, which is equivalent to a linear decay in the log-amplitude spectrum against a log frequency axis. However, the $1/f$ decay is too simplistic to accurately model the complex noise spectral envelope. The implementation of a similar noise sharing method to [146], by extracting the spectral envelope using linear predictive coding (LPC)[147, 157], will now be described.

The basic premise of linear predictive coding, which is widely used for speech analysis, coding and synthesis, is that the signal is an AR process (eqn. 5.1). Whereas the AR model was used as a time-domain method in section 5.1.1, the AR model has an equivalent

frequency-domain interpretation where the coefficients $a[m]$ determine the poles of an all-pole filter[147]. The z-transform of eqn. 5.1 results in:

$$A[z] = \frac{E[z]}{1 - \sum_{m=1}^{M} a[m]\ z^{-m}} \tag{5.20}$$

where $A[z]$ and $E[z]$ are the z-transforms of the signal $x[n]$ and error signal $e[n]$. The term $e[n]$ is also referred to as the excitation signal. Eqn. 5.20 therefore shows that the observed spectrum $A[z]$ is the result of filtering an excitation $E[z]$ with a filter $H[z]$:

$$H[z] = \frac{1}{1 - \sum_{m=1}^{M} a[m]\ z^{-m}}. \tag{5.21}$$

This source-filter model is a fairly good approximation to the human voice, where the excitation signal is a noise source or pseudo-periodic series of glottal impulses, and the filter $H[z]$ describes the frequency shaping of the vocal tract and lips. LPC relates to the analysis, coding and transmission of these quantities, and synthesis from them. Here, we use LPC simply as a spectral envelope estimation method for the noise and harmonic spectra.

Let $x^{res}[n]$ be the residual signal after the harmonic components, $x^p[n]$, of a set of P notes have been extracted from the original mix $x[n]$. A time-varying spectral envelope shape can be computed for each of the signals $x^{res}[n]$ and $x^p[n]$, $p = 1, \ldots, P$, by calculating an AR fit in a sliding window. In the r^{th} time frame we denote $A^{res}[k.r]$ as the amplitude response of the LPC filter in eqn. 5.21, where the AR coefficients $a[m]$ have been estimated from the r^{th} windowed segment of $x^{res}[n]$ by solving the Yule-Walker equations using the Levinson-Durbin algorithm[147, 148]. Similarly, denote the amplitude response of the LPC filter corresponding to $x^p[n]$ as $A^p[k, r]$ in the r^{th} time frame. Fig. 5.13 illustrates the shape of the estimated spectrum of a periodic signal in a single frame, for different values of the model order M. Eqn. 5.20 shows that the number of poles of the all-pole filter is equal to M, which explains why in fig. 5.13 the LPC spectrum envelope is smoother at the lower model order.

We now make the assumption that the spectral envelope of the noise component of note p in frame r should have roughly the same shape as the harmonic spectral envelope $A^p[k, r]$. In other words, we assume that the noise content is concentrated in roughly the same spectral regions as the harmonic content, which is confirmed by general observations that the noise content is stronger around the partial locations. However, we will introduce a constant of proportionality, ξ_b^p, to account for the fact that each note will have a different overall contribution to the overall noise spectrum in band b. If the mix contains only the P notes and no other unknown content, then the noise energy in a particular frequency band should be roughly the sum of the noise energy contributions of the individual notes

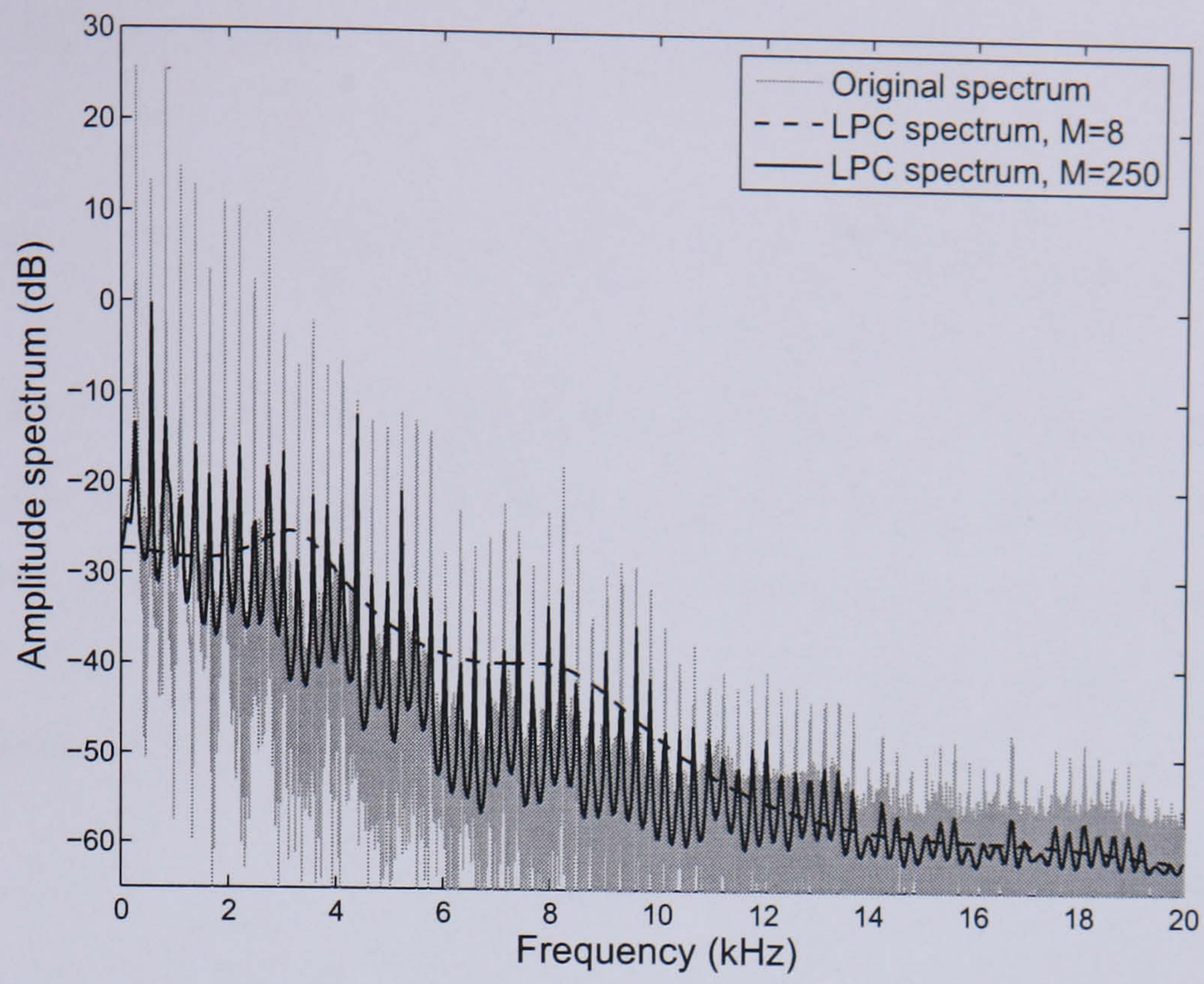

Figure 5.13: Estimation of the log-amplitude spectral envelope using LPC at two different model orders.

in that band. The shape of the noise energy spectrum in frame r can be approximated by the squared shape of the amplitude envelope: $A^{res}[k,r]^2$. Therefore, we can approximate:

$$A^{res}[k,r]^2 \simeq \sum_{p=1}^{P} \left(\xi_b^p \, A^p[k,r]\right)^2 \tag{5.22}$$

where the parameters ξ_b^p are to be determined by optimisation. To clarify, $b \equiv b(k)$, i.e. $b(k)$ gives the band in which frequency bin k falls. The set of constants $\{\xi_b^p;\ p = 1, \ldots, P\}$ can be estimated by minimising the median of the difference energy spectrum $E[k,r]$:

$$\arg\min_{\{\xi_b^p\}} \sum_r \text{median}\left\{E[k,r];\ k = k_b^{min}, \ldots, k_b^{max}\right\}. \tag{5.23}$$

The sum over r extends over the entire signal, although it would be possible to determine $\{\xi_b^p\}$ for shorter time segments, thereby allowing the noise energy to vary relative to the harmonic energy over the duration of the note. Of course, this would make the parameter estimation more time-consuming. k_b^{min} and k_b^{max} are the upper and lower frequency bins of band b, and:

$$E[k,r] = \left| A^{res}[k,r]^2 - \sum_{p=1}^{P} (\xi_b^p)^2 \, A^p[k,r]^2 \right|. \tag{5.24}$$

The median of $E[k,r]$ was chosen as a measure of the goodness of fit as opposed to using a LSE estimation. This was decided because the median is less influenced by outlying large values of $E[k,r]$, and as the LPC amplitude spectra can have poles which are very prominent and large in amplitude, it would not be desirable to place too much significance on the exact amplitude of these poles when estimating the goodness of fit. Once the set

of parameters $\{\xi_b^p\}$ have been determined for all frequency bands, a quantity $w^p[k,r]$ is defined:

$$w^p[k,r] \;=\; \left[\frac{(\xi_b^p)^2\; A^p[k,r]^2}{\sum_{q=1}^{P}\;(\xi_b^q)^2\; A^q[k,r]^2}\right]^{1/2} \;=\; \frac{\xi_b^p\; A^p[k,r]}{\sqrt{\sum_{q=1}^{P}\;(\xi_b^q)^2\; A^q[k,r]^2}} \tag{5.25}$$

such that $w^p[k,r]^2$ is a measure of the proportion of noise energy in the original mix at frequency bin k in frame r that is contributed by note p. Thus, similarly to section 5.2, we can interpret the set $\{w^p[k,r];\; p=1,\ldots,P\}$ as a set of time-frequency weighting functions that share the noise spectral energy between the P overlapping sources. It is clear that they are normalised so that the energy proportions sum to unity:

$$\sum_{p=1}^{P}\; w^p[k,r]^2 \;=\; 1. \tag{5.26}$$

Finally, we can estimate the discrete STFT noise spectrum of the p^{th} note as:

$$\hat{R}^p[k,r] \;=\; \frac{w^p[k,r]}{\sum_{q=1}^{P}\;w^q[k,r]}\; F^{res}[k,r] \tag{5.27}$$

where $F^{res}[k,r]$ is the (complex) STFT of the residual waveform $x^{res}[n]$. The normalisation above ensures that the residual spectrum is split entirely between the P notes:

$$\sum_{p=1}^{P}\; \hat{R}^p[k,r] \;=\; F^{res}[k,r]. \tag{5.28}$$

Finally, the estimated noise waveform of the p^{th} note can be calculated from $\hat{R}^p[k,r]$ using an overlap-add process, as discussed in section 2.1.1. As $\hat{R}^p[k,r]$ is in phase with $F^{res}[k,r]$, the subtraction of the noise component of a single note from the mix yields a smaller noise residual attributed to the remaining notes.

Two issues remain that have not been discussed in much detail. Firstly, the choice of AR model order M is important. If the order is low, the envelope is smooth and contains few poles, as shown in fig. 5.13 for $M = 8$. The poles are usually too far apart to align themselves with the harmonic frequencies, and thus, the weighting functions $w^p[k,r]$ are themselves fairly smooth functions that do not have much selectivity in terms of being concentrated around individual harmonic frequencies, and have a closer resemblance to formant structures. For large values of M in relation to the average period of $x^p[n]$, the poles of the AR model mostly align themselves with the harmonic frequencies, also shown in fig. 5.13 for $M = 250$. Thus, the weighting functions $w^p[k,r]$ tend to selectively extract noise content from $F^{res}[k,r]$ at the harmonic frequencies of this note. It was found empirically that moderately large orders ($128 \leq M \leq 256$) tended to perform best.

The second issue relates to the decision to estimate $\{\xi_b^p\}$ within frequency bands rather than providing a single constant for each note that reflects its overall noise contribution to

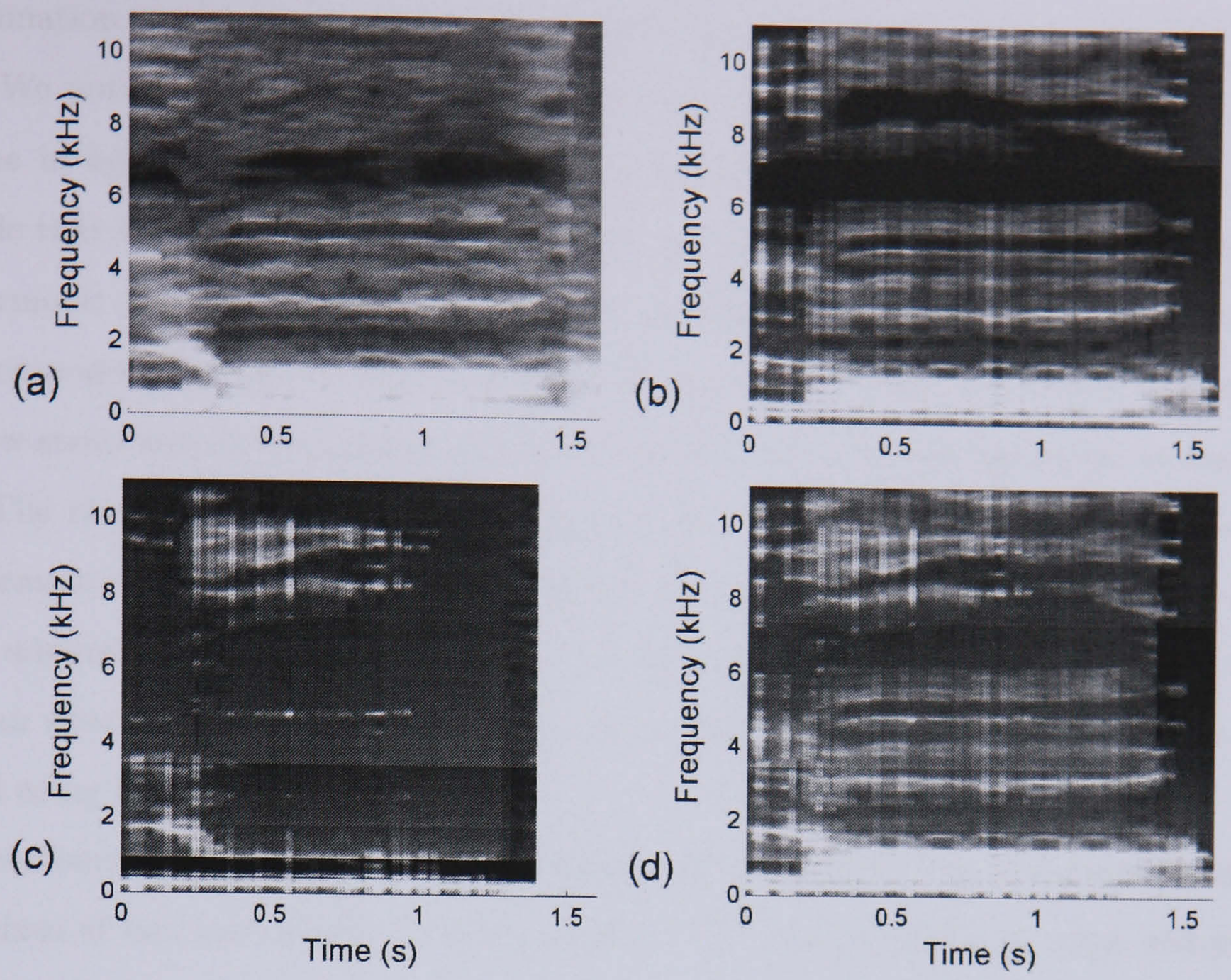

Figure 5.14: Separation of the residual of a mix of two sung vowels: 'laa' and 'loo'. **(a)** The shape of the AR noise energy spectrum of the residual mix, $A^{res}[k,r]^2$, **(b)** The weighted AR energy spectrum of the harmonic component of 'laa', $(\xi_b^1 A^1[k,r])^2$, **(c)** The weighted AR energy spectrum of the harmonic component of 'loo' $(\xi_b^2 A^2[k,r])^2$, **(d)** Sum of the AR harmonic energy spectra in b and c.

the mix. The main reason for bandwise processing is, it is possible that the ratio of noise energy in a note to harmonic energy is frequency-dependent, and general observations of clean instrument samples have indicated that a larger relative noise level exists at higher frequencies than at lower frequencies. If the number of bands chosen is too large, then the optimisation in eqn. 5.23 is likely to be cumbersome as the number of parameters $\{\xi_b^p\}$ increases, and less robust given that there would be fewer frequency bins over which to estimate each parameter. It was decided to use frequency bands equally spaced on the Bark scale, similarly to section 5.2.1, except that a smaller number of Bark bands is used here. It was found that between 1 and 10 Bark bands lead to useful results.

Fig. 5.14 illustrates the result of fitting the shapes of the harmonic LPC energy spectra $(A^p[k,r])^2$ in figs. 5.14b,c to the shape of the residual energy spectrum $A^{res}[k,r]^2$ in fig. 5.14a. Starting with two female sung voice samples, 'laa' and 'loo', the harmonic content of each was extracted, and then the residuals of each were summed together. The two vowels have the same pitch but a different formant structure. The optimisation in eqn. 5.23 was used to find the set $\{\xi_b^p\}$ where $b = 1,\ldots,6$ and $M = 128$. Effectively, fig. 5.14d, which shows the sum of the weighted noise energy spectra, $\xi_b^1\ A^1[k,r] + \xi_b^2\ A^2[k,r]$, is as close an

approximation as possible to fig. 5.14a obtained by adjusting the parameters $\{\xi_b^p\}$ in each band. We notice that there is a clear correlation between the shape of the noise energy envelope in fig. 5.14a and the shapes of the harmonic energy envelopes. Thus we can conclude that aspects of the formant structure of each vowel are evident in both the noise and harmonic components. In this example, the first vowel, 'laa', seems to have a larger noise component, hence the close similarity of figs. 5.14a and 5.14b. Nevertheless, there are a few structures in the mixed noise spectrum that are more similar to the second vowel, 'loo'. The re-synthesised noise components of each vowel extracted from the mix had a clear formant structure, and sounded similar to the original unmixed residual components. As the original unmixed residuals are out of phase with the separated residuals, the SRR is not an effective measure of separation performance, and so a formal evaluation of the method using SRRs was not made.

It was found on a few other occasions that the re-synthesised noise components extracted from mixes of two instruments/voices resembled the original unmixed noise components, and retained the natural noise-like characteristics of the original instruments. However, the method is of course not foolproof: there is generally some cross-over between the noise components of each note, for example, a vibrato effect of one note bleeding into the other. This can be partly offset by a careful choice of the AR model order and the number of frequency bands. Another limitation of the method is that the optimisation in fig. 5.22 assumes that the noise mix is an exclusive sum of P note residuals, i.e. it is does not make allowances for other background noise sources. It would be necessary to adapt the method if the task required was to extract the noise component of a single note from an unknown mix of other sources. Nevertheless, it is interesting that a perceivable separation of the mixed residual component is possible at all, which supports the original assumption that the harmonic content and noise content of a note are correlated.

5.4 Conclusions

The chapter focused on separating the transient and noise content of a note from a mix of notes or recording, as these components of the note are, in addition to partial content, important to the integrity of the timbre of the separated note[158], and necessary to avoid artifacts being left in the residual recording. Whereas the existence of a discrete series of harmonics usually results in a fair degree of isolation in the time-frequency representation between the harmonic content of overlapping notes, the task of separating overlapping noise or transient content is more difficult. There is virtually no time-frequency region of the noise/residual representation of the mix that can be identified uniquely with a single note. Furthermore, the highly nonlinear behaviour that is responsible for transient/noise

content in acoustic instruments is difficult to model even for a single instrument type, and so effectively, there is no equivalent of the sinusoidal model for transient or noise content.

Three practical approaches to the problem have been developed, two of these methods based upon energy considerations. The first method is intended for separating the transient content of a note from a recording, where the attack of the note is sufficiently far apart from other notes to be separable in the time-domain. A time-domain AR model was described for separating the transient attack/event from the recording. The method works well at separating impulsive or percussive onsets from the recording, and is not as successful for tonal onsets characterised by a rapid change in timbre but without a large corresponding change in the energy profile. As long as the transient events in the recording are indeed separated in time (by roughly > 100 ms), the method extracts transient events that sound similar in timbre to note/percussion onsets in the mixed recording. A simple effect can be implemented which adjusts the 'level of staccato' in the recording, simply by attenuating or amplifying the transient component of the recording.

When transient events of multiple notes are overlapping in time, the above method would most likely interpret these as a single transient event, and so another approach is needed. A bandwise energy interpolation method was developed that attempts to separate multiple transient attacks by interpolating the energy in each attack, within individual frequency bands, across the duration in which there is an overlap between more than one transient. We assume that the attack onset times are known *a priori*, and that the notes are well modelled as decaying energy envelopes within each band. Although the assumption of decaying envelopes is a simplification, it is this which makes the method applicable generically to a variety of percussive and pitched instrument types. The notes are required to be separated by a short time interval, so that each note is in a state of energy decay before the onset of the following note. The method was applied to separating percussive sounds in [7], but has also had some success at separating overlapping pitched note attacks from instruments with sharp attacks, such as the piano or guitar. A potential improvement to the method would be to incorporate a joint estimation of the bandwise energy envelopes of all sources for each frequency band. The current iterative subtractive method gives slightly different results depending on the order in which the note energy envelopes are subtracted from the original mixed envelope. Aside from this relatively small potential modification, it is difficult to see how the transient separation method could be substantially improved without incorporating instrument or context specific information.

The third and last method attempted to separate the noise content of a note from a mix of notes, and was originally proposed in a simpler form in [146]. Whilst the separation in time between attacks was integral to the two transient separation methods, the noise

components of mixed notes are overlapping in both time and frequency, meaning that it is virtually impossible to identify any region of the time-frequency plane with a single instrument. We therefore proposed using the harmonic content of each note, which is more easily separable from the mix, as an indication of where the noise component of this note would be concentrated in the time-frequency plane. It was argued that the noise and harmonic content of a note are likely to be correlated in time and concentrated in the same frequency regions. A method was developed for fitting a weighted sum of spectral envelopes of the harmonic components of each note to the spectral envelope of the mixed residual. This allowed the construction of time-frequency weighting functions for sharing the noise/residual signal between the set of overlapping notes. The method is not particularly accurate; an average performance for separating two notes could be described as: both separated noise components sounding like a mix of the two original unmixed note residuals, but with the correct note residual louder. In a test sample consisting of a mix of two female sung vowels of the same pitch, the separated residuals were similar to the original residuals, and could easily be identified with the correct vowel. The partial success of the method at least supports the idea that the harmonic and noise component of a note are correlated, and this can be useful for estimating temporal and spectral attributes of the noise component when mixed with other notes. Again, we could expect gains in performance by incorporating instrument-specific information, such as formant structures or general noise measures. One possible route for further investigation, which would not be instrument-specific, would be to study the nature of spectral line broadening due to microfluctuations of sinusoidal frequency and amplitude. An analytical expression for the shape of spectral peaks due to microfluctuating partials would allow a better estimation of the noise spectrum surrounding partials.

Although there has been a fair amount of work towards transient extraction methods[29, 68, 67, 71, 69, 70, 71, 72] and also noise envelope estimation methods[65, 151, 159, 156, 160], these have tended to focus on audio modelling applications where a succinct and flexible model of the non-sinusoidal component of the mixed recording is required, rather than on the problem of separating overlapping residual and noise components from multiple sources. The latter task is difficult for the reasons discussed above, and so it seems unlikely that the quality of separation achievable using transient/noise separation methods can equal the performance of methods for separating harmonic content, such as sinusoidal modelling or harmonic filtering. In certain applications though, such as when separating an instrument that contains prominent attacks from a recording, the extraction of the non-sinusoidal component from the mix, in a way that does not leave artifacts in the residual, is perceptually important and deserves further investigation.

Chapter 6

Grouping of Separated Notes

> "The melody of a piece [of music] is not composed without order and without reason– it is formed of several segments that each have a complete meaning..."
>
> **- Michel de Saint-Lambert, 'Les Principes du Clavecin' (1702)**

We now turn our attention towards a problem of a rather different nature to chapters 4 and 5 which dealt, respectively, with the separation of harmonic and non-partial content of a note from a polyphonic recording. Given an automated system for separating individual notes from the recording, it would be enormously beneficial to automatically categorise these notes into different sources or instruments. As human listeners tend to perceive individual notes not as isolated entities, but as transitory events within a phrase or instrumental part attributed to a single source, it would be more natural and useful in an automatic system to interpret the music contents in a similar manner. Aside from more closely emulating the behaviour of the human auditory system, the ability to separate different sources from a mix would facilitate the extraction of valuable content-based information. Some functions could be, for example, extraction of melodies or instrumental solos, which would be helpful for music information retrieval, audio segmentation and markup according to source/instrument type, and de-mixing the recording into different source parts. It would also enable source-specific audio transformations such as re-mixing a recording with different volumes or spatial locations of each instrument. Although corrections to the note categorisation could be made by hand in some cases, some of these applications would not require 100% accuracy. For example, even if the majority of notes in a melody played by a single source could be distinguished from notes of other sources, this might be sufficient to identify the piece of music in a query-by-humming system. In fact, it has been observed in the note grouping system described here, that the 'key notes' perceived in a melody/solo tend to be grouped correctly, whereas difficult cases, which include short transition notes, are perhaps not quite as important from the point of view of music content description.

Although we always try to avoid simplifications that restrict music processing systems to simple test cases or real recordings that are relatively easy to deal with, there are always difficult cases that fall outside the processing paradigm. In contemporary electroacoustic music, for example, it is not uncommon to encounter a piece containing hardly any harmonic or melodic content, and no identifiable real sources or repetitive structures. However, the music is not random, and if we have any inclination to listen to it, we are still usually capable of recognising as distinct entities, the 'building blocks' or sound events from which the music is constructed. Thus, although artificial systems have demonstrated similar performance to human listeners in isolated tasks, the dexterity, complexity and intelligence of the human auditory system is far from being equalled. In the particular task of identifying a source or instrument type, we generally do not perceive a note as consisting of separate partial, transient and noise components, but rather as a complete unit, the source of which is identified based upon a holistic judgement (although there are conflicting views as to the exact importance of the note transient in identification, as discussed in section 6.1). It seems likely that a combination of capabilities related to musical contextual information, and identification of transients or the physical excitation mechanism, is what allows even very short notes in a recording to be correctly identified by human listeners, in the sense that they are placed in the correct perceptual stream.

In many perceptual tasks, humans deal with incomplete information by making a best judgement based upon the limited information that is available to them. This work does not propose to replicate this complex behaviour, and is still essentially a data-driven or 'bottom-up' approach without, what some might argue, a proper integration of 'top-down' information based upon higher level inferences (see for example [161]). In some informal tests, the inclusion of higher-level musical contextual information was found to enhance note grouping performance (an increase in sound source identification accuracy due to musical context integration was also observed in [162]), but this will be left as an area for future research. Hence, this chapter will explore data-driven methods for grouping isolated pitched notes, as long as they can be clearly perceived as distinct events within the recording, into clusters of notes based entirely upon features computed from the individual notes.

In real music pitched notes can be shortened, articulated in an unconventional way, post-processed, and overlapping with other sources. Furthermore, each note may arise from one of a huge number of acoustic instruments, or from a synthesised instrument where the variation in sound parameters is essentially infinite. The problem of note grouping is thus of a very different nature to previous studies on instrument classification (section 6.2) from clean acoustic note samples, which assume a finite set of instruments and have usually been tested on clean samples. What we wish to do instead is to group notes into

clusters, such that the notes in each cluster are similar, in the sense that they would be perceived by a human as arising from the same acoustic or synthetic source. The sound variation within each source/class is thus assumed to be smaller than that between samples of different classes. If this turns out to be false, it is probably necessary to incorporate some additional information to perform note grouping, for example, by including the musical context of each note within the recording. It is useful to review previous work on timbre discrimination (section 6.1), where timbre will be defined as that which distinguishes two sounds being equal in pitch, loudness and subjective duration, and presented in a similar manner[163]. Work in timbre discrimination has led to the identification of some common perceptual axes that humans use to differentiate between different timbres. Loudness, pitch and duration are also factors that humans are likely to use for source identification, so what is actually desired is a set of axes for source discrimination rather than timbral discrimination. However, the similarity between the two tasks is close enough, that a good starting point for a note grouping system would be to adapt the features and methods developed in timbre discrimination to source discrimination.

One theme that arises in this work is that of 'understanding without separation'[47]. This is primarily a theory of music perception that views the listening experience, not as a highly analytical recognition of a structured music representation, but as a response to a continuous sound input. The listener is able to perform certain functions in response to this input, such as foot-tapping to the beat, and make certain judgements about the musical quality, such as recognition of genre, composer or instrument type. In fact, it is argued in [47] that cognitive models of music listening constructed upon an abstracted or symbolic representation of music are actually misleading. However, our primary interest here is not music cognition, and as there are many creative and content retrieval-based applications for symbolic transcription-based separation systems (section 1.1), this work is directed at these applications rather than at an accurate model of music cognition. It will be interesting to see, however, in relation to the theme of 'understanding without separation', in which of the following two cases the particular task of grouping notes into sources is performed better. Firstly, features that do not imply a symbolic musical structure are computed directly on a segment of the original waveform containing a note, and secondly, features that imply a structured representation (i.e. that are dependent on the pitch value or harmonic trajectories) are computed in addition to the 'unstructured' features on a separated note from the mix. In other words, does the act of separating a set of notes from a recording make it easier or more difficult to perform the task of grouping these notes into sources? Put simply, can separation aid understanding, at least from a signal processing rather than perceptual point of view? Results will be given in section 6.8 for both cases.

6.1 Timbre discrimination

We wish to find a set of features, computable on separated note waveforms or note segments of the original recording, that is suitable for categorising notes into a finite number of classes, where ideally these classes represent the different sources/instrument types as perceived by the listener. In an ideal situation in which the recording is monophonic and note samples are relatively free of interference, it would be reasonable to assume that the various physical correlates of the 'perceptual axes' that have been identified in the timbral discrimination literature (e.g. spectral centroid, spectral flux and attack rise time) would be useful in our source discrimination problem. However, as we are dealing with real polyphonic music, this is not necessarily the case. The existence of interfering sources has the consequence that the features computed can be misleading, since they become more indicative of the background and less indicative of the desired source. In other words, it cannot be assumed that the features identified in the timbral discrimination literature (for example [158, 1, 164]), or in the area of instrument classification (reviewed in [165]), are the best features for source discrimination in mixtures. Some features can lose their discriminatory power in mixtures. For example, attack or rise time, computed as the time measured from when the amplitude envelope reaches 2% of its maximum until it reaches the maximum, can produce abnormal results when the amplitude of an interfering source is larger than this threshold. As a treatment of timbre discrimination or instrument classification in polyphonic real recordings was not found in the literature, it was decided to determine the best set of features for note grouping from training data. The approach in section 6.3 is to select a subset of 'good' features out of a large set of features. The feature subset should be chosen such that the different source/instrument classes are separated well, given a training set of note samples from real recordings embodying a diverse selection of timbres and source properties.

It is true that idiosyncrasies in articulation of the instrument/source and musical context within a melodic phrase provide perceptual cues for note identification, that would not exist when dealing with isolated instrument tones. It was found in [166] that instrument identification based upon phrases was better than that on isolated notes. A different study of timbre discrimination compared the ability to discriminate pairs of isolated tones with that of musical patterns[167]. The results of the listening experiment were inconclusive in showing whether musical context is advantageous or detrimental to timbre discrimination, but nevertheless did show a marked difference between the two cases, which turned out to be source dependent. Comparable results for timbre discrimination between isolated tones and melodies were obtained in [168]. The integration of musical context into the note grouping system will be left as a potentially fruitful area for further research.

For completeness, the remainder of this section will review various studies in timbre dis-

crimination in humans, and attempts at quantifying this behaviour using multidimensional timbral spaces. The motivation for modelling timbre as multidimensional is that timbre is effectively defined by what it is not, encompassing all the possible variation in sound unaccounted for by differences in loudness, pitch or subjective duration[163, 169]. Whereas loudness and pitch can each be measured on a single scale, timbre can be manipulated in a multitude of different ways, including all kinds of temporal and spectral transformations, and hence it is impossible to quantify it using a single variable.

The importance to sound perception of the relative amplitudes of partials is well known, originating in the work of Helmholtz[170]. The correlation between timbre and spectral shape has since been confirmed in numerous studies of timbre perception, for example [169, 171, 172, 1, 173, 164, 174]. Temporal attributes of tones were also found to be useful in [175, 166, 174]. The importance of the initial attack of the sound was highlighted in studies on instrument recognition such as [176, 177, 178, 179], where the attack was removed or modified, such as by time-stretching. However, in [180] it was found that timbral similarity judgements between instrument tones did not change significantly when the attack was removed and the residual was listened to, or when only the attack was listened to, indicating that the main attributes used in similarity judgements were present in both the attack and remainder, and consequently, the attack is not essential for timbral similarity judgements. A possible explanation for the above seemingly contradictory views concerning the role of the attack, according to [180], is that instrument identification and similarity judgements are based upon different attributes. For example, the acoustic attack of an instrument may be quite important for detecting the type of physical excitation, and hence be informative of the source, whereas this may not be as important for overall timbral similarity judgements. Alternatively, the disagreement may in part be due to the fact that there is no clear distinction between the transient and steady state components[181]. Thus, a separated attack may contain portions of the steady-state component, and vice versa. Furthermore, some of the discrepancies between the various studies may be due to using different and selected sets of instrument samples. A large set of wide ranging pitched and percussive timbres was analysed in [164], with the result that both spectral (centroid) and temporal (rise time) features were correlated with the principal timbral dimensions. There is also some evidence in [164] of a tendency to group timbres according to source properties, such as the type of resonator, excitation method, material type and instrument shape. In the 3-dimensional space determined in [158], there was also some evidence of grouping of stimuli on the basis of instrument family. It would seem difficult to quantify to what extent these ecologically motivated factors[182] influence our perception of timbral similarity.

A common approach to constructing timbral spaces is to use multidimensional scal-

ing (MDS) analysis[183, 184, 185]. In this context, MDS attempts to find the principal attributes that listeners use for timbre discrimination[158, 1, 164, 169, 172, 186, 187]. Listeners are asked only to judge the dissimilarity between pairs, or on occasion triads[169, 188] of sounds, where these sounds may be synthesised or complex real sources. MDS analysis finds the spatial configuration or distances between these stimuli (in the simplest case, Euclidean distance) in a multidimensional space (depicted in fig. 6.1) of specified dimensionality, that observe as closely as possible the subjective similarities between the stimuli judged by human listeners. The derived orthogonal axes of this space are then sometimes given a physical interpretation (e.g. spectral centroid, attack rise time) when they are closely correlated with features computed from or used to synthesise the stimuli.

On most occasions MDS studies of timbre perception have yielded either 2-dimensional [172, 164] or 3-dimensional spatial solutions[169, 158, 186, 1, 168]. In [164] a 2-dimensional solution was determined for a diverse set of timbres from both orchestral and percussive instruments, whereas a 3-dimensional solution was determined for the subset of percussive sounds. In [1] an extension of the CLASCAL MDS algorithm[189] was used to determine the psychophysical dimensions of timbre. The algorithm accounts for variation between ratings of different subjects by modelling latent classes or sub-populations of subjects that weight the various timbral dimensions differently. It also incorporates a 'specificity' for each stimulus, which can be thought of as the intrinsic perceptual strength of each stimulus unaccounted for in the dissimilarities along the common dimensions. One way of looking at specificity is as an extra dimension or set of dimensions particular to this stimulus. Both a six dimensional model without specificities, and a three dimensional model with specificities turned out to be comparably close fits to the dissimilarity ratings. The acoustic correlates of the latter were log rise time, spectral centroid and spectral flux. Specificities indicated the possibility of additional dimensions along which only certain types of sounds vary. Spectral centroid also appeared as a principal timbral dimension in [171, 186, 164, 190] and its importance was confirmed in [174]. A principal timbral feature related to temporal aspects or synchronicity of the attack was also determined in [158, 172, 186, 1, 164]. Finally, a method for calculating the perceptual distance between two sounds was given in [191], determined by applying linear regression to the perceptual axes derived in the MDS analyses of [186, 1].

A creative application of MDS analyses of timbre is described in [172]. Here, the emphasis was towards musical synthesis of tones by using the timbral space as an expressive control structure. Control over perceptually important aspects of the tones, relating to spectral energy distribution and various temporal parameters, could be achieved by specifying points within this timbral space, coupled to a synthesis model based upon additive

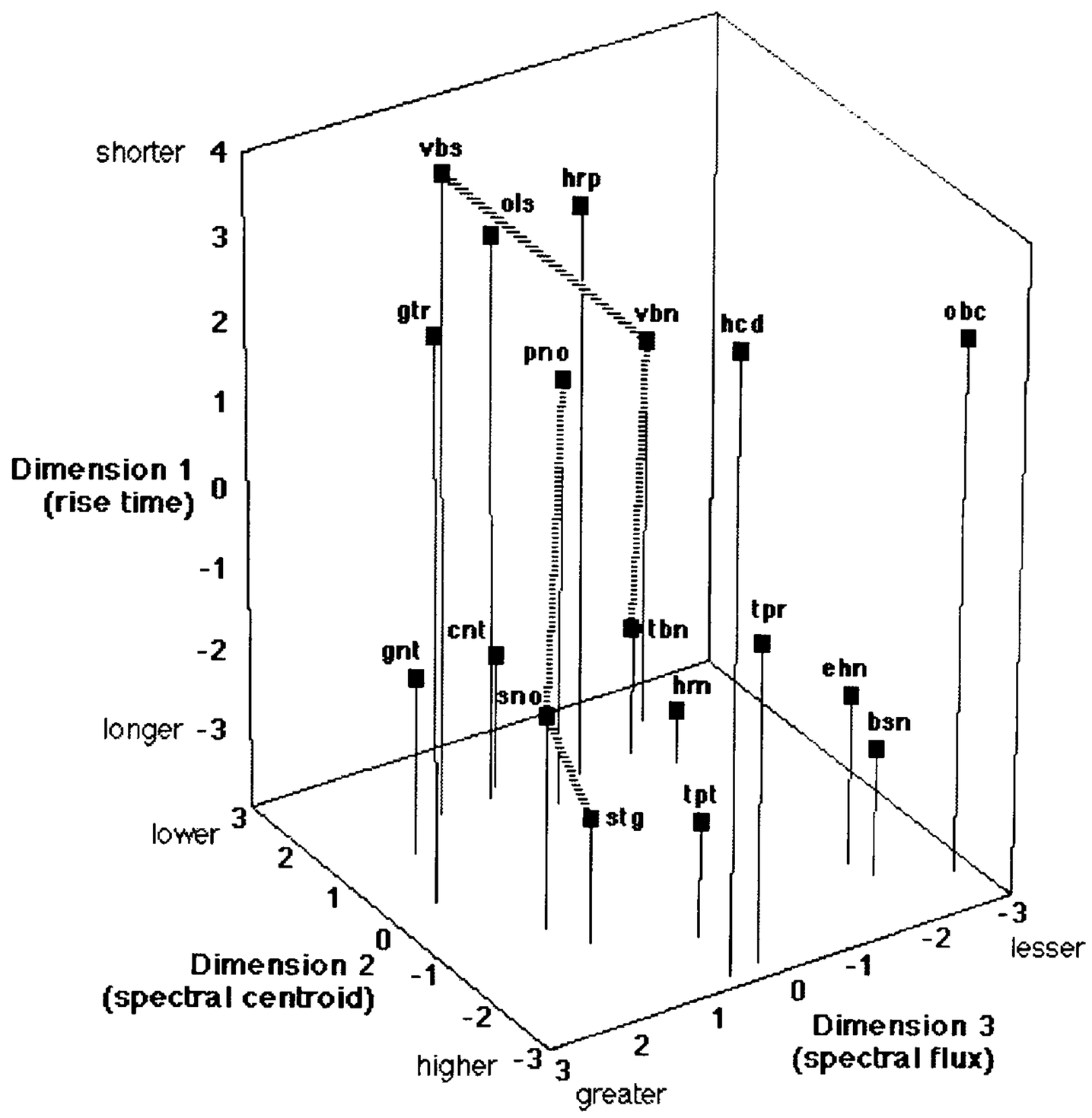

Figure 6.1: The 3-dimensional timbral space (with specificities) obtained in [1] by MDS analysis of dissimilarity ratings of 18 timbres by 88 subjects, indicating the acoustic correlates of the perceptual dimensions:
'Rise time' – logarithm of the time measured from when the amplitude envelope reaches a threshold of 2% of the maximum amplitude to the time it attains its maximum amplitude.
'Spectral centroid' – the average over the duration of the tone of the instantaneous spectral centroid within a running time window of 12 ms.
'Spectral flux' – the average of the correlation between amplitude spectra in adjacent time windows.
(bsn - bassoon, cnt - clarinet, ehn - English horn, gnt - guitar/clarinet, gtr - guitar, hcd - harpsichord, hrn - French horn, hrp - harp, obc - oboe/harpsichord, ols - oboe/celesta, pno - piano, tbn - trombone, tpr - trumpet/guitar, tpt - trumpet, sno - bowed string/piano, stg - bowed string, vbs - vibraphone, vbn - vibraphone/trombone). *Reproduced with permission of main author.*

synthesis. The timbral space should be consistent in the sense that sounds synthesised using interpolated parameters between two other points in the space are judged to have appropriately interpolated perceptual coordinates in the timbral space.

The study of timbre discrimination feeds into the area of auditory scene analysis, that is, the psychophysical study of how humans infer meaning from a continuous sound stream of independent and simultaneously sounding sources. The emulation of these processes or functionalities is known as computational auditory scene analysis (CASA). It is therefore logical that some systems have incorporated timbral discrimination cues into the CASA framework[162, 192]. The blackboard system for CASA in [192] describes how note hypotheses are organised into melodic tracks based upon a combination of musical contextual information and a timbral similarity measure. This measure is based upon the similarity of 'timbral tracks', which represent changes in the spectral centroid against changes in the average amplitude over time.

What can be extracted from this review, although different studies are not always in complete agreement, is some idea of the relative importance of various temporal and spectral features in timbre similarity judgements. Also, as MDS studies have concluded that two or three dimensional representations account for most of the variation in timbral similarity experiments, then if loudness, pitch and possibly duration are to be included as extra dimensions for source discrimination, it will be wise to use a feature space of dimensionality at least four.

6.2 Instrument classification

It has been argued that as the number of potential sources or instrument classes is not finite when dealing with music as a whole, the note grouping task is not a classification problem. Nevertheless, it may be true that some features or techniques that have been shown to be beneficial in classification of instrument sounds could also be useful if applied to a note grouping or clustering task. Thus, we review some work in this area.

A computational approach to auditory sound recognition of musical instruments is presented in [193]. Instrument recognition is performed within a hierarchical taxonomy of musical instruments. A tree-based classifier proceeds from the recognition of instrument family (e.g. bowed strings) to the recognition of the particular instrument type (e.g. violin, viola, cello, double bass). At any particular node of the tree, a set of features is selected for classification that provide maximum discriminatory power between the 'children' or descendants of this node, incorporating also a measure of feature reliability. This flexible approach to feature selection is reported as providing stability when features may be unreliable due to, for example, masking of certain audio attributes in real recordings. This resulted in a

greater robustness to classification across different sample databases. Features were derived from the 'weft': a 2-dimensional representation extracted from the 3-dimensional correlogram, where the latter is an auditory inspired audio representation with axes of frequency, time and time lag. The philosophy with regards to feature extraction is that sound sources are recognised primarily by perception of their excitatory and resonance structures[193]. The classification results tested on a database of 25 instrument types approached that of human listeners. The correct instrument type/family was identified with an accuracy of 38.7%/75.9% for isolated tones, and with an accuracy of 56.9%/74.5% for 10 s excerpts from solo recordings, respectively. A study of 14 orchestral instruments selected from a single database of clean instrument samples was given in [181], employing a slightly different instrument taxonomy to the one above. Fisher multiple discriminant analysis was used at each decision point or node of the taxonomy, to project the entire set of features onto a lower dimensional feature space, where the classes to be discriminated were maximally separated. A maximum *a posteriori* estimator was employed for classification based upon a single Gaussian distribution for each class in the lower dimensional feature space. Classification accuracies of 71.6%/85.3% were obtained for recognition of instrument type/family respectively, although no cross-validation was performed on other sample databases, which would be likely to decrease performance. A limitation of the approaches in [181, 193] is that the front-end representation is suited to monophonic sources, which means that it is probably unfeasible to adapt the feature extraction process to polyphonic material.

In [162] a template adaptation and matching procedure was used to identify instrument type, as part of a complete sound source identification system incorporating musical contextual information. It is interesting that the integration of musical contextual information improved instrument classification rates significantly (from 67.8% to 88.5% on average). The template adaptation and matching process is an improvement on that presented in [58]. Instrument types were identified in [58] using a combination of 'tone memory' or stored spectral models of each instrument, and 'timbre models' which use a probabilistic distance measure for classification within an 11-dimensional feature space in which the different instrument types are well separated. The adaptation method in [162] adapts stored templates or sample waveforms of each instrument to the polyphonic audio, using a combination of phase tracking and FIR filter design. The template adaptation was designed in order to alleviate classification problems arising from variation, encountered even between different samples of the same pitch and instrument type, due to differences in expressivity, recording environment, etc.. A disadvantage of this approach is that it requires a fairly comprehensive set of templates or pre-recorded samples for each instrument. One template waveform for each semitone of each instrument was stored in [162]. Along with several other

approaches focused on selected instrument databases, the template-based approach in [162] for polyphonic source identification seems impractical for analysis of general polyphonic recordings, due to this reliance on stored instrument templates or models. The approach discussed in this chapter is aimed at music as a whole, and proposes that given the huge variability in the acoustic or synthetic qualities of the instruments and recording environment, it is more appropriate to consider the instruments within any particular recording to be unique to that recording. As the tendency in popular music production shifts from traditional instrumentations to synthetic sound design, instrumentations that are common to more than one piece of music become harder to find. It seems that the instrument classification problem has direct relevance only to specific musical contexts. The broad approach taken in this chapter is to group a set of recurring timbral sound objects, or in the particular context of this work, notes, into classes that are unique to each recording, where the objects within a particular class share some common attributes which would render them as being derived from the same source, as judged by the listener.

An 8-class musical instrument identifier was described in [194] for classification using Gaussian mixture models (GMMs) or support vector machines (SVMs) and three feature sets: linear prediction coefficients, cepstral coefficients and mel-frequency cepstral coefficients (MFCCs). The best classification results (30% error rate) for 0.2 s samples were obtained for the SVM classifier using MFCCs when the test and training samples were taken from different recordings. Classification results in [195] produced no errors in a 5-class instrument classification experiment with 150 samples. A maximum likelihood estimator based upon a single multivariate Gaussian for each instrument type was used for classification, with a perceptually inspired 16-dimensional feature set. The sample database consisted of notes played with varying intensity and playing styles, although a single recording environment and instrument was used for each instrument type. These results would be more convincing if cross-validated on independent sample databases.

One of the earlier studies of musical instrument classification[196] reported accurate classification results for a set of four acoustically contrasting instruments played in a one octave range, using a nearest neighbour classifier and artificial neural network classifier. However, the limited set of instruments and recording conditions prevents any quantitative comparison with other systems. A more general classification system incorporating additional instruments and providing cross-validation results on different sample databases was described in [197]. Instrument taxonomies based upon single-stage, hybrid and hierarchical tree structures were used in classification, and a decision was made at each node of the tree using the combined results of five k-nearest neighbour classification (k-NN) algorithms computed on different feature sets. The k-NN algorithm is a rather simple and popular

technique for instance-based learning, which classifies a test feature vector (FV) into the class which occurs most often amongst the k-nearest neighbouring labelled FVs in the feature space. k-NN classifiers were also tested in [181, 193, 198, 199, 200, 201, 202] for various purposes in musical instrument classification. Overall classification results reported in [197] using a single-stage classifier were 55.6% and 57.4% when training and test samples were selected from different databases (with more than 350 samples in each database), where the number of instruments used in the classification was 13 and 9 respectively. Vibrato detection provided significant classification improvement by using the presence of vibrato in a test sound to restrict the detected instrument type to a subset of flute, cello or violin. A similar classification system that integrated several k-NN classifiers based on different feature sets (constant-Q spectrum and cepstral coefficients, MDS trajectories, RMS amplitude envelope and spectral centroid) was trained on a single instrument database of 19 musical instruments played at a single dynamic level within a three octave range[203]. This achieved a much better instrument classification accuracy of 90% and 93% for a single-stage and hierarchic classifier, respectively, which in comparison to the results of the similar system in [197], provides some indication that cross-validation on different databases can have a significant effect on overall classification accuracy.

A review of work related to music instrument classification and separation of duet sounds is given in [204]. Classification results based upon an artificial neural network classifier and three different feature sets (specified in greater detail in [205]) are presented, including an experiment where features were derived from sounds after the application of duet separation methods. These particular results were given only for a few artificially mixed sound examples, and so it is unclear how much bearing they have on our current problem of grouping notes separated from real recordings.

Results reported in [202] obtained classification accuracies of up to 68.4% using a k-NN classifier and 18 instrument categories, with a combination of spectral and temporal descriptors. Features derived from a wavelet coefficients were used in [206], with classification accuracies of 42.8% and 57.9% achieved using a classifier based upon rough sets and a decision-tree classifier, respectively. Samples from both these studies were drawn from a single sample database of musical instrument sounds.

In [207] a set of 18 cepstral coefficients based upon a constant-Q transform[11] was used to differentiate between oboe and saxophone samples. A Gaussian mixture model was constructed for each instrument class, with means and variances obtained using the k-means algorithm, and class assignment of each test sample was performed using a maximum likelihood criterion. Similar classification rates were obtained between the artificial classifier and human subjects, despite features being based entirely upon spectral information, and

the advantage that human listeners have in being able to assimilate musical context. A wider ranging study of four woodwind instrument types (oboe, saxophone, clarinet and flute) using a similar classification approach was described in [208]. A number of spectral and temporal features were used in this study in addition to the cepstral coefficients. However, constant-Q cepstral coefficients still emerged as effective features for classification, as did bin-to-bin differences of the constant-Q coefficients.

A comparison of features for instrument classification is given in [200] for a set of more than 5000 samples amalgamated from multiple sample databases, and containing a total of 29 instrument types. A set of 12 MFCCs augmented with selected individual features relating to the excitation characteristics (e.g. spectral centroid, strength and frequency of AM, and onset synchrony of different frequency components) was determined to have the best performance. Results were also given for linear-predictive derived cepstral coefficients, and it was found that a frequency warping approximation to the Bark scale led to improved classification rates using the warped linear predictive cepstral coefficients. In later work[209] using a feature set of MFCCs and delta-MFCCs, a hidden Markov model was used to model some temporal variation, arguing that a musical note can in general be characterised by an onset, steady state and decay segment, each of which has different spectral properties. Independent component analysis was also tested as a means of reducing the feature set to a smaller basis with maximal statistical independence, and led to consistent improvements in classification accuracy of a few percent.

Another study based entirely upon spectral characteristics of isolated tones obtained a classification accuracy of 70% for 27 instruments, outperforming human listeners[201]. Of the implemented classifiers (quadratic discriminant analysis, SVM, k-NN and canonical discriminant analysis), SVM and quadratic discriminant analysis produced comparably good results. Spectral centroid, inharmonicity, and the energy in the first partial emerged as the most relevant features. After comparison of the results of the different classification algorithms, the authors were of the opinion that the choice of feature set is more critical than the choice of classifier.

Also of interest is a real-time instrument classifier based upon a k-NN architecture, which uses a genetic algorithm to compute optimal feature weights off-line[198]. A classification accuracy of 68% was achieved using this system on a set of 39 timbral classes (deriving from 23 orchestral instruments with different playing styles). An enhanced temporal segmentation of each tone was reported to increase performance. We mention too a method described in [210] for classification into instrument category of 'non-registered musical instruments' (i.e. music instruments not present in the training data).

Although we have by no means covered the area of musical instrument classification in

its entirety, the interested reader is referred to a comprehensive review given in [165] for additional material. Also, given that pitched instruments are the main focus of this work, we have largely omitted to review in any detail the classification of non-pitched instruments. A review of work in this field can be found in [211].

6.3 Overview of the note grouping method

The next four sections describe a system for grouping individual pitched notes within a polyphonic recording into groups or clusters of notes, where ideally these clusters correspond to different source types. Firstly, the construction of a database of notes on which to test the system is described in section 6.4. A set of features are calculated on each note, which is outlined in section 6.5, and detailed formulations of each feature are given in appendix A. Section 6.6 describes the feature selection algorithm, which is used to reduce the entire set of features into a subset, where the separability between different source types in the lower-dimensional feature space is maximised for the training data. Lastly, a description of the clustering algorithm is given in section 6.7. This is used to cluster test samples within the feature space, where the number of source classes may or may not be known. Furthermore, the sources within the training data and test data are always kept separate.

6.4 Sample database

The method for note grouping was evaluated on a set of notes extracted from a fairly eclectic mix of commercially available recordings, summarised in table 6.1. In total, 20 different sources were selected from 16 different recordings of typical CD quality, giving a complete set of 574 individual note samples. Of the sources that are labelled with a common instrument type, these notes are sufficiently different between recordings to be easily recognised as arising from different sources. The number of notes selected per source/recording (P) is also given in table 6.1. As all of these recordings are in stereo, the channel in which each instrument was loudest was used as the original mono recording. The particular selection of notes from each recording does of course influence any subsequent results. Thus the following guidelines were adopted in the note selection process:

- Notes should be of duration 0.3 s or longer (the longest note is of length 6.7 s)
- Each note should have an easily identifiable pitch
- Each note should have clearly delineated boundaries within which the pitch remains relatively constant

Table 6.1: Recordings and instrument types used for evaluating the note grouping method.

Instrument	P*	Song Title	Artist/Composer
Bagpipe	22	'Ondes do Mar de Vigo'	M. Codax
Bass guitar	37	'Say Goodbye Hollywood'	Eminem
Double bass	28	'Blues for Dad'	Tananas
Guitar	31	'Chan Chan'	Buena Vista Social Club
Acoustic guitar	34	'Blues for Dad'	Tananas
Electric guitar	35	'Win My Train Fare Home'	Robert Plant
Horn	31	'Horn Concerto Nr. 4 (K495)'	W.A. Mozart
Laoud	26	'Veinte Años'	Buena Vista Social Club
Keyboard synth	42	'Only You'	Portishead
Piano	40	'Trois Gymnopedies: II'	Erik Satie
Organ	32	'Caballo Viejo'	Ry Cooder/Manuel Galbán
Tenor sax	23	'Moanin''	Art Blakey
Trumpet	30	''Round Midnight'	Miles Davis
Trumpet	30	'Moanin''	Art Blakey
Trombone	27	'At the River'	Groove Armada
Violin	23	'Erbarme dich, mein Gott'	J.S. Bach
Female voice	22	'Erbarme dich, mein Gott'	J.S. Bach
Female voice	24	'Only You'	Portishead
Male voice	24	'Tonight I'll Be Staying Here With You'	Bob Dylan
Male voice	13	'The Small Print'	Muse

* - number of notes per instrument type/recording

- Each note should be easily perceivable within the original recording and louder than any other pitched instruments in that particular segment
- Subject to the above rules, the selection should be random

Thus, although we tried to avoid biasing the results to easy cases, the sample set is on the whole representative of the more salient notes within the recording. For some applications such as music information retrieval, this slight bias could well be reflective of the context in which a note grouping system would be used. In a query-by-humming task for example, the query is more likely to be structured in terms of so-called 'key' notes, which tend to be those that fit the criteria above.

The note grouping algorithms were tested on both the raw note samples which are unaltered segments of the original recording, and on the separated notes from each raw sample. The latter were obtained using the harmonic filtering method described in chapter 4. To recapitulate, this consists of an initial pitch tracking stage, followed by harmonic frequency and amplitude tracking, and finally harmonic filtering using time-varying comb-

like filters.

6.5 Feature set

As argued in the introduction to this chapter, it would not be wise to assume that the various features that have been discovered to be optimal for instrument classification, or to be correlated with principal perceptual axes, are necessarily the best features for source discrimination in polyphonic mixtures. No study has been found of timbral discrimination of different tones in the presence of external interfering sources, and almost all studies of musical instrument classification have used either isolated tones or solo passages. Furthermore, instrument classification is a very different task from perceptual grouping of sounds of which the listener may not have any prior exposure. Thus, we have adopted an approach which consists of calculating a large number of features, and then selecting or extracting a subset of these features which provide good separability of the sources in the training data. The feature selection procedure will be described in section 6.6.

A total of 185 features were calculated on each raw or separated note sample. These have been divided into the categories listed below. As there are sometimes slight inconsistencies within the literature in the definitions of certain features, the procedures and formulas used for calculating each one is given in appendix A. Most of the features have been amalgamated from these sources: [212, 213, 214, 215]. The features can be categorised as one of two types: global or frame-by-frame (FBF). Global features are computed over the entire note, and FBF features are computed in each time frame and then summarised by three statistics: the mean, variance and linear slope/gradient of the feature measured over all time frames for that note. $f_1[r]$ is the frequency of the fundamental component which varies with time frame index r.

6.5.1 Waveform features

1 feature

The smallest category contains only one feature: the zero-crossing-rate of the waveform. This has been normalised with respect to the mean of $f_1[r]$ over all time frames.

6.5.2 Harmonic features

39 features

These are computed from the estimated harmonic frequency and amplitude trajectories and consist of 39 features. The global features include the mean-crossing-rate of $f_1[r]$ (the frequency at which $f_1[r]$ crosses its mean value), jitter (related to the frequency variation

of $f_1[r]$) and shimmer (related to the variation of the harmonic amplitudes). The FBF features include the logarithm of $f_1[r]$, amplitude ratio of odd-to-even harmonics, harmonic centroid, tilt, spread, skewness and kurtosis (all related to the rough shape of the harmonic amplitude spectrum), three harmonic tristimuli, the harmonic irregularity ('roughness' of the amplitude spectrum), and harmonic deviation (overall measure of inharmonicity).

6.5.3 Temporal/dynamic features

13 features

The temporal features encode the dynamic characteristics of the time-domain waveform and are computed on its amplitude envelope. The envelope is obtained by performing the Hilbert transform of the waveform and low-pass filtering the magnitude of the result using a IIR Butterworth filter of cut-off frequency 20 Hz. The temporal features include the maximum amplitude, decrease slope (related to the rate of decay), effective duration (percentage of time amplitude is greater than 40% of the maximum), amplitude ratio (ratio of the time that the amplitude is greater than 80% of the maximum to the effective duration), centre of gravity (temporal centroid), spread, skewness and kurtosis. Appendix A also describes a method for calculating the logarithm of the note attack, sustain and release times. A method for calculating the two remaining features in this category, the vibrato AM frequency and amplitude, can also be found in Appendix A.

6.5.4 Spectral features

48 features

The spectral features are all computed on a FBF basis and mostly using a logarithmic frequency axis. They consist of the spectral tilt and decrease (both related to the rate of decrease of the amplitude spectrum), roll-off (frequency below which 95% of the spectral energy is contained), flatness (ratio of geometric mean to arithmetic mean of amplitude spectrum in four frequency bands), crest (ratio of maximum amplitude to arithmetic mean of amplitude spectrum in four frequency bands), variation (variation of amplitude spectrum between time frames), centroid, spread, skewness and kurtosis.

6.5.5 Energy features

13 features

The energy features simply measure the total energy in the signal and the spectral energy within four logarithmically-spaced bands between 50 and 400 Hz.

6.5.6 Perceptual features

64 features

The perceptual features include 13 means of the MFCCs, the means of the delta-MFCCs (time derivative of the MFCCs), and the means of the delta-delta-MFCCs (second time derivative of the MFCCs). An auditory-motivated measure of relative specific loudness within 24 Bark bands, and the total loudness, are also included.

6.5.7 MPEG-7 features

7 features

The MPEG-7 musical timbre description tools are a set of descriptors aimed at describing the perceptual features of instrument sounds[216]. They are divided into two groups: the HarmonicInstrumentTimbre descriptor is designed for sustained harmonic sounds and contains four descriptors computed on the harmonic trajectories plus the LogAttackTime descriptor. The second group, the PercussiveInstrumentTimbre descriptor, is suited to percussive sounds and contains only the LogAttackTime, SpectralCentroid and TemporalCentroid descriptors. In total, all seven MPEG-7 low-level descriptors have been incorporated into our framework. Although the words 'feature' and 'descriptor' have been used interchangeably here, according to the MPEG-7 standard[216], a descriptor defines the syntax and semantics of each feature, whereas a feature is simply a distinctive characteristic of the data.

To assist manual feature selection whilst developing code and for informally evaluating the usefulness of various features, a graphical user interface was developed in Matlab, of which a screen-shot is shown in fig. 6.2. This provided some elementary functions for selecting individual features or groups of features, and a utility to allow the user to create their own mid-level features based upon low-level descriptors, although the latter function was not used in generating any of the results in this chapter.

6.6 Feature selection and extraction

The are obvious computational limitations on the number of features that can be used in the clustering algorithm, and given that the full set encompasses 185 features, a method for dimensionality reduction is essential. Aside from this though, two other reasons exist for dimensionality reduction. The first is given the unnerving title 'the curse of dimensionality': as the dimensionality increases, the data becomes more sparsely distributed. Thus, for a given sample size, there is usually an optimum dimensionality above which the data is too sparse to accurately classify new data points, and below which we do not make full use of

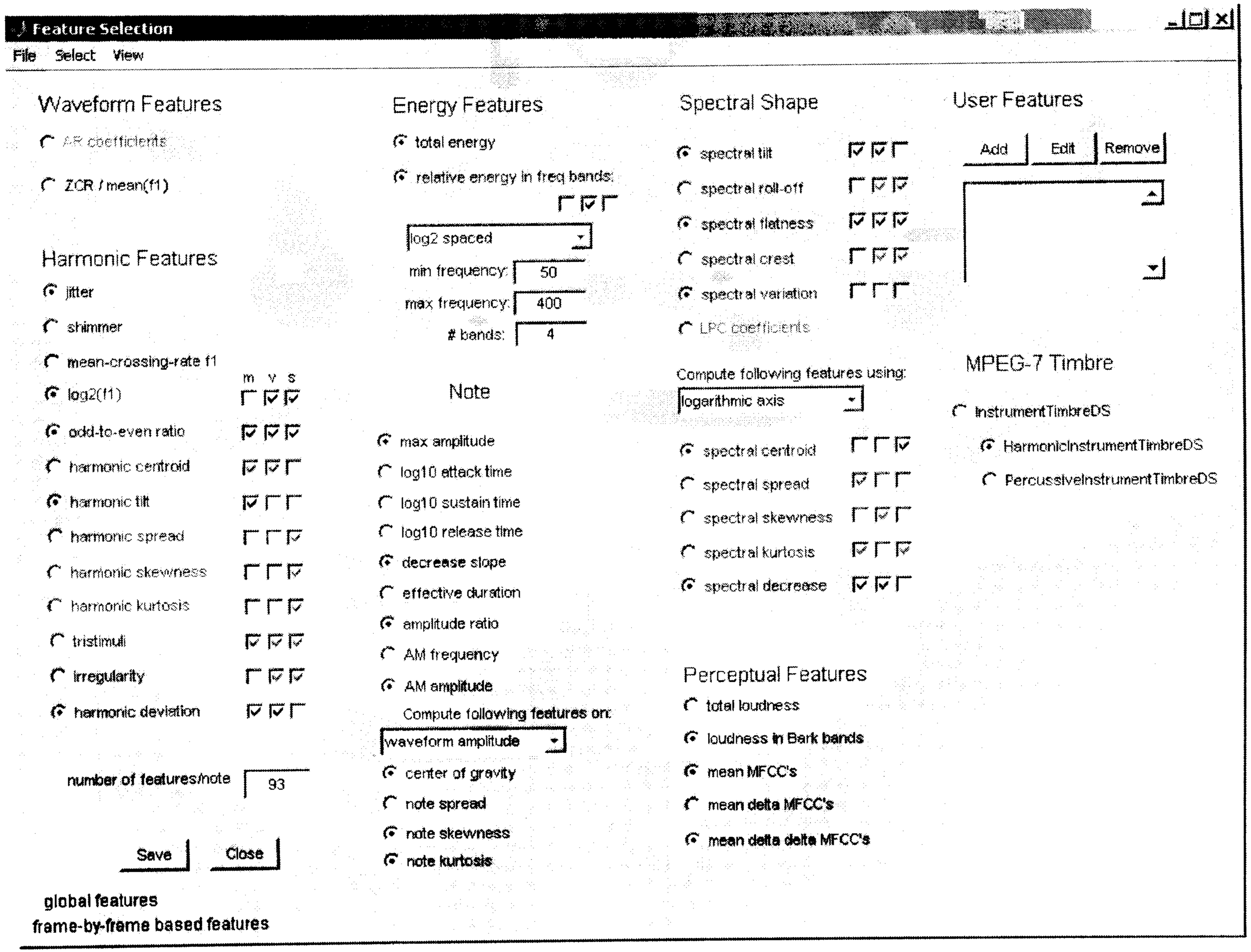

Figure 6.2: Screen-shot of the Matlab user interface for manual feature selection.

the separability of the classes. Secondly, some of these features may not provide any clear discrimination between the classes in the training data. It has generally been observed here that a smaller set of 'good' features performs better than a larger set of 'mediocre' features, so it is of benefit to remove 'weak' features from further consideration.

The distinction between feature selection and feature extraction must be made explicit [217]. A feature selection algorithm attempts to select out of D features, the best subset of $d < D$ features, where the best subset maximises some criterion function $J(X)$ over all possible subsets of d features, called X. Thus, the interpretability of the data is maintained, in the sense that each axis in the lower dimensional space corresponds exactly to a single original feature. There is a computational benefit in this approach, as once a subset of features has been decided, it is no longer necessary to acquire the remaining features from further samples. On the other hand, its ability to maximise class separability is less than that of feature extraction. In the latter case, the objective is to transform the original feature set to a lower dimensionality, where the discriminatory power in the lower-dimensional space is maximised. Principal component analysis, independent component analysis and linear discriminant analysis fit into this framework. Feature selection is actually a specific case of feature extraction, corresponding to a transformation matrix with "0"s and "1"s along the main diagonals.

As we would like to make some direct inferences about the types of features that are useful in note grouping, which would also greatly simplify data acquisition in any future work, a feature selection approach has been taken. To acquire some idea of the optimal number of dimensions d, note clustering results have been reported for different values of d.

One way of finding the optimal subset of d features would be to make an exhaustive search. However, this would require ${}_DC_d$ combinations of features to be evaluated, which is totally impractical for large feature sets. The branch-and-bound method[218] and its later improvements[219] allow the optimal subset to be found more quickly than by an exhaustive search. It exploits the monotonicity condition of the criterion function: as features are added to the subset, if the criterion function increases monotonically with increasing dimensionality, we can exploit this fact to exclude certain feature combinations from the search that could not possibly be optimal. The criterion function described in section 6.6.2 does behave monotonically, so the branch-and-bound algorithm would be an option, but a suboptimal feature selection algorithm has been chosen for computational efficiency.

6.6.1 Sequential forward floating search method

A sub-optimal feature selection algorithm known as the sequential forward floating selection (SFFS) algorithm has been used[220]. It was found to have better computationally efficiency than the branch-and-bound method and other sub-optimal methods even for large scale feature selection problems[217]. Its potential in situations where the criterion function is not monotonic is also commended in [221]. The SFFS can be seen as an improvement on the sequential forward selection method[222]. In the latter, the subset of features is built using a 'bottom up' approach, i.e. by sequentially adding to the current feature subset or empty set, the best feature (i.e. that which maximises the criterion function $J(X)$) amongst the remaining features. The related 'top down' method, sequential backward selection, begins with the entire feature set, and sequentially removes features until the required dimensionality d is reached. A drawback of the sequential forward selection method is that once a feature has been selected, it is not possible to remove it at any later point, where in combination with new inclusions of features it may be considered suboptimal. SFFS corrects this problem by allowing the size of the subset to 'float', i.e. features can be included or excluded from the current subset, allowing the correction of wrong decisions made in previous steps. Nevertheless, there is still a predominant direction of forward search. Predictably, in the sequential backward floating selection algorithm, the predominant direction of search is backwards.

A step-forward algorithm was used in [181] to determine the 10 best features for classification at different nodes of an instrument hierarchy. It emerged that some features were salient across instrument families, whereas others were useful only for discriminating between instruments of a particular family. The overall classification accuracy for instrument family was improved using this method.

6.6.2 The criterion function

We now describe the criterion or objective function, which for any subset of d features computed on the training data, measures the separability of classes within the d-dimensional feature space. A better criterion function would optimise predictive accuracy by explicitly relating to the error rate, for example, $J(X) = 1 - e(X)$, where $e(X)$ is the percentage of training samples that would be misclassified after the clustering process for a particular feature subset X. In other words, by maximising the criterion function, we would be minimising the likelihood of misclassification if the training data was resubstituted as test data. This type of pattern recogniser is called a 'wrapper'. However, it would be far too computationally expensive for our particular clustering algorithm (section 6.7) to repeatedly evaluate the clustering performance for all permutations of the feature subset. Instead, a

criterion function is used which attempts to find a tight clustering of within-class FVs whilst achieving maximum separability between classes, defining it as a so-called 'filter' pattern recognition algorithm.

Firstly, we define the within-class and between-class scatter matrices[223]. Suppose that X_c is the set of n_c FVs belonging to cluster/class c, and $\mathbf{x}$ will denote an individual FV. Then the mean FV for class c is:

$$\mathbf{m}_c = \frac{1}{n_c} \sum_{\mathbf{x} \in X_c} \mathbf{x}. \tag{6.1}$$

If n is the total number of feature vectors summed over all N_c classes, then the total mean vector is:

$$\mathbf{m} = \frac{1}{n} \sum_{c=1}^{N_c} n_c \mathbf{m}_c = \frac{1}{n} \sum_{\mathbf{x} \in X} \mathbf{x} \tag{6.2}$$

where $X = \{X_1, \ldots, X_{N_c}\}$. We define a scatter matrix that measures the covariance of the set of FVs in class c as:

$$S_c = \sum_{\mathbf{x} \in X_c} (\mathbf{x} - \mathbf{m}_c)(\mathbf{x} - \mathbf{m}_c)^T. \tag{6.3}$$

The within-class scatter matrix is the sum of the individual scatter matrices:

$$S_w = \sum_{c=1}^{N_c} S_c. \tag{6.4}$$

The between-class scatter matrix measures a weighted covariance between class means, and is given by:

$$S_b = \sum_{c=1}^{N_c} n_c (\mathbf{m}_c - \mathbf{m})(\mathbf{m}_c - \mathbf{m})^T. \tag{6.5}$$

Finally, the total scatter matrix is:

$$S_t = \sum_{x \in X} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T = S_w + S_b. \tag{6.6}$$

As one notices there is no dependence on c in eqn. 6.6, this means that the total scatter is a property of the data rather than any particular clustering. We also see that if the within-class scatter is minimised by a particular clustering, a consequence is that the between-class scatter is also maximised. This is fortunate, as ideally the criterion function should achieve maximum separability between classes, at the same time as obtaining tight clustering within classes. It can be shown that the eigenvalues $\lambda_1, \ldots, \lambda_D$ of $S_w^{-1} S_b$ are invariant to linear transformations of the data. Thus, a suitable criterion function that is also linearly invariant to transformations, is the trace of $S_w^{-1} S_b$:

$$J(X) = \mathrm{tr}(S_w^{-1} S_b) = \sum_{d=1}^{D} \lambda_d. \tag{6.7}$$

A number of alternative invariant criterion functions can be defined using other functions of the eigenvalues λ_d[223], but the above is appropriate for our needs.

Table 6.2: Reduced feature sets obtained by applying the SFFS algorithm to the full database of separated note waveforms. (*see appendix A for a description of the features*)

Best 3 features:	**Best 20 features:**
loudnessBark1	zcr
loudness	loudnessBark1
mean harmtilt	loudnessBark2
Best 5 features:	loudnessBark3
loudnessBark1	loudnessBark7
loudness	loudness
mmfcc3	mmfcc1
mean log2f	mmfcc2
mean harmtilt	mmfcc3
Best 10 features:	mmfcc4
zcr	mean log2f
loudness	mean harmtilt
mmfcc1	mean Ebands1
mmfcc2	mean Ebands3
mmfcc3	var Ebands1
mean log2f	var Ebands2
mean harmtilt	mean specvariation
mean Ebands1	var specflatness1
var Ebands1	mean speccrest4
mean specvariation	mean specspread

6.6.3 Results of feature selection

As a large number of features has been implemented, and many of these may be redundant or inappropriate for the clustering task, it is of interest to discover which features are best at discriminating between the different sources/classes. This has been attempted by applying the SFFS algorithm to features derived from the entire note database. As stated in section 6.4, the database consists of 574 samples drawn from 20 different sources, and the full set of 185 features was calculated on each note. It was suspected that the best features when calculated on separated note waveforms (i.e. the harmonic content of each note has been filtered from a segment of the recording) would be different to those computed on the original note segments. Table 6.2 lists the best 3, 5, 10 and 20 features that emerged from the SFFS algorithm when maximising the separability between the entire set of classes using eqn. 6.7, when features where calculated on separated note waveforms. Similarly, fig. 6.3 gives the best features when calculated on original note segments.

We start by summarising some general observations of tables 6.2 and 6.3:

Table 6.3: Reduced feature sets obtained by applying the SFFS algorithm to the full database of original note segments.

Best 3 features:	**Best 20 features:**
loudnessBark23	zcr
loudnessBark24	loudnessBark23
mean log2f	loudnessBark24
Best 5 features:	loudness
loudnessBark23	mmfcc1
loudnessBark24	mmfcc2
loudness	mmfcc3
mean log2f	mmfcc6
mean spectilt	mmfcc7
Best 10 features:	mmfcc8
loudnessBark23	mean log2f
loudnessBark24	mean tristim1
loudness	mean Ebands1
SpectralCentroid	mean specvariation
mean log2f	mean specflatness1
mean harmtilt	mean specflatness2
mean specvariation	mean speccrest4
mean specflatness1	var speccent
mean spectilt	mean spectilt
mean specspread	mean specspread

1. For the separated notes, features that relate to low frequency content, for example, 'loudnessBark1', 'loudnessBark2', 'loudnessBark3', 'mean log2f', "mean Ebands1', 'var Ebands1', appear more often than those relating to higher frequency energy. This is not the case for the original note segments, where for example, 'loudnessBark23' and 'loudnessBark24' are very dominant features.

2. There is very little importance assigned to temporal features. Of the best features, the only ones directly encoding any temporal information are: 'zcr', 'mean specvariation', 'var Ebands1', 'var Ebands2', 'var specflatness1' and 'var speccent'. However, none of these are part of the subset of temporal/dynamic features (section 6.5.3).

3. The lower MFCCs tend to be more important than higher MFCCs, which suggests that the smooth shape of the amplitude spectrum is more important than its detail.

Most of these results are straightforward to interpret. With regards to the first observation, it is generally more difficult to separate higher frequency harmonics from the recording than lower harmonics. Thus, although higher frequency content is important in the original note segment case, the separation of this high frequency content may not have been entirely reliable, resulting in lower-frequency content being relatively more important for discrimination in the separated note case.

The lack of importance of the temporally-based features can be explained by the fact that the amplitude envelope describes the shape of the polyphonic waveform rather than the shape of a single source. Almost all of the note samples contained interference from other sources, creating unusually shaped note amplitude envelopes, and there was also a large amount of dynamic and expressive variation between notes belonging to the same source. It could also be that temporal features are generally not very useful even in clean sample conditions, which is also an issue arising in the timbral discrimination literature, which was reviewed in section 6.1.

The third point puts an emphasis on the smooth shape of the spectrum rather than the detail. The detail of the spectral shape may to some extent be masked by other sources, whereas the general shape of the spectrum is not as sensitive to interference. An alternative explanation could be that the source is typically producing a rapidly time-varying sound. It may take a more sustained or stationary sound for spectral detail to be discernable. However, test conditions in which samples are clean, excited in a conventional manner, of relative long duration and are allowed to decay naturally, are simply not reflective of music as a whole.

6.7 Clustering method

Once a subset of d features has been selected, the next task is to cluster unseen test data in the d-dimensional feature space into sources/classes. In a fully automatic system we would probably not know the number of sources in a recording *a priori*. So ideally, what is required is a clustering algorithm that cannot only find natural clusters in the FV data, but will also estimate the number of these clusters. Thus, the task is an unsupervised clustering problem. Training data was used in the previous section only for the initial feature selection, and the training and test data have been kept separate. The note samples contain interference, and on top of this there is a range of expressive variation amongst notes from the same source. As a consequence, it is generally not true that the clusters from the different sources are well-separated and compact.

6.7.1 Model-based clustering

A number of approaches to unsupervised clustering are described in [223] including maximum likelihood and Bayesian methods, hierarchical clustering and self-organising feature maps. It was decided to use model-based clustering[2, 8] for the following reasons. It allows objective comparisons to be made between the performances of different cluster models and numbers of clusters using the Bayesian information criterion. It is a fairly intuitive model of the data, comprising a finite mixture of multivariate normal distributions, and model parameters can be estimated efficiently using the expectation-maximisation (EM) algorithm.

The Model-Based Clustering Toolbox[2] performs clustering based upon the following model. A random FV, $\mathbf{x}$, belonging to a cluster k, has a single multivariate normal probability distribution centred on the mean of this cluster μ_k:

$$p_k(\mathbf{x}|\mu_k, \sum_k) = \frac{\exp[-\frac{1}{2}(\mathbf{x}-\mu_k)^T \sum_k^{-1}(\mathbf{x}-\mu_k)]}{(2\pi)^{d/2} \mid \sum_k \mid^{1/2}} \tag{6.8}$$

where $\sum_k$ is the covariance matrix for the k^{th} cluster. The covariance matrices can be parameterised in terms of an eigenvalue decomposition[224]:

$$\sum_k = \lambda_k D_k A_k D_k^T \tag{6.9}$$

where D_k is the matrix of eigenvectors, A_k is a matrix with the eigenvalues of $\sum_k$ on the diagonal, and λ_k is a scalar determining the volume of the ellipsoid. In the most general form above, each cluster is an ellipsoid with a unique volume, orientation and shape, resulting in a large number of parameters that should be estimated from the data. A number of simpler parameterisations of the covariance matrix can be used to simplify the parameter estimation problem at some detriment to accuracy. Four of these have been implemented in the MBC

Table 6.4: Covariance matrices implemented in the MBC Toolbox[2].

Covariance matrix $\sum_k$	Description
λI	Distributions are spherical and have the same volume
$\lambda_k I$	Distributions are all spherical but have different volumes
$\lambda D A D^T$	Distributions are ellipsoid, but all of the same size and orientation
$\lambda_k D_k A_k D_k^T$	Distributions are ellipsoid and have unique orientations and sizes

Toolbox and their covariance matrices are specified in table 6.4. A model of each cluster as a single multivariate Gaussian may at times be over-simplified. For example, an instrument can sometimes be played with several different styles, e.g. staccato, pizzicato, legato, which would suggest that a single multivariate Gaussian distribution is an insufficiently complex model to account for these distinctive timbral classes. However, although we have not deliberately chosen it so, there is no obvious evidence of any sources in the sample database using multiple playing styles.

Given that the source labels for the data $\mathbf{x} \in X$ are unknown in unsupervised clustering, we are required to find the set of parameters that maximise the likelihood for a Gaussian mixture model (GMM):

$$\prod_{j=1}^{n} \sum_{k=1}^{N_c} \tau_k \; p_k(\mathbf{x}_j | \mu_k, \sum_k) \tag{6.10}$$

where τ_k is the mixing proportion (the probability that a sample is from class k), n is the total number of FVs, and it has been assumed that the data are statistically independent samples from the GMM. An EM algorithm usually converges to the maximum likelihood estimate given an initial estimate of the component parameters and mixing proportions, which is obtained using an agglomerative hierarchical clustering algorithm. It also requires that the number of clusters and model for the covariance matrix be specified. The reader is referred to [2] for more information on the EM optimisation process.

A convenient method for comparing the clustering performance of different model types and number of clusters is provided by the Bayesian information criterion (BIC). This is the value of the maximum log likelihood plus a penalisation term that depends on the number of parameters in the model. By maximising the BIC we arrive at a solution that finds the closest fit to the data with the least amount of complexity. This solves the familiar overfitting problem, that when the parameterised model becomes more complicated, it fits the data more accurately, but loses its potential to generalise to unseen data.

6.8 Note grouping results

Note grouping/clustering experiments were designed with the following questions in mind:

1. Is the validity of this clustering approach, which treats each note as an isolated entity, justified by results?

2. What is a reasonable number of features/dimensions for clustering? (This is likely to have some dependence on the particular clustering algorithm.)

3. Is it better to calculate features on the segment of the recording containing the note, or on the harmonic content of the note that has been separated from the recording? Related to this, is there any reason to believe that 'separation' can be an aid to 'understanding' (as discussed in the introduction to this chapter)?

It was decided from the point of view of most intended musical applications, and due to practical limitations, that the clustering performance should be measured for mixes of between $N_c = 2$ and 6 sources. In other words, the results are more reflective of the expected performance on a piece of popular music (e.g. a rock group) than the almost impossible task of a symphony orchestra. In order to maintain a clear separation between the data used for feature selection and that for testing the clustering algorithm, in each iteration of the experiment, the notes from the remaining (20-N_c) sources were used as input to the feature selection algorithm. The clustering experiment was repeated 100 times for each value of N_c, randomly choosing N_c sources out of the 20 sources in each iteration. The average number of note samples used in the clustering experiment was therefore $n\frac{N_c}{20}$ where $n = 574$ is the total number of samples in the database. Results are given for both of the following cases: the number of clusters is unknown to the clustering algorithm, and the number of clusters is provided *a priori*.

To answer the third point in the list above, three sets of results were obtained. The first measures the clustering accuracy when all 185 features are computed on the original segment of the recording containing the note. In the second, a reduced set of 141 features is computed on the original segments. The reduced feature set consists of all features that do not require any intensive parameter estimation, which means we exclude features that depend on the harmonic frequency trajectories. There are two justifications for using a reduced feature set: in an automatic system it may not be possible or may be computationally expensive to estimate the harmonic trajectories accurately, so it would be desirable to use only those features that can be reliably and easily computed. Secondly, we wish to see if features that point to an underlying symbolic or structured content (i.e. harmonic trajectories imply the existence of pitch) are of benefit to the clustering task. The philosophy in [47] is that humans are able to perform various functions in response to a musical stimulus without the need for a highly structured or symbolic internal representation of the sound. Perceptual grouping of notes from a common source is indeed a function that humans are capable of.

It would be interesting then, to apply this reasoning to machine recognition of music. In other words, how important could an implicit structured/symbolic representation be to note grouping in an automatic system? This has an implication for music content description: if there is not much to be gained in extracting a symbolic representation of music, then this makes various functions in music content description or information retrieval much easier, e.g. segmenting a recording according to instrument type. Finally, the third set of results computes the full set of 185 features on the note waveforms whose harmonic content has been filtered from the original recording. The same feature selection algorithm was used in all three cases. Table 6.5 summarises the six possible test scenarios:

Table 6.5: Conditions for feature computation for the six sets of results.

Case	N_c known *a priori*	Features computed on	Symbolic features allowed?
1	no	Original waveform	yes
2	no	Original waveform	no
3	no	Separated waveform	yes
4	yes	Original waveform	yes
5	yes	Original waveform	no
6	yes	Separated waveform	yes

The clustering accuracy will be defined as the percentage of notes that have been grouped into the correct cluster. As the clusters determined using unsupervised clustering do not have any labels assigned to them, there is a slight permutation problem, meaning that it is not at all obvious which cluster the 'correct' cluster is. The problem is solved by permuting cluster labels. Denote y_k as the known source label of note k, which is in the range $[1, N_c]$, and let y'_k be the nominal label of the cluster which it gets identified with in the range $[1, C]$, where the number of clusters determined is not necessarily equal to the number of original sources. The correct labelling of clusters is determined by finding all permutations of the cluster labels over a range $[1, \max(N_c, C)]$, and choosing the permutation which gives the maximum overall clustering accuracy:

$$\text{Clustering accuracy (\%)} \quad = \quad \frac{1}{n} \sum_{k=1}^{n} d_k \tag{6.11}$$

$$\text{where} \quad d_k \quad = \quad \begin{cases} 0 & ; \quad y_k \neq y'_k \\ 1 & ; \quad y_k = y'_k \end{cases}$$

The permutation algorithm has the statistical effect of increasing the baseline accuracy, which is the accuracy that would result if clustering were completely random. Thus, the true baseline accuracy has been calculated numerically for the all values of N_c, both when N_c is known and unknown in advance. In the latter case an assumption is made that all values of C in the range [2,6] are equally likely to be determined using the clustering

algorithm, although in practice there is a tendency for C to be chosen as 2 or 6 more often than 3, 4 or 5.

Table 6.6 shows the average clustering accuracy when the number of clusters/sources is unknown (cases 1–3 in table 6.5), and table 6.7 gives the results when N_c is provided *a priori* to the clustering algorithm (cases 4–6 in table 6.5). At the bottom of tables 6.6 and 6.7 the clustering accuracy is averaged over all values of N_c. Figs. 6.3 and 6.4 display the general characteristics of the clustering performance for the separated note cases (cases 3 and 6 in table 6.5).

6.9 Discussion

The main observation of the results is that there is a clear advantage in computing features on the original note segments, both when N_c is unknown (cases 1–3 in table 6.5) or when N_c is given (cases 4–6 in table 6.5). Secondly, although their difference is not as large, there is definitely some gain to be had in using the full feature set (cases 1 and 4 versus cases 2 and 6, respectively). Consequently, when N_c is unknown and the full feature set is used, the average clustering accuracy for the original note segments is around 10–16% higher than for the separated notes. Even when the reduced feature set is used for the original note segments, results are 7–13% higher than in the separated note case. Similarly, when N_c is known *a priori*, clustering is 9–15% better for original note segments, or 9–12% better for the reduced features calculated on original note segments.

To compromise between clustering accuracy and computational efficiency, the latter of which decreases steadily as the number of features increases, the best feature space has around $d = 10$ dimensions. For this dimensionality, the full feature set calculated on the original note segments gives the best results averaged over all values of N_c. Notably, around 74% and 82% when N_c is unknown and known, respectively.

Can any conclusion be made with regard to the importance of structural/symbolic features (i.e. features depending on harmonic trajectories)? From the point of view of feature selection, if there are sufficient training samples available so that the robustness of the feature selection algorithm is capable of excluding all features that do not provide any discriminatory power, then it can be of no detriment to include extra symbolic features. The results demonstrate that the harmonic-dependent features implemented provided a small improvement in performance of around 1–4%. Given the general absence of harmonic features from the feature subsets in tables 6.2 and 6.3, it is not surprising that the difference in accuracy is moderately small. Even so, we expect that the parameterised harmonic content could have been condensed into a more efficient feature set, which may have made a more marked difference to clustering performance, although further tests would be necessary

Table 6.6: Clustering performance on separated notes when N_c is unknown (cases 1–3 in table 6.5).

Nc	Baseline	Dimensionality	Case 1	Case 2	Case 3
2	43.5	3	86.4	83.6	78.1
		5	89.6	84.0	70.2
		10	86.9	86.9	59.7
		20	83.8	83.8	54.1
3	39.3	3	76.9	77.0	65.9
		5	80.5	77.4	64.0
		10	80.0	77.8	62.0
		20	73.8	73.3	62.1
4	35.1	3	71.3	69.9	56.2
		5	72.1	68.3	57.5
		10	72.8	67.4	61.0
		20	66.0	63.9	63.0
5	31.3	3	62.9	61.0	52.2
		5	66.0	63.1	54.3
		10	65.3	62.2	55.3
		20	58.1	55.4	58.7
6	27.7	3	55.1	55.4	48.2
		5	59.5	58.3	48.6
		10	63.6	58.2	51.4
		20	57.0	50.5	52.7
Average		3	70.5	69.4	60.1
		5	73.6	70.2	59.0
		10	73.7	70.5	57.9
		20	67.8	65.4	58.1

Table 6.7: Clustering performance on separated notes when N_c is known *a priori* (cases 4–6 in table 6.5).

Nc	Baseline	Dimensionality	Case 4	Case 5	Case 6
2	59.0	3	91.6	90.8	83.3
		5	93.7	92.0	81.8
		10	92.9	89.8	81.3
		20	92.5	90.8	82.0
3	45.5	3	77.5	79.0	70.7
		5	85.2	82.4	71.0
		10	86.5	83.7	73.9
		20	83.4	84.4	71.5
4	38.5	3	72.6	72.0	60.8
		5	78.1	74.5	62.6
		10	81.2	75.2	65.2
		20	79.6	79.2	66.1
5	34.2	3	66.7	64.4	56.2
		5	73.3	70.9	58.2
		10	74.4	71.5	59.5
		20	74.4	73.5	62.3
6	30.9	3	61.8	60.1	52.0
		5	70.3	67.9	53.5
		10	75.1	70.6	56.1
		20	71.5	68.4	59.1
Average		3	74.0	73.3	64.6
		5	80.1	77.5	65.4
		10	82.0	78.1	67.2
		20	80.3	79.3	68.2

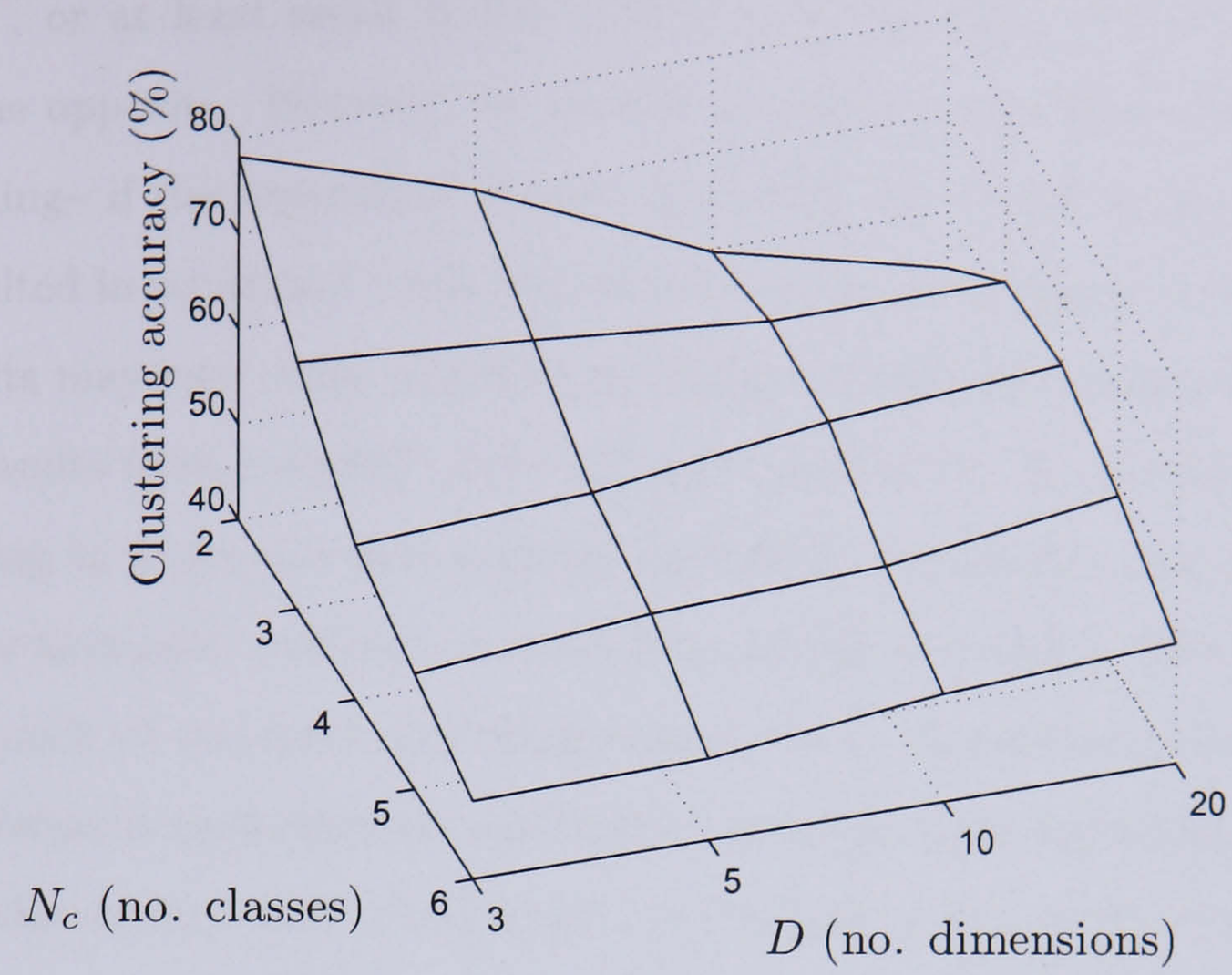

Figure 6.3: Note clustering accuracy calculated on separated notes from the recording and given the full feature set. N_c is not given to the clustering algorithm.

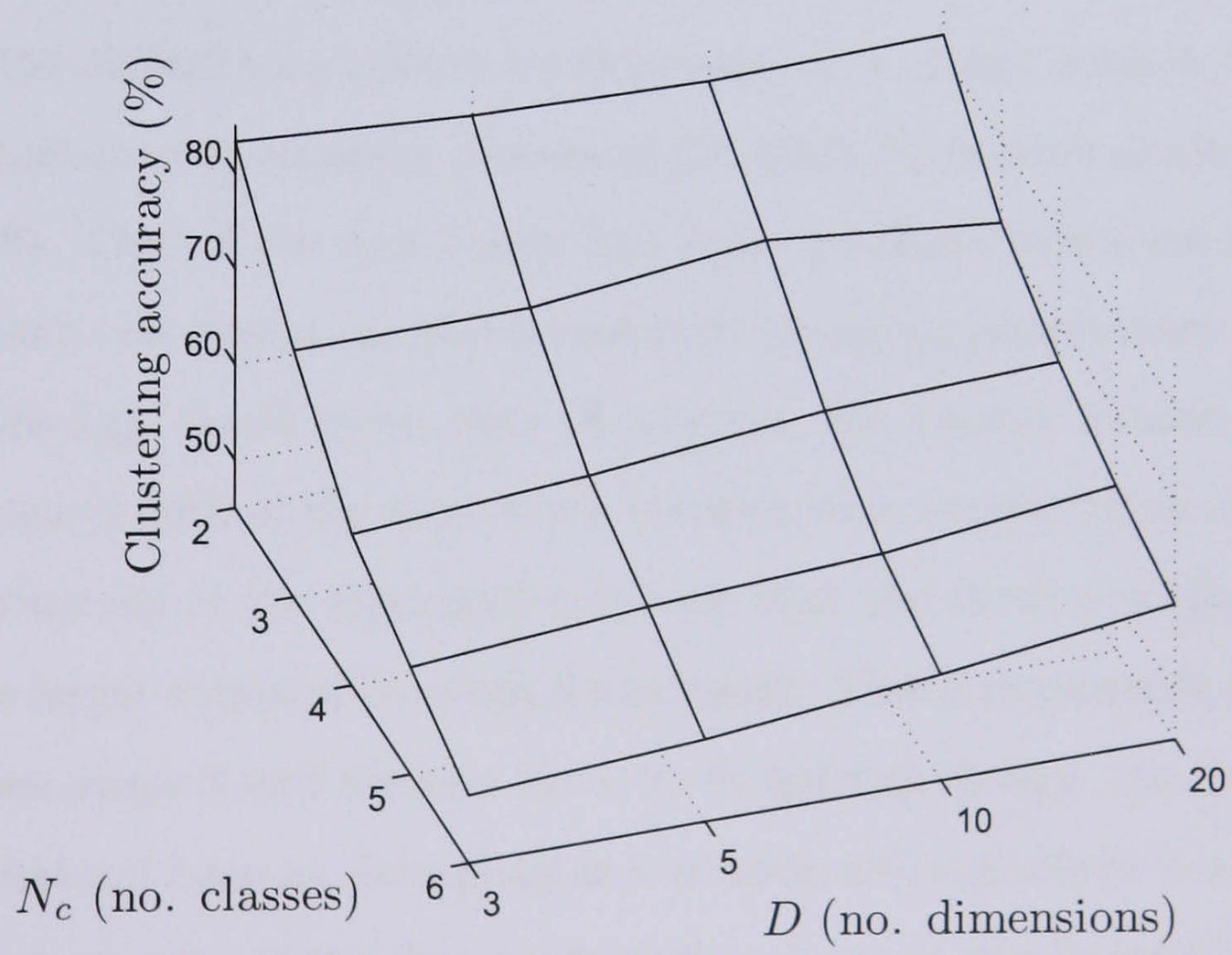

Figure 6.4: Note clustering accuracy calculated on separated notes from the recording and given the full feature set. N_c is provided *a priori*.

to evaluate this hypothesis.

With regards to the question of whether separation of the harmonic content might aid 'understanding', or at least assist in the task of note grouping, the results quite clearly demonstrate the opposite. However, we cannot conclusively say that separation does not aid understanding– if the separation of each note from the recording had been performed better and resulted in separated notes that were closer to the original unmixed sources, the clustering results may have been significantly better on separated notes. However, we can look at these results from a slightly different angle. In chapter 4, the motivation for using harmonic filtering in preference to sinusoidal modelling of harmonics was that we intended to filter out the harmonic content of a note from a mixture, rather than introduce signal artifacts as a result of incorrect modelling assumptions. If clustering accuracy decreases when note clustering is performed on a filtered component of the signal, then there must be information in the residual that is important for this task, which we have neglected to filter out. This additional information may be contained in the transient or noise component of each note. This hypothesis could be tested by extracting a transient component for each note, and adding this to the separated harmonic content of each note before feature computation. One other possibility which we have not explored at all, is to approach the problem not with the aim of separating the note from the mixture, but with the aim of suppressing any interfering sources in the signal. This way we might retain all the useful note content, but improve the reliability of some features, especially the temporal/dynamic features, which are vulnerable to strong interference.

Comparing the clustering accuracy between cases 1, 2, 3 and cases 4, 5, 6 respectively, there is an improvement in accuracy of around 8% when N_c is given at the optimal dimensionality ($d = 10$). Thus, if the user knows how many pitched sources are contained within the recording, they can expect an improvement in grouping performance. This difference is actually surprisingly small given that on average, the correct number of clusters was detected only around 30% of the time when features were computed on original note segments. It was observed in the separated note case that the clustering algorithm generally tended to choose larger values of N_c when d was larger. This is responsible for the difference in results between cases 3 and 6 when $N_c = 2$. In general though, the so-called 'curse of dimensionality' has not been as distressing as it sounds, although there is a slight indication in the results that sparsity of the data is starting to become problematic for $d = 20$.

On the whole, the clustering performance is a large margin above the baseline results in tables 6.6 and 6.7. This demonstrates that the note grouping approach is well founded, but there is still some room for improvement. Although we could expect moderate gains in accuracy through various refinements of the feature selection and clustering algorithms,

increasing the size of the training data and extensive testing, it is not felt that this effort would be worthwhile without making concurrent improvements in the following two main areas. Firstly, it would be useful to define a better set of features suitable for polyphonic signals. The presence of interfering sources means that some features lose much of their discriminatory power. Human listeners are able to concentrate on certain characteristics of a source, and ignore uncorrelated content arising from interfering sources. This selectivity is lacking in the current feature set, but it seems a challenging task to correct this. On the one hand, we could argue for more complex or structured features with some spectral and/or temporal selectivity, on the other hand, we would not like to lose generality by unjustifiably imposing models derived from the training set onto a set of unheard instruments in the test data. The second area for improvement which has not been explored in any great detail is the inclusion of musical contextual information. For humans, the integration of contextual information is certainly an aid in a timbral association task. The notes within a continuous musical phrase tend to converge to the same perceptual stream. Ultimately, these source clustering methods are seen as being most appropriately used as an aid for extracting instrumental parts from recordings, mostly at times when uncertainties in the integration of contextual information, such as at the start or ends of musical phrases, would benefit from a greater reliance on lower-level source properties.

Upon reflection, there is an ambiguity in this study that might have been avoided through a wiser choice of sample material. That is, it is difficult to say whether the feature selection algorithm has learnt to find features that class all notes within the same recording as being similar, or features that class notes from a common source but in different musical contexts as being similar. For example, suppose that all piano note samples were taken from a section in the recording where percussion was also playing. Do we expect that piano notes in a different section of the recording without percussive accompaniment would be recognised as being the same instrument? Ideally we would like to select features that are specific to each source type, rather than the context within the recording. It would thus have made sense to choose more examples for training in which several instruments were extracted from the same recording. This way, the selected features would probably be more source specific, i.e. better at discriminating different sources within the same recording environment.

6.10 Conclusions

This chapter described one aspect of an automatic system for de-mixing mono recordings into separate pitched sources or instrumental parts. This work is also relevant to the problem of automatic music transcription (AMT). Transcription is often viewed as being

simply a combination of pitch tracking and onset/offset detection. However, in the case of multiple sources in a recording, if a human was to transcribe a piece of music without separating the notes produced by each instrument into different parts, this would generally not be considered a proper transcription. Thus, the association of transcribed notes with sources is an important component of AMT.

The tools that were presented in chapters 4 and 5 for note separation are intended for use with an AMT-based system or one where the written/symbolic score is aligned with audio (section 3). To be useful for most of the intended applications of this system (section 1.1), the separated notes are required to have at least nominal class labels. Hence, the intent in this chapter was to develop a system for forming natural groups of individual notes within the polyphonic recording, where notes within a single group ideally arise from the same source. It has been emphasised that this is not an instrument classification problem, as it would be impossible to define a limited set of instrument types that are representative of music as a whole.

The system for note grouping is based upon finding subsets of notes which are similar to each other, where 'similarity' is related to the distance between samples within a multidimensional feature space. An obvious issue is to determine what are the axes of this feature space. This is basically the same concern that underlies numerous timbral discrimination studies (reviewed in section 6.1), where timbre is viewed as a multidimensional attribute of audio perception. Whilst our problem is not strictly speaking restricted to notes with equal pitches and loudnesses, and our intention is focused on practical applications rather than an understanding of perception, there are still commonalities between the two approaches. A second area of research that is of relevance is musical instrument classification (section 6.2). Although we are not implementing a classifier, the field is of interest mainly in terms of identifying effective features for music, that are at least partially invariant to expressive and other kinds of variation, which can be observed even between notes from the same instrument and recording environment.

This chapter described the design, computation and selection of features that make up the multidimensional feature space, and the clustering algorithm that has been used for forming groups of notes within the feature space. The clustering algorithm is unsupervised, and in the absence of prior information, must also determine the number of source types. It also has some selectivity concerning the shape of the statistical models of these clusters. Issues affecting clustering accuracy were addressed, such as dimensionality, the use of symbolic features, and whether to calculate features on note segments within the original recording, or on the filtered harmonic content of these notes. It was found that a 10 dimensional feature space was appropriate, symbolic features demonstrated a gain in

clustering accuracy of 1–4%, and the performance increased by around 8% using original note segments at this dimensionality. We have left the integration of musical contextual information as a topic for future research, and expect significant gains in note grouping accuracy to be obtained even through simple melodic and harmonic rules.

Chapter 7

Conclusions and Further Work

The thesis has argued for an empirical and heuristic approach to source separation from single-channel polyphonic real recordings. It has focused mainly on separating pitched notes, although much of the work on note clustering and transient and noise extraction is relevant to other kinds of sounds. We have selected source material from a wide range of musical genres and contexts, keeping in mind a host of potential applications in music information retrieval, source spatialisation, restoration and others in production and effects processing. It has been important not to be reliant on source-specific information or prior knowledge of the recording, as this would restrict the application of these methods to a few applications where this information is readily available. Although MIDI information has been used to provide note timing and pitch information, this would eventually be replaced by an automatic music transcription system, resulting in a completely automatic source separation system.

Firstly, we outline the modelling framework and objectives of the source separation system. A source, which may be a real or synthetic instrument, plays a set of notes, which we define as finite durations of relatively stable pitch, and these are scattered throughout the recording and mixed with notes from other sources. The goal is to separate all audio content belonging to the set of notes produced by a particular source from the recording. By the term 'separation', it is not only implied that each separated source sounds perceptually similar to its unmixed original, but also that the residual after subtraction of a particular source contains little remaining trace of this source and is free of disturbing artifacts of the separation.

Clearly, a note is in general a complex sound event, containing both deterministic content and having non-stationary and stochastic behaviour. A predominant feature of a pitched note is the large amount of energy contained within harmonics, which are often distinguishable from the harmonics of other notes. It is no surprise then that a lot of effort in speech and music signal separation and restoration has been directed at extracting the

harmonic content of a desired source from a mixture. Some developments in this area were presented in chapter 4. One focus of this chapter was the separation of overlapping harmonics, which is more pertinent to music separation than speech separation due to the tendency for harmonically related notes to be played together. It was found in test mixes of pitched notes, that the harmonic filtering methods developed here usually produced quantitatively better separation results than sinusoidal modelling techniques.

In the hypothetical situation that a perfect harmonic separation algorithm exists, it is still necessary to consider the non-harmonic content of each note that would still remain in the recording. The separation of non-harmonic content from polyphonic music is a relatively unexplored area. This work was conducted within a framework that has become fairly wide-spread in the area of music signal processing, in which the music signal is considered to contain three main types of content: partials, transients and noise. 'Transients' or 'transient events' typically occur at the start of an excitation, and are characterised by sharp increases in broad-band energy. Everything that is not partial or transient content is treated as noise. Although it is slightly misleading to treat partials, transients and noise as distinct components, this is still a generic and useful characterisation of many kinds of musical signals. Chapter 5 presented work towards extracting the transient attack of a note from a mixture, in both the cases where the transient is isolated, and where the transient is overlapping in time with other transient events. It also described a method for separating the spectral noise content of a mix of overlapping notes, which relied upon the assumption that the harmonic and noise content of a note are correlated in both the temporal and spectral domains. This assumption seems to have some validity, both from the perspective of the sound production mechanisms in acoustic instruments, and based upon general observations of the spectra of musical signals.

Lastly, in chapter 6 we translated a series of techniques for separating individual notes from a recording into a system for separating complete sources from a mixture. As the notes produced by each source are scattered all over the recording, and as an automatic transcription system is unlikely to provide a source label for each detected note, some way of categorising these notes into sources is required. This is not a source or instrument classification task, as basing the system upon a limited set of predefined instruments would restrict the usefulness of the system to only certain musical contexts. In music as a whole, we might as well consider that an infinite number of instruments or sources exist, particularly when dealing with synthetic sounds, and considering that the 'instrument' we listen to in a recording is a product not only of the acoustic excitation, but also of the recording environment and post-production. Thus, an automatic note grouping system should seek to identify information that is generically useful for discriminating between different

source types. In this respect, it has much in common with timbre description studies that attempt to find physical sound attributes that are correlated with perceptual judgements of timbral similarity. A small subset of features (3, 5, 10 or 20 features) was identified for source discrimination by applying a feature selection method to a larger set consisting of 185 features, given a training database of 574 labelled note segments from commercial recordings. The note grouping accuracy is the percentage of notes that are identified with the correct cluster/source. The accuracy obtained when grouping sets of notes from 2, 3, 4, 5 and 6 sources was 60%, 62%, 61%, 55% and 51% when the number of sources in the recording was unknown, and 81%, 74%, 65%, 60% and 56% when the number of sources was provided *a priori*, when the number of features was chosen as 10. This system makes no use of musical contextual information, treating individual notes as completely isolated entities, whereas in reality they fit into a musical context with other notes. Musical context potentially provides strong cues that would aid note grouping, such as the tendency for notes in a musical phrase to be played by the same instrument, or percussive sounds to occur at roughly the same position within each bar.

7.1 Further Work

Due to the interdisciplinary nature of research in music processing, which combines elements of signal processing, music analysis, statistics, physical modelling, perception, psychoacoustics and other disciplines, there are ample opportunities for lateral diversions into relatively unexplored territory. What follows is a list of a few specific areas in the main chapters that may be fruitful to investigate further, and could provide improvements to the current implementation of the source separation system. After this, some general criticisms of the reductionist view of music embodied in this work and suggestions for a more flexible and integrated approach will be given.

Chapter 3 The source separation system is dependent on a pre-processing stage that aligns prior note pitch and timing information in the form of a MIDI score with the recording. A fully automatic system would, however, require note pitches and timing information to be inferred from the actual recording. Thus, an obvious practical extension of the system would be to replace this pre-processing with an automatic music transcription (AMT) system, or better yet, to develop a switchable pre-processing system that is capable of reading MIDI or any other score-like input, and in the absence of this prior information, reverts to AMT mode. If the score or MIDI information differs in tempo from the recording, which is typical in many score-following applications, the current MIDI-to-audio alignment system would need to be adapted to perform dynamic time-warping. This improvement could be implemented within the same or similar dynamic programming framework that has already

been chosen. Score following is an area of active research with the two main approaches to time-warping being through the use of dynamic programming[94, 95, 96, 97] or dynamic time warping algorithms[98] and hidden Markov models (HMMs)[99, 100]. One weakness of the pre-processing stage arises from the assumption that music consists of quasi-stationary pitches over short durations in time. Although pitches in the score were allowed to adapt slightly to align themselves with the recording, which was sufficient to account for vibrato and small FM effects, we were not able to adequately separate notes with a large change in pitch over their duration, such as notes played in a glissando style, or melismatic music (a sequence of several notes sung to one syllable of text, as in Gregorian chant). It seems that a better description of a pitched sound is as a pitch envelope over time, where the essential requirement is time continuity of the pitch contour rather than adherence to a static pitch.

Chapter 4 A couple of small improvements could be beneficial to the harmonic filtering stage. Firstly, it would be useful to include a method that decides how many harmonics to track for each note. In the current system it has been assumed that all notes have a preset number of harmonics, which has usually been set to 40. However, it has been observed that for some instruments, especially bass instruments, only the first few harmonics can be detected in the mixed recording. If we refrained from tracking harmonics above the highest-frequency harmonic of each note which we could be confident did not actually arise from noise or another source, we could avoid misidentifying some higher frequency partials as the higher harmonics of low pitched tones. The effect of setting a fixed number of harmonics is a tendency for separated lower pitched notes to accumulate more energy than higher pitched notes. One solution that has been briefly experimented with but requires further development and evaluation is a HMM approach using the Viterbi algorithm, which tries to find the optimal change-point below which all harmonics are significant, and above which all harmonics contain negligible energy. A second possibility for further development in this chapter is to adapt the harmonic extraction stage to combine the different strengths of both harmonic filtering and sinusoidal subtraction. An intelligent switching mechanism that performs sinusoidal subtraction of non-overlapping harmonics generally at low SNRs and harmonic filtering at high SNRs is envisaged. Finally, it is acknowledged that a difficultly lies in suppressing percussive interference when separating harmonic content, but we are unable to offer any effective solutions to this problem. The sharp increase in broad-band noise at percussive/impulsive onsets completely overlaps all partial content in the spectral-domain, and as this noisy interference is rapidly time-varying, it is very difficult to apply established noise subtraction methods on a sufficiently short time scale.

Chapter 5 This chapter presented three methods focused on different aspects of a rather difficult task: the extraction of the non-harmonic content of a note from a polyphonic

recording. Due to the fact that these methods are fairly novel, and have consequently not benefited from extensive testing and refinement over time and comparison with alternative methods, it was not deemed to be an opportune moment to integrate the three methods into a single system, although this may be a longer term goal. However, some more detailed avenues for further work can be identified. A more accurate method for calculating event onset times would be beneficial to the autoregressive (AR) transient separation algorithm. The most realistic transient separations occur when the starting point for the forwards AR interpolation is at most 10 ms before the true transient onset. Secondly, in relation to the band-wise noise interpolation method for separating overlapping transients, it was already mentioned that a global estimate of the noise envelopes of each transient in the overlapping region would be preferable to the current iterative-subtractive method. We cannot offer any immediate solutions to the case where overlapping transients do not contain uniformly decaying noise envelopes, without introducing prior knowledge of the source characteristics. Within a more focused musical context, however, the inclusion of prior information into a transient separation system may be a fruitful area for future work. With regards to the last section on extracting the noise content of a note from a polyphonic mix, additional studies of auditory processing would be needed to evaluate what characteristics of the spectrum are most important for our perception of 'noisiness' or 'naturalness' in an instrument tone. If these characteristics are indeed correlated in some way with harmonic content, it has been shown that it may be possibile to partially extract this noise component from the mixture.

Chapter 6 We discussed a basic mechanism for grouping notes into sources, which allowed the number of sources in the recording to be estimated. Some obvious directions forward are to conduct further experiments to provide more quantitative evidence of the efficacy of the various features, to see whether alternative criterion functions for feature selection such as the clustering error rate lead to better feature selection, and to test other clustering methods. With regards to the first point, it is of immediate interest to redesign the harmonic feature set, as surprisingly, these features did not emerge as being very useful for source discrimination. A number of alternative unsupervised clustering methods[223] could be tested, such as other statistical parameterised models, hierarchical clustering and self-organising feature maps, although it is expected that greater gains in accuracy can be achieved in the immediate future by reformulating the feature set. If a more flexible description of music in terms of pitch envelopes rather than quasi-stationary pitches is adopted, some of the features will require modification anyway. Whether it would be advantageous to extract, in addition to the harmonic content, a transient or noise component for each note, and then to compute features on the mixture of these components, is an open question. As discussed in this chapter, a larger and more appropriate sample

database may also enlighten us as to whether the features learnt in feature selection are indicative of some background information in the recording, or exclusively of the actual notes within the recording.

One potential extension of this work that has relevance to all of the above areas, is the integration of spatial information in multi-track recordings into the current structural framework. As spatial information is not always available or easily intelligible, a basic separation system based upon structural information is envisaged, that is augmented with a spatial de-mixing system when reliable spatial information is available.

We conclude with a general remark relating to music content description. On the one hand, from the perspective of auditory scene analysis, it may be that a structural or symbolic representation of music has no great value for modelling audio perception[47]. On the other hand, some kind of reduction of the music signal into components is a practical requirement for the various applications listed in section 1.1. In the latter case, the objective is not necessarily to emulate the auditory system, but to produce perceptually believable results. It has already been argued that without even taking into consideration non-pitched sounds, it is rather limiting to describe music as a sum of quasi-stationary pitched notes. So the question arises: what kinds of structures would be flexible, amenable to analysis, and perceptually representative of the signal? One example is a pitch envelope, which has already been proposed as an alternative to a note with relatively static pitch. These still contain onset and offset times, and would be constructed using temporal periodicity-continuity constraints. Aside from this, it seems that the Gestalt grouping cues involved in audio perception (e.g. common onsets, temporal continuity, common amplitude and frequency modulation, harmonic coherence of partials, and other higher level processing such as pattern repetition), of which an excellent account is given in [61], have not yet been exploited to their full potential. The incredible robustness and speed of the auditory system give reason to believe that these cues are an excellent source of inspiration in the construction of a representational framework for audio. In other words, each so-called object or structure within the representation should maintain a certain cohesiveness when perceived by the listener, i.e. it should consist of a partition of temporal and spectral content that fuses through auditory processing into what is perceived as a single event or auditory stream. Actually, this idea has been used, for example, in studies where partials have been grouped into 'notes' based upon auditory grouping cues (section 2.2.1). However, there is definitely room for investigation into incorporating non-partial content into the framework, and in finding a way to fuse partial and non-partial content.

One can argue that the fact that the results of systems broadly classified as music content processing are often judged by 2-dimensional visual inspection before listening,

using a spectrogram for example, is proof that we have not yet made the best use of the information contained within our time-frequency representation. One hopes that the structural information and patterns that can be discerned within a visual display such as a spectrogram, will correspond to emergent objects or structures within an auditory or otherwise inspired audio signal representation.

The above discussion provides some speculations about the future theoretical developments of this work, identifying auditory-inspired audio processing as an interesting focal point. With regards to commercial exploitation of these ideas, further technical developments are likely to be determined by specific needs and the availability of prior information concerning the recording/s.

Appendix A

Detail on feature computations

This appendix describes in greater detail the methods for calculating the features used in chapter 6 for note clustering. The different categories of features (and number of feature dimensions per category) are: waveform features (1), harmonic features (39), temporal/dynamic features (13), spectral features (48), energy features (13), perceptual features (64) and MPEG-7 features (7), making up a total of 185 feature dimensions. They are categorised into two types: global features which are computed over the entire note, and frame-by-frame (FBF)-based features computed in each time frame. The information in each FBF feature is summarised by three global features: the mean, variance and slope of the FBF feature over all time frames for a particular sample. The variance and slope encode basic characteristics of the time-varying behaviour of the FBF feature. A multiplicative factor is given in brackets following each feature name, indicating the dimensionality of each class of feature. 'tristimuli', for example, has a (3) besides it, as it actually consists of three components: 'first tristimulus', 'second tristimulus' and 'third tristimulus'. Thus, if the means, variances and slopes of each of these were found, there would be a total of 9 features. Most features have been taken or adapted from these sources: [195, 212, 213, 216].

Symbols:

$$
\begin{aligned}
s[n] &= \text{note waveform } (n = 0, \ldots, L-1) \\
x_{env}[n] &= \text{note amplitude envelope } (n = 0, \ldots, L-1) \\
r &= \text{time frame index } (r = 0, \ldots, R-1) \\
M &= \text{number of harmonics in note} \\
f_m[r] &= \text{frequency in Hz of } m^{th} \text{ harmonic in frame } r \\
a_m[r] &= \text{amplitude of } m^{th} \text{ harmonic in frame } r \\
f_0[r] &= \text{pitch in Hz (approximated as } f_1[r]) \\
\bar{f}_0 &= \text{mean pitch } (\tfrac{1}{R}\textstyle\sum_{r=0}^{R-1} f_0[r]) \\
\hat{f}_m[r] &= \text{relative harmonic frequency } f_m[r]/f_0[r] \\
\mathrm{mcr}(s[n], d) &= (\text{number of crossings of } d \text{ by } s[n])/L \\
A[k, r] &= k^{th} \text{ bin in frame } r \text{ of the amplitude spectrogram}
\end{aligned}
$$

A.1 Waveform features

This feature is calculated directly on the note waveform.

Zero-crossing-rate (1) zcr

The division by $\bar{f}_0$ attempts to remove the rough dependence of the number of zero-crossings on pitch. The zero-crossing-rate is affected by both the sample rate and level of background noise, although if these are constant across all samples, then these dependencies could be less important after pre-scaling of this feature (e.g. to zero mean and unity variance).

$$\text{zcr} = \frac{\text{mcr}(s[n], 0)}{\bar{f}_0} \tag{A.1}$$

A.2 Harmonic features

These features are calculated on the extracted harmonic frequency and amplitude trajectories.

Jitter (1) jitter

Jitter measures the stability of the fundamental frequency over time. It normalises the difference between the pitch in the current frame and the three-point moving average centred about it. It correlates well with voice roughness and 'breathiness' qualities[213].

$$\text{jitter} = \frac{\sum_{r=1}^{R-2} \left| f_0[r] - \frac{f_0[r-1]+f_0[r]+f_0[r+1]}{3} \right|}{\sum_{r=0}^{R-1} f_0[r]} \tag{A.2}$$

Shimmer (1) shimmer

Shimmer[225] has also been found to correlate with vocal properties. It is defined here as the relative variation of the sum of the harmonic amplitudes over time[213].

$$\begin{aligned} A[r] &= \sum_{m=1}^{M} a_m[r] \\ \text{shimmer} &= \frac{\sum_{r=1}^{R-2} \left| A[r] - \frac{A[r-1]+A[r]+A[r+1]}{3} \right|}{\sum_{r=0}^{R-1} A[r]} \end{aligned} \tag{A.3}$$

Mean-crossing rate of f_0 (1) mcrf

This provides a rough estimate of the FM/vibrato rate.

$$\text{mcrf} = \text{mcr}(f_0, \bar{f}_0) \tag{A.4}$$

Log_2 of f_0 (1) FBF log2f

The pitch is certainly a perceptually significant sound attribute, and has been converted to an octave scale to correspond with musical tuning scales.

$$\text{log2f} = \log_2(f_0[r]) \tag{A.5}$$

Log2 odd-to-even ratio (1) FBF oddevenratio

The odd-to-even ratio[213] measures the ratio of energy in odd harmonics to even harmonics.

It helps to isolate certain instruments which contain most of their energy in odd harmonics. e.g. clarinet, from others that have more equally-spread harmonic energies, e.g. trumpet. The fundamental component has been excluded from calculations in accordance with [195. 213].

$$\text{oddevenratio} = \log_2 \left\{ \frac{\sum_{m=1}^{M/2} a_{2m+1}[r]^2}{\sum_{m=1}^{M/2} a_{2m}[r]^2} \right\} \tag{A.6}$$

Harmonic centroid (1) FBF harmcent

The harmonic centroid is the 'centre-of-mass' or barycentre of the harmonic energy spectrum. It was expected that this would be a useful feature due to its similarity with spectral centroid. However, it is formulated here using a linear frequency axis rather than a logarithmic axis for the spectral centroid. Furthermore, as we might expect the harmonic centroid to be proportional to pitch as a very rough approximation, it has been written in terms of relative frequency, $\hat{f}_m[r]$, to avoid having too many features that behave similarly to pitch.

$$\text{harmcent} = \frac{\sum_{m=1}^{M} a_m[r]^2 \ \hat{f}_m[r]}{\sum_{m=1}^{M} a_m[r]^2} \tag{A.7}$$

Harmonic spread (1) FBF harmspread

This measures the spread of the harmonic spectrum around the harmonic centroid. In timbral discrimination tests, the feature described as harmonic spectral spread in [226] was interpreted as an alternative third dimension for the timbral space of [1].

$$\text{harmspread} = \frac{\sum_{m=1}^{M} a_m[r]^2 \cdot [\hat{f}_m[r] - \text{harmcent}]^2}{\sum_{m=1}^{M} a_m[r]^2} \tag{A.8}$$

Harmonic skewness (1) FBF harmskewness

This measures the asymmetry of the harmonic spectrum around the harmonic centroid.

harmskewness $< 0 \rightarrow$ there is more energy above the harmonic centroid,

harmskewness $> 0 \rightarrow$ there is more energy below the harmonic centroid.

$$\text{harmskewness} = \frac{\sum_{m=1}^{M} a_m[r]^2 \cdot [\hat{f}_m[r] - \text{harmcent}]^3}{\sum_{m=1}^{M} a_m[r]^2 \cdot \text{harmspread}^{3/2}} \tag{A.9}$$

Harmonic kurtosis (1) FBF harmkurtosis

This represents the flatness of the harmonic spectrum around the harmonic centroid.

$$\text{harmkurtosis} = \frac{\sum_{m=1}^{M} a_m[r]^2 \cdot [\hat{f}_m[r] - \text{harmcent}]^4}{\sum_{m=1}^{M} a_m[r]^2 \cdot \text{harmspread}^2} \tag{A.10}$$

Harmonic tristimuli (3) FBF tristim

The tristimuli are the relative amplitudes of the first harmonic, sum of the second to fourth harmonics, and sum of the remaining harmonics[227, 195]. The timbral interpretation of the

tristimuli was preceded by their use for describing colour in images, in terms of a tristimulus value for each of the three primary colours.

$$\begin{aligned} \text{tristim1} &= \frac{a_1[r]}{\sum_{m=1}^{M} a_m[r]} \\ \text{tristim2} &= \frac{\sum_{m=2}^{4} a_m[r]}{\sum_{m=1}^{M} a_m[r]} \\ \text{tristim3} &= 1 - \text{tristim1} - \text{tristim2} \end{aligned} \tag{A.11}$$

Harmonic tilt (1) FBF harmtilt

The harmonic tilt is the gradient of the linear regression of the harmonic amplitudes (in dB) onto $\log_2(f_m)$, and describes the rate of decrease of the harmonic amplitudes.

Harmonic irregularity (1) FBF irr

This measures the roughness of the harmonic amplitude spectrum by finding the difference between the m^{th} harmonic amplitude and the three-point average of the $m-1, m$ and $(m+1)^{th}$ harmonic amplitudes.

$$\text{irr} = \frac{\sum_{m=1}^{M} \left| a_m[r] - \frac{a_{m-1}[r]+a_m[r]+a_{m+1}[r]}{3} \right|}{\sum_{m=1}^{M} a_m[r]} \tag{A.12}$$

Harmonic deviation (1) FBF harmdev

The harmonic deviation is the overall deviation of the harmonic frequencies from perfect harmonicity. This may help to distinguish instruments like the piano, whose harmonics are noticeably stretched, from more harmonic instruments.

$$\text{harmdev} = \frac{\sum_{m=1}^{M} a_m[r] \cdot \left| \hat{f}_m - m \right|}{\sum_{m=1}^{M} a_m[r]} \tag{A.13}$$

A.3 Temporal/dynamic features

The temporal features are calculated on the amplitude envelope of the waveform, $x_{env}[n]$. In practice this can be sampled at a fraction of the sample rate of the original waveform. The envelope is calculated by Hilbert transforming the signal, and then filtering the result with a Butterworth low-pass filter of order 3 and cutoff frequency $f_c = 20$ Hz. We also define the normalised amplitude envelope:

$$p[n] = \frac{x_{env}[n]}{\sum_{n=0}^{L-1} x_{env}[n]} \tag{A.14}$$

Maximum amplitude (1) max

This is simply the maximum of the amplitude envelope.

$$\text{max} = \max(x_{env}[n]) \tag{A.15}$$

Centre of gravity (1) centgrav

This is similar to the temporal centroid, which has been used in studies of timbre discrimination[226] and is one component of the MPEG-7 PercussiveInstrumentTimbre descriptor.

It differs from the MPEG-7 TemporalCentroid descriptor due to the divide by L below which normalises the feature range to $[0, 1]$.

$$\text{centgrav} = \sum_{n=0}^{L-1} p[n] \frac{n}{L} \tag{A.16}$$

Note spread (1) notespread

This finds the spread of the amplitude envelope about the centre of gravity.

$$\text{notespread} = \sum_{n=0}^{L-1} p[n] \left(\frac{n}{L} - \text{centgrav}\right)^2 \tag{A.17}$$

Note skewness (1) noteskewness

This measures the asymmetry of the amplitude envelope about the centre of gravity.

$$\text{noteskewness} = \frac{\sum_{n=0}^{L-1} p[n] \left(\frac{n}{L} - \text{centgrav}\right)^3}{\text{notespread}^{3/2}} \tag{A.18}$$

Note kurtosis (1) notekurtosis

The kurtosis measures the flatness of the amplitude envelope relative to the centre of gravity.

$$\text{notekurtosis} = \frac{\sum_{n=0}^{L-1} p[n] \cdot \left(\frac{n}{L} - \text{centgrav}\right)^4}{\text{notespread}^2} \tag{A.19}$$

Log attack time (1) logattacktime

The method for calculating log attack time, sustain time and release time was modified from the adaptive threshold method given in [212]. As the real note samples being dealt with in chapter 6 typically contain significant interference from other sources, the fixed threshold method for calculating attack time (e.g. by finding the time difference between the point where the envelope crosses 20% of the maximum and 90% of the maximum) becomes fairly unreliable. Furthermore, for clean note samples, the adaptive method usually provides a closer fit to the envelope attack. The method is illustrated in fig. A.1 for a fairly non-uniform shaped note envelope, i.e. where the attack, sustain and decay are not clearly defined. However the method should be robust to this type of occurrence, as it is quite typical to find non-standard envelope shapes in polyphonic music due to interference from other sources.

Firstly, a series of thresholds are defined at 10. 20, ..., 90% of the maximum envelope amplitude. We denote the time at which the envelope crosses the threshold j by t_j, and the j^{th} 'effort', w_j, as the time taken for the amplitude envelope to increase from threshold j to $j + 1$. $\bar{w}$ is the median of $\mathbf{w}$, and is an overall measure of the rate of attack. The threshold for the start or end of the attack is then chosen as the first threshold for which $w_j < \frac{\bar{w}}{3}$ or $w_j > 3\ \bar{w}$, respectively. The times at which these thresholds are crossed are denoted T_1 and T_2 respectively, as depicted in fig. A.1.

$$\text{logattacktime} = \log_{10}(T_2 - T_1) \tag{A.20}$$

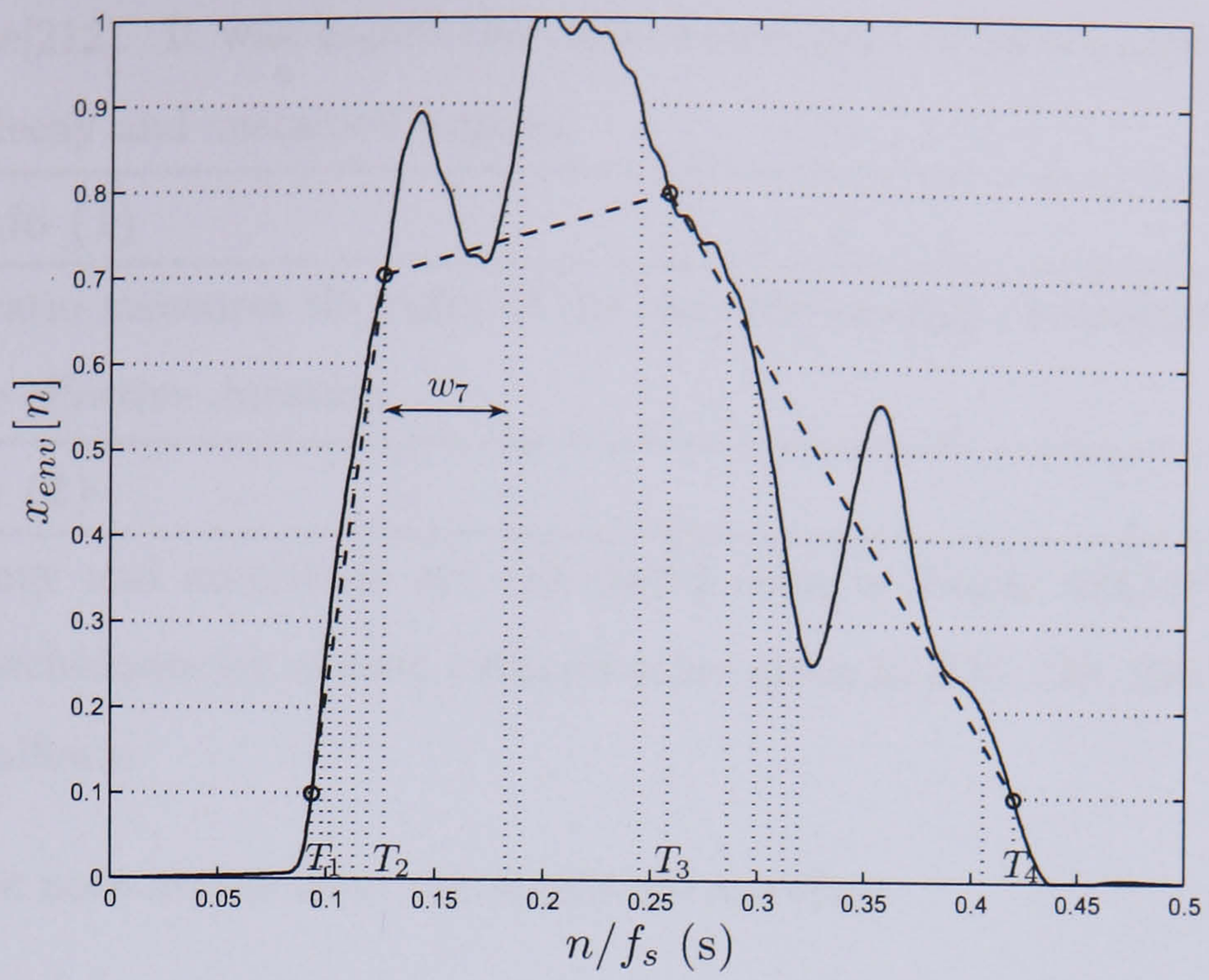

Figure A.1: Adaptive threshold method for estimating the log attack, sustain and release times from the amplitude envelope.

Log release time (1) logreleasetime

The log release time is calculated in the same fashion as the log attack time, with the only major difference being a time reversal of the amplitude envelope. T_3 and T_4 denote the start and end of the release respectively, as depicted in fig. A.1.

$$\text{logreleasetime} \;=\; \log_{10}(T_4 - T_3) \tag{A.21}$$

Log sustain time (1) logsustaintime

The log sustain time is simply taken to be the time between the end of the attack, T_2, and the start of the release, T_3.

$$\text{logsustaintime} \;=\; \log_{10}(T_3 - T_2) \tag{A.22}$$

Decrease slope (1) decreaseslope

The decrease slope is the negative gradient of the linear regression of $\log(x_{env}[n])$ on n in the decay region[212]. It is based upon a temporal model of the amplitude decay of the form:

$$x_{env}[n] \;=\; A\ \exp[-\text{decreaseslope} \cdot (n - n_{max})] \tag{A.23}$$

where the maximum of the envelope occurs at $n = n_{max}$. It may be useful for distinguishing between sounds with different rates of decay.

Effective duration (1) duration

The effective duration attempts to find the proportion of time that the sound is perceptually meaningful. It is the ratio of the time the envelope is greater than 40% of its maximum,

to the total time[212]. It was hoped that this would help to distinguish between sounds having a rapid decay and sustained sounds.

Amplitude ratio (1)	ampratio

The amplitude ratio measures the ratio of the time the envelope is greater than 80% of its maximum to the effective duration.

AM frequency (1)	AMfreq

The AM frequency and amplitude are calculated using a simple AM/vibrato estimation method. Other techniques for vibrato extraction are given in [228, 229, 230]. The procedure used here is as follows:

1. Remove the note attack from the amplitude envelope.
2. Fit a 4^{th} order polynomial to the remaining envelope.
3. Subtract the interpolated polynomial from the envelope to produce a flattened envelope, $x_{res}[n]$.
4. Find the maximum bin j of the DFT amplitude of $x_{res}[n]$, $|F[k]|$, in the range $f \in [0, 10]$ Hz.
5. If $|F[j]| > 0.5 \max\{|F[k]|;\ k = 0, \ldots, N/2\}$, the vibrato effect is assumed to be significant, and the AM frequency is $f(j)$. Otherwise, the AM frequency is zero.

AM amplitude (1)	AMamp

If the AM effect is found to be significant above, the AM amplitude is chosen as:

$$\text{AMamp} = \sqrt{\text{mean}({x_{res}}^2[n])} \tag{A.24}$$

Otherwise, it is set to zero.

A.4 Spectral features

The spectral features are all FBF features and are calculated from the amplitude spectrogram $A[k, r]$. $\sum_k$ is short for the sum over all frequency bins up to the Nyquist frequency bin: $k = 0, \ldots, N_q$. Spectral centroid, tilt, spread, skewness, kurtosis, decrease and roll-off are all calculated with respect to a logarithmic frequency axis, i.e. $f(k)$ below is the logarithm of the frequency in Hz of the k^{th} frequency bin.

Spectral centroid (1)	FBF	speccent

The spectral centroid is the 'centre-of-mass' or barycentre of the amplitude spectrum. It has been found to correlate very well with one of the perceptual dimensions derived from

multidimensional scaling (MDS) analysis, and is sometimes referred to as 'brightness' (see section 6.1).

$$\text{speccent} = \frac{\sum_k f(k)\ A[k,r]}{\sum_k A[k,r]} \tag{A.25}$$

Spectral tilt (1)	FBF	spectilt

The spectral tilt is the gradient of the linear regression of $A[:,r]$ in dB onto the logarithmic frequency axis, and it describes the rate of decrease of the amplitude spectrum.

Spectral spread (1)	FBF	specspread

The spectral spread measures the spread or 'bandwidth' of the amplitude spectrum around the spectral centroid[212].

$$\text{specspread} = \frac{\sum_k A[k,r]\ [f(k) - \text{speccent}]^2}{\sum_k A[k,r]} \tag{A.26}$$

Spectral skewness (1)	FBF	specskewness

This measures the asymmetry of the amplitude spectrum around the spectral centroid, and has the same properties as the harmonic skewness[212].

$$\text{specskewness} = \frac{\sum_k A[k,r]\ [f(k) - \text{speccent}]^3}{\text{specspread}^{3/2}\ \sum_k A[k,r]} \tag{A.27}$$

Spectral kurtosis (1)	FBF	speckurtosis

The spectral kurtosis is a measure of the flatness/peakedness of the amplitude spectrum around the spectral centroid[212].

$$\text{speckurtosis} = \frac{\sum_k A[k,r] \cdot [f(k) - \text{speccent}]^4}{\text{specspread}^2\ \sum_k A[k,r]} \tag{A.28}$$

Spectral decrease (1)	FBF	specdecrease

This measures the rate of decay of the amplitude spectrum[212].

$$\text{specdecrease} = \frac{1}{\sum_{k=1}^{N_q} A[k,r]} \sum_{k=1}^{N_q} \frac{A[k,r] - A[0,r]}{f(k) - f(0)} \tag{A.29}$$

Spectral roll-off (1)	FBF	specrolloff

The spectral roll-off is the frequency below which 95% of the signal energy is contained[212].

$$\text{specrolloff} = f(K) \quad \text{s.t.} \quad \sum_{k=1}^{K} A[k,r]^2 = 0.95 \cdot \sum_{k} A[k,r]^2 \tag{A.30}$$

Spectral variation (1)	FBF	specvariation

The spectral variation (or spectral flux) is a measure of the amount of variation of the amplitude spectrum between consecutive time frames. This feature appeared to be perceptually important from MDS experiments in both [186] and [1].

$$\text{specvariation} = 1 - \frac{\sum_k A[k,r-1] \cdot A[k,r]}{\sqrt{\sum_k A[k,r-1]^2} \cdot \sqrt{\sum_k A[k,r]^2}} \tag{A.31}$$

Spectral flatness (4)	FBF	specflatness

The spectral flatness and crest are each estimated in four separate frequency bands: [250,500] Hz, [500,1000] Hz, [1000,2000] Hz and [2000,4000] Hz. Spectral flatness is the ratio of the geometric mean to the arithmetic mean of the amplitude spectrum, and measures the sinusoidality/noisiness of the spectrum in each band[212]. For band b with boundary frequency bins k_b^L and k_b^R, and K_b number of bins in band b:

$$\text{specflatness} = \frac{\prod_{k=k_b^L}^{k_b^R} A[k,r]^{1/K_b}}{\sum_{k=k_b^L}^{k_b^R} A[k,r]/K_b} \tag{A.32}$$

Spectral crest (4)	FBF	speccrest

The spectral crest is the ratio of the maximum value of the amplitude spectrum in band b to the arithmetic mean of the amplitude spectrum in band b[212].

A.5 Energy features

Total energy (1)	Etotal

This is simply the total energy in the signal.

$$\text{Etotal} = \sum_{n=0}^{L-1} x[n]^2 \tag{A.33}$$

Energy in frequency bands (4)	FBF	Ebands

The relative energy in frequency bands was computed using 4 non-overlapping frequency bands spaced equally on a logarithmic axis between 50 and 400 Hz. For band b with boundary frequency bins k_b^L and k_b^R:

$$\text{Ebands(b)} = \frac{\sum_{k=k_b^L}^{k_b^R} A[k,r]^2}{\text{Etotal}} \tag{A.34}$$

A.6 Perceptual features

Total loudness (1)	loudness

The total loudness is the sum of the loudnesses in Bark bands.

Loudness in Bark bands (24)	loudnessBark

The relative specific loudness in Bark band b is defined as:

$$\text{loudnessBark(b)} = \frac{{E_b}^{0.23}}{\sum_{b=1}^{24} E_b} \tag{A.35}$$

where the energy in Bark band b has been calculated simply as the energy within one of 24 non-overlapping Bark bands between zero and the Nyquist frequency:

$$E_b = \sum_{k=k_b^L}^{k_b^R} A[k,r]^2$$

The lower and upper frequency bins for band b are given by k_b^L and k_b^R, and the Bark scale is described in section 5.2.1.

MFCC coefficients (13)	mmfcc

The perceptually motivated mel-frequency-cepstral coefficients (MFCCs) decorrelate and compactly encode the shape of the amplitude spectrum, and are computed as described in [214]. MFCCs have been useful on many occasions for speech and music description, e.g.[194, 200, 231]. The means of the MFCCs over all time frames, denoted 'mmfcc', result in 13 independent features.

delta-MFCC coefficients (13)	mdmfcc

The delta-MFCC coefficients are the time derivative of the MFCCs in each band. The means of the delta-MFCCs over all time frames, denoted 'mdmfcc', provide 13 additional features.

delta-delta-MFCC coefficients (13)	mddmfcc

Likewise, the delta-delta-MFCC coefficients are the second time derivatives of the MFCCs, and their means result in 13 additional features denoted 'mddmfcc'.

A.7 MPEG-7 features

The MPEG-7 instrument timbre descriptors have also been included[216]. These aim at describing the perceptual features supporting listeners' judgements of timbral similarity between different instrument sounds. They are split into the HarmonicInstrumentTimbre Descriptor and the PercussiveInstrumentTimbre Descriptor, containing 5 and 3 component features respectively. However, as LogAttackTime is an element of both categories, a total of 7 features would result if the two feature sets were combined.

HarmonicInstrumentTimbre Descriptor(5)	HarmonicInstrumentTimbre

These are intended to be used for harmonic coherent sustained sounds and contain the 4 harmonic timbral spectral descriptors (HarmonicSpectralCentroid, HarmonicSpectralDeviation, HarmonicSpectralSpread, HarmonicSpectralVariation) and the LogAttackTime descriptor.

PercussiveInstrumentTimbre Descriptor(3)	PercussiveInstrumentTimbre

The PercussiveInstrumentTimbre Descriptor is intended to be used with non-sustained percussive sounds, and consists of the timbral temporal descriptors (LogAttackTime, TemporalCentroid) and the SpectralCentroid descriptor.

Acronyms

AMT – Automatic Music Transcription

AR – Autoregressive

BIC – Bayesian Information Criterion

CASA – Computational Auditory Scene Analysis

CWT – Continuous Wavelet Transform

DCT – Discrete Cosine Transform

DDS – Damped and Delayed Sinusoid

DFT – Discrete Fourier Transform

DFT^{-1} – Inverse Discrete Fourier Transform

DWT – Discrete Wavelet Transform

DWT^{-1} – Inverse Discrete Wavelet Transform

EDS – Exponentially Damped Sinusoid

EM – Expectation-Maximisation

ENBW – Equivalent Noise Bandwidth

FBF – Frame-By-Frame

FFT – Fast Fourier Transform

FV – Feature Vector

GMM – Gaussian Mixture Model

HMM – Hidden Markov Model

k-NN – k-Nearest Neighbour

LPC – Linear Predictive Coding

LSE – Least-Squares Error

MBC – Model-Based Clustering

MDS – Multidimensional Scaling

MFCC – Mel-Frequency-Cepstral Coefficient

MP – Matching Pursuit

MQ – McAulay-Quatieri

MSRR – Mean Signal-to-Residual Ratio

NLS – Nonlinear Least Squares

QMF – Quadrature Mirror Filter

RMS – Root-Mean-Square

SFFS – Sequential Forward Floating Selection

SMS – Spectral Modeling Synthesis

SNR – Signal-to-Noise Ratio

SRR – Signal-to-Residual Ratio

STFT – Short-Time Fourier Transform

SVM – Support Vector Machine

WPT – Wavelet Packet Transform

WPT^{-1} – Inverse Wavelet Packet Transform

WVD – Wigner-Ville Distribution

References

[1] S. McAdams, S. Winsberg, S. Donnadieu, G. De Soete, and J. Krimphoff, "Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes," *Psychological Research*, vol. 58, no. 3, pp. 177–192, 1995.

[2] A. R. Martinez and W. L. Martinez, "Model-Based Clustering Toolbox for Matlab," tech. rep., Naval Surface Warfare Center, Dahlgren Division, Jan. 2004. [Online] Accessed 14 Nov. 2005. http://www.stat.washington.edu/mclust/.

[3] Caruso, Enrico and Vienna Radio Orchestra, conducted by Gottfried Rabi, "Italian songs." Audio CD, Feb. 2002. RCA 82569.

[4] "WWW Ircam: European projects - CUIDADO." [Online] Accessed 14 Nov. 2005. http://www.ircam.fr/projets_europeens.html?L=1.

[5] "Semantic HIFI." [Online] Accessed 14 Nov. 2005. http://shf.ircam.fr/.

[6] "SIMAC - Semantic Interaction with Music Audio Contents." [Online] Accessed 14 Nov. 2005. http://www.semanticaudio.org/.

[7] M. R. Every and J. E. Szymanski, "Separation of overlapping impulsive sounds by bandwise noise interpolation," in *Proc. 8th Int. Conf. on Digital Audio Effects (DAFx'05)*, (Madrid, Spain), Sep. 20-22 2005.

[8] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, vol. 97, no. 458, pp. 611–631, 2002.

[9] A. Cohen and J. Kovacevic, "Wavelets: The Mathematical background," *Proc. of the IEEE*, vol. 84, no. 4, pp. 514–522, 1996.

[10] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, vol. 8, no. 4, pp. 14–38, 1991.

[11] J. C. Brown, "Calculation of a constant Q spectral transform," *J. Acoust. Soc. Am.*, vol. 89, no. 1, pp. 425–434, 1991.

[12] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *J. Acoust. Soc. Am.*, vol. 92, no. 5, pp. 2698–2701, 1992.

[13] M. M. Goodwin, *Adaptive Signal Models: Theory, Algorithms, and Audio Applications*. PhD thesis, University of California, Berkeley, 1997.

[14] L. Cohen, "Time-frequency distributions - A review," *Proc. of the IEEE*, vol. 77, no. 7, pp. 941–981, 1989.

[15] F. Auger, P. Flandrin, P. Gonçalves, and O. Lemoine, "Time-Frequency Toolbox, for use with Matlab," 1997. [Online] Accessed 14 Nov. 2005. http://tftb.nongnu.org/.

[16] W. Pielemeier, G. Wakefield, and M. Simoni, "Time-frequency analysis of musical signals," *Proc. of the IEEE*, vol. 84, no. 9, pp. 1216–1230, 1996.

[17] P. Masri, A. Bateman, and N. Canagarajah, "A review of time-frequency representations, with application to sound/music analysis-resynthesis," *Organised Sound*, vol. 2, no. 3, pp. 193–205, 1997.

[18] J. Flanagan and R. Golden, "Phase vocoder," *Bell System Technical Journal*, pp. 1493–1509, Nov. 1966.

[19] M. R. Portnoff, "Implementation of the digital phase vocoder using the fast Fourier transform," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 24, no. 3, pp. 243–248, 1976.

[20] M. Puckette and J. Brown, "Accuracy of frequency estimates using the phase vocoder," *IEEE Trans. Speech and Audio Processing*, vol. 6, no. 2, pp. 166–176, 1998.

[21] R.E.Crochiere, "A weighted overlap-add method of short-time Fourier analysis/synthesis," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-28, no. 1, pp. 99–102, 1980.

[22] "FFTW Home Page." web. [Online] Accessed 14 Nov. 2005. http://www.fftw.org.

[23] G. Evangelista, "Pitch-synchronous wavelet representations of speech and music signals," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3313–3330, 1993.

[24] H. Muta, T. Baer, K. Wagatsuma, T. Muraoka, and H. Fukuda, "A pitch-synchronous analysis of hoarseness in running speech," *J. Acoust. Soc. Am.*, vol. 84, no. 4, pp. 1292–1301, 1988.

[25] S. Mallat, *A Wavelet Tour of Signal Processing.* Academic Press, 2nd ed., 1999.

[26] Y. Meyer, *Wavelets: Algorithms & Applications.* Philadelphia, PA.: Society for Industrial and Applied Mathematics, May 1993.

[27] I. Daubechies, *Ten Lectures on Wavelets.* No. 61 in CBMS-NSF Regional Conf. Series in Applied Mathematics, Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992.

[28] J. R. Beltrán and F. Beltrán, "Additive synthesis based on the continuous wavelet transform: A sinusoidal plus transient model," in *Proc. 6th Int. Conf. on Digital Audio Effects (DAFx'03)*, (London, U.K.), pp. 123–128, Sep. 8-11 2003.

[29] K. N. Hamdy, M. Ali, and A. H. Tewfik, "Low bit rate high quality audio coding with combined harmonic and wavelet representations," in *Proc. IEEE Int. Conf. on*

Acoustics, Speech, and Signal Processing (ICASSP'96), vol. 2, (Atlanta, Georgia), pp. 1045–1048, May 1996.

[30] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[31] P. Sathidevi and Y. Venkataramani, "Perceptual audio coding using sinusoidal/optimum wavelet representation," *Circuits, Systems & Signal Processing*, vol. 21, no. 5, pp. 511–524, 2002.

[32] G. Evangelista and P. Polotti, "Analysis and synthesis of pseudoperiodic 1/f -like noise by means of multiband wavelets," in *Proc. of 12th Int. Meeting of Computer Music (CIM'98)*, (Genova, Italy), pp. 35–38, 1998.

[33] P. Polotti and G. Evangelista, "Fractal additive synthesis via harmonic-band wavelets," *Computer Music Journal*, vol. 25, no. 3, pp. 22–37, 2001.

[34] R. Kronland-Martinet, "The wavelet transform for analysis, synthesis, and processing of speech and music sounds," *Computer Music Journal*, vol. 12, no. 4, pp. 11–20, 1988.

[35] D. Darlington, L. Daudet, and M. Sandler, "Digital audio effects in the wavelet domain," in *Proc. of the 5th Int. Conf. on Digital Audio Effects (DAFx'02)*, (Hamburg, Germany), pp. 7–12, Sep. 26–28 1992.

[36] C. Schremmer, T. Haenselmann, and F. Bömers, "A wavelet based audio denoiser," in *IEEE Int. Conf. on Multimedia and Expo (ICME'01)*, (Tokyo, Japan), pp. 145–148, Aug. 22-25 2001.

[37] P. Wolfe and S. Godsill, "Audio signal processing using complex wavelets," in *Preprint 5829, presented at the 114th Convention of the Audio Engineering Society*, (Amsterdam, The Netherlands), Mar. 22-25 2003.

[38] C. Delfs and F. Jondral, "Classification of transient time-varying signals using DFT and wavelet packet based methods," in *Proc. 1998 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'98)*, vol. 3, (Seattle, WA), pp. 1569–1572, May 1998.

[39] T. Lambrou, P. Kudumakis, R. Speller, M. Sandler, and A. Linney, "Classification of audio signals using statistical features on time and wavelet transform domains," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'98)*, vol. 6, (Seattle, WA), pp. 3621–3624, May 12-15 1998.

[40] G. Tzanetakis, G. Essl, and P. R. Cook, "Audio analysis using the discrete wavelet transform," in *Proc. WSES Int. Conf., Acoustics and Music: Theory and Applications (AMTA)*, (Skiathos, Greece), Sep. 27-30 2001.

[41] P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, D. Martinez-Munoz, and F. Lopez-Ferreras, "Transient modeling by matching pursuits with a wavelet dictionary for parametric audio coding," *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 349–352, 2004.

[42] S. Mallat, "Multiresolution approximation and wavelet orthonormal bases of $L^2(R)$," *Trans. American Mathematical Society*, vol. 315, pp. 69–87, Sep. 1989.

[43] Y. Meyer, "Orthonormal wavelets," in *Wavelets, Time-Frequency Methods and Phase Space* (J. Combes, A. Grossmann, and P. Tchamitchian, eds.), Berlin: Springer-Verlag, 1989.

[44] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[45] K. Ramchandran, M. Vetterli, and C. Herley, "Wavelets, subband coding, and best bases," *Proc. of the IEEE*, vol. 84, no. 4, pp. 541–560, 1996.

[46] R. Coifman and M. Wickerhauser, "Entropy–based algorithms for best basis selection," *IEEE Trans. Information Theory*, vol. 38, no. 2, pp. 713–718, 1992.

[47] E. D. Scheirer, *Music-Listening Systems*. PhD thesis, MIT Media Laboratory, Apr. 2000.

[48] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[49] T. Quatieri and R. Danisewicz, "An approach to co-channel talker interference suppression using a sinusoidal model for speech," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 38, no. 1, pp. 56–69, 1990.

[50] X. Serra, *Musical Signal Processing*, ch. Musical Sound Modeling with Sinusoids plus Noise. Studies on new music research, Lisse [Netherlands]: Swets & Zeitlinger Publishers, 1997.

[51] S. N. Levine, *Audio Representations for Data Compression and Compressed Domain Processing*. PhD thesis, Department of Electrical Engineering, Stanford University, Dec. 1998.

[52] X. Rodet, "Musical sound signal analysis/synthesis: Sinusoidal+residual and elementary waveform models," in *Proc. IEEE Time-Frequency and Time-Scale Workshop (TFTS'97)*, (Coventry, UK), Aug. 27-29 1997.

[53] P. Masri, *Computer Modelling of Sound for Transformation and Synthesis of Musical Signals*. PhD thesis, Department of Electrical and Electronic Engineering, University of Bristol, UK, Dec. 1996.

[54] M. Lagrange, S. Marchand, M. Raspaud, and J.-B. Rault, "Enhanced partial tracking using linear prediction," in *Proc. 6th Int. Conf. on Digital Audio Effects (DAFx'03)*, (London, UK), pp. 141–146, Sep. 8-11 2003.

[55] P. Depalle, G. Garcia, and X. Rodet, "Tracking of partials for additive sound synthesis using hidden Markov models," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'93)*, vol. 1, (Minneapolis, MN), pp. 225–228, Apr. 27-30 1993.

[56] M. Marolt, "Networks of adaptive oscillators for partial tracking and transcription of music recordings," *J. New Music Research*, vol. 33, no. 1, pp. 49–59, 2004.

[57] A. Sterian and G. H. Wakefield, "A model-based approach to partial tracking for musical transcription," in *Proc. SPIE Int. Symp. On Optical Science, Engineering, and Instrumentation*, (San Diego, CA), July 19-24 1998.

[58] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka, "Application of Bayesian probability network to music scene analysis," in *Working Notes of Int. Joint Conferences on Artificial Intelligence, Workshop of Computational Auditory Scene Analysis (IJCAI-CASA)*, pp. 52–59, Aug. 1995.

[59] G. Peeters and X. Rodet, "Sinusoidal characterization in terms of sinusoidal and non-sinusoidal components," in *Proc. COST-G6 Workshop on Digital Audio Effects (DAFx'98)*, (Barcelona, Spain), Nov. 19-21 1998.

[60] M. Lagrange, S. Marchand, and J.-B. Rault, "Sinusoidal parameter extraction and component selection in a non stationary model," in *Proc. 5th Int. Conf. on Digital Audio Effects (DAFx'02)*, (Hamburg, Germany), pp. 59–64, Sep. 26-28 2002.

[61] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound.* Cambridge, MA: MIT Press, Nov. 1990.

[62] T. Virtanen and A. Klapuri, "Separation of harmonic sound sources using sinusoidal modeling," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'00)*, (Istanbul, Turkey), pp. 765–769, June 2000.

[63] H. Viste and G. Evangelista, "A method for separation of overlapping partials based on similarity of temporal envelopes in multi-channel mixtures," *to be published in IEEE Trans. Speech and Audio Processing.*

[64] A. Sterian, *Model-based Segmentation of Time-Frequency Images for Musical Transcription.* PhD thesis, University of Michigan, Ann Arbor, 1999.

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

[66] X. Serra and J. Smith III, "Spectral Modelling Synthesis: A Sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[67] T. S. Verma and T. H. Meng, "Extending spectral modeling synthesis with transient modeling synthesis," *Computer Music Journal*, vol. 24, no. 2, pp. 47–59, 2000.

[68] J. Nieuwenhuijse, R. Heusens, and E. F. Deprettere, "Robust exponential modeling of audio signals," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'98)*, vol. 6, (Seattle, WA), pp. 3581–3584, May 12-15 1998.

[69] R. Boyer and K. Abed-Meraim, "Audio modeling based on delayed sinusoids," *IEEE Trans. Speech and Audio Processing*, vol. 12, no. 2, pp. 110–120, 2004.

[70] R. Boyer and K. Abed-Meraim, "Damped and delayed sinusoidal model for transient signals," *IEEE Trans. Signal Processing*, vol. 53, no. 5, pp. 1720–1730, 2005.

[71] X. Rodet and F. Jaillet, "Detection and modeling of fast attack transients," in *Proc. Int. Computer Music Conf. (ICMC'01)*, (Havana, Cuba), Sep. 18-22 2001.

[72] C. Duxbury, M. Davies, and M. Sandler, "Separation of transient information in musical audio using multiresolution analysis techniques," in *Proc. COST-G6 Conf. on Digital Audio Effects (DAFx'01)*, (Limerick, Ireland), pp. 1–4, Dec. 6-8 2001.

[73] M. Dörfler, *Gabor Analysis for a Class of Signals called Music*. PhD thesis, Institut für Mathematik, Universität Wien, July 2002.

[74] P. J. Wolfe, M. Dörfler, and S. J. Godsill, "Multi-Gabor dictionaries for audio time-frequency analysis," in *Proc. IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA'01)*, (New Paltz, NY), pp. 43–46, Oct. 21-24 2001.

[75] T. Verma and T. Meng, "Sinusoidal modeling using frame-based perceptually weighted matching pursuits," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'99)*, vol. 2, (Phoenix, AZ), pp. 981–984, Mar. 15-19 1999.

[76] M. M. Goodwin, "Multiscale overlap-add sinusoidal modeling using matching pursuit and refinements," in *Proc. IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA'01)*, (New Paltz, NY), pp. 207–210, Oct. 21-24 2001.

[77] R. Heusdens, R. Vafin, and W. B. Kleijn, "Sinusoidal modeling using psychoacoustic-adaptive matching pursuits," *IEEE Signal Processing Lett.*, vol. 9, no. 8, pp. 262–265, 2002.

[78] A. Nesbit, M. Sandler, and M. Davies, "A short review of two-channel source separation for music signals," in *Proc. Digital Music Research Network Summer Conf.*, (Glasgow, UK), pp. 5–8, July 23-24 2005.

[79] A. Jourjine, S. Rickard, and O. Yılmaz, "Blind separation of disjoint orthogonal signals: demixing N sources from 2 mixtures," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP'00)*, vol. 5, (Istanbul, Turkey), pp. 2985–2988, June 5-9 2000.

[80] O. Yılmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Trans. Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.

[81] D. FitzGerald, *Automatic Drum Transcription and Source Separation*. PhD thesis, Conservatory of Music and Drama, Dublin Institute of Technology, Ireland, 2004.

[82] R. C. Maher, "Evaluation of a method for separating digitized duet signals," *J. Audio Eng. Soc.*, vol. 38, no. 12, pp. 956–979, 1990.

[83] K. D. Martin, "A blackboard system for automatic transcription of simple polyphonic music," Tech. Rep. Technical Report No. 385, M.I.T. Media Laboratory Perceptual Computing Section, 1996.

[84] J. P. Bello Correa, *Towards the Automated Analysis of Simple Polyphonic Music: A Knowledge-Based Approach.* PhD thesis, Department of Electronic Engineering, Queen Mary, University of London, UK, Jan. 2003.

[85] P. Walmsley, S. Godsill, and P. Rayner, "Bayesian modelling of harmonic signals for polyphonic music tracking," in *Cambridge Music Processing Colloquium*, (Cambridge, UK), Sep. 1999.

[86] P. J. Walmsley, *Signal Separation of Musical Instruments- Simulation-Based Methods for Musical Signal Decomposition and Transcription.* PhD thesis, Department of Engineering, University of Cambridge, UK, Sep. 2000.

[87] M. Goto, "A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.

[88] S. Abdallah and M. Plumbley, "Polyphonic transcription by non-negative sparse coding of power spectra," in *Proc. 5th Int. Conf. on Music Information Retrieval (ISMIR'04)*, (Barcelona, Spain), pp. 318–325, Oct. 10–14 2004.

[89] M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler, "Automatic music transcription and audio source separation," *Cybernetics and Systems*, vol. 33, no. 6, pp. 603–627, 2002.

[90] A. Klapuri, *Signal Processing Methods for the Automatic Transcription of Music.* PhD thesis, Tampere University of Technology, Mar. 2004.

[91] A. Klapuri, "Automatic transcription of music," in *Proc. Stockholm Music Acoustics Conf. (SMAC'03)*, (Stockholm, Sweden), Aug. 6-9 2003.

[92] A. P. Klapuri, "Musical meter estimation and music transcription," in *presented at the Cambridge Music Processing Colloquium*, (Cambridge, UK), Mar. 28 2003.

[93] A. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003.

[94] N. Orio and D. Schwarz, "Alignment of monophonic and polyphonic music to a score," in *Proc. Int. Computer Music Conf. (ICMC'01)*, (Havana, Cuba), Oct. 12 2001.

[95] R. B. Dannenberg and N. Hu, "Polyphonic audio matching for score following and intelligent audio editors," in *Proc. Int. Computer Music Conf. (ICMC'03)*, (Singapore), Sep. 29- Oct. 4 2003.

[96] F. Soulez, X. Rodet, and D. Schwarz, "Improving polyphonic and poly-instrumental music to score alignment," in *Proc. 4th Int. Conf. on Music Information Retrieval (ISMIR'03)*, (Washington D.C. and Balitmore, MD), Oct. 26-30 2003.

[97] R. J. Turetsky and D. P. Ellis, "Ground-truth transcriptions of real music from force-aligned MIDI syntheses," in *Proc. Int. Conf. on Music Information Retrieval (ISMIR'03)*, (Washington D.C. and Baltimore, MD), Oct. 26-30 2003.

[98] S. Dixon, "Live tracking of musical performances using on-line time warping." in *Proc. of the 8th Int. Conf. on Digital Audio Effects (DAFx'05)*, (Madrid, Spain), pp. 92–97. Sep. 20-22 2005.

[99] N. Orio, S. LeMouton, D. Schwarz, and N. Schnell, "Score following: State of the art and new developments," in *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME'03)*, (Montreal, Canada), pp. 36–41, May 22-24 2003.

[100] D. Schwarz, N. Orio, and N. Schnell, "Robust polyphonic MIDI score following with hidden Markov models," in *Proc. Int. Computer Music Conf. (ICMC'04)*, (Miami, FL.), Nov. 1-6 2004.

[101] H. Thornburg and F. Gouyon, "A flexible analysis-synthesis method for transients," in *Proc. Int. Computer Music Conf. (ICMC'00)*, (Berlin, Germany), Aug. 27-Sep. 1 2000.

[102] A. Klapuri, "Sound onset detection by applying psychoacoustic knowledge," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'99)*, vol. 6, (Phoenix, AZ), pp. 3089–3092, Mar. 15-19 1999.

[103] M. Marolt, A. Kavčič, and M. Privošnik, "Neural networks for note onset detection in piano music," in *Proc. Int. Computer Music Conf. (ICMC'02)*, (Götheborg, Sweden), Sep. 2002.

[104] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[105] N. Collins, "A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions," in *presented at the 118th AES Convention*, (Barcelona, Spain), May 28-31 2005.

[106] C. Duxbury, J. P. Bello, M. Davies, and M. Sandler, "Complex domain onset detection for musical signals," in *Proc. 6th Int. Conf. on Digital Audio Effects (DAFx'03)*, (London, UK), pp. 90–93, Sep. 8-11 2003.

[107] C. Duxbury, J. Bello, M. Sandler, and M. Davies, "A comparison between fixed and multiresolution analysis for onset detection in musical signals," in *Proc. 7th Int. Conf. on Digital Audio Effects (DAFx'04)*, (Naples, Italy), pp. 207–211, Oct. 5-8 2004.

[108] J. Bello, C. Duxbury, M. Davies, and M. Sandler, "On the use of phase and energy for musical onset detection in the complex domain," *IEEE Signal Processing Lett.*, vol. 11, no. 6, pp. 553–556, 2004.

[109] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–53, 1970.

[110] N. Fletcher and T. Rossing, *The Physics of Musical Instruments*. New York: Springer-Verlag, 2nd ed., 1998.

[111] D. Schwarz, "Spectral Envelopes in Sound Analysis and Synthesis," Diplomarbeit Nr. 1622, Universität Stuttgart, Fakultät Informatik, 1998.

[112] S. Marchand, *Modélisation Informatique du Son Musical (Analyse, Transformation, Synthèse)*. PhD thesis, École doctorale de Mathématiques et d'Informatique, L'Université Bordeaux 1, Dec. 2000.

[113] M. Desainte-Catherine and S. Marchand, "High precision Fourier analysis of sounds using signal derivatives," *J. Audio Eng. Soc.*, vol. 48, no. 7/8, pp. 654–667. 2000.

[114] F. Auger and P. Flandrin, "Improving the readability of time-frequency and time-scale representations by the reassignment method," *IEEE Trans. Signal Processing*, vol. 43, no. 5, pp. 1068–1089, 1995.

[115] K. Fitz and L. Haken, "On the use of time-frequency reassignment in additive sound modelling," *J. Audio Eng. Soc.*, vol. 50, no. 11, pp. 879–893, 2002.

[116] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[117] M. Donadio, "How to interpolate frequency peaks," revised 5/5/1999. [Online] Accessed 14 Nov. 2005. http://www.dspguru.com/howto/tech/peakfft2.htm.

[118] S. Marchand and R. Strandh, "InSpect and ReSpect: Spectral Modeling, Analysis and Real-Time Synthesis Software Tools for Researchers and Composers," in *Proc. of the Int. Computer Music Conf. (ICMC'99)*, (Beijing, China), pp. 341–344, Oct. 1999.

[119] F. Keiler and S. Marchand, "Survey on extaction of sinusoids in stationary sounds," in *Proc. 5th Int. Conf. on Digital Audio Effects (DAFx'02)*, (Hamburg, Germany), pp. 51–58, Sep. 26-28 2002.

[120] H. A. Conklin Jr., "Generation of partials due to nonlinear mixing in a stringed instrument," *J. Acoust. Soc. Am.*, vol. 105, no. 1, pp. 536–545, 1999.

[121] L. Ortiz-Berenguer and F. Casajús-Quirós, "Polyphonic transcription using piano modeling for spectral pattern recognition," in *Proc. 5th Int. Conf. on Digital Audio Effects (DAFx'02)*, (Hamburg, Germany), pp. 45–50, Sep. 26-28 2002.

[122] L. Ortiz-Berenguer, F. Casajús-Quirós, M. Torres-Guijarro, and J. Beracoechea

[123] A. Klapuri, T. Virtanen, and J.-M. Holm, "Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals," in *Proc. COST-G6 Conf. on Digital Audio Effects (DAFx'00)*, (Verona, Italy), Dec. 2000.

[124] T. Virtanen and A. Klapuri, "Separation of harmonic sounds using multipitch analysis and iterative parameter estimation," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'01)*, (New Paltz, NY.), 2001.

[125] T. Tolonen, "Methods for separation of harmonic sound sources using sinusoidal modeling," in *presented at AES 106th Convention*, (Munich, Germany), May 8-11 1999.

[126] T. Virtanen and A. Klapuri, "Separation of harmonic sounds using linear models for the overtone series," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'02)*, vol. 2, (Orlando, FL.), pp. 1757–1760, May 13-17 2002.

[127] M. Every and J. Szymanski, "Separation of synchronous pitched notes by spectral filtering of harmonics," *to be published in IEEE Trans. Speech and Audio Processing*, 2006.

[128] T. W. Parsons, "Separation of speech from interfering speech by means of harmonic selection," *J. Acoust. Soc. Am.*, vol. 60, no. 4, pp. 911–918, 1976.

[129] A. P. Klapuri, "Multipitch estimation and sound separation by the spectral smoothness principle," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'01)*, vol. 5, (Salt Lake City, UT), pp. 3381–3384, May 7-11 2001.

[130] H. Viste and G. Evangelista, "Separation of harmonic instruments with overlapping partials in multi-channel mixtures," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'03)*, (New Paltz, NY.), pp. 25–28, Oct. 19-22 2003.

[131] P. Depalle and T. Hélie, "Extraction of spectral peak parameters using a short-time Fourier transform modeling and no sidelobe windows," in *Proc. IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'97)*, (New Paltz, NY), Oct. 19-22 1997.

[132] M. Kazama, K. Yoshida, and T. M., "Signal representation including waveform envelope by clustered line-spectrum modeling," *J. Audio Eng. Soc.*, vol. 51, no. 3, pp. 123–137, 2003.

[133] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis*. Upper Saddle River, NJ.: Prentice Hall, 1st ed., 1997. pp. 319.

[134] T. Tolonen, "Object-based sound source modeling for musical signals," in *Proc. 109th Audio Engineering Society Convention*, (Los Angeles, CA.), Sep. 22-25 2000.

[135] "Melody Separation Demonstrations." [Online] Accessed 14 Nov. 2005. http://www-users.york.ac.uk/~jes1/Separation2.html.

[136] "Note Separation Demonstrations." [Online] Accessed 14 Nov. 2005. http://www-users.york.ac.uk/~jes1/Separation1.html.

[137] "Percussive Separation Demonstrations - DAFx'05." [Online] Accessed 14 Nov. 2005. http://www-users.york.ac.uk/~jes1/Separation3.html.

[138] D. Ellis, "Sinewave and Sinusoid+Noise Analysis/Synthesis in Matlab." web, Mar. 2003. [Online] Accessed 14 Nov. 2005. http://www.ee.columbia.edu/~dpwe/resources/matlab/sinemodel/.

[139] "University of Iowa, Electronic Music Studios, Musical Instrument Samples." [Online] Accessed 14 Nov. 2005. http://theremin.music.uiowa.edu/.

[140] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[141] H. Hirsch and C. Ehrlicher, "Noise estimation techniques for robust speech recognition," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'95)*, vol. 1, (Detroit, MI), pp. 153–156, May 9-12 1995.

[142] V. Stahl, A. Fischer, and R. Bippus, "Quantile based noise estimation for spectral subtraction and Wiener filtering," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'00)*, vol. 3, (Istanbul, Turkey), pp. 1875–1878, June 5-9 2000.

[143] I. Cohen and B. Berdugo, "Noise estimation by minima controlled recursive averaging for robust speech enhancement," *IEEE Signal Processing Lett.*, vol. 9, no. 1, pp. 12–15, 2002.

[144] K. Yamashita and T. Shimamura, "Nonstationary noise estimation using low-frequency regions for spectral subtraction," *IEEE Signal Processing Lett.*, vol. 12, no. 6, pp. 465–468, 2005.

[145] H.-T. Hu, F.-J. Kuo, and H.-J. Wang, "Supplementary schemes to spectral subtraction for speech enhancement," *Speech Communication*, vol. 36, no. 3-4, pp. 205–218, 2002.

[146] M. R. Every, "Separating harmonic and inharmonic note content from real mono recordings," in *Proc. Digital Music Research Network Summer Conf. 2005*, (Glasgow, U.K.), pp. 9–13, July 23-24 2005.

[147] J. Makhoul, "Linear prediction: A tutorial review," *Proc. of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[148] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ.: Prentice Hall PTR, 2nd ed., 1999.

[149] U. Zölzer, ed., *DAFX - Digital Audio Effects*. John Wiley and Sons, Feb. 2002.

[150] S. L. Marple Jr., *Digital Spectral Analysis: With Applications*. London: Englewood Cliffs: Prentice-Hall, 1987.

[151] M. Goodwin, "Residual modeling in music analysis-synthesis," in *Proc. 1996 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'96)*, vol. 2, (Atlanta, GA.), pp. 1005–1008, May 7-10 1996.

[152] E. Zwicker and E. Terhardt, "Analytical expression for critical-band rate and critical bandwidth as a function of frequency," *J. Acoust. Soc. Am.*, vol. 68, no. 5, pp. 1523–1525, 1980.

[153] H. Traunmüller, "Analytical expressions for the tonotopic sensory scale," *J. Acoust. Soc. Am.*, vol. 88, no. 1, pp. 97–100, 1990.

[154] F. Avanzini, *Computational Issues in Physically-based Sound Models*. PhD thesis, Dept. of Computer Science and Electronics, University of Padova, Italy, 2001.

[155] P. Polotti and G. Evangelista, "Analysis and synthesis of pseudo-periodic 1/f-like noise by means of wavelets with applications to digital audio," *EURASIP Journal on Applied Signal Processing*, vol. 1, pp. 1–14, 2001.

[156] K. Fitz, L. Haken, and P. Chirstensen, "A new algorithm for bandwidth association in bandwidth-enhanced additive sound modeling," in *Proc. Int. Computer Music Conf. (ICMC'00)*, (Berlin, Germany), 2000.

[157] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice.* Upper Saddle River, NJ: Prentice Hall, 2002.

[158] J. M. Grey, "Multidimensional perceptual scaling of musical timbres," *J. Acoust. Soc. Am.*, vol. 61, no. 5, pp. 1270–1277, 1977.

[159] Y. Ding and X. Qian, "Processing of Musical Tones Using a Combined Quadratic Polynomial-Phase Sinusoid and Residual (QUASAR) Signal Model," *J. Audio Eng. Soc.*, vol. 45, no. 7/8, pp. 571–584, 1997.

[160] N. Laurenti and G. De Poli, "A method for spectrum separation and envelope estimation of the residual in spectrum modeling of musical sound," in *Proc. COST-G6 Conf. on Digital Audio Effects (DAFx'00)*, (Verona, Italy), pp. 233–236, Dec. 7-9 2000.

[161] D. P. Ellis, *Prediction-driven Computational Auditory Scene Analysis.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1996.

[162] K. Kashino and H. Murase, "A sound source identification system for ensemble music based on template adaptation and music stream extraction," *Speech Communication*, vol. 27, no. 3/4, pp. 337–349, 1999.

[163] R. Plomp, *Frequency Analysis and Periodicity Detection in Hearing*, pp. 397–414. Leiden: Sijthoff, 1970.

[164] S. Lakatos, "A common perceptual space for harmonic and percussive timbres," *Perception & Psychophysics*, vol. 62, no. 7, pp. 1426–1439, 2000.

[165] P. Herrera-Boyer, G. Peeters, and S. Dubnov, "Automatic classification of musical instrument sounds," *J. New Music Research*, vol. 32, no. 1, pp. 3–21, 2003.

[166] R. Kendall, "The role of acoustic signal partitions in listener categorization of musical phrases," *Music Perception*, vol. 4, no. 2, pp. 185–214, 1986.

[167] J. M. Grey, "Timbre discrimination in musical patterns," *J. Acoust. Soc. Am.*, vol. 64. no. 2, pp. 467–472, 1978.

[168] S. Samson, R. Zatorre, and J. Ramsay, "Multidimensional scaling of synthetic musical timbre: Perception of spectral and temporal characteristics." *Canadian Journal of Experimental Psychology*, vol. 51, no. 4, pp. 307–315, 1997.

[169] R. Plomp, *Aspects of Tone Sensation: A Psychophysical Study.* London: Academic Press, 1976.

[170] H. Helmholtz, *Théorie Physiologique de la Musique Fondée sur l'Étude des Sensations Auditives.* Paris: Masson, 1868–1874.

[171] J. M. Grey and J. W. Gordon, "Perceptual effects of spectral modifications on musical timbres," *J. Acoust. Soc. Am.*, vol. 63, no. 5, pp. 1493–1500, 1978.

[172] D. L. Wessel, "Timbre space as a musical control structure," *Computer Music Journal*, vol. 3, no. 2, pp. 45–52, 1979.

[173] S. McAdams, J. Beauchamp, and S. Meneguzzi, "Discrimination of musical instrument sounds resynthesized with simplified spectrotemporal parameters," *J. Acoust. Soc. Am.*, vol. 105, no. 2, pp. 882–897, 1999.

[174] A. Caclin, S. McAdams, B. K. Smith, and S. Winsberg, "Acoustic correlates of timbre space dimensions: A confirmatory study using synthetic tones," *J. Acoust. Soc. Am.*, vol. 118, no. 1, pp. 471–482, 2005.

[175] J. M. Grey and J. A. Moorer, "Perceptual evaluations of synthesized musical instrument tones," *J. Acoust. Soc. Am.*, vol. 62, no. 2, pp. 454–462, 1977.

[176] K. W. Berger, "Some factors in the recognition of timbre," *J. Acoust. Soc. Am.*, vol. 36, no. 10, pp. 1888–1891, 1964.

[177] M. Clark Jr., P. Robertson, and D. Luce, "A preliminary experiment on the perceptual basis for musical instrument families," *J. Audio Eng. Soc.*, vol. 12, no. 3, pp. 199–203, 1964.

[178] E. Saldanha and J. Corso, "Timbre cues and the identification of musical instruments," *J. Acoust. Soc. Am.*, vol. 36, no. 11, pp. 2021–2026, 1964.

[179] L. Wedin and G. Goude, "Dimension analysis of the perception of instrumental timbre," *Scandinavian Journal of Psychology*, vol. 13, no. 3, pp. 228–240, 1972.

[180] P. Iverson and C. L. Krumhansl, "Isolating the dynamic attributes of musical timbre," *J. Acoust. Soc. Am.*, vol. 94, no. 5, pp. 2595–2603, 1993.

[181] K. D. Martin and Y. E. Kim, "Musical instrument identification: A patter-recognition approach," in *136th Meeting of the Acoustical Society of America*, (Norfolk, VA.), Oct. 1998.

[182] J. Gibson, *The Senses Considered as Perceptual Systems.* Boston: Houghton Mifflin, 1966.

[183] R. Shepard, "The analysis of proximities: Multidimensional scaling with an unknown distance function. Part I.," *Psychometrika*, vol. 27, no. 2, pp. 125–140. 1962.

[184] R. Shepard, "The analysis of proximities: Multidimensional scaling with an unknown distance function. Part II.," *Psychometrika*, vol. 27. no. 3, pp. 219–246, 1962.

[185] J. Kruskal, "Multidimensional-scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29. no. 1. pp. 1–27. 1964.

[186] C. Krumhansl, *Structure and Perception of Electroacoustic Sound and Music*. pp. 43–53. Amsterdam: Elsevier, 1989. Excepta Medica 846.

[187] J. R. Miller and E. C. Carterette, "Perceptual space for musical structures," *J. Acoust. Soc. Am.*, vol. 58, no. 3, pp. 711–720, 1975.

[188] R. Plomp and H. Steeneken, "Effect of phase on the timbre of complex tones," *J. Acoust. Soc. Am.*, vol. 46, no. 2B, pp. 409–421, 1969.

[189] S. Winsberg and G. De Soete, "A latent class approach to fitting the weighted Euclidean model, CLASCAL," *Psychometrika*, vol. 58, pp. 315–330, 1993.

[190] S. Handel and M. L. Erickson, "Sound source identification: The possible role of timbre transformations," *Music Perception*, vol. 21, no. 4, pp. 587–610, 2004.

[191] N. Misdariis, B. Smith, D. Pressnitzer, P. Susini, and S. McAdams, "Validation of a multidimensional distance model for perceptual dissimilarities among musical timbres," in *16th Int. Congress on Acoustics and 135th Meeting Acoustical Society of America*, (Seattle, WA.), June 20-26 1998.

[192] D. Godsmark and G. J. Brown, "A blackboard architecture for computational auditory scene analysis," *Speech Communication*, vol. 27, pp. 351–366, 1999.

[193] K. D. Martin, *Sound Source Recognition: A Theory and Computational Model*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA., June 1999.

[194] J. Marques and P. J. Moreno, "A study of musical instrument classification using Gaussian mixture models and support vector machines," tech. rep., Cambridge Research Laboratory, June 1999. CRL 99/4.

[195] K. Jensen, *Timbre Models of Musical Sounds*. PhD thesis, Department of Computer Science, University of Copenhagen, 1999. Report no. 99/7.

[196] I. Kaminskyj and A. Materka, "Automatic source identification of monophonic musical instrument sounds," in *Proc. IEEE Int. Conf. Neural Networks*, vol. 1, (Perth, Australia), pp. 189–194, 1995.

[197] I. Kaminskyj, "Multi-feature musical instrument sound classifier w/user determined generalisation performance," in *Proc. Australasian Computer Music Conf.*, (Melbourne, Australia), pp. 53–62, July 2002.

[198] I. Fujinaga and K. MacMillan, "Realtime recognition of orchestral instruments," in *Proc. 2000 Int. Computer Music Conf.*, (Berlin, Germany), pp. 141–143, 2000.

[199] G. Agostini, M. Longari, and E. Pollastri, "Musical instrument timbres classification with spectral features," in *Proc. 2001 IEEE Fourth Workshop on Multimedia Signal Processing*, (Cannes, France), pp. 97–102. Oct. 3-5 2001.

[200] A. Eronen, "Comparison of features for musical instrument recognition," in *Proc. IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA '01)*, (New Paltz, NY.), pp. 19–22, Oct. 21-24 2001.

[201] G. Agostini, M. Longari, and E. Pollastri, "Musical instrument timbres classification with spectral features," *EURASIP Journal of Applied Signal Processing*, vol. 1, pp. 5–14, 2003.

[202] A. Wieczorkowska, J. Wróblewski, P. Synak, and D. Ślęzak, "Application of temporal descriptors to musical instrument sound recognition," *Journal of Intelligent Information Systems*, vol. 21, no. 1, pp. 71–93, 2003.

[203] I. Kaminskyj and T. Czaszejko, "Automatic recognition of isolated monophonic musical instrument sounds using kNNC," *Journal of Intelligent Information Systems*, vol. 24, no. 2-3, pp. 199–221, 2005.

[204] B. Kostek, "Musical instrument classification and duet analysis employing music information retrieval techniques," *Proc. of the IEEE*, vol. 92, no. 4, pp. 712–729, 2004.

[205] B. Kostek and A. Czyzewski, "Representing musical instrument sounds for their automatic classification," *J. Audio Eng. Soc.*, vol. 49, no. 9, pp. 768–785, 2001.

[206] A. Wieczorkowska, "Musical sound classification based on wavelet analysis," *Fundamenta Informaticae*, vol. 47, no. 1-2, pp. 175–188, 2001.

[207] J. Brown, "Computer identification of musical instruments using pattern recognition with cepstral coefficients as features," *J. Acoust. Soc. Am.*, vol. 105, no. 3, pp. 1933–1941, 1999.

[208] J. C. Brown, O. Houix, and S. McAdams, "Feature dependence in the automatic identification of musical woodwind instruments," *J. Acoust. Soc. Am.*, vol. 109, no. 3, pp. 1064–1072, 2001.

[209] A. Eronen, "Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs," in *Proc. 7th Int. Symp. on Signal Processing and its Applications*, vol. 2, (Paris, France), pp. 133–136, July 2003.

[210] T. Kitahara, M. Goto, and H. G. Okuno, "Category-level identification of non-registered musical instrument sounds," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP'04)*, vol. 4, (Montreal, Canada), pp. 253–256, May 2004.

[211] P. Herrera, A. Dehamel, and F. Gouyon, "Automatic labeling of unpitched percussion sounds," in *Proc. Audio Engineering Society, 114th Convention*, (Amsterdam, The Netherlands), March 22-25 2003.

[212] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," tech. rep., CUIDADO I.S.T. Project Report, 2004. [Online] Accessed 14 Nov. 2005. http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf.

[213] F. Thibault, "High-level control of singing voice timbre transformations," Master's thesis, Faculty of Music, McGill University, Canada, Aug. 2004.

[214] M. Slaney, "Auditory toolbox, version 2," Tech. Rep. 1998-010, Interval Research Corporation, 1998.

[215] M. Casey, "MPEG-7 Multimedia Software Resources, ISO15938-4 (MPEG-7) Feature Extraction Matlab Source." web, 2003. [Online] Accessed 14 Nov. 2005. http://mpeg7.doc.gold.ac.uk/mirror/index.html.

[216] "MPEG-7 Overview," Tech. Rep. ISO/IEC JTC1/SC29/WG11 N6828. Int. Organisation for Standardisation, Palma de Mallorca, Oct. 2004. Martinez, J.M. (ed.).

[217] P. Pudil and J. Novovičová, "Novel methods for subset selection with respect to problem knowledge," *IEEE Intelligent Systems and Their Applications*. vol. 13, no. 2. pp. 66–74, 1998.

[218] P. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Computers*, vol. 26, no. 9, pp. 917–922, 1977.

[219] P. Somol, P. Pudil, and J. Kittler, "Fast branch & bound algorithms for optimal feature selection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 900–912, 2004.

[220] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[221] A. Jain and D. Zongker, "Feature selection: evaluation, application, and small sample performance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[222] A. Whitney, "A direct method of nonparametric measurement selection," *IEEE Trans. Computing*, vol. C-20, pp. 1100–1103, 1971.

[223] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience, 2nd ed., Nov. 2000.

[224] J. D. Banfield and A. E. Raftery, "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, vol. 49, pp. 803–821, Sep. 1993.

[225] Y. Horii, "Vocal shimmer in sustained phonation," *J. Speech and Hearing Research*, vol. 23, no. 1, pp. 202–209, 1980.

[226] G. Peeters, S. McAdams, and P. Herrera, "Instrument sound description in the context of MPEG-7," in *Proc. Int. Computer Music Conf. (ICMC'00)*, (Berlin, Germany), Aug./Sep. 2000.

[227] H. Pollard and E. Jansson, "A tristimulus method for the specification of musical timbre," *Acustica*, vol. 51, pp. 162–171, 1982.

[228] P. Herrera and J. Bonada, "Vibrato extraction and parameterization in the Spectral Modeling Synthesis framework," in *Proc. COST-G6 Conf. on Digital Audio Effects (DAFx'98)*, (Barcelona, Spain), Nov. 19-21 1998.

[229] S. Rossignol, P. Depalle, J. Soumagne, X. Rodet, and J.-L. Colette, "Vibrato: detection, estimation, extraction, modification," in *Proc. COST-G6 Workshop on Digital Audio Effects (DAFx'99)*, (Trondheim, Norway), Dec. 9-11 1999.

[230] I. Arroabarren, M. Zivanovic, J. Bretos, A. Ezcurra, and A. Carlosena, "Measurement of vibrato in lyric singers," *IEEE Trans. on Instrumentation and Measurement*, vol. 51, no. 4, pp. 660–665, 2002.

[231] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. Int. Symp. on Music Information Retrieval*, (Plymouth, MA.), Oct. 23-25 2000.