TAMPERE UNIVERSITY OF TECHNOLOGY

Department of Information Technology

TUOMAS VIRTANEN

AUDIO SIGNAL MODELING WITH SINUSOIDS PLUS NOISE

Master of Science Thesis

The subject was approved by the Department of Information Technology on the 23th of August 2000.
Thesis supervisors: Professor Jaakko Astola
MSc Anssi Klapuri

Preface

This work has been carried out in the Signal Processing Laboratory of Tampere University of Technology, Finland. The working in the Audio Research Group of TUT has been very enjoyable, and I would like to thank Mr. Jari Yli-Hietanen and Mr. Anssi Klapuri for giving me this possibility. I also would like to thank my other colleagues for the refreshing atmosphere.

I wish to thank Professor Jaakko Astola for his valuable advice, and Anssi Klapuri for all his guidance and encouragement in the first steps of my scientific career.

Tampere, March 2001

Tuomas Virtanen

Tiivistelmä

TAMPEREEN TEKNILLINEN KORKEAKOULU
Tietotekniikan osasto
Signaalinkäsittely

VIRTANEN, TUOMAS: Audiosignaalin mallintaminen sineillä ja kohinalla
Diplomityö, 62 s., 6 liites.
Tarkastajat: Prof. Jaakko Astola, DI Anssi Klapuri
Rahoittajat: Tampereen teknillinen korkeakoulu, Signaalinkäsittelyn laitos
Maaliskuu 2001
Avainsanat: spektrin mallintaminen, välitason esitysmuoto, sinimalli, äänilähteiden erottelu

Audiosignaalien spektrin mallintamisessa tavoitteena on muuntaa signaali helpommin analysoitavaan muotoon poistaen kuulon kannalta merkityksetön informaatio. Sini- ja kohinamalli on spektrimalli, jossa äänen jaksolliset komponentit esitetään sineillä joiden taajuudet, amplitudit ja vaiheet muuttuvat ajan funktiona. Jäljelle jäävät ei-jaksolliset komponentit esitetään kaistoittain suodatettuna kohinana. Sinimalli hyödyntää musikaalisten instrumenttien fyysisisiä ominaisuuksia ja kohinamalli ihmiskuulon epätarkkuutta kohinaspektrin tarkan muodon tai vaiheen suhteen.

Sinien parametrien estimointi polyfonisista musiikkisignaaleista on hankalaa johtuen siitä että jaksolliset komponentit ovat vain harvoin täysin stabiileja. Myös riittävää aika- ja taajuusresoluutiota on hankala saavuttaa yhtä aikaa. Suuri osa diplomityöstä käsittelee jaksollisten komponenttien havaitsemista sekä niiden parametrien estimointia useilla eri algoritmeilla. Vanhojen algoritmien lisäksi työssä esitetään uusi iteratiivinen algoritmi, joka perustuu lähekkäisten sinien yhdistämiseen.

Sinimallia on työssä sovellettu päällekkäisten äänten erotteluun sekä manipulointiin. Erottelussa käytetään uutta etäisyysmittaa yksittäisten sinien välillä. Etäisyysmitta jäljittelee ihmiskuulon tapaa ryhmitellä ääniä. Työssä selostetaan myös lyhyesti uusi erottelumenetelmä joka käyttää hyväkseen estimaattia äänten korkeudesta. Äänten nopeuden ja korkeuden muuttaminen laatua heikentämättä sini- ja kohinamallin avulla on myös käsitelty lyhyesti.

Abstract

TAMPERE UNIVERSITY OF TECHNOLOGY
Department of Information Technology
Signal Processing Laboratory

VIRTANEN, TUOMAS: Audio Signal Modeling with Sinusoids Plus Noise
Master of Science Thesis, 62 pages, 6 enclosure pages
Examiner: Prof. Jaakko Astola, MSc Anssi Klapuri
Financial support: Tampere University of Technology, Signal Processing Laboratory
March 2001
Keywords: Spectrum modeling, mid-level representation, sinusoidal modeling, sound source separation

In audio signal spectrum modeling, the aim is to transform a signal to a more easily applicable form, removing the information that is irrelevant in signal perception. Sinusoids plus noise model is a spectral model, in which the periodic components of the sound are represented with sinusoids with time-varying frequencies, amplitudes and phases. The remaining non-periodic components are represented with a filtered noise. The sinusoidal model utilizes the physical properties of musical instruments and the noise model the humans' inability to perceive the exact spectral shape or phase of stochastic signals.

In the case of polyphonic music signals, the estimation of the parameters of sinusoids is a difficult task, since the periodic components are usually not stable. A sufficient time and frequency resolution is also difficult to achieve at the same time. A big part of this thesis discusses the detection and parameter estimation of periodic components with several algorithms. In addition to already existing algorithms, a new iterative algorithm is presented, which is based on the fusion of closely spaced sinusoids.

The sinusoidal model is applied in the separation of overlapping sounds and manipulation. In the sound separation, a new perceptual distance measure between sinusoids is used. The perceptual distance measure is based on the humans' way to associate spectral components into sound sources. Also a new separation method based on the multipitch estimation is explained. The modification of the pitch and time scale of sounds with the sinusoid plus noise model without affecting the quality of the sound is explained shortly, too.

1. Introduction

This thesis describes methods for sinusoids+noise signal modeling, with the aim of applying them in machine hearing and in the content analysis of musical signals. Until quite recently, most of the work in machine hearing has been done in the area of speech recognition. In the last few years, more interest has started to emerge towards the general computational auditory scene analysis. Recent studies in this area have shown that only in very limited cases we can achieve results comparable to human hearing. In general, the human auditory system is still superior compared to computer for example in recognition tasks. It is therefore natural to try to design systems which process signals more in the way that our own auditory system does.

In most cases, the standard pulse code modulated (PCM) signal which basically describes the sound pressure levels reaching the ear is not a good presentation for the analysis of sounds. A general approach is to use spectrum modeling, or a suitable middle-level representation to transform the signal into a form that can be generated easily from the PCM signal, but from which also the higher level information can be more easily obtained. The sinusoids+noise model is one of these representations. The sinusoidal part utilizes the physical properties of general resonating systems by representing the resonating components by sinusoids. The noise model utilizes the inability of humans to perceive the exact spectral shape or phase of stochastic signals.

Automatic transcription is one interesting application area of machine hearing. The large number of different kinds of instruments and their wide pitch range, variety of spectra or other characteristics make the problem very challenging. The main focus of our sinusoids+noise system is in musical signals. The system is built upon ideas taken from several other sinusoids+noise modeling systems, with some original algorithms proposed here. The system was designed modular in order that different algorithms could be tested in each stage of processing. The system was implemented in Matlab environment.

Since the sinusoids+noise model has the ability to remove irrelevant data and encode signals with lower bit rate, it has also been successfully used in audio and speech coding. Even though the perceptual quality of the synthesized sounds was one criteria when this system was built, the main emphasis was the usability in the selected applications, sound separation and signal analysis.

1.1 Sinusoids plus noise signal model

The sounds produced by musical instruments and other physical systems can be modeled as a sum of deterministic and stochastic parts, or, as a sum of a set of sinusoids plus noise residual [Serra 1997]. Sinusoidal components are produced by a vibrating system, and are usually harmonic. The residual contains the energy produced by the excitation mechanisms and other components which are not result of periodic vibration.

In the standard sinusoidal model, the deterministic part of the signal $x(t)$ is represented as a sum of sinusoidal trajectories (see Table 1 for term definitions) with time-varying parameters:

$$x(t) = \sum_{i=1}^{N} a_i(t)\cos(\theta_i(t)) + r(t), \quad (1)$$

where $a_i(t)$ and $\theta_i(t)$ are amplitude and phase of sinusoid i at time t, and $r(t)$ is a noise residual, which is represented with a stochastic model. We assume that the sinusoids are locally stable, which means that the amplitudes do not exhibit arbitrarily rapid changes, and that the phases are locally linear. The whole signal is modeled with either a sinusoidal or a stochastic model, thus the residual $r(t)$ contains all the components of signal $x(t)$ that are not modeled with sinusoids, including sinusoids that have not been detected.

The human sound perception is not sensitive to the detailed spectral shape or phase of non-periodic signals. Assuming that the residual contains only stochastic components, it can be represented with filtered white noise. Neither instantaneous amplitude nor phase of the residual is retained, but instead it is modeled with a time-varying frequency-shaping filter or with short-time energies within certain frequency bands such as Bark bands. Taking into account these facts, the sinusoids+noise model can be considered as a model arising from both physical and physiological properties.

Table 1: Term definitions

term	definition
trajectory, track	sinusoidal components with time-varying frequencies, amplitudes and phases, appearing as trajectories in the time-frequency spectogram
harmonic (partial)	modes of a vibrating system, the frequencies of which are whole number multiples of the fundamental frequency
(noise) residual	what is left when the deterministic part of the signal has been removed
sound separation	process where two a more sound mixed in one signal are separated from the signal and synthesized alone

1.2 General structure of the sinusoids plus noise analysis/synthesis system

There are many existing implementations of the standard sinusoids plus noise model, and a number of improvements to it. The implementation of the standard model is presented here, and improvements are discussed in Chapters 3 and 4. The block diagram of the sinusoids plus noise system is illustrated in Figure 1. At first, the input signal is analyzed to obtain time-varying amplitudes, frequencies and phases of the sinusoids. Then, the sinusoids are synthesized and subtracted from the original signal to obtain the noise residual. The stochastic analysis is applied to the residual to obtain short-time Bark-band energies. The stochastic signal can be resynthesized and added to the synthesized sinusoids to obtain the whole resynthesized signal.

In the parametric domain, we can make modifications to produce effects like pitch shifting or time stretching. The synthesized signals or the residual can be further analyzed, or analysis can be performed directly on the parametric data. For example, we can recognize acoustic noise mixtures like drums using the short-time Bark-band energies [Sillanpää et al. 2000].

The analysis of sinusoids is the most complex part of the system. Firstly, the input signal is divided into partly overlapping and windowed frames. Secondly, the short-time spectrum of the frame is obtained by taking a discrete Fourier transform (DFT). The spectrum is analyzed, prominent spectral peaks are detected and their parameters, amplitudes, frequencies, and phases, are estimated.The methods for peak detection and parameter estimation are discussed in detail in Chapter 3.

Once the amplitudes, frequencies and phases of the detected sinusoidal peaks are estimated, they are connected to form interframe trajectories. A peak continuation algorithm tries to find the appropriate continuations for existing trajectories from among the peaks of the next frame. The obtained sinusoidal trajectories contain all the information required for the resynthesis of the sinusoids. The sinusoids can be synthesized by interpolating the parameters of trajectories and summing the resulting waveforms up in time domain. Peak continuation algorithms and the sinusoidal synthesis are discussed in Chapter 4.

The stochastic part of the signal is obtained by subtracting the synthesized sinusoids from the original signal in time domain. This residual is represented with filtered noise. Since human auditory perception can not difference the change of energy inside certain frequency bands called Bark bands for noise-like, stationary signals, the exact spectral shape is not required. For stochastic processes, the phases are not perceptually irrelevant, too, and can therefore be discarded. As a consequence, the only information needed for noise-like signals is the short-time energies within each Bark band. In stochastic analysis, the complex spectrum of the residual is calculated and short-time energies within each Bark band are estimated. In synthesis, we generate the complex spectrum by generating a random phase for amplitudes that are obtained from the Bark-band energies. Adjacent frames are combined using overlap-add synthesis. The stochastic model is discussed in Chapter 5.

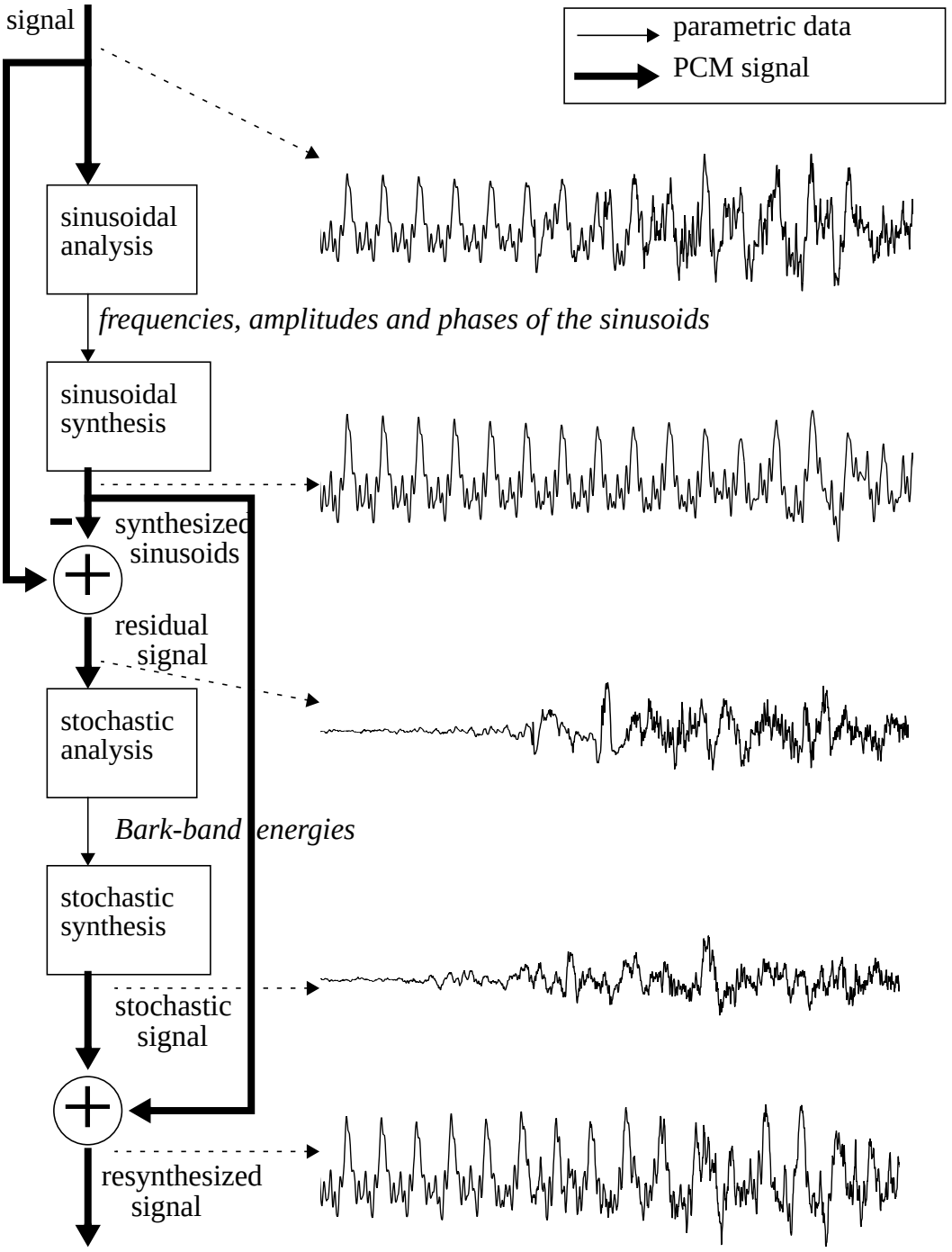

Figure 1: General implementation of the standard sinusoids+noise system. In the right half of the figure there is a plot of an example signal at each stage of the processing. The example signal is a mixture of bowed violin and a snare drum.

2. Literature Review

2.1 Mid-level representations

Human perception of audio signals can be viewed as a sequence of presentations from “low” to “high” [Ellis&Rosenthal 1995]. Low-level presentations correspond to signals before the inner ear. High-level representations are those to which we have a cognitive access, like “Lasse playing the bass guitar while a mobile phone rings in the background”. Between these two levels we have representations that are called mid-level. The sinusoids+plus noise model can be considered as one rather efficient choice to carry out the functions of a mid-level representation.

The idea of spectrum modeling is to discard any information that is useless in human audio perception. The original signal reaching human ear has elements like phase, which in general are not needed in the monaural sound perception.

The knowledge of the mid-level representations of the human auditory system is somewhat limited. In computational auditory scene analysis we try to build models which have the same properties as human auditory system. Ellis and Rosenthal list the following desirable properties for auditory mid-level presentations [Ellis&Rosenthal 1995]:

1. Sound source separation. Natural sounds overlap with each other, and our hearing has the ability to organize the sounds to their sources of production from the complex mixture.
2. Invertibility. From a parametric representation, we can regenerate the original signal, although to a perceptual rather than bit-wise criterion.
3. Component reduction. The original signal reaching the eardrum can be considered as an array of air pressure levels. As we represent it, the number of objects should decrease while the meaningfulness of each should increase.
4. Abstract salience of attributes. The features that the representation uses should correspond to the physical characteristics rather than algorithmic details.
5. Physiological plausibility from the human auditory physiology point of view.

Sinusoids+noise model meets well the second and third criteria, since we can synthesize the analyzed signal from the obtained parameters and the number of parameters is usually quite low. The sinusoidal model allows separation of sound sources, and one possible approach is presented in Chapter 7. In general, the noise model does not allow sound source separation. Sinusoidal model also meets the fourth criteria somehow: an onset of a sinusoid corresponds to an onset of a sound, and the frequencies of the sinusoids correspond to resonant frequencies of sound sources.

The physiological plausibility of the whole sinusoids+noise model is poor. The sinusoidal model is more physically than physiologically oriented. However, this can be considered also as an advantage, too. The model produces oversimplified data, for which only a minimum amount of deduction has been done. If higher level information is desired, the data can be easily analyzed using an upper level analysis, which for example combines the sinusoids into separate sound sources.

2.2 Spectral models related to the sinusoids+noise model

Additive synthesis is a traditional sound synthesis method that is very close to the sinusoidal model. It has been used in electronic music for several decades [Roads 1995]. Like the sinusoidal model, it represents the original signal as a sum of sinusoids with time-varying amplitudes, frequencies, and phases [Moorer 1985]. However, it does not make any difference between harmonic and non-harmonic components. To represent non-harmonic components it requires a very large amount of sinusoids, therefore giving best results for harmonic input signals.

Vocoders are an another group of spectral models. They represent the input signal at multiple parallel channels, each of which describes the signal at a particular frequency band [Tolonen et al. 1998]. Vocoders simplify the spectral information and therefore reduce the amount of data. Phase vocoder is a special type of vocoder, which uses a complex short-time spectrum, thus preserving the phase information of the signal. The phase vocoder is implemented with a set of bandpass filters or with a short-time Fourier transform. The phase vocoder allows time and pitch scale modifications, like the sinusoidal model does [Dolson 1986].

Source–filter synthesis uses a time-varying filter and an excitation signal, which is either a train of impulses or white noise. While the desired signal is obtained by filtering the broadband excitation, the method is also called subtractive synthesis. This method approximates human speech production system, and it is often used in speech coding [Moorer 1985]. The filter coefficients can be obtained e.g. by the linear predictive analysis. For voiced speech, a periodic pulse train is used as an excitation, and white noise is used for unvoiced speech. Naturally, the voiced excitation can be used only for monophonic signals. However, the idea of using filtered noise for non-harmonic signals is quite close to the stochastic synthesis used in our system. Our system uses Bark-band energies instead of time-varying filters, which in general case is psychoacoustically better justified.

2.3 Sinusoids + noise modeling systems

The sinusoidal model was originally proposed by McAulay+Quatieri for speech coding purposes and by Smith+Serra [McAulay&Quatieri 1986; Smith&Serra 1987] for the representation of musical signals. Even though the systems were developed independently, they were quite similar. Some parts of the systems such as the peak detection were slightly different, but both systems had all the basic ideas needed for the sinusoidal analysis and synthesis: the original signal was windowed into frames, and the short-time spectrum was

examined to obtain the prominent spectral peaks. The frequencies, amplitudes and phases of the peaks were estimated and the peaks were tracked into sinusoidal tracks. The tracks were synthesized using linear interpolation for amplitudes and cubic polynomial interpolation for frequencies and phases.

Serra [1989] was the first to decompose the signal into deterministic and stochastic parts, and to use a stochastic model with the sinusoidal model. Since then, this decomposition has been used in several systems. The majority of the noise modeling systems use two kind of approaches: either the spectrum is characterized by a time-varying filter or the short-time energies within certain frequency bands.

2.4 Transient modeling

While sinusoids and noise can be used to model a large variety of sounds, they perform poorly with very rapidly changing signals components called transients. One could use sinusoids to model transients, but since transients often have a large bandwidth, the number of sinusoids required is large. Also, the time-resolution used normally in the sinusoidal analysis is not good enough for transients, because the window length can be much larger than the length of a transient. Using a long window with transients results in an effect often encountered in audio coding: pre-echo.

The mentioned problems can be avoided using a separate model for transients. A transient detector determines where the transients are located. While other parts of the signal are represented with the parametric sinusoids+noise model, the detected transients are represented with non-parametric transform coding [Levine 1998]. Transform coding is used only for a short amount of time (66 ms). Transient model has been used together with sinusoids+noise model in the systems presented in [Ali 1996], [Levine 1998] and [Verma 1999].

The transient model is not included in our system, because it was considered that from the auditory scene analysis point of view this information would not give significant improvement. If we consider the quality of the synthesized sound, it is clear that adding the transient model would improve the quality a lot. However, our main purpose was to construct a good mid-level representation for audio content analysis, not an audio coder.

2.5 Pitch-synchronous analysis

The estimation of the sinusoidal modeling parameters is a difficult task in general. Most of the problems are related to the analysis window length. If the input signal is monophonic, or consist of harmonic voices that do not overlap in time, it advantageous to synchronize the analysis window length to the fundamental frequency of the sound. Usually the frequencies of the harmonic components of voiced sounds are integral multiples of the fundamental frequency.

The advantage of the pitch-synchronous analysis is most easily seen in the frequency domain: the frequencies of the harmonic components correspond exactly to the frequencies of the DFT coefficients. The estimation of the parameters is very easy, since no interpolation is needed, and the amplitudes and phases can be obtained directly from the complex spectrum. Also, pitch-synchronous analysis allows the use of window lengths as small as one period of the sound, while non-synchronized windows have to be 2-4 times the period depending on the estimation method. This means that a much better time resolution is gained by using the pitch synchronous analysis.

Unfortunately, pitch-synchronous analysis can not be utilized in the case where several sounds with different fundamental frequencies occur simultaneously. In general, monophonic recordings represent only a small minority among musical signals and therefore pitch-synchronous analysis typically can not be used. To keep the complexity of the system low, the pitch-synchronous analysis was not included in our system.

Adaptive window length has been successfully used in modern audio coding systems, but in a quite different manner: a long window is used for stationary parts of the signal, and when rapid changes occur, the window is switched into a shorter one. This enables good frequency resolution for the stable parts and a good time resolution in rapid changes.

3. Peak Detection and Parameter Estimation

In this thesis, the basic principles and theoretical background of sinusoidal analysis algorithms are presented in Chapters 3 and 4. The practical performance of the algorithms is studied in Chapter 6. Based on the simulations and general knowledge gained during the implementation of the algorithms, two algorithm sets were chosen to be used in practical applications. These sets are described in Chapter 6.5.

The sinusoidal analysis constitutes an integral part of the overall sinusoids+noise system, as depicted in Figure 1. The sinusoidal analysis module can be further divided into four steps, which are presented in Figure 2. At first, meaningful peaks in the incoming signal are detected. Second, the peaks are interpolated to obtain better frequency resolution. Third, the amplitudes and phases of the detected peaks are estimated, and finally the peaks are connected into trajectories.

Several alternative methods exists for each of the four analysis steps. In this chapter, we present the algorithms of the first three phases tested in our sinusoidal analysis system. There are two peak-detection algorithms, two peak interpolation algorithms and two iterative parameter estimation methods. The continuation part is discussed in Chapter 4.

Peak detection is a crucial part in a sinusoidal modeling system, since sinusoidal synthesis is done using the detected peaks only. There are many fundamental problems in the estimation of the meaningful peaks and their parameters. Most of these problems are related to

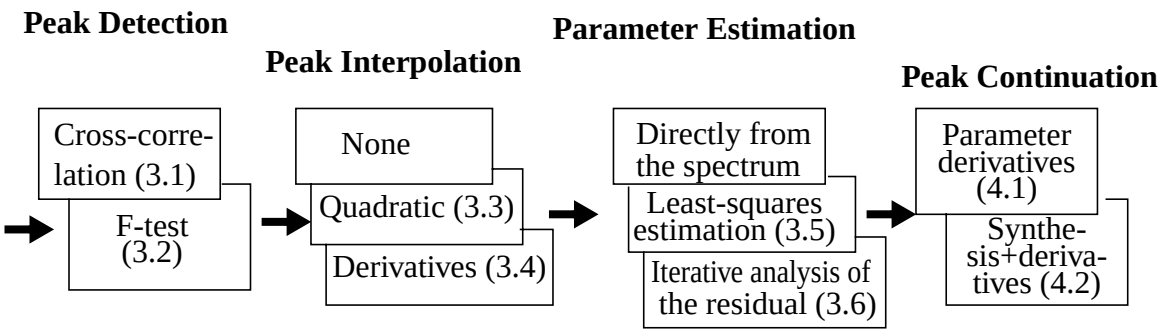

Figure 2: The phases and algorithms of the sinusoidal analysis. The numbers in parenthesis refer to chapters where each algorithm is explained.

the length of the analysis window: a short window is required to follow rapid changes in the input signal, but a long window is needed to estimate accurate frequencies of the sinusoids or to distinguish spectrally close sinusoids from each other.

What is a “meaningful peak” is a fundamental question. If fast changes in the amplitude and frequency are allowed, even the stochastic part of the signal can be modeled using a large number of sinusoids. In general, that is not what we want in sinusoidal modeling. Instead we want to use sinusoids to represent the harmonic partials of a periodic sound.

In almost all the sinusoidal analysis systems the peak detection and parameter estimation is done in the frequency domain using the DFT. This is natural, since each stable sinusoid corresponds to an impulse in the frequency domain. Because natural sounds are never infinite-duration stable sinusoids, we have to analyze the time-domain signal at several time instants using a sliding window and a short-time Fourier transform (STFT).

Usually zero-padding is used to increase the frequency resolution of the short-time spectrum [Laroche 1998]. If N is a power of two, we can use the fast Fourier transform (FFT) algorithm, which is a computationally efficient implementation of the DFT. The signals, spectrum and sinusoidal peaks in each stage of processing are illustrated in Figure 3.

A peak or a local maximum in the magnitude of the STFT indicates the presence of a sinusoid at a nearby frequency. The simplest method for detecting sinusoids in the signal is therefore to choose a fixed number of local maxima in the magnitude of the STFT. This method is very fast and produces a fixed bit rate, this why it is often used in audio coding applications. For analysis purposes, a fixed number of sinusoids is not practical: in the case of non-harmonic sounds, the method picks peaks caused by noise, which causes problems in subsequent analysis. In the case of polyphonic signals, the number of harmonic partials is large, and a fixed number of sinusoids may not suffice to model all of them.

A natural improvement of the method is to use a threshold for peak detection: all local maxima of the magnitude of the STFT above the threshold are interpreted as sinusoidal peaks. This method produces a variable number of peaks. However, it does not remove the problem that some peaks in magnitude spectrum can be caused by noise, or other non-harmonic sounds. Also, it does not take into account the overall spectral shape and amplitudes of the harmonics, which in the case of natural sounds are usually decreasing as a function of frequency, or, natural sounds have most of their energy at lower frequencies. As a consequence, higher harmonic partials often fall below the fixed threshold, and are not detected. For these reasons, we concentrated on two more sophisticated peak detection algorithms, the cross-correlation method and f-statistics.

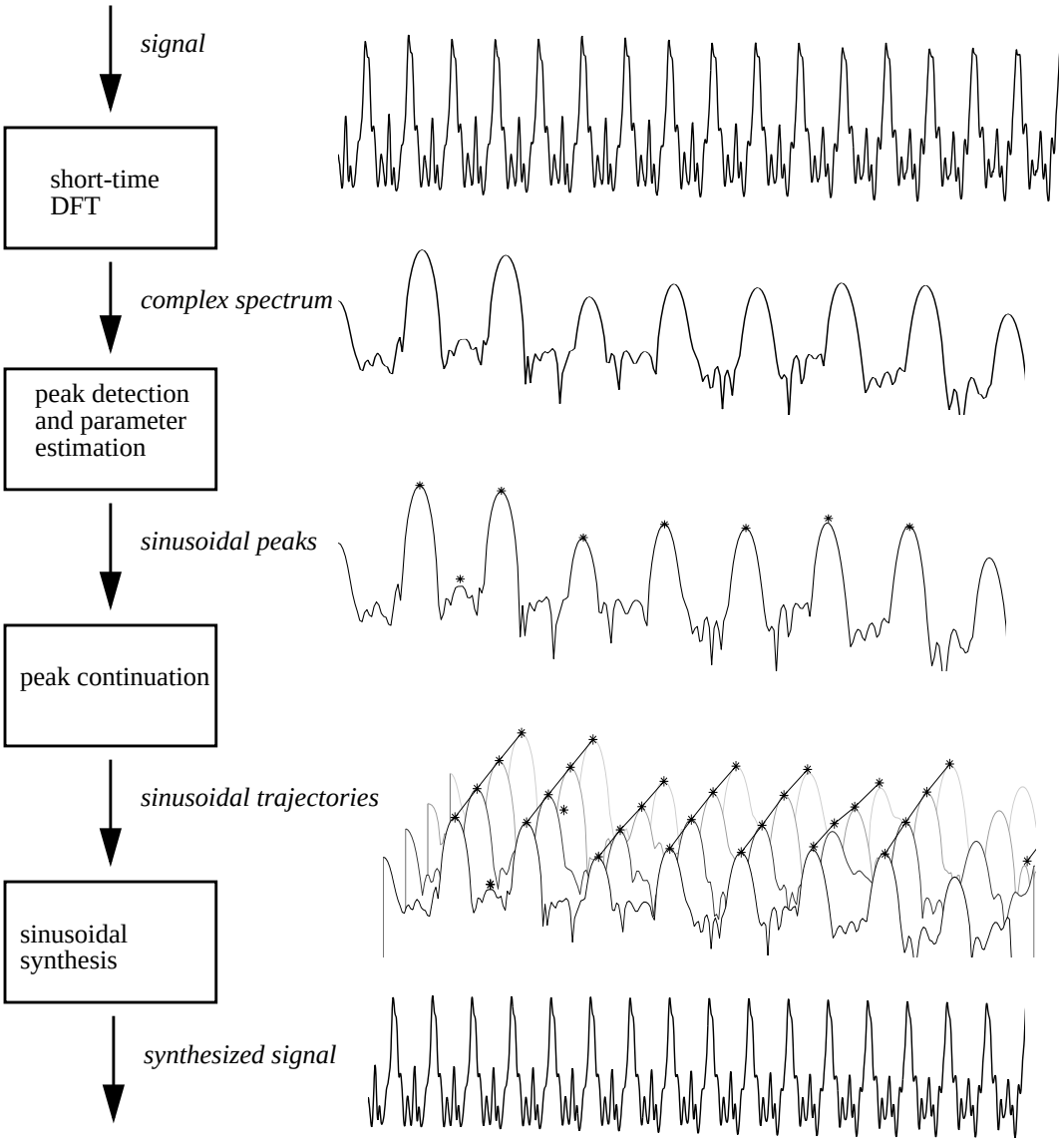

Figure 3: Block diagram of the general implementation of the sinusoidal analysis-synthesis process. The example signal in the right side of the figure is a bowed violin.

3.1 Cross-correlation method

Sinusoids can be defined as frequency components which have significantly more energy than the adjacent frequencies. The cross-correlation method makes use of this idea by calculating the cross-correlation between the short-time spectrum of the signal and the spectrum resulting from an ideal sinusoid, and scaling the result by the overall spectral shape. The obtained result is called sinusoidal likeness measure.

The cross-correlation method has been used successfully e.g. in speech coding [Griffin&Lim 1985], where the voicing index is similar to sinusoidal likeness measure. In musical signals, several sinusoidal components can be close to each other in frequency, so it is difficult to obtain a measure how voiced/unvoiced an individual components it. However, the method is able to detect a large number of sinusoids in many different conditions.

The spectrum $S(\omega_k)$ of a single sinusoid is a scaled and phase-shifted shape of $H(\omega_k - \Omega)$, which is the spectrum of the analysis window translated at frequency Ω [Rodet 1997]. In harmonic sounds, we have a sum of several $H(\omega_k)$ translated, scaled and phase-shifted to several different frequencies, amplitudes and phases. It is therefore natural to look at the cross-correlation function between $H(\omega_k)$ and $X(\omega_k)$, the STFT of the windowed signal. Usually $H(\omega_k)$ values are very small at high frequencies, so we can calculate the cross-correlation using only a narrow bandwidth [-*W*,*W*] of $H(\omega_k)$:

$$r(\omega) = \sum_{k,\,|\omega - \omega_k| < W} H(\omega - \omega_k) X(\omega_k). \tag{2}$$

If we define norms for $H(\omega_k)$ and $X(\omega_k)$ at frequency Ω by:

$$|H|_\Omega^2 = \sum_{k,\,|\omega - \omega_k| < W} \left|H(\Omega - \omega_k)\right|^2 \text{ and } |X|_\Omega^2 = \sum_{k,\,|\omega - \omega_k| < W} \left|X(\Omega - \omega_k)\right|^2, \tag{3}$$

we get an estimate v_Ω of the likeness between the observed peak and the peak that would result from an ideal sinusoid:

$$\nu_\Omega = \frac{|r(\Omega)|}{|H|_\Omega |X|_\Omega}. \tag{4}$$

v_Ω is always between 0 and 1, $v_\Omega = 1$ resulting from an ideal sinusoid, no noise present.

We also get an estimation of amplitude *A* and phase φ of a sinusoid at frequency Ω by:

$$A(\Omega) = \frac{|r(\Omega)|}{|H|_\Omega^2} \text{ and} \tag{5}$$

$$\varphi(\Omega) = Arg[r(\Omega)]. \tag{6}$$

We can use v_Ω to detect sinusoids and their frequencies by setting a fixed limit, which is between 0 and 1, and choosing frequency points that are local maxima of v_Ω and above the fixed threshold.

In Figure 4 we have the amplitude spectrum of a windowed violin sample and the sinusoidal likeness measure calculated for the same sample. As we can see from the amplitude spectrum, the overall spectral level is lower at higher frequencies. In sinusoidal likeness measure this is taken into account: the harmonics at high frequencies have large sinusoidal likeness measure even though their amplitudes are about 20 dB lower than those of the low

harmonics. The sinusoidal likeness measure is a bit lower for higher harmonics, but this is explained by the fact that the violin has some high-frequency noise caused by the exciting bow, and therefore the higher harmonics are not ideal sinusoids.

Cross-correlation is a convolution where the time scale of the other signal is inverted [Hartmann 1997]. The cross-correlation of the frequency domain signals can therefore be implemented using a multiplication for the time-domain signals. For a large bandwidth *W*, $r(\omega)$ can be calculated more efficiently using the FFT for $x(t)$ windowed twice with the analysis window $h(t)$. Because the calculation of $|X|_{\Omega}^{2}$ can be viewed as an filtering operation with an FIR filter which coefficients are all one, the FIR filter can be replaced with an IIR filter which has only two coefficients not zero: one delay takes a cumulative sum of the incoming signal and the other delay subtracts the values at the end of the window. This makes the computation of v_{Ω} very efficient.

The sinusoidal likeness measure assumes that there is only one sinusoid inside the bandwidth *W*. In most cases, we have to use a small bandwidth to handle dense groups of harmonic partials. On the other hand, noisy conditions require that the threshold is small

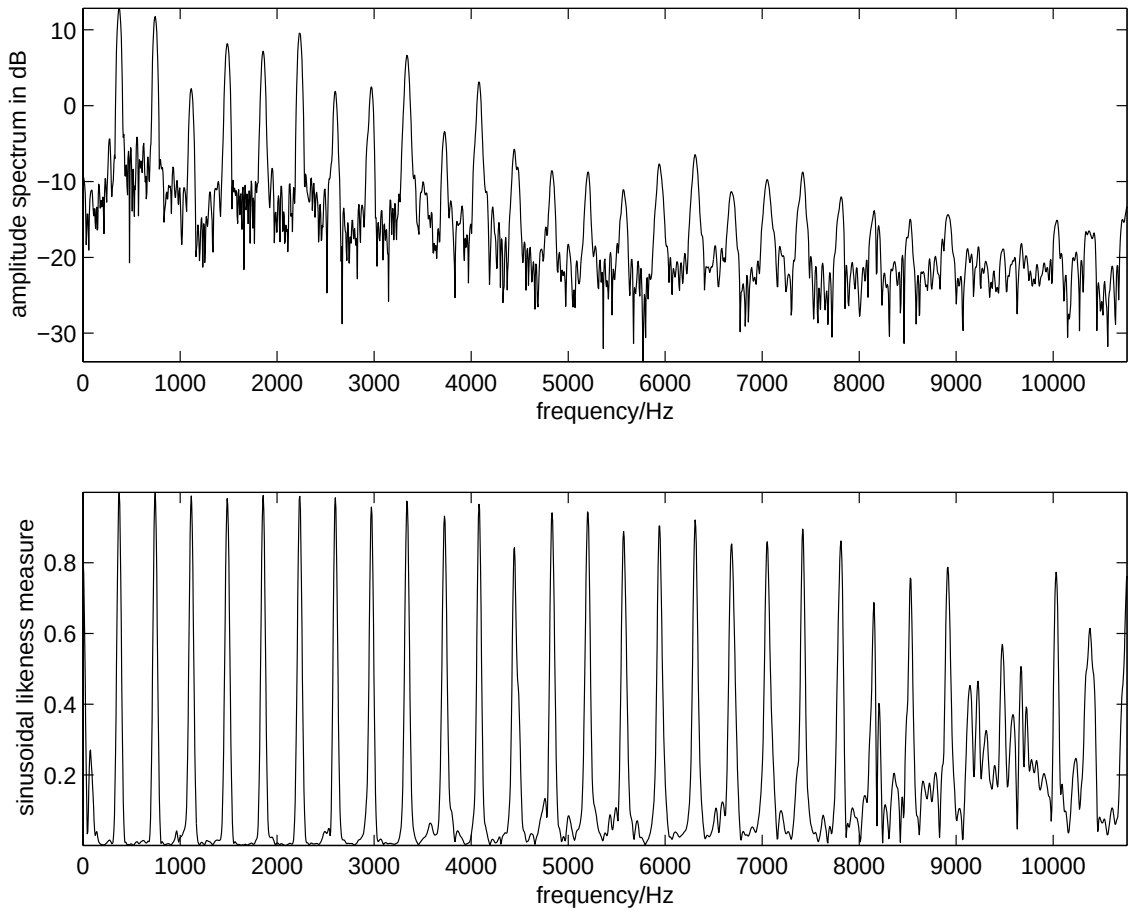

Figure 4: Upper plot: the amplitude spectrum of a bowed violin. The length of the sample is 45 ms. Lower plot: sinusoidal likeness measure for the same sample. The violin has also noise components at high frequencies and therefore the sinusoidal likeness measure is not unity at high frequencies.

enough to detect sinusoids with low amplitudes. It follows that in frequency areas where there is no sinusoids, small bandwidth and threshold cause peaks caused by noise or sidelobes of other sinusoids to be interpreted as sinusoids. Therefore, inside one small frame, we are not able to judge reliably if there exists a sinusoid at certain frequency, and information from adjacent frames is needed for a reliable sinusoidal analysis.

From a psychoacoustic point of view, it can be stated that representing peaks caused by noise with sinusoids is not always an error. In some cases human auditory system tries to assign pitch for signal components that are not periodic, for example for delayed broadband noise or for repeated noise pulses [Meddis&Hewitt 1991]. However, our goal is to represent only periodic components with sinusoids. In practise the cross-correlation method was found to perform robustly for dynamic parameters, especially time-varying frequencies.

3.2 F-test

A statistical test developed by Thomson [1982] has been originally developed in geophysics, but it has been successfully used in audio in the detection of sinusoids for example in [Ali 1996] and [Levine 1998]. The method employs a set of orthogonal windows called discrete prolate spheroidal sequences. To treat bias and smoothing problems, an estimate of the spectrum is calculated as a weighted average of several data windows.

Like the cross-correlation method, the F-test gives a value for each frequency component, which tells how probable it is that there is a sinusoid at this frequency. In the case of the F-test, this value is called the f-value. We can set a fixed threshold so that frequency coefficients which f-value are local maxima and larger than the threshold are interpreted as a sinusoid at this frequency. Like the cross-correlation method, F-test also measures the ratio of harmonic components to continuous, non-harmonic part of spectrum. The spectrum of the residual is assumed to be smooth.

The calculation of the discrete prolate spheroidal sequences is explained in [Thomson 1982; Verma 1999]. These sequences are used as windowing functions, and in a finite frequency interval, the energy of these windows is most concentrated [Verma 1999]. The input signal is windowed with each sequence and several estimations of the spectrum are obtained by taking the FFT of each windowed signal. As the sequences are orthogonal, they do not correlate with each other. The harmonic mean of all the estimations is used as a more reliable estimation of the spectrum.

The variance of the estimated mean depends on the local continuous part of the spectrum, and gives an estimate of the background spectrum. By comparing the power at a particular frequency to the continuous part of the spectrum we get f-value.

Because the F-test requires several FFTs, it is computationally more expensive than the cross-correlation method. In ideal conditions, it is very reliable, and is able to detect sinusoids without picking noise peaks. In non-ideal conditions, such as closely spaced sinuso-

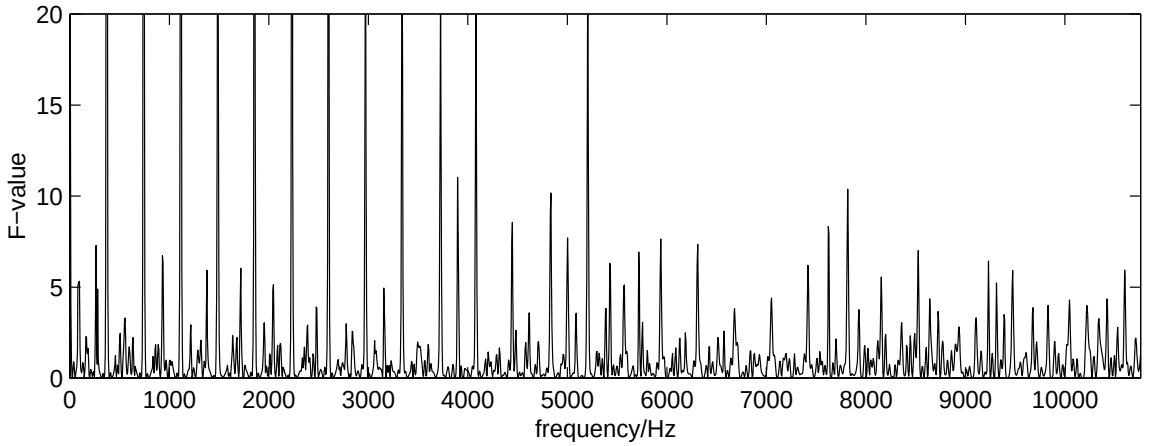

Figure 5: F-value of the violin signal, which amplitude specrum is in the upper plot of Figure 4. At low frequencies the F-value is very large and was therefore left out of the figure.

ids or rapidly changing amplitudes or frequencies, it does not perform as well as the cross-correlation method. If the frequencies of sinusoids are close to each other, they may "cancel" each other out. If the window length is small compared to the wavelength of the sinusoid, the performance of the F-test reduces dramatically.

The f-value of the violin sample, the amplitude spectrum of which is in Figure 4, is illustrated in Figure 5. The bowed violin has some vibrato, or, frequency modulation, which affects more f-value than sinusoidal likeness measure. Usually the differences between F-test and sinusoidal likeness measure are not as clear as in Figures 4 and 5.

3.3 Quadratic interpolation

According to the Heisenberg Uncertainty principle, the frequency resolution is limited in a finite time frame. However, if a sinusoid is the only significant component in its vicinity, zero-padding can be used to get a better resolution of the DFT. This makes the spectral shape and place of the sinusoid more clear and enables more accurate parameter estimation.

Each DFT coefficient represents a frequency interval of F_s / N, where F_s is the sampling frequency and N is the length of the DFT. One semitone, i.e, interval between adjacent notes in the Western musical scale can be less than one Hz at bass frequencies. For high quality sampling frequencies, a DFT length of tens or even hundreds of thousands of samples would be required. This is not practical, so a different method is needed to obtain the accurate frequencies of sinusoids. Originally in [Smith&Serra 1987], a method is described which applies a quadratic function to obtain the accurate frequencies of the sinusoidal components.

A local maximum of $|X(\omega)|$, the magnitude of the spectrum of a windowed signal, indicates the presence of a sinusoid at a nearby frequency. The shape of a windowed sinusoid is $|H(\omega - \Omega)|$, the sampled shape of the DFT of the window function translated at frequency Ω. If the window function $h(t)$ is symmetric, a quadratic function centered at Ω gives a good approximation of the of $|H(\omega - \Omega)|$ around Ω [Rodet 1997]. We can estimate the parameters of the function using only three points of the DFT spectrum. For the window functions used, the logarithm of H gave better results than using just H. This assumes that H is gaussian near zero, so the logarithm of H is quadratic. If $|X(\omega_{\lambda-1})|$, $|X(\omega_\lambda)|$ and $|X(\omega_{\lambda+1})|$ are adjacent values of the magnitude of the spectrum, $|X(\omega_\lambda)|$ being a local maximum, the quadratic function is:

$$f(\omega) = a\omega^2 + b\omega + c = \log|X(\omega)|,\ \omega \approx \omega_\lambda . \tag{7}$$

The values for a, b and c are obtained by find a quadratic function that goes through the points, and setting the derivative of the quadratic function equal to zero we get estimations for the amplitude and frequency:

$$a_{peak} = |X(\omega_\lambda)| + \frac{1}{8}\frac{\log|X(\omega_{\lambda+1})| - \log|X(\omega_{\lambda-1})|}{\log|X(\omega_{\lambda+1})| + \log|X(\omega_{\lambda-1})| - 2\log|X(\omega_\lambda)|} \tag{8}$$

$$\omega_{peak} = \omega_\lambda + \frac{\log|X(\omega_{\lambda+1})| - \log|X(\omega_{\lambda-1})|}{\log|X(\omega_{\lambda+1})| + \log|X(\omega_{\lambda-1})| - 2\log|X(\omega_\lambda)|}(\omega_{\lambda+1} - \omega_\lambda) . \tag{9}$$

Using the obtained frequencies, the phase spectrum can be interpolated for example using the weighted average of two DFT coefficients that are nearest to the exact frequencies. The quadratic interpolation of a single sinusoid is illustrated in Figure 6

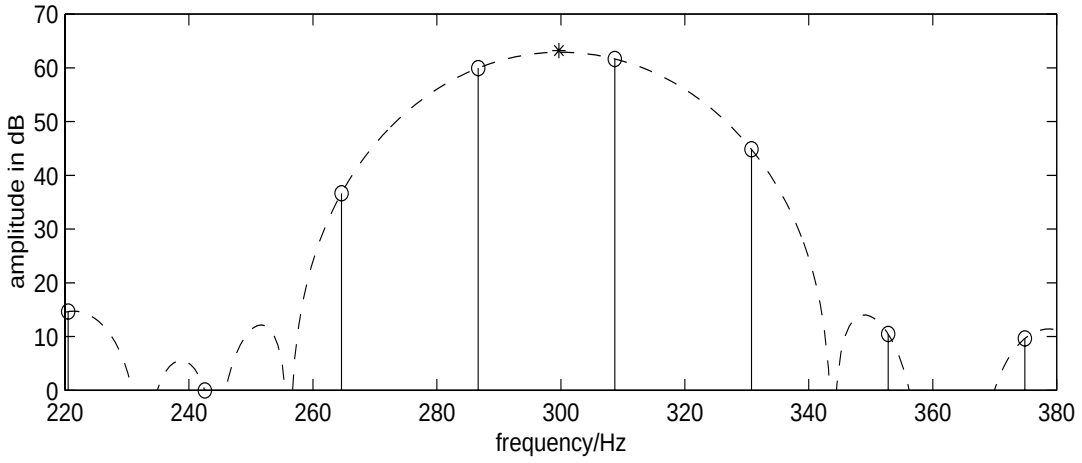

Figure 6: The quadratic interpolation of a single 300 Hz sinusoid. The stems with circles are coefficients of the original amplitude spectrum. Using the three largest coefficients and quadratic interpolation we obtain peak which is marked with a star. The dashed line is the amplitude spectrum calculated with zero-padded FFT.

Even though the method is based on the ideal sinusoid assumption, or, no noise or other sinusoids present, it is usually a good way of interpolating frequencies even in polyphonic signals. It should be noticed that the quadratic interpolation assumes that the center amplitude $|X(\omega_\lambda)|$ is larger than adjacent amplitudes $|X(\omega_{\lambda-1})|$ and $|X(\omega_{\lambda+1})|$. In the case that a more complex peak detection algorithm than just choosing the local maxima of the amplitude spectrum is used, it is possible that the center amplitude is not the largest. In that case, the quadratic interpolation cannot be used.

3.4 Signal derivative interpolation

Desainte-Catherine and Marchand have shown that the DFTs of the signal and its derivatives can be utilized to obtain the exact frequencies and amplitudes of the spectral components [Desainte-Catherine&Marchand 2000]. Taking the derivative of a signal does not affect the frequencies of the sinusoids, and ideally the change of amplitudes is linearly dependent on the frequencies. If $v(t)$ is the derivative of $x(t)$ which Fourier transform is $X(\omega)$, the Fourier transform $V(\omega)$ of $v(t)$ is

$$V(\omega) = j\omega X(\omega). \tag{10}$$

Coefficient $j\omega$ is only a theoretical gain which does not apply in the case of discrete time processing. The derivative of the signal has to approximated by the first-order difference. The first-order difference can be viewed as a filtering operation with a first-order linear filter. The error between the ideal and approximated spectrum of the derivative of the signal can be corrected by a scaling factor F [Desainte-Catherine&Marchand 2000]:

$$F(\omega) = \frac{\omega}{2\sin(\omega/2)}. \tag{11}$$

When DFT^1, the DFT of the derivative of the signal has been corrected by the scaling factor F, the frequency ω_{peak} of a sinusoid can be approximated by dividing the DFT^1 at frequency ω by DFT^0, the original DFT of the signal:

$$\omega_{peak} = \frac{1}{2\pi}\frac{\mathrm{DFT}^1(\omega)}{\mathrm{DFT}^0(\omega)}. \tag{12}$$

Naturally, both signals are windowed before taking the DFT.

In our preliminary simulations this method did not give quite as accurate estimates of frequencies as the quadratic interpolation, even though a remarkable improvement was achieved compared to the estimation of parameter without any interpolation. Especially in noisy conditions the quadratic interpolation performed better. As the quadratic interpolation is very commonly used, our emphasis was on it. However, a more detailed test with a generated test a signal showed (see Chapter 6.2) that the performance of the derivative interpolation is almost equal to the quadratic interpolation.

3.5 Iterative least-square estimation

Even with the most advanced methods, it is difficult to estimate the sinusoidal parameters of a complex sound by analyzing the signal only once in each time frame. One possibility is to estimate the parameters with a simple estimation method, and then iteratively improve the parameter set [Depalle & Helie 1997; Tolonen 1999].

If we assume that the amplitudes a_k and frequencies ω_k of sinusoids remain constant inside one frame, the sinusoidal model for one frame is:

$$\hat{s}(n) = \sum_{k=1}^{K} a_k \cos(2\pi\omega_k n + \phi_k), \tag{13}$$

where K is the number of the sinusoids and ϕ_k is the initial phase of the k^{th} sinusoid. An estimation of the STFT of the model $\hat{s}(n)$ is given by [Depalle&Helie 1997]:

$$\hat{S}(\omega) = \sum_{k=1}^{K} \frac{a_k}{2}(e^{j\phi_k}H(\omega - \omega_k) + e^{-j\phi_k}H(\omega + \omega_k)), \tag{14}$$

where $H(\omega)$ is the Fourier transform of the analysis window. Our goal is to find parameters a_k, ω_k and ϕ_k which minimize the least-square error $\|S - \hat{S}\|$ between the true STFT and the estimated STFT. Both STFTs are measured at N equally spaced frequencies $\omega_i = i/N$ for i=0,...N-1. The expression 14 for $\hat{S}$ is nonlinear in terms of ω_k, and even if the dependence between $\hat{S}$ and ω_k was linearized, the expression for $\hat{S}$ contains products of unknown parameters. Also K, the number of sinusoids is unknown. Therefore, there is no analytical solution to the least-square problem.

Starting from the estimates of a_k, ω_k and ϕ_k which are obtained using some other estimation method, we can iteratively improve accuracy of the estimates. First, the amplitudes and phases are solved, assuming that the frequencies are correct. Then, the frequency estimates are improved assuming that the amplitudes and phases are correct. This procedure is repeated several times, resulting in better estimates for the parameters at each iteration. During the iteration process, the number of sinusoids can be altered, so we can remove and add sinusoids when necessary.

Amplitude and phase estimation

Assuming that the number of sinusoids and the frequency of each are known, the spectrum estimate of Equation 14 can be rewritten as:

$$\hat{X}(\omega) = \sum_{k=1}^{2K} p_k R_k(\omega), \tag{15}$$

where parameters p_k are

$$\begin{cases} p_k = \frac{a_k}{2}\cos\phi_k & k \in [1, K] \\ p_{K+k} = \frac{a_k}{2}\sin\phi_k & k \in [1, K] \end{cases} \tag{16}$$

and the known 2*K* expressions related to Fourier transform of the window function

$$\begin{cases} R_k(\omega) = H(\omega - \omega_k) + H(\omega + \omega_k) \\ R_{K+k}(\omega) = jH(\omega - \omega_k) - H(\omega - \omega_k) \end{cases}. \tag{17}$$

If we define a matrix $\Re$ of dimensions *N*x2*K* where $\Re_{i,k} = R_k(\omega_i)$, and vector $\underline{p}$ of the unknown parameters $[p_1, \ldots, p_{2K}]^T$, the spectrum estimate can be written as

$$\underline{\hat{X}} = \Re\underline{p}. \tag{18}$$

Least-square solution for this is [Kay 1993]

$$\underline{p} = (\Re^H\Re)^{-1}\Re^H\underline{X}, \tag{19}$$

from which we get amplitudes by

$$a_k = 2\sqrt{p_k^2 + p_{K+k}^2} \tag{20}$$

and phases by

$$\phi_k = \arg(p_k + jp_{K+k}). \tag{21}$$

For known frequencies, this method gives good results, particularly in a situation where the frequencies of sinusoids are close the each other. In this case, other methods usually perform poorly. However, if the frequencies are too close to each other, or, inside the same F_i interval, $\Re$ becomes singular, the solution does not exist.

Frequency estimation

If we know the amplitudes and phases of the sinusoids, and have rough approximations of the frequencies, the dependence of the model on the frequencies can be linearized. Our goal is to estimate $\Delta_k = \omega_k - \hat{\omega}_k$, the distance between approximations of frequencies $\hat{\omega}_k$ and correct frequencies ω_k. For each frequency measurement point ω_i, we linearize the frequency dependence using a first-order limited expansion of $H(\omega)$. The Fourier Transform of the analysis window can now be written as:

$$H(\omega \mp \omega_k) = H(\omega \mp \hat{\omega}_k) \mp H'(\omega \mp \hat{\omega}_k)\Delta_k + o(\Delta_k^2), \tag{22}$$

where the derivative $H'(\omega)$ can be estimated at discrete frequency points either by using the first order difference of $H(\omega)$ or by taking a DFT of the product $h(t)t$. If we define matrix Ω by

$$(\Omega)_{i,k} = \frac{a_k}{2}(-e^{j\phi_k}H'(\omega_i - \hat{\omega}_k) + e^{-i\phi_k}H'(\omega_i + \hat{\omega}_k)), \quad (23)$$

we can rewrite the spectrum estimate as

$$\hat{X} = \tilde{X} + \Omega\Delta, \quad (24)$$

where $\tilde{X}$ is the STFT model evaluated with frequencies $\hat{\omega}_k$. The least squares solution for the frequencies is:

$$\omega = \hat{\omega}_k + (\Omega^H\Omega)^{-1}\Omega^H(X - \tilde{X}). \quad (25)$$

Because a first-order expansion of the $H(\omega)$ is used, this estimation method is very sensitive to the shape of the analysis window $h(t)$. In practise, this means that the Fourier transform of the analysis window should not have sidelobes. In [Depalle&Helie 1997], a method is presented to design windows with a small bandwidth and a small effective duration.

In an ideal case, or for signals that consist of synthesized sinusoids, the iterative analysis can find good estimates of the parameters even if the initial values are far from correct values. However, in more problematic cases, such as closely spaced sinusoids or complex polyphonic signals, the algorithm performed poorly. If the frequency estimates are close to the correct ones, the method gives good estimates for the amplitudes and phases, but if the frequency estimates are not correct, the algorithm cannot find better estimates for complex signals. Also, the algorithm is computationally very expensive for a large number of sinusoids. This problem can be somehow avoided by splitting the spectrum into separate frequency bands and by solving the parameters separately at each band.

The LSQ algorithm was found most useful for amplitudes and phases only, using a non-iterative implementation. When the frequencies are obtained using a peak detection algorithm, the amplitudes and phases can be solved using the LSQ algorithm in a one pass. Even in case of closely spaced sinusoids, the algorithm outputs the correct parameters provided that the frequencies are correct.

3.6 Iterative analysis of the residual

Another iterative approach is to perform iterative analysis of the residual. Combined with a parameter fusion algorithm, this parameter estimation procedure has two advantages: it decreases the number of sinusoidal components, and gives more accurate parameters for single sinusoids. Since the iterative analysis requires several passes of traditional analysis, it is computationally more expensive. This work is originally presented in [Virtanen 2001].

The iterative analysis proceeds as follows. First, we detect the sinusoids from the original signal with some simple detection method. Second, we synthesize the sines and then subtract them from the original signal in time domain to obtain the residual. Then we detect sinusoids from the remaining residual, and synthesize and subtract them again. This can be repeated a fixed amount of iterations, or until the desired amount of sinusoids is obtained, or no significant harmonic components are left in the residual. While this algorithm produces perceptually good results, the number of sinusoids usually becomes large. The parameters obtained at each single iteration are usually not exactly correct, which results in estimation errors in the residual. The estimation errors of sinusoidal components are also sinusoids the frequencies of which are close to the original, and amplitudes usually smaller than the original amplitude. At subsequent iterations we detect these estimation errors, thus each harmonic component in the original signal is finally represented with more than one sinusoid. While this not desirable, our algorithm combines the sinusoids after each iteration as illustrated in Figure 7.

The parameter fusion is based on the assumption that two closely spaced sinusoids have arisen from the same source, thus we can combine the sinusoids in such a way that the resulting sinusoid represents the underlying harmonic component better than either of the original ones alone. The parameters of the new sinusoid are calculated so that the new sinusoid represents the sum of the original sinusoids. For simplicity, let us assume that we operate around time t=0. Let the amplitudes, frequencies and phases of the two original sinusoids be a_1, a_2, ω_1, ω_2, φ_1 and φ_2. The sum of the sinusoids at time t is

$$x(t) = a_1 \sin(\omega_1 t + \varphi_1) + a_2 \sin(\omega_2 t + \varphi_2) \ . \tag{26}$$

In Appendix A it is shown that $x(t)$, the sum of the sinusoids, can be represented with a single sinusoid, the amplitude and frequency of which are time-varying:

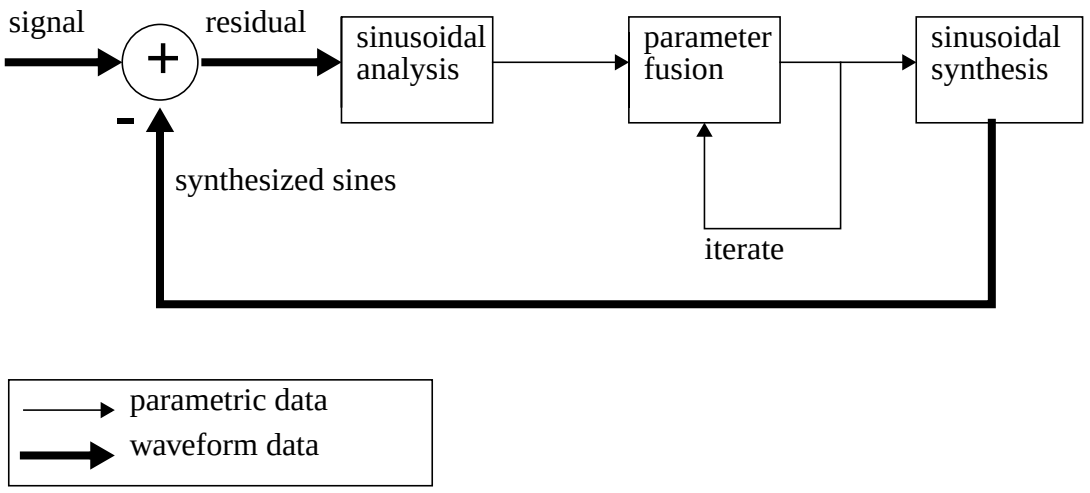

Figure 7: Block diagram of iterative parameter estimation algorithm.

$$x(t) = a_3(t)\sin\left(\frac{(\omega_2+\omega_1)t+\varphi_2+\varphi_1}{2}+\int_0^t \omega_3(u)\mathrm{d}u+\varphi_3(0)\right), \quad (27)$$

where the new amplitude $a_3(t)$, frequency $\omega_3(t)$ and initial phase $\varphi_3(0)$ are

$$a_3(t)= \sqrt{a_1^2+a_2^2+2a_1a_2\cos((\omega_2-\omega_1)t+\varphi_2-\varphi_1)}, \quad (28)$$

$$\omega_3(t)= \frac{\left[1+\tan^2\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\right]}{1+\tan^2\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\left[\frac{(a_2-a_1)}{(a_2+a_1)}\right]^2}\left(\frac{\omega_2-\omega_1}{2}\right)\frac{(a_2-a_1)}{(a_2+a_1)}, \quad (29)$$

$$\varphi_3(0) = \begin{cases} \operatorname{atan}\left(\tan\left(\frac{\varphi_2-\varphi_1}{2}\right)\frac{(a_2-a_1)}{(a_2+a_1)}\right)+\pi & \frac{\pi}{2}<\left(\frac{\varphi_2-\varphi_1}{2} \bmod 2\pi\right)<\frac{3\pi}{2} \\ \operatorname{atan}\left(\tan\left(\frac{\varphi_2-\varphi_1}{2}\right)\frac{(a_2-a_1)}{(a_2+a_1)}\right) & \text{otherwise} \end{cases}. \quad (30)$$

Neither the time-varying frequency of the Equation 27 nor the time-varying phase also derived in Appendix A can be directly utilized in our sinusoidal model, since our model assumes that amplitudes and frequencies are constant and phases linear inside one frame. However, in certain conditions we can approximate time varying amplitude and frequency with constants. The conditions are:

1. Time *t* is near zero. This means that the approximated values are valid only in a small time frame. The parameters of the sinusoidal model are updated from frame to frame, so this condition is fulfilled. The shorter the time, the better to approximation is.

2. The frequencies are close the other. When conditions 1 and 2 hold, term $(\omega_2-\omega_1)t$ in Equations 28 is 29 becomes negligible.

3. The amplitude envelope of the sum of the two sinusoids does not have a local maximum or minimum inside the time frame. This depends on the phases and frequencies of the original sinusoids. As it is later shown, this condition is fulfilled if $0 \le (\omega_2-\omega_1)S/F_s + (\varphi_2-\varphi_1+\pi/2) \bmod \pi \le \pi$, where S is the length of the frame in samples and F_s is the sampling frequency.

4. The ratio of the amplitudes is large. This happens in situations where the first sinusoid is obtained on the first analysis pass and the second one is the error remaining from the first one. If this condition is fulfilled, the term $\left[\frac{(a_2-a_1)}{(a_2+a_1)}\right]^2$ in Equation 29 is near unity.

If these conditions are fulfilled, the sinusoid with time-varying parameters can be approximated with a sinusoid with constant parameters:

$$= a_n \sin(\omega_n t + \varphi_n), \tag{31}$$

where constants a_n, ω_n, and φ_n are the parameters of the new sinusoid which replaces the old ones. The approximations are:

$$a_n = \sqrt{a_1^2 + a_2^2 + 2a_1 a_2 \cos(\varphi_1 - \varphi_2)}, \tag{32}$$

$$\omega_n = \frac{\omega_1 a_1 + \omega_2 a_2}{a_1 + a_2}, \tag{33}$$

$$\varphi_n = \begin{cases} \operatorname{atan}\left(\tan\left(\frac{\varphi_2 - \varphi_1}{2}\right)\frac{(a_2 - a_1)}{(a_2 + a_1)}\right) + \frac{\varphi_2 + \varphi_1}{2} + \pi & \frac{\pi}{2} < \left(\frac{\varphi_2 - \varphi_1}{2} \bmod 2\pi\right) < \frac{3\pi}{2} \\ \operatorname{atan}\left(\tan\left(\frac{\varphi_2 - \varphi_1}{2}\right)\frac{(a_2 - a_1)}{(a_2 + a_1)}\right) + \frac{\varphi_2 + \varphi_1}{2} & \text{otherwise} \end{cases}. \tag{34}$$

An example of the approximation is illustrated in Figure 8. It can be seen clearly that near zero the approximation is better.

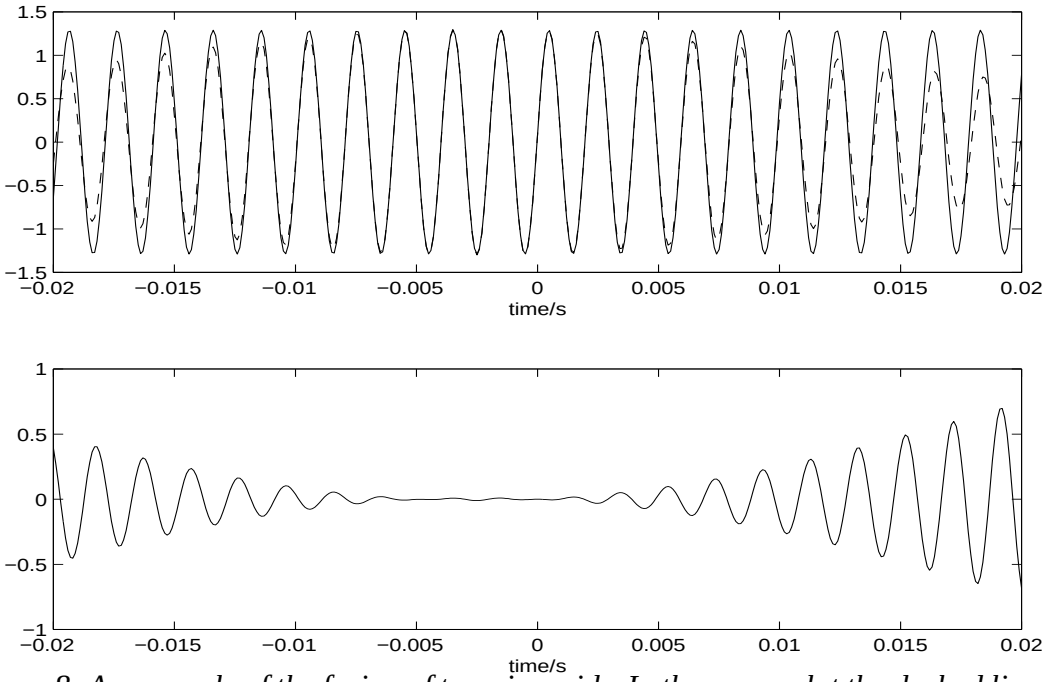

Figure 8: An example of the fusion of two sinusoids. In the upper plot the dashed line is a sum of two sinusoids, the frequencies of which are 500 and 520 Hz and the amplitudes 1 and 0.3. The solid line is the results of the approximation. In the lower plot is illustrated the error between the two original sinusoids and the approximated sinusoid.

In parameter fusion, we detect sinusoid pairs that fulfill all the conditions. The parameters of the new sinusoid are estimated, and then the old sinusoids are replaced with the new one. The parameters for the new sinusoid are calculated for each frame.

In synthesis, the parameters of the sinusoids are interpolated from frame to frame. Therefore, it is difficult to measure the validity of the approximation in a single time frame. The amplitudes are interpolated linearly, and if there is no local maxima or minima between the frames, the interpolation should work well.

When the formulas for the exact amplitude (Equation 28) and the approximated amplitude (Equation 32) are considered, we can roughly assume that the fusion of two sinusoids is valid if the sign of the derivative of the amplitude envelope in Equation 28 does not change. This is illustrated in Figure 9. The derivative of cosine is minus sine, which changes its sign at $\pm n\pi$, n=0,1,2... . Therefore, the validity of the approximation can be formulated by:

$$\mathrm{sgn}[\sin((\omega_2 - \omega_1)t + \varphi_2 - \varphi_1)] = \mathrm{sgn}[\sin(\varphi_2 - \varphi_1)], \forall t\left(0 \le t \le \frac{S}{F_s}\right) \qquad (35)$$

which is achieved if the argument $(\omega_2 - \omega_1)t + \varphi_2 - \varphi_1$ is inside the same interval $[-\pi/2 + n\pi, \pi/2 + n\pi]$ at the beginning and the end of the frame, n being any integer. By adding $\pi/2$ to the interval and to argument values at points $t = 0$ and $t = S/F_s$, the argument values at the beginning and the end of the frame become $(\omega_2 - \omega_1)S/F_s + \varphi_2 - \varphi_1 + \pi/2$ and $\varphi_2 - \varphi_1 + \pi/2$, the interval becomes $[n\pi, \pi + n\pi]$.

Now we can solve value for n:

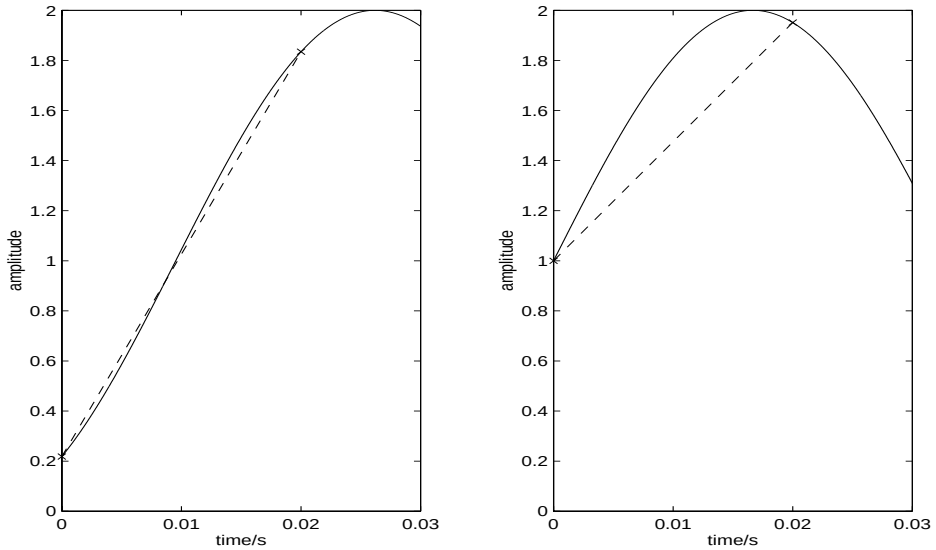

Figure 9: Linear approximating of the amplitude envelope of two combined sinusoids. The solid line is the original amplitude envelope and the dashed line is linear approximation. In left plot the sign of the slope of the amplitude envelope does not change so the approximation is valid. In right plot the sign changes so the approximation is not valid.

$$n = \left\lfloor \frac{\varphi_2 - \varphi_1}{\pi} + \frac{1}{2} \right\rfloor,$$

which gives the relation

$$n\pi \le (\omega_2 - \omega_1)\frac{S}{F_s} + \varphi_2 - \varphi_1 + \frac{\pi}{2} \le \pi + n\pi \tag{36}$$

from which we get by subtracting $n\pi$:

$$0 \le (\omega_2 - \omega_1)\frac{S}{F_s} + \varphi_2 - \varphi_1 + \frac{\pi}{2} - \pi\left\lfloor \frac{\varphi_2 - \varphi_1}{\pi} + \frac{1}{2} \right\rfloor \le \pi. \tag{37}$$

This is simplified to:

$$0 \le (\omega_2 - \omega_1)\frac{S}{F_s} + \left(\varphi_2 - \varphi_1 + \frac{\pi}{2}\right) \bmod \pi \le \pi. \tag{38}$$

For a large set of sinusoids, we can firstly filter out sinusoid pairs using a constant limit for the difference between the frequencies:

$$\left|(\omega_2 - \omega_1)\right|\frac{S}{F_s} \le \pi \Leftrightarrow \left|(\omega_2 - \omega_1)\right| \le \frac{\pi F_s}{S}. \tag{39}$$

This equation gives a threshold $\pi F_s / S$ for the difference between the frequencies, which can be used to filter out most of the sinusoid pairs using only the frequency information. Then, we can examine all the remaining pairs if they fulfill the rest conditions.

Since there are not reliable methods to judge numerically the accuracy of the sinusoidal analysis for unknown signal contents, the iterative algorithm was compared to other analysis methods using a generated test signal. The test results are presented in the Chapter 6.

3.7 Multiresolution approach

Since the frequency resolution of a short-time spectrum is linearly dependent on the analysis window length, a long window is needed to determine accurately the frequencies of the sinusoids. Also, a long analysis window is needed to detect low-frequency sinusoids, because the window length has to be 2-4 times the wavelength of the sinusoid, somewhat depending on the analysis method. A natural drawback of using a long window is a poor time resolution. Real sounds often exhibit rapid changes in their amplitudes and frequencies, so the assumption that sinusoids are stable inside one window does not hold. Obviously, a trade-off between the time and frequency resolution has to be made.

At low frequencies, we definitely need a long window, because the wavelengths of the sinusoids are long. At high frequencies, the wavelengths are short and components usually have rapid changes, thus a short window is needed there. At middle frequencies the situation is something between these two. Also, the frequencies that have been chosen to make up the scale of Western music are geometrically spaced. Considering all these facts together, a transform like constant-Q transform (CQT), would be a perfect choice. The frequency coefficients of CQT are geometrically spaced, and window length is inversely proportional to the frequency [Brown 1991]. The ratio between center frequency and frequency resolution is constant, thus the name constant-Q transform. The practical implementation of CQT just combines several coefficients of FFT [Brown&Puckette 1992], so compared to the use of several FFTs, we actually do not gain anything,

The bounded-Q transform (BQT) approximates a logarithmic frequency scale by using a different resolution and window for each octave so that the number of frequency coefficients is constant in each octave. This approach was used e.g. in the filterbank implementation of Levine [Levine 1998]. He used the sinusoidal analysis only for frequencies from 0 to 5 kHz and three octaves the frequency ranges of which were 0-1250, 1250-2500 and 2500-5000 Hz and window lengths 46, 23 and 11.5 ms, respectively.

In our system, sinusoidal analysis is used up to 10 kHz. The system was made flexible in such a way that the analysis bands do not need to be octaves, or they can be positioned at arbitrary positions. The lowest fundamental frequencies in our test musical samples were basses at about 30 Hz. A 46 ms analysis window is not long enough to detect reliably that low frequencies. Since basses have most of their energy at low frequencies, we found that it is enough to use a longer, 80 ms analysis window for frequencies from 0 to 200 Hz. A 46 ms window was used from 200 Hz up to 5 kHz. Above 5 kHz the characteristics of sounds are very different from the lower frequencies. The 46 ms analysis window was used also for these frequencies, but the parameters of the analysis algorithms were slightly different. To make further analysis easier, all the windows at different frequency bands were positioned at the same time. The frame rate was constant and therefore longer windows at low frequencies overlapped more than the shorter windows at high frequencies.

In harmonic sounds there is one property which disagrees against the use of the logarithmic frequency scale: the harmonic partials are spaced linearly. A sound with fundamental frequency 50 Hz has a period of 20 ms. It's 10^{th} harmonic partial has frequency 500 Hz and a wavelength of 2 ms. However, the distance between adjacent harmonics is always 50 Hz. Since the window length needed to discriminate two sinusoids does not only depend on the frequencies of the sinusoids, but the on difference of the frequencies too, we need a long analysis window also for higher harmonic partials of a low sound. In the case of polyphonic signals, the number of harmonic partials can be large at middle frequencies, so a long window is needed even though the wavelengths of the sinusoids are small.

Even though linear spacing of harmonic components disagrees against the use of multiresolution analysis, we found that it is still advantageous to use different window lengths for different frequency bands. Because the properties of sounds are not the same at different

frequency bands, the detection algorithms can be optimized individually for each band. The flexibility of our system made it possible to try different sinusoidal detection algorithms for different frequency bands, mainly F-test and cross-correlation method with different parameters.

4. Sinusoidal Continuation and Synthesis

As illustrated in Figure 2 in the Chapter 3, the last step in the estimation of the sinusoids is peak continuation analysis. In this chapter, the theory of two peak continuation algorithms and sinusoidal synthesis is presented. The performance of the algorithms is compared in Chapter 6, where experimental results are presented.

Once the meaningful sinusoidal peaks and their parameters have been estimated, the peaks are tracked together into interframe trajectories. At each frame, a peak continuation algorithm tries to connect the sinusoidal peak into the already existing trajectories at the previous frame, resulting into a smooth curve of frequencies and amplitudes. The continuation was tested with two algorithms: the traditional one which uses only the parameters of the sinusoids to obtain smooth trajectories and one original method which synthesizes the possible continuations inside certain deviation limits and compares them to the original signal. There is also other systems which use more advanced methods, for example the Hidden Markov Models [Depalle et. al 1993] to track the trajectories, but they were not tested.

4.1 Continuation based on the derivatives

The smoothness is obtained by using the derivatives of frequencies and amplitudes: for each pair of peaks a smoothness coefficient is calculated as a weighted sum of the first, second, etc. derivatives of the parameters. The algorithm assumes that the parameters are slowly-varying and that the trajectories do not cross each other.

Since human pitch perception is close to logarithmic over most the hearing range, and also fundamental frequencies produced by most musical instruments are logarithmically spaced, we take logarithm of the frequencies. Because peak continuation is done on the frame level, the differences between adjacent values are used as estimates of the derivatives. As a subtraction of logarithms is a logarithm of a division, the factor describing the smoothness of the frequencies becomes the logarithm of the ratio of the frequencies:

$$\log(\omega_{n-1}(i)) - \log(\omega_n(j)) = \log\left(\frac{\omega_{n-1}(i)}{\omega_n(j)}\right).$$

The perception of amplitude differences is also more logarithmic and the same procedure was used for amplitudes.

Since frequencies are derivatives of phases, the smoothness of phases is dependent on the frequencies too. The smoothness of phases is estimated using the interpolation coefficients α and β used in sinusoidal synthesis (Chapter 4.4). An absolute value is taken of all the factors, because negative deviation is as unwanted as positive deviation. If we use only the first derivatives, the smoothness coefficient between *i* and *j* at frames *n*-1 and *n* is

$$s_n(i, j) = w_f \left| \log\left(\frac{\omega_{n-1}(i)}{\omega_n(j)} \right) \right| + w_a \left| \log\left(\frac{a_{n-1}(i)}{a_n(j)} \right) \right| + w_\alpha |\alpha_n(i, j)| + w_\beta |\beta_n(i, j)|, \quad (40)$$

where $\omega_n(i)$ and $a_n(i)$ are the frequency and amplitude of i^{th} peak at frame *n*, $\alpha_n(i, j)$ and $\beta_n(i, j)$ are the phase interpolation coefficients between the peaks and w_a, w_f, w_α and w_β are the weights. It is advantageous to set maximum limits for the frequency and amplitude deviation so that the number of possible trajectory-peak pairs is limited.

In general, evaluating all the possible combinations between peaks in adjacent frames is not possible because the number of combinations is too large even though a maximum limit was set for the deviation of the parameters. We have used a greedy algorithm, which evaluates the smoothness for all single trajectory-peak pairs, and then chooses the continuation that is the smoothest, i.e. which has the smallest $s_n(i,j)$. Then, peaks *i* and *j* that had the smoothest continuation are removed, and algorithm is repeated for remaining peaks. For some generated test signals this algorithm may produce erroneous results, but for natural signals it seems to work quite well.

If a suitable continuation for some peak cannot be found, that means that the sound that produced that frequency component has faded out and the trajectory dies. If a peak in current frame does not represent a continuation to any of the already existing trajectories, that means a new component onsets and a new trajectory is born, as illustrated in Figure 10.

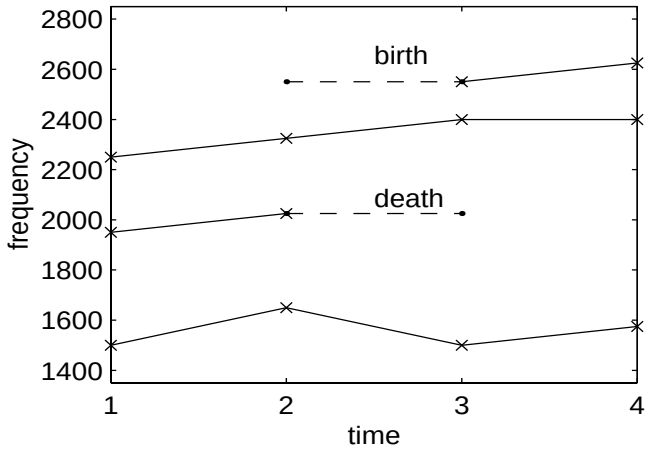

Figure 10: Continuation of sinusoidal trajectories. At time=2, no suitable continuation is found for the trajectory around 2000 Hz, so it dies. At time=3, a new trajectory is born around 2500 Hz.

After calculating the peak continuations, we have a set of sinusoidal trajectories with time-varying amplitudes, frequencies, and phases. Each trajectory has an onset and offset time, which define the time range in which the trajectory exists. To obtain a smooth transition to zero level, an extra peak is added into the beginning and the end of a trajectory. These extra peaks have the same frequencies as the next and previous peaks of the trajectory, but amplitudes are zero. This ensures that onset or offset does not produce modeling artefacts.

4.2 Continuation based on synthesis

The continuation based only on amplitude and frequency deviation and phase interpolation is usually not very robust method. This is because of the following reasons: To detect low-amplitude harmonic components in noise, the peak detection threshold has to be set to a quite low value. Naturally, this means that we also get many peaks that are caused by noise. Even though we set strict limits for amplitude and frequency deviation, some peaks caused by noise will be so close to each other, that they are connected into a sinusoidal trajectory.

Our solution to this problem is to synthesize all the possible continuations inside deviation limits, and to compare the result to the original signal. Sinusoidal synthesis is described in Chapter 4.4. If a synthesized sinusoid captures enough of the energy of the original signal, we assume that the sinusoid corresponds to a component that truly exists in the original signal. We use a greedy algorithm that always picks the continuation that minimizes the remaining energy. Then, the synthesized sinusoid is subtracted from the original signal, and the residual is compared to the synthesized continuation possibilities that are left. This repeated until none of the remaining synthesized continuations reduces the energy of the residual enough. The whole procedure is done in time domain.

This algorithm turned out to be significantly more robust than the continuation based on only deviation of amplitudes and phases. Of course, some continuated noise peaks still appear, because the continuations happen to match well with the original signal. However, the number of ‘noisy’ continuations is much smaller than with the simpler algorithm. The only drawback of continuation-by-synthesis is its computational load, which is huge compared to an algorithm that uses only the deviations of the parameters. Synthesis uses third order polynomial to interpolate the phases and linear interpolation for amplitudes, so synthesizing all the time-domain sinusoids is computationally expensive. The DFT of one synthesized sinusoid could be approximated for example using a series development, but doing the whole process in frequency domain would not help very much because anyway we have to calculate the remaining energy of each residual, which is computationally expensive for a large number of sinusoids in both time and frequency domains.

4.3 Trajectory filtering

We know from everyday experience, that the human auditory system has the property that a weak sound can be rendered inaudible in the presence of another loud sound. This effect is called masking [Moore 1997]: a sound is masked by another, masking sound. In other

words, masking can be defined as a process which raises the threshold of hearing: if a sound falls below the threshold, it is not audible. Masking occurs in both frequency and time domains, and they are called simultaneous and non-simultaneous masking, respectively. In the frequency-domain masking, the closer the simultaneous sounds are to each other in frequency, the stronger the masking effect. In time-domain, a loud sound can mask a quieter sound which occurs after, or even before the masking sound.

The masking effect is utilized in audio coding by removing the components that would be masked. This could be used in the sinusoidal modeling, too. In addition to that, the threshold of hearing can be used to judge which sinusoidal peaks or trajectories are caused by noise. As previously mentioned, not all the sinusoidal peaks are resulted from stable sinusoids. Since the number of sinusoidal peaks in polyphonic signals is very large, it is probable that some of the false peaks match well with the original signal and therefore become continuated into a trajectory. These false trajectories are usually very short, only a couple of frames long. If a sinusoid is short, it is possible that human auditory system is not fast enough to determine the pitch of the sinusoid, especially if the amplitude of the sinusoid is small and other signal components are present. Therefore, there is no need to model the sinusoid with the sinusoidal model, but it can be left to the residual to be modeled with the stochastic model. If a sinusoidal peak is clearly below the threshold of hearing, or masking threshold, it is probable that the component is not a result of sinusoid.

Scott Levine used a method which uses the average distance to the masking threshold and length of each sinusoid to determine if the sinusoid is kept or filtered out [Levine 1998]. The average signal-to-mask (SMR) ratio is calculated by comparing the amplitude of the sinusoid to the masking threshold which has computed in each frame. The sinusoid *i* is removed, if $\mathrm{SMR}(i) < 6 - 96 \cdot \mathrm{len}(i)$, where SMR(*i*) is the average signal-to-mask ratio of the sinusoid *i* in dB and len(*i*) is the length of the sinusoid in milliseconds. This implies that a short sinusoid requires a large SMR not to be filtered. The longer the sinusoid is, the lower the SMR can be and the sinusoid is still retained.

Our system computes the masking threshold in a way similar to that in MPEG model 2 [Colomes et al. 1995]. For each sinusoid, an excitation pattern is calculated in the frequency domain, which has a resolution of 1/25 Bark, which makes about 620 frequency bands between 0 and 22.5 kHz. Thus, the energy of a sinusoid is distributed along 620 frequency bands using a spreading function, which is triangular in Bark domain, but non-symmetric in frequency domain. The excitation patterns of all sinusoids are combined using the exponential law

$$e = \left(\sum_i e(i)^{\alpha} \right)^{1/\alpha},$$

where $e(i)$ is excitation of i^{th} component and α is between 1 and 2. We used value 1.5 for α. Our system does not utilize non-simultaneous masking.

4.4 Synthesis

Sinusoidal trajectories contain all the information needed for the reconstruction of the harmonic parts of input signals: amplitudes, frequencies and phases of each trajectory at each frame. To avoid discontinuities at frame boundaries, the amplitudes, frequencies and phases are interpolated from frame to frame. Amplitudes are linearly interpolated, so instantaneous amplitude of trajectory p_i at frame n is

$$a_{i,n}(m) = a_{i,n} + (a_{i,n+1} - a_{i,n})\frac{m}{S} \qquad m = 0, 1, \ldots, S-1 \tag{41}$$

where S is the frame length in samples.

Phase interpolation is more complicated, because instantaneous frequencies are derivatives of phases and four parameters (frequencies and phases at two adjacent frames) have to be taken into account. Smooth phase as a function of time is obtained by using a cubic polynomial interpolation function [McAulay&Quatieri 1986]:

$$\theta(t) = \zeta + \gamma t + \alpha t^2 + \beta t^3, \tag{42}$$

where $\theta(t)$ is the interpolated instantaneous phase at time t, and ζ, γ, and β are interpolation coefficients. Setting the instantaneous phase and frequency at points t=0 and t=S equal to the known frequencies and phases ω_n, ω_{n+1}, θ_n and θ_{n+1}, we obtain solution to the cubic polynomial function:

$$\zeta = \theta_n,$$

$$\gamma = \omega_n, \text{ and}$$

$$\begin{bmatrix} \alpha(M) \\ \beta(M) \end{bmatrix} = \begin{bmatrix} \frac{3}{S^2} & -\frac{1}{S} \\ -\frac{2}{S^3} & \frac{1}{S^2} \end{bmatrix} \begin{bmatrix} \theta_{n+1} - \theta_n - \omega_n S + 2\pi M \\ \theta_{n+1}\theta_n \end{bmatrix}, \tag{43}$$

for any integer M.

Maximally smooth phase, or phase which second derivative is minimized, is obtained using

$$M = round\left(\frac{1}{2\pi}\left[(\theta_n + \omega_n S - \theta_{n+1}) + (\omega_{n+1} - \omega_n)\frac{S}{2}\right]\right). \tag{44}$$

When instantaneous amplitudes and phases have been calculated for all the sinusoidal trajectories at each time instant in a frame, the reconstructed sines are obtained by summing up all the trajectories:

$$s(t) = \sum_{i=1}^{N} a_i(t)\cos(\theta_i). \quad (45)$$

The interpolation of the parameters works well if the assumptions of the model are valid: the harmonic components are slowly-varying and therefore almost stable in a single frame. Even though frequencies vary a little, quadratic interpolation seems to work very well. In the case of sharp attacks, linear interpolation of amplitude values does not fit the actual amplitudes of the harmonic components. This is because a relatively long window is needed to distinguish closely spaced frequencies and because only one value represents the behavior of amplitude of sinusoids inside the window, we can not get exact amplitudes. There has been some attempt to extract more information of the parameter variation inside a single window [Rodet 1997], such as amplitude and frequency modulation, but for real musical signals the methods are not robust enough in the presence of other interfering sounds.

Phaseless Reconstruction

Phase is not perceptually very important, so in audio coding applications we no need to transmit it. In the decoder a random initial phase can be generated for each sinusoidal trajectory, and then get rest phases as an integral of the frequency:

$$\theta_{n+1} = \theta_n + \omega_{n+1}S. \quad (46)$$

If the phaseless reconstruction is used, the synthesized signal is not phase aligned with the original signal any more. If we want to obtain the residual, we have to use also the phase information when removing the sinusoids from the original signal. After that, the phase information can be discarded.

5. Stochastic Modeling

When the synthesized sinusoids are subtracted from the original signal in time domain, we get a residual signal, which ideally contains non-harmonic components only. Analysis and synthesis of the stochastic signal components is significantly easier than that of the deterministic part. Because human monaural sound perception is not sensitive to phase, the only information needed to represent the residual is the time-varying spectral shape. In psychoacoustic experiments, it has been found out that the ear is not sensitive to variations of energy inside the Bark bands for stationary, noise-like signals. Between 0 and 20 kHz there are 25 Bark bands, or critical bands, which are not linearly spaced. Assuming that the residual is noise-like, it can be modeled by calculating the short-time energies within each Bark band.

5.1 Analysis

The stochastic analysis process is illustrated in Figure 11. The residual is segmented into frames, and STFT is taken in each frame. The power spectrum is obtained by taking the square of the magnitude of the STFT. Then, energy within each Bark band is calculated by integrating the power spectrum values over the Bark band.

We denote the residual signal by $r(n)$ and its STFT at frequency ω and time t by $R(\omega, t)$. The short-time power spectrum of $r(n)$ is $|R(\omega, t)|^2$. The Bark band z corresponding to frequency f in Hz is approximated by [Zwicker&Fastl 1999]:

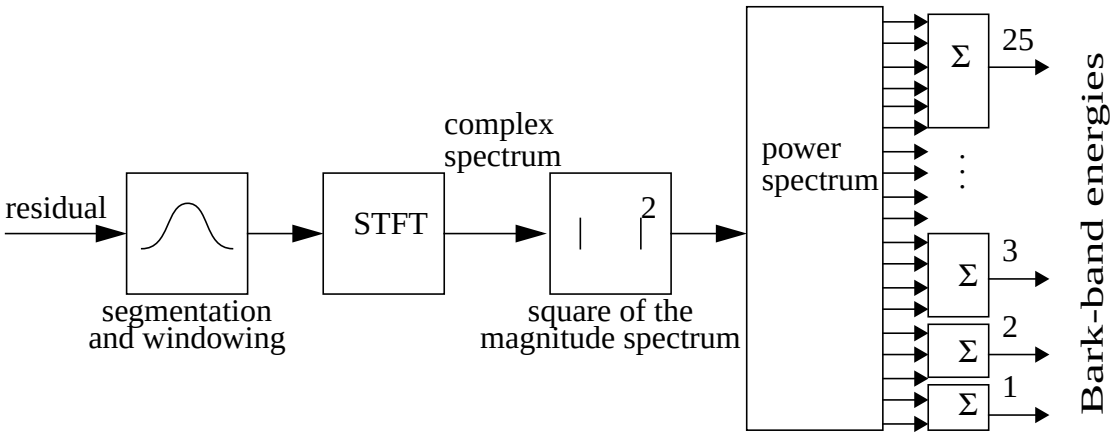

Figure 11: Block diagram of the stochastic analysis.

$$z(f) = \left\lfloor 13\,\mathrm{atan}(0.00076\,f) + 3.5\,\mathrm{atan}\left[\left(\frac{f}{7500}\right)^2\right] \right\rfloor . \tag{47}$$

The relation between the angular frequency and the frequency depends on the sampling rate F_s as follows

$$\omega = 2\pi \frac{f}{F_s} . \tag{48}$$

For each Bark band, we calculate the short-time energy inside the band. The short-time energy for band *b* is:

$$E(b, t) = \frac{1}{M} \sum_{z(\omega) = b} |R(\omega, t)|^2 , \tag{49}$$

where M is the length of the STFT. The short-time energies and the frame rate are all the information needed to represent the residual.

5.2 Synthesis

In stochastic synthesis, we construct a complex short-time spectrum using a piecewise uniform Bark band magnitude and random phase. The synthesis procedure is illustrated in Figure 12. The magnitude of spectrum is obtained by dividing each Bark band energy by corresponding bandwidth, and taking a square root:

$$|S(\omega, t)| = \sqrt{\frac{E(b, t)}{\beta_b}} , \tag{50}$$

where β_b is the bandwidth of band *b*, in samples of the synthesized spectrum $S(\omega, t)$. The division by β_b can be done also at analysis stage, so that we do not calculate the energy within each band but the mean of power spectrum coefficients within each band. To elimi-

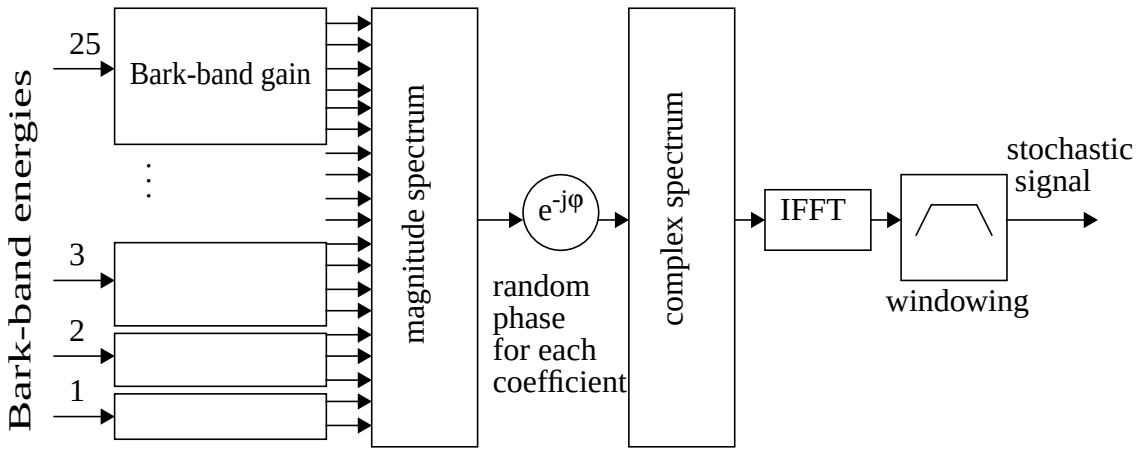

Figure 12: Block diagram of the stochastic synthesis.

nate sharp band boundaries, the spectrum can be slightly smoothed, but usually this is not necessary because time-domain windowing in the overlap-add phase causes smoothing in frequency domain.

The spectrum is made stochastic by creating a random vector for the phases. The random phase vector $\varphi(\omega)$ is uniformly distributed at the interval $[-\pi, \pi]$. The complex spectrum is the product of the magnitude spectrum and the random phases:

$$S(\omega, t) = |S(\omega, t)|e^{i\varphi(\omega)} \tag{51}$$

Stochastic signal is obtained by taking inverse STFT of each short-time complex spectrum. To prevent clicks at frame boundaries windowing and overlap-add is used. The window function is chosen to sum unity when the overlap adjacent frames is taken into account. The Bark-band energies of the residual of a particular music sample are illustrated in Figure 13. Drums dominate the residual signal: regular bass and snare drum hits can be recognized from the energies.

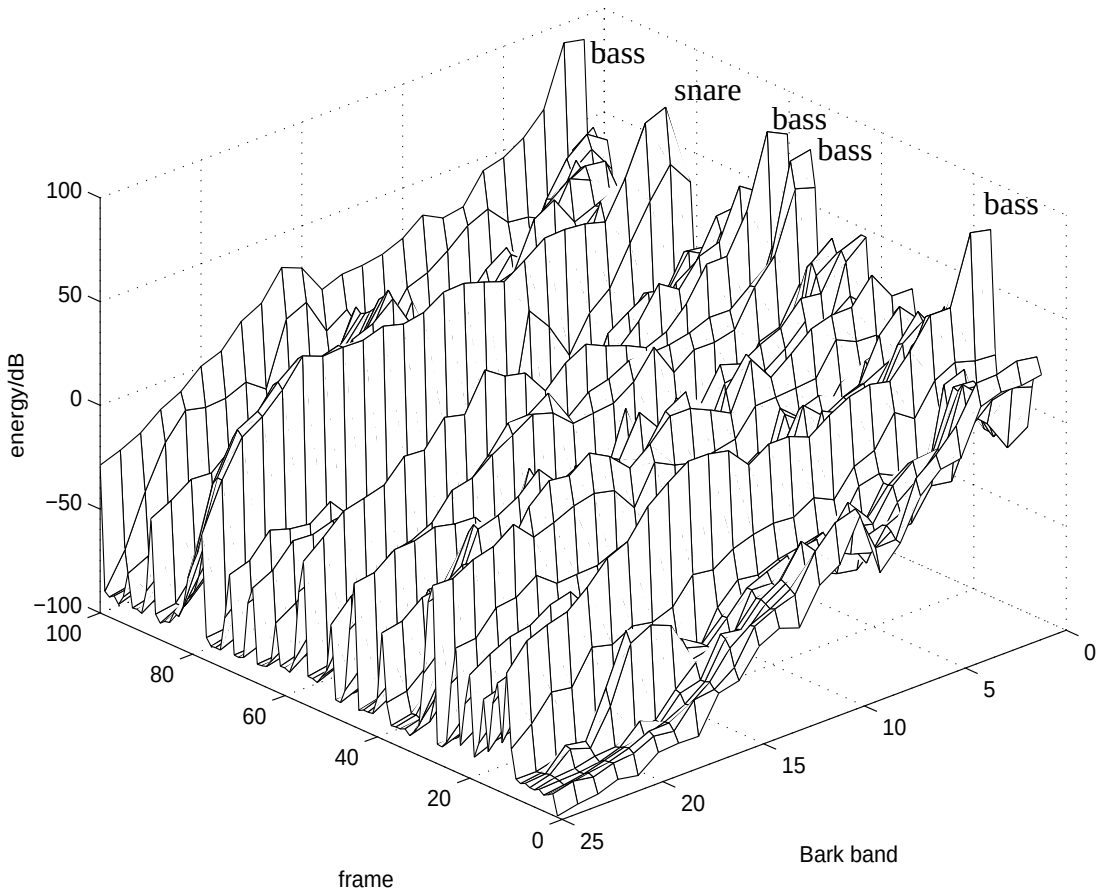

Figure 13: Bark band energies of the music sample "What a Friend We Have in Jesus". Bass and snare drum hits are marked on the figure.

Compared to the sinusoidal analysis and synthesis, the processing of the stochastic part is significantly simpler. Basically the only parameters of the stochastic analysis that can be adjusted are the window length and frame rate. Naturally, multiresolution analysis makes it possible to use several different windows. Some experiments were made to remove the possibly remaining harmonic components from the residual by non-linear filtering of the magnitude spectrum, but it seems that nothing can be gained with this method.

After the stochastic part and the sinusoids have been synthesized, they can be linearly added in time domain to obtain a complete resynthesized signal. In some systems both the signals are synthesized in the frequency domain: a random spectra is generated using the Bark band energies, and the sinusoids are added to the spectra. With this method the quadratic interpolation of the phases is not possible.

6. Experimental Results

In complex real-world signals, the density of sinusoidal components can be very high, and there are no obvious numerical ways to measure the performance of a sinusoids+noise analysis system. During the implementation of our system, visual and auditory evaluation was used by plotting the sinusoidal peaks and their parameters and obtained trajectories of each algorithm and by listening to the obtained sinusoids and residuals. This information is very difficult to present with numerical or even verbal means, since the difference between the analysis algorithms are almost inaudible. Therefore the performance of the analysis algorithms was studied by calculating some statistics from analysis and synthesis results obtained from a set of music samples and from a generated test signal.

Since the peak detection is a crucial part of the analysis system, a large part of the test was to compare the peak detection algorithms. Another crucial parameter of the system is the window length, which is always a compromise between the time and frequency resolutions. The effect of the different window length was studied during the implementation and is not included in this work, but the found optimal trade-offs are used. The same window length was used with all the algorithms.

6.1 Comparison of the peak detection algorithms with musical signals

Usually it is very difficult to estimate the performance of a peak detection algorithm in a single time frame. Therefore, a continuation algorithm was used to unite the peaks into sinusoidal trajectories, and the performance analysis was based on the trajectory data. Most of the false peaks that the estimation algorithm produce are discarded in the continuation phase. Since the false peaks can be removed after the peak detection, we are more interested in the undetected harmonic components, because it is much more difficult to detect the missing components in the latter phases of the sinusoidal analysis.

The performance of the two best peak detection methods, F-test and the cross-correlation method, were tested with musical signals, which were 10 to 20 second excerpts from five musical performances listed in Table 2. Three parameter sets were used with both methods: one set that was tuned optimal for musical signals by hand during the implementation and testing of the algorithms, second set that picked more peaks than the optimal one and third set which picked fewer peaks.

Table 2: Musical test signals

song title	artist	style	instruments
Blowing In The Wind	Bob Dylan	pop	male vocals, acoustic guitar, harmonica
Danda da Solidao	Marisa Monte	latin pop	female vocals, bass, accordion, percussion
Kova luu	Tuomari Nurmio	rock	male vocals, distorted electric guitar
The Four Seasons / Spring	Antonio Vivaldi	classical	symphony orchestra, mainly strings
What a Fried We Have in Jesus	Brentwood Jazz Quartet	jazz	piano, electric bass, drums, electric guitar, keyboards

The parameters of the sinusoids were obtained using the best methods found, quadratic interpolation for the frequencies and least-squares method for the amplitudes and phases. The continuation was based on the comparison of the synthesized sines. The frequencies of the sinusoidal trajectories found from “Blowing In The Wind” using the cross-correlation with normal parameters are illustrated in Figure 14. Once the sinusoidal trajectories were obtained with both algorithms and all parameter sets, the sinusoids were synthesized and residuals obtained by subtracting the synthesized sines from the original signal. The

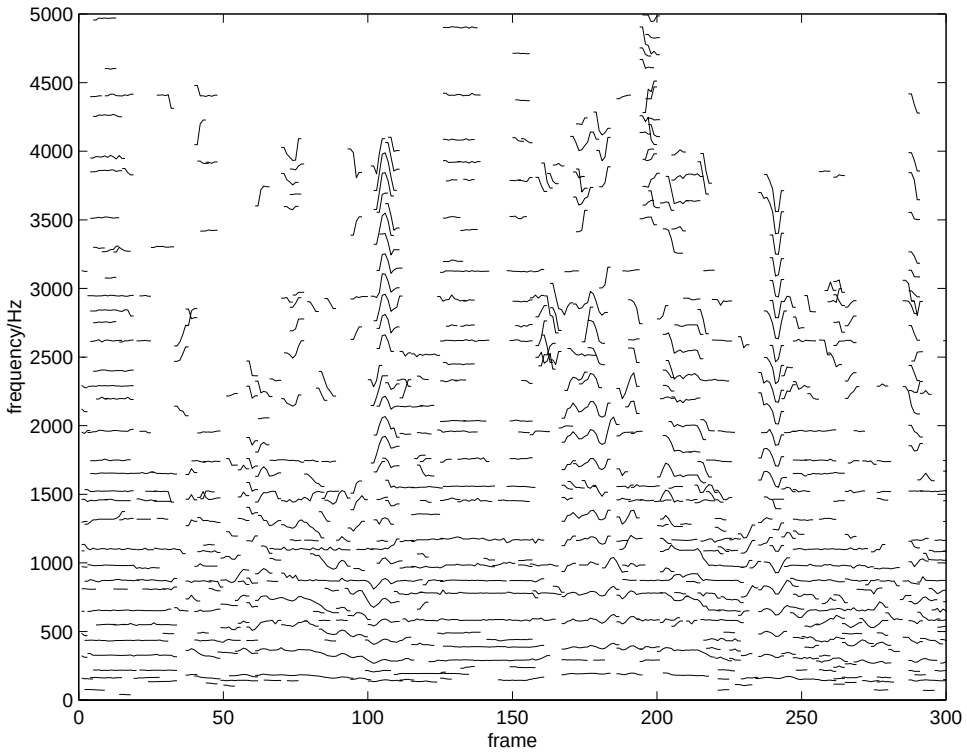

Figure 14: The frequencies of the sinusoids obtained from an excerpt of ‘Blowind In The Wind’. Above 5 kHz there is not many sinusoids so that part was left out of the plot.

signal to residual -ratios (SRR) were calculated as ratio of the energies of the signals. The SRRs measure how well the sines have been removed from the signal, but also overall characteristics of the signal: if there is a lot of non-harmonic components like drums in the signal, the SRR is low even though the sinusoidal analysis was perfect. In general, the more sinusoids has been detected, the better the SRR, no matter if the sinusoids are correct or erroneously modeled noise. Therefore, the SRRs alone do not measure the quality of the sinusoidal analysis.

The quality of the analysis was also examined by comparing the original and synthesized residuals. The residuals were synthesized by calculating the short-time energies within each Bark band and then performing the stochastic synthesis as usual. If all sinusoids have been removed from a residual, its amplitude spectrum should be smooth, and therefore the amplitude spectra of the original and synthesized residual should be close to each other. The mean square error between the short-time amplitude spectra of the original and synthesized residual is calculated within each Bark band in each frame. The error was averaged over time and all 25 Bark bands. The spectrum of the synthesized residual is smooth, so the resulting error measures the irregularity of the amplitude spectra. Therefore, the error between the synthesized and original residual is an estimate of the amount of harmonic components left in the residual, trying to discard all noise-like components.

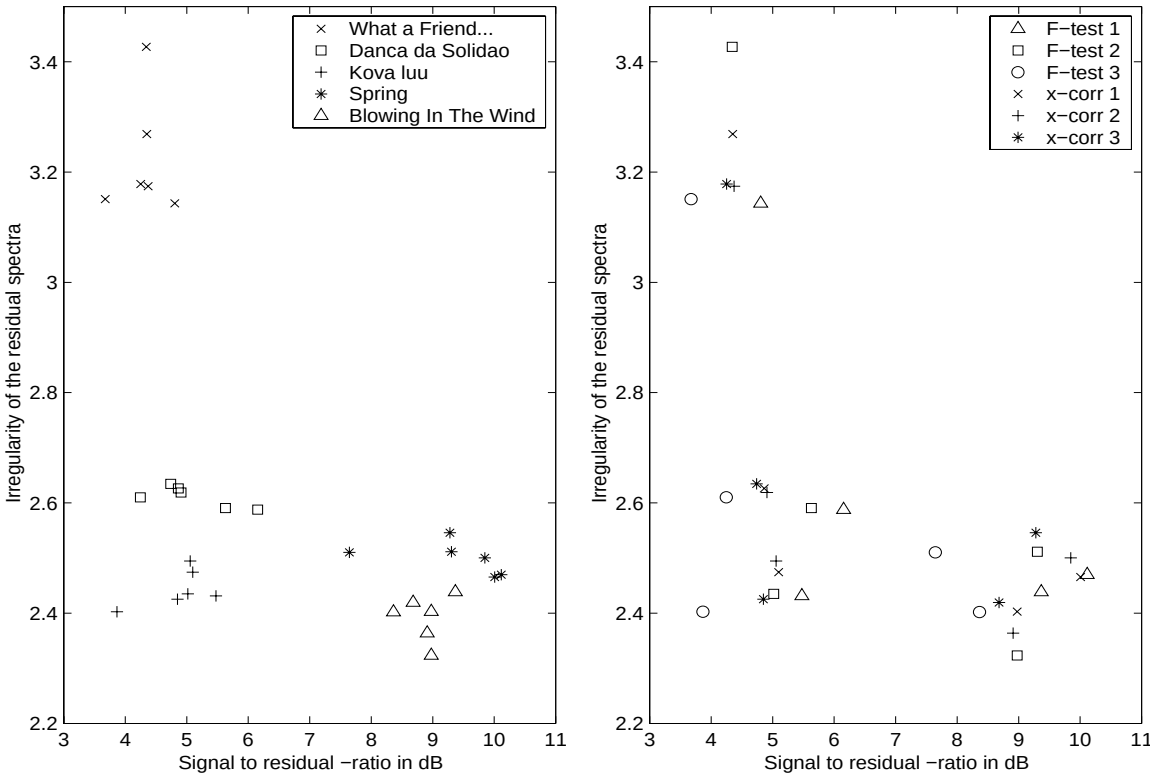

Figure 15: The performance of sinusoidal analysis with musical signals. In the left plot, each signal has it's own symbol and different values are obtained with different algorithms and parameters. In the right plot, different algorithms are marked with different symbols.

The obtained spectral irregularities and SRRs are illustrated in Figure 15. As can be seen, the differences between different music samples are much larger than the differences between the analysis algorithms and different parameters sets. We tried to compare different algorithms and parameters by removing the mean values obtained for each signal. Naturally, parameter sets that produced extracted the biggest number of sinusoids resulted in slightly better SRRs and lower spectral irregularities. Still the differences between different algorithms were quite small. The F-test produced more varying results than the cross-correlation method.

The results were also studied by listening to the synthesized signals and residuals, and by scanning through the obtained frequency and amplitude curves of the sinusoidal trajectories. The differences between the algorithms were almost inaudible, even though the listening tests gave the impression that the cross-correlation method would be slightly better than F-test, especially in signal sections that contained fast frequency changes or vibrato. This was confirmed by examining the frequency curves.

6.2 Comparison of the sinusoidal analysis algorithms using a generated test signal

As mentioned in the beginning of this section, there are no good numerical criteria in measuring the goodness of sinusoidal analysis algorithms for complex real-world signals. We tried to overcome this problem by generating a test signal that comprised only sinusoids. The test signal introduces phenomena usually encountered in musical signals: different kinds of changes in amplitude and frequency, harmonic sounds composed of sinusoids that overlap with each other, colliding sinusoids etc. The signal was divided into ten sections, which are described in Table 3.

The generated test signal was analyzed in three different noise conditions: The levels of additive white noise were no noise, low -14 dB noise and loud +6 dB noise. The reference level 0 dB is a single sinusoid with unity amplitude, and the noise levels are over the whole 0-22 kHz frequency range.

For each step of the sinusoidal analysis we have 2-3 possible algorithms and the performance of each step is affected by the preceding steps. For example, if the frequency of a detected peak is wrong, it is impossible to obtain correct amplitude and phase. Moreover, peaks with wrong parameters are easily continuated wrong. Even the parameter estimation and continuation of correct peaks is affected by the false peaks. Therefore, it would be ideal to test each algorithm with all possible preceding algorithm combinations. However, the number of possible combinations is 48 and therefore only a limited set of algorithms was used.

Eight different sinusoidal analysis systems were compiled by selecting among the alternatives. With these sets, we can compare each algorithm to other possible algorithms of each analysis stage. The algorithm sets are described in Table 4.

Table 3: Description of the generated test signal

Section	Signal description. Amplitude is unity (0 dB) unless otherwise stated.
1	Stable sinusoids at different frequencies, one sinusoid at a time.
2	Frequency sweep of a sinusoid from 20 Hz to 10 kHz. The speed of the sweep was constant on an exponential frequency scale.
3	Single sinusoid the amplitude of which fades exponentially from 0 dB to -40 dB.
4	Mix of sinusoids with different amplitude and frequency modulations (tremolo and vibrato). The modulation frequencies vary from 0 to 20 Hz, amplitude deviaton from 0 to 1 and frequency deviation from 0 to 1.5 semitones (0 to 9.05% of the center frequency).
5	Frequency crossing of two sinusoids at several different frequencies.
6	Stable harmonic sounds at different fundamental frequencies. All the sounds had 10 first harmonic partials, with unity amplitudes.
7	A frequency sweep of a harmonic sound, ten harmonic partials.
8	Vibrato of a harmonic sound. The modulation frequency and depth of the vibrato were time-varying like in section 4.
9	Different kind of sharp attacks of a Shephard tone. The harmonics were at frequencies 100, 200, 400,..., 3200, 6400 Hz.
10	Frequency sweep of a harmonic sound, mixed with a constant harmonic sound.

Algorithm set 2 corresponds to the standard McAulay-Quatieri algorithm. It picks peaks directly from the amplitude spectrum. This methods does not taken into account the overall level of the spectrum, therefore the user has to define the threshold for the detection. Since the threshold has to be adjusted to the test signal, algorithm sets 1 and 2 have a slight advantage to the other peak detection algorithms, which are signal-independent. The parameters of the other algorithms were tuned for music signals and then fixed.

Table 4: Analysis algorithm sets.

set	peak detection	peak interpolation	parameter estimation	peak continuation
1	fixed [1]	none	STFT [6]	param. derivatives [8]
2	fixed	quadratic [4]	STFT	param. derivatives
3	cross-corr.[2]	quadratic	STFT	param. derivatives
4	cross-corr.	quadratic	LSQ [7]	param. derivatives
5	cross-corr.	quadratic	LSQ	synthesis [9]
6	cross-corr.	signal derivatives [5]	LSQ	synthesis
7	F-test [3]	quadratic	LSQ	synthesis
8	The same algorithms as in set 5, with one iterative analysis pass (Chapter 3.6).			

1) Local maxima above a fixed threshold of the amplitude spectrum
2) Cross-correlation (Chapter 3.1)
3) F-test (Chapter 3.2)
4) Quadratic interpolation (Chapter 3.3)
5) Derivative interpolation (Chapter 3.4)
6) STFT coefficients directly
7) Least-squares estimation, amplitudes and phases only (Chapter 3.5)
8) Parameter derivatives (Chapter 4.1)
9) Compare synthesized continuations to the original signal (Chapter 4.2)

All the algorithms apply the same 46 millisecond analysis window for all the frequencies. The test signal contains frequencies the wavelengths of which are about the length of the analysis window. To detect a sinusoidal component, the window length has to be 2-4 times the period of the sinusoid, depending on the analysis method. Therefore, with the multiresolution analysis better performance would have been obtained especially in the low frequencies. However, the same analysis window was used for all the frequencies for simplicity.

Since the test signal was generated using sinusoids, the 'correct' frequencies, amplitudes and phases of each sinusoid are known. The parameters of the sinusoids obtained from the analysis were compared to those of the correct ones. Several statistics were calculated after the analysis. These include the percentages of sinusoids not found, extra peaks found, breaks in sinusoids, erroneous continuations and mean frequency, amplitude, and phase errors. The statistics were averaged over the three noise levels and combined into four tables which are presented in Appendix B. The most important information of each table was extracted and collected into Tables 5 to 8.

The percentage of missed plus extra peaks per the number of sinusoids in the original signal is presented in Table 5. Algorithm set 7, which uses F-test in peak detection has clearly more errors than the others in some sections. Most of the errors were caused by F-test's inability to detect sinusoids at extremely low frequencies when the window length is small. F-test was also clearly worse in sections 4 and 5 which contained vibratos, tremolos and colliding sinusoids. F-test was better than cross-correlation in sections 6 and 10 which contained harmonic tones. The amplitude-spectrum thresholding used in algorithm sets 1 and 2 worked surprisingly well. It was worse than average only in sections 3 and 9, which is natural since these sections contained sounds which amplitudes were different from the overall level.

The mean frequency errors are presented in Table 6. Algorithm sets 1 and 2 are otherwise similar but set 2 uses quadratic interpolation. The mean frequency error indicates clearly that the quadratic interpolation improves the analysis. The quadratic interpolation is compared to the derivative-interpolation in sets 5 and 6. The performance is almost similar, so we cannot say if some of the interpolation methods is better.

The errors in amplitude and phase estimation are measured by calculating the distance to the correct points in imaginary space. The mean distances to the correct points are presented in Table 7. By examining the performance of algorithm sets 3 and 4 we can see the difference of simpler method to the LSQ estimation. In the first section the simpler method is clearly better, which is explained by the fact that LSQ makes large error in low frequencies. The simpler method is also better in section 5, which is surprising because LSQ should be especially good in the case of closely spaced sinusoids. In general, the LSQ is still better than the simpler method.

Table 5: Peak detection: percentage of missed and extra peaks

	signal section									
algorithm set	1	2	3	4	5	6	7	8	9	10
2 (fixed)	0	6	77	4	18	23	10	0	40	17
5 (cross-corr.)	0	4	28	1	23	64	64	21	31	74
7 (F-test)	25	49	25	22	53	47	82	32	33	63
8 (iterative)	0	5	28	1	18	62	57	10	31	67

Table 6: Peak interpolation: average frequency error / Hz

	signal section									
algorithm set	1	2	3	4	5	6	7	8	9	10
1 (no interpolation)	1.4	3.0	2.4	4.5	2.1	4.8	3.6	6.4	1.7	3.3
2 (quadratic interp.)	0.3	1.9	0.3	3.9	1.2	3.9	2.8	6.2	0.5	2.7
5 (quadratic interp.)	0.5	1.3	0.8	3.6	1.5	0.3	2.4	5.1	0.7	1.8
6 (derivative interp.)	0.4	1.4	0.7	3.8	1.2	0.3	2.5	5.2	0.9	1.8
8 (iterative)	0.5	1.3	0.8	3.6	1.6	0.4	2.3	5.4	0.7	1.8

Table 7: Parameter estimation: average amplitude and phase errors (distance to correct point in imaginary space)

	signal section									
algorithm set	1	2	3	4	5	6	7	8	9	10
3 (spectrum coeff.)	0.2	1.1	0.2	0.7	1.1	0.2	1.9	0.3	0.4	1.7
4 (LSQ)	0.7	0.6	0.2	0.4	1.3	0.2	0.7	0.2	0.3	0.9
8 (iterative)	0.7	0.7	0.2	0.4	1.7	0.3	0.8	0.3	0.3	1.0

Table 8: Peak continuation: percentage of false continuations and breaks in trajectories

	signal section									
algorithm set	1	2	3	4	5	6	7	8	9	10
2 (param. derivatives)	0	4	0	8	7	9	10	3	30	17
4 (param. derivatives)	0	2	2	1	10	0	2	4	30	2
5 (synthesis)	0	2	2	2	11	0	4	8	30	4
6 (synthesis)	0	0	2	2	11	0	4	8	30	4
7 (synthesis)	0	1	2	1	2	13	1	6	30	6
8 (iterative, synthesis)	0	1	2	2	9	0	5	8	30	7

The percentage of false continuations or breaks in the sinusoidal trajectories is presented in Table 8. Sets 5-7 use the continuation by synthesis, the others use derivatives. It is notable how much the performance varies inside the same continuation algorithm but with different peak detection and parameter estimation. Especially the amplitude spectrum detection with the simple parameter estimation causes clearly errors in the continuation stage. The differences in the continuations with and without synthesis are quite small, and it cannot

be said if either of the methods is better. However, in the case of crossing partials, where the continuation based on the parameter derivatives often makes errors, the continuation by synthesis is more likely to make correct continuations.

Algorithm set 8 is a bit different than the others, since it uses iterative analysis of the residual once. In peak detection, it performed slightly better than the non-iterative version. When it comes to mean frequency, amplitude, and phase errors, the performance is almost similar. Also the continuation errors are comparable to the non-iterative algorithm sets. This is natural since most of the continuations are done in the first iteration. On the second pass, we just extract peaks that have not been found, or improve the parameters found in the first iteration, so it is not likely that already made continuations would be changed. These statistics considered, the iterative analysis is not better than non-iterative algorithms. However, if the number of additional components is increased, the iterative algorithm can produce better results than the other algorithms in all the sections [Virtanen 2001].

6.3 Computational efficiency considerations

Considering the whole analysis/synthesis process of the sinusoids+noise model, sinusoidal analysis is clearly the most time-consuming part, taking more than 50% of the overall processing time. In the Figure 17, the analysis and synthesis times of the algorithm set 5 are illustrated. It should be noted that sinusoidal analysis and synthesis times depend very much on the signal: if no sinusoids are found, analysis and synthesis are be very fast, whereas in the case of a rich harmonic sound they take a longer time. The times in Figure 16 are obtained using the generated test signal. The complexity of the stochastic analysis and synthesis is signal-independent, since they are simply based on the calculation of energy at certain frequency bands.

Naturally, the sinusoidal analysis time depends on the algorithms used, which is illustrated in Figure 17. With the three first algorithm sets, the analysis time is only about 4 times the real time when implemented in Matlab. These sets use the simplest peak detection algorithms, amplitude spectrum thresholding and the cross-correlation method, and the param-

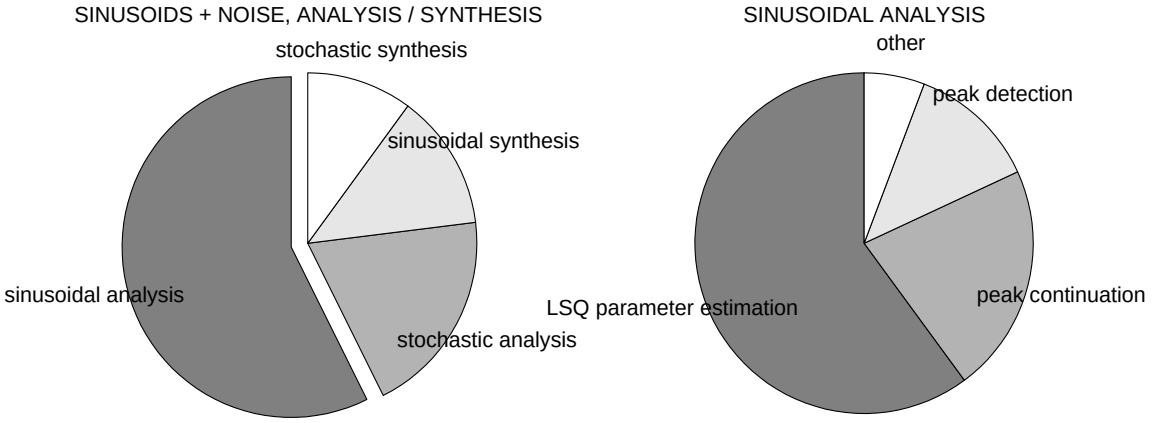

Figure 16: Percentage of times used in the sinusoids+noise analysis/synthesis process and details of the sinusoidal analysis for algorithm set 5.

eters are obtained directly from the spectrum. Therefore, the sets require efficiently implemented only one FFT per analysis frame. With these algorithm sets the sinusoidal analysis time is about the same as the sinusoidal synthesis and stochastic analysis and synthesis times.

As we switch to the more sophisticated LSQ parameter estimation (set 4), the analysis time is multiplied. This is natural since the LSQ requires one inverse of a large matrix per time frame. When synthesis is used in the peak continuation (set 5), the analysis time becomes even longer. The derivative-interpolation used in set 6 takes a bit longer time than quadratic interpolation used in sets 2-5. F-test, which is used in set 7, requires several FFTs and several other operations that are computationally expensive, and is therefore clearly slower than cross-correlation method used in sets 2-6. The iterative algorithm set 8 analyses the signal twice, therefore requiring about twice longer analysis time. Its analysis time includes the synthesis and subtraction of sinusoids after the first analysis pass.

The algorithms were implemented using the Matlab programming environment. Naturally, we tried to use fast matrix operations whenever possible. Usually, most of the computation time is spend in several computationally expensive FFTs, mean square errors or matrix inverses. Since the loop operations in the Matlab are very slow, at least the greedy continuation algorithm could be speeded up using some other programming language.

6.4 Comparison to other sinusoids+noise systems

Even though the number of different sinusoids+noise systems is large, there are not so many that are freely available. Our system was compared to two other systems by listening to the synthesized signals. Our system uses several similar algorithms that Scott Levine's

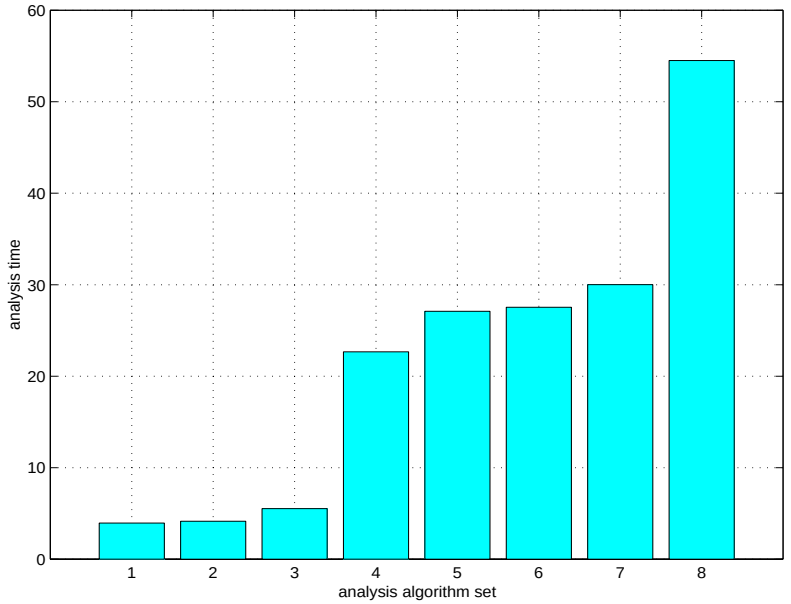

Figure 17: Comparison of sinusoidal analysis times with different analysis algorithm sets. Analysis time is scaled to the length of the original signal so that analysis time=20 means that the analysis takes 20 times the length of the signal.

system uses. Therefore, it was natural to compare our system with his one. A software called SNDAN includes an implementation of the standard McAulay-Quatieri algorithm and it is freely available. This was chosen as the other system.

Levine's system includes the transient model which is not used in our system. Therefore, the sound qualities are not directly comparable. Since we did not have access to Levine's system, we tested our system with two musical signals that were available on his web page.

The perceptual difference between the systems is surprisingly small, considering that our system does not include the transient model. The difference is audible, but not as large as one might expect. In both systems, the synthesized noise sounds somewhat similar, which is natural since both systems use Bark bands in the stochastic model. The overall perceptual quality of Levine's system is better, but it is very difficult to say what is the effect of the transient model.

SNDAN uses much larger frame rate than our system, and its peak detection threshold has to be set by hand. If the threshold is low, the system detects a huge amount of sinusoids and produces a bit phase vocoder-like sound; it represents all the components in the input signal, even drums, with sinusoids. If the threshold is high, only the harmonic components are represented with sinusoids, which is the desired situation. Then the sound quality depends on the characteristics of the signal. If there is a lot of dynamic changes in the signal, the fixed threshold does not work: in quiet parts, all the components fall below the threshold and no peaks are detected.

In some cases, the quality of the synthesized signals was comparable to our system, but in most cases it was clearly worse. In some cases, the fast frame rate caused an annoying audible effect. In SNDAN, the sinusoidal trajectories could not cross each other, even though some implementations of the McAulay-Quatieri algorithm allow that.

6.5 Selected "lightweight" and "quality" algorithm combinations

Our sinusoidal analysis system is based on several ideas taken from other sinusoidal analysis systems, modified with a couple of our own ideas. Our main emphasis was on the quality of the resulting sinusoids from computational auditory scene analysis point of view. In some algorithms, we had to make some minor compromises to keep the computational complexity in practical limits.

During the implementation and testing of the algorithms it became clear that in the analysis process, none of the algorithm combinations is the ultimate answer, for none of the algorithms performs well for all signals. This can be clearly seen in the results presented in Chapter 6. Therefore, the system was built so that different algorithms can be used, depending on the application. Two default algorithm sets were chosen to be used in situations where the user does not want to specify the algorithms himself: a 'lightweight' ver-

sion which is fast but still produces applicable results, and a ‘quality’ version, where the quality of the analysis is priority, but the analysis time is still tolerable. The chosen combinations use algorithms sets corresponds to algorithms sets 3 and 5 in Table 4.

Both systems use cross-correlation method to detect peaks and quadratic interpolation in peak interpolation. In the lightweight version, a single 46 millisecond window is used for all the frequencies. In the quality version, multiresolution analysis is used by using three frequency bands: 20-200 Hz, 200-5000 Hz, and 5-10 kHz and window length 86, 46, and 46 ms, respectively. Two highest bands use the same window length, but with slightly different parameters. The characteristics of the sounds are different in the highest band, and it was found advantageous to use different analysis parameters there.

In the lightweight version, the parameters are obtained directly from the interpolated spectra, and the obtained peaks are continuated using the derivatives of the parameters. In the quality version, the parameters are estimated using the LSQ method for amplitudes and phases, and continuation by synthesis is used to continuate the peaks. In both versions, a masking curve is calculated and erroneous trajectories are filtered out, based on the masking threshold.

Both versions are causal. The algorithmic delay is caused mainly by the long analysis windows and trajectory filtering. Practically, the longest trajectories that can be removed are length about 65 milliseconds in length. The longest analysis window used is 86 milliseconds, so the algorithmic delay is less than 100 milliseconds. Therefore, both versions can be implemented in real-time if enough computation resources are available.

If especially good quality is desired, and the number and length of input signals is small, iterative analysis can be used to obtain better results. Again, the algorithms used should depend on the application: if we are interested in the noise part and want to remove all harmonic components, we do not need the parameters of the sinusoids, and therefore the parameter fusion is not needed.

7. Application to Sound Separation and Manipulation

In this chapter we apply the implemented sinusoids+noise model as a middle-level representation for sound separation. The chapter discusses mostly sound separation using a perceptual distance between the trajectories, which has been originally presented in [Virtanen&Klapuri 2000]. Also, a more reliable sound separation method is shortly described which is based on a multi-pitch estimation model, originally presented in [Klapuri et al. 2000].

Separation of mixed sounds has several applications in the analysis, editing and manipulation of audio signals. These include e.g. structured audio coding, automatic transcription of music, audio enhancement, and computational auditory scene analysis. Until now, main part of the research in sound separation has taken place in the area of computational auditory scene analysis.

The sinusoids+noise model allows the manipulation of the separated sounds in parametric domain. The pitch and time scale of the signals can be modified without any change in the quality of the synthesized sound. The theory behind these modifications is shortly described in the end of this chapter.

7.1 Sound separation

When two sounds overlap in time and frequency, separating them is difficult and there is no general method to resolve the component sounds. However, if we can make some assumptions of the mixed sounds, we can synthesize sounds that are perceptually close to the original before mixing. Our assumption is that the underlying sounds are harmonic, and they have different fundamental frequencies. Using the sinusoidal model we can decompose an input signal into spectral components, assign them to sound sources using a set of perceptual association cues, and then synthesize the sounds separately.

Calculations proceed as follows. First, the system uses sinusoidal modeling to represent signals with sinusoidal trajectories. Second, some breaks caused by amplitude modulation, transients or noise in resulting sinusoids are removed by interpolating trajectories. Third, the system estimates the perceptual closeness of the trajectories by calculating the difference of scaled amplitudes and frequencies and the harmonic concordance of the trajectories. Then the trajectories are classified into sound sources. The system can determine

which of the trajectories are result of colliding harmonics, and then split these trajectories in two. Finally, after the trajectories have been classified and split, the system is able to synthesize the two sounds separately.

The classification part itself is currently the most undeveloped part of the system. In simulations, the classifier assumes two sound sources and uses their different onset times to initialize both classes. Thus the onset difference of the sounds had to be at least 100 ms. This constraint could be removed by calculating the perceptual distances between all the sinusoids and then classifying them with a generic clustering algorithms.

The methods themselves can be used also in more complex tasks, and for simultaneously onsetting sounds. As long as only sinusoidal modeling is used, it is difficult to obtain good results for a large number of mixed sounds, because some of their harmonic partials are likely to be undetected.

7.2 Modifications to the standard sinusoidal model

All the trajectories that result from the sinusoidal modeling are not usually representing entire partials of the sound. The most common estimation errors are breaks in trajectories. They can be caused by transients or noise occurring at the same time, or the amplitude of the harmonic itself is so low that the trajectory can not be estimated. This happens usually for signals with strong amplitude modulation, where the amplitude actually can go to zero. This phenomena can be easily seen in higher harmonics of violin of Figure 18.

If time difference between two frequency components is small and their frequencies and amplitudes are close to each other, human auditory system connects the sounds. Our system tries to model this by connecting trajectories which are close to each other. The breaks between trajectories are interpolated. The interpolation also increases the robustness of the system, as one harmonic is represented with one long trajectory instead of many short ones.

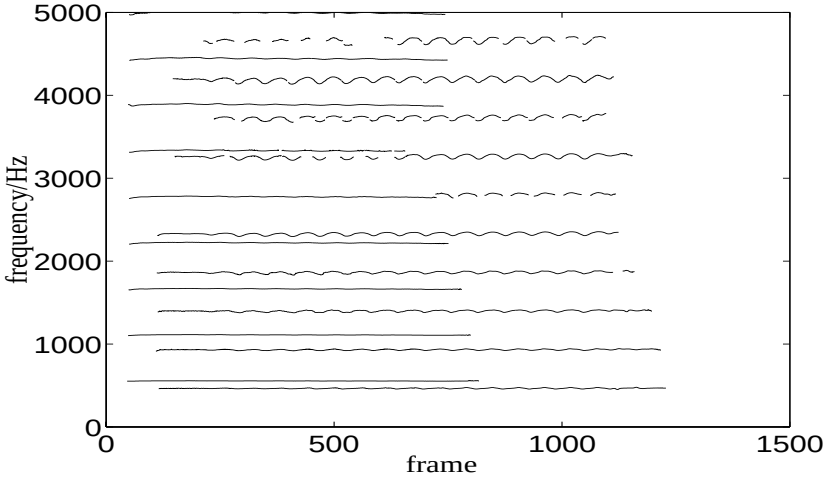

Figure 18: Sinusoidal trajectories of a signal consisting of oboe and violin sounds starting at times 100 and 300 ms.

The trajectories to be connected are detected by comparing the onset and offset times, frequencies and amplitudes near the breaks and finding those trajectories which are most probable belonging to the same partial. Then the breaks are removed by interpolating the frequencies and amplitudes over the break. In our system, we are using linear interpolation since it seems to work well enough for separation purposes. Not every break is interpolated, because allowing too long breaks or big amplitude/frequency differences causes wrong trajectories to be connected. In this example, the total number of trajectories reduced from 87 to 48.

7.3 Measure of perceptual distance

In his work, Bregman [1990] lists the following association cues in human auditory organization:

1. Spectral proximity (closeness in time or frequency)
2. Harmonic concordance
3. Synchronous changes of the components: a) common onset, b) common offset, c) common amplitude modulation, d) common frequency modulation, e) equidirectional movement in spectrum
4. Spatial proximity.

In this study, we focus on the synchronous changes of the components, together with the harmonic concordance, which is taken into account to some extent, too.

Measuring the amplitude and frequency changes

When the measurement of common amplitude and frequency modulation was studied, we found out that in some cases, modulation can be expressed with two quantities, modulation frequency and index. However, to present amplitude or frequency modulation only with two quantities is usually not enough. Because modulation usually varies in time domain, we would need several measurements to cover the changes within time. Also, the changes of the overall long-time intensity of the sound sometimes makes it hard to measure the modulation characteristics of the sound.

Different harmonic partials have a wide range of amplitudes values and sometimes their long-time progress is not similar. However, by scaling the amplitude of each partial by its average, the resulting curves are quite close to each other. In the case of frequencies this method is even more accurate, because frequencies do not change so much over time as amplitudes, as illustrated in Figure 19. The mean square error between these scaled frequencies measures the frequency distance between the sinusoidal trajectories:

$$d_f(i, j) = \frac{1}{t_2 - t_1 + 1} \sum_{t = t_1}^{t_2} \left(\frac{f_i(t)}{f_i} - \frac{f_j(t)}{f_j} \right)^2 , \tag{52}$$

where $f_i(t)$ is frequency of trajectory p_i at time t. Times t_1 and t_2 are chosen so that both trajectories p_i and p_j exists at times $t_1<t<t_2$. Scaling coefficients f_i and f_j are the average frequencies of trajectories p_i and p_j, calculated over times t_1 and t_2.

The same principle is used to obtain the perceptual distance caused by amplitude differences $d_a(i,j)$ using amplitudes $a_i(t)$ and $a_j(t)$ and their averages a_i and a_j.

These quantities measure all the synchronous changes of the components which were listed in the beginning of this chapter. Of course, long-term movements in the spectrum are weighted more, but if the frequencies or amplitudes do not change a lot, which is often the case for musical sounds, the quantity measures well the smaller changes like frequency modulation.

Measure of harmonic concordance

The frequency f_p of a harmonic partial is close to an integral multiple of the fundamental frequency f_0 of the harmonic sound. In our system we do not know the fundamental frequencies of the sounds even though they could be estimated in the original signal [Klapuri 1998]. In our system, we do not try to calculate fundamental frequencies of the sounds in the mixture. Instead, we developed a measure for modeling the harmonic concordance of any two sinusoidal trajectories. If we have two sinusoidal trajectories belonging to one harmonic source, the ratio of the frequencies of the trajectories is a ratio of small positive integers:

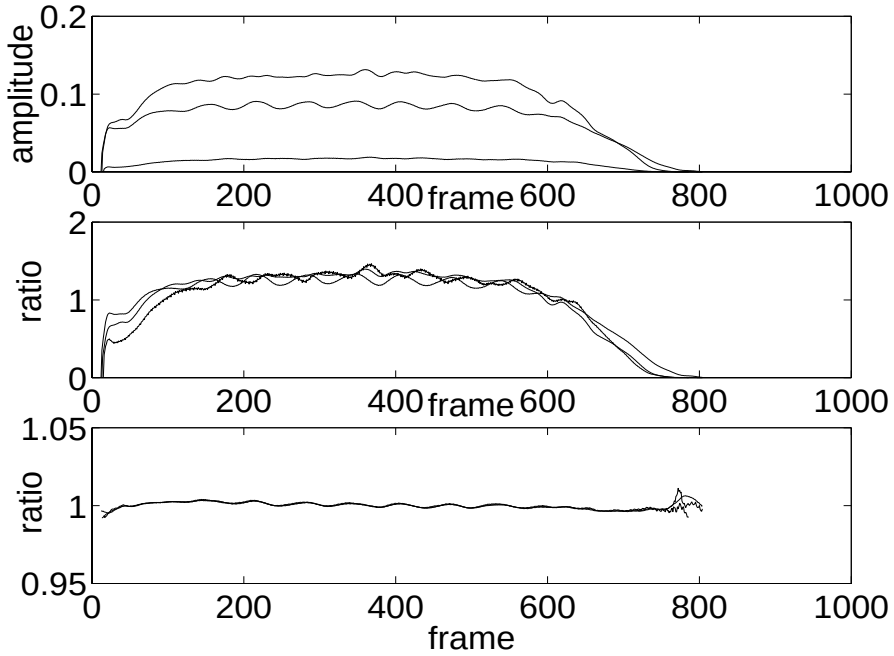

Figure 19: The upper plot shows the amplitudes of three first harmonics of the oboe. In the middle plot are scaled curves of these amplitudes. In the third plot are scaled curves of the frequencies of the same harmonics.

$$\frac{f_i}{f_j} = \frac{a}{b}, \tag{53}$$

where f_i and f_j are frequencies of the sinusoidal trajectories p_i and p_j and which are a^{th} and b^{th} harmonic of a sound.

Because we do not know which trajectory belongs to which sound or which trajectory is which harmonic partial, we assume that the fundamental frequency can not be smaller than the minimum frequency found in sinusoidal modeling. That way, we obtain upper limits for *a* and *b*:

$$a = 1, 2, \ldots, \left\lfloor \frac{f_i}{f_{min}} \right\rfloor, \; b = 1, 2, \ldots, \left\lfloor \frac{f_j}{f_{min}} \right\rfloor, \tag{54}$$

where f_{min} is the minimum frequency found in sinusoidal modeling.

After determining the limits for *a* and *b*, we calculate all the ratios for possible *a* and *b* and choose the one which the best estimate for the ratio of the frequencies. The harmonic distance quantity is then the error between ratio of the frequencies and the ratio of *a* and *b.* To give equal weight for ratios below and above unity, we use the absolute value of the logarithm of the ratio to measure the harmonic distance between trajectories:

$$d_h(i, j) = min\left|\log\left(\frac{f_i / f_j}{a / b}\right)\right|, \tag{55}$$

with *a* and *b* having the restrictions described in Equation 54.

Overall perceptual distance between trajectories

The overall perceptual distance between any two trajectories is a weighted sum of frequency, amplitude and harmonic distances:

$$d_{all}(i, j) = w_f d_f(i, j) + w_a d_a(i, j) + w_h d_h(i, j). \tag{56}$$

Due to the physics of sound production, frequencies usually do not vary as much as amplitudes. Thus the frequency distance has to be weighted more than the amplitude distance. The harmonic distance is calculated in a different way than the others, thus it has a different scaling. Because perceptual organization of simultaneous trajectories is largely based of harmonic concordance, the harmonic distance is weighted in a way to have the biggest effect.

Onset times are not taken into account directly when calculating these overall distances. Amplitude curves themselves contain the information of onset and offset times. Before onset, the amplitude of a trajectory is always zero. For natural sounds, the amplitude curve usually rises rapidly after the onset, and then starts to slowly decay. After the offset, the amplitude is again zero. The amplitude distance takes all this behavior into account.

7.4 Trajectory classification

After estimating the closeness between each pair of sinusoidal trajectories, they should be classified into separate sound sources. Because we do not have any coordinates common to all the trajectories, but only the distances between each pair of trajectories, we have to classify these trajectories into classes having minimum error between trajectories inside a class:

$$min\left(\frac{1}{|S_1|}\sum_{i,\, j \in S_1} d_{all}(i, j) + \frac{1}{|S_2|}\sum_{k,\, l \in S_2} d_{all}(k, l)\right), \tag{57}$$
$$S_1 \cup S_2 = S,\ S_1 \cap S_2 = \varnothing$$

where S_1 and S_2 are trajectory sets of two sounds, S is set of all trajectories, and $|S|$ is the cardinality of a set.

An ideal solution for choosing these classes would be to calculate all the possible permutations and choose the best one. However, for a practical number of sinusoidal trajectories, the number of calculations becomes very large (2 to the power of number of trajectories), so some other of classification approach must be taken. An efficient solution to this problem is to choose an initial set of trajectories for each class (or sound, in this case) and then add trajectories one by one to the classification by choosing the trajectory which has the minimum distance to the previous ones.

A good initial set of trajectories can be obtained by choosing all the trajectories whose onset times are close enough to an estimated onset time t_0 of a sound, and then evaluating all the possible subsets of certain number of trajectories. The subset which minimizes the error contains usually sinusoids which have been tracked well and do not contain any estimation errors or colliding sinusoids. To emphasize long, stable trajectories, the length of a trajectory can be used as a scaling factor. The number of trajectories close to onset time is usually so small that all the permutations can be evaluated:

$$e = min\left(\sum_{i,\, j \in S_1} \frac{d_{all}(i, j)}{\sqrt{length(p_i)}}\right), \tag{58}$$
$$(i \in S_1; |t_1 - t(i)| < t_{limit}),\ |S_1| = c,\ S_1 \subset S,$$

where t_1 is estimated onset time of sound 1, $t(i)$ is onset time of trajectory i, t_{limit} is maximum distance of trajectory to the onset and c is the size of initial subset. The onset times times can be obtained in many ways. In this system, we used the sum of difference of amplitudes at the beginning of each trajectory which were smoothed using a triangular window. The smoothed amplitudes were summed and local maxima of the resulting curve were selected as onset times. This onset detection method can detect onsets of sounds which do not have a strong transient in the beginning, for example violin.

When we have estimated the initial subsets for each sound, we start adding the rest of the trajectories into them one by one, always choosing a subset and trajectory whose distance is the smallest. The distance between a subset and a trajectory is simply the average of distances between the trajectory and the trajectories of the subset. Iteration is continued until all the trajectories are classified. The result of the classification is presented in Figure 20. One colliding trajectory belongs to both sounds. Detection of these colliding trajectories is discussed in the next section.

7.5 Colliding trajectories

As mentioned, the harmonics of two sounds often overlap in musical signals. Sinusoidal trajectories which are the result of overlapping harmonics, are called colliding trajectories. Which harmonics are overlapping, depends on the interval of the sounds. With harmonic intervals [Bregman 1990, Klapuri 1998] like major third and perfect fifth, many of the low harmonics are overlapping, because the ratio of the fundamental frequencies is a ratio of small integers. In the case of dissonant intervals, the low harmonics are not overlapping, but they can still be quite close to each other, which may cause estimation errors in sinusoidal modeling.

It is easy to see that an efficient way to detect colliding sinusoids is to find trajectories which are harmonically suitable for both sounds, or, whose harmonic distance to both sounds is small enough:

$$\frac{1}{|S_1|}\sum_{j \in S_1} d_h(i, j) + \frac{1}{|S_2|}\sum_{k \in S_2} d_h(i, k) < c_{limit}, \tag{59}$$

where S_1 and S_2 are sets of trajectories belonging to sounds 1 and 2 and c_{limit} is a constant. If equation is true for trajectory p_i, then it is probable that p_i contains harmonic partials from both sounds.

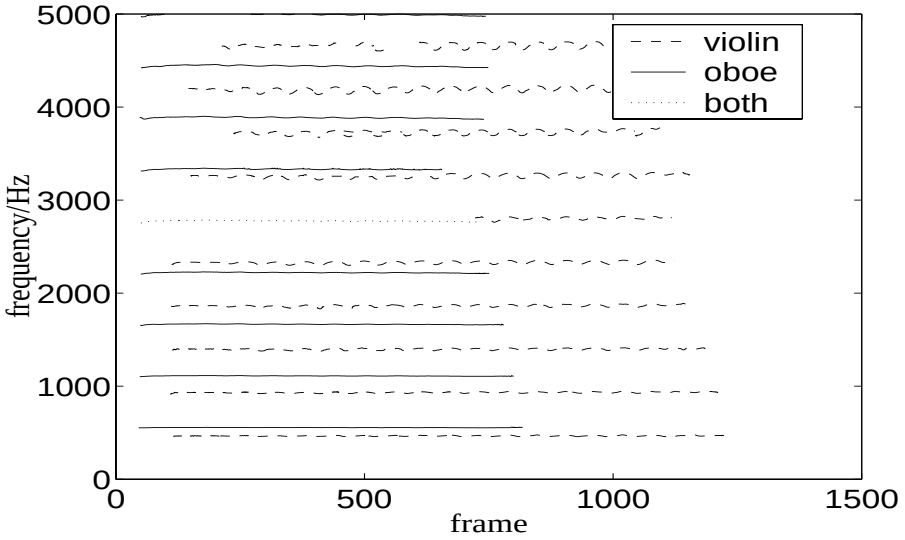

Figure 20: Classified trajectories.

The detected amplitude that results from two or more sinusoids close to each other in frequency is affected by the phase difference of the sinusoids. Because of the frequency and amplitude modulation, which is usually present in natural voices, estimation of the exact amplitudes and frequencies is very complicated, because after sinusoidal modeling we do not have the exact spectral information, only the detected sinusoids.

General solution of the colliding sinusoids problem is above the scope of this paper. This has been addressed e.g. in [Tolonen 1999]. However, to achieve better perceptual quality, we approximate the underlying harmonics. The system interpolates the amplitudes of the colliding trajectories using amplitude curves of other, not-colliding sinusoids. The frequencies are left intact.

Finally, when we have detected and split the colliding trajectories, we can represent the separated signals and synthesize them. The separated trajectories are presented in Figure 21.

Validation experiments and the perceptual quality of separated sounds demonstrate that the presented methods can be used to yield practically applicable results. Remaining problems to be addressed in the future include dynamic detection of the number of the mixed sounds, better estimation of amplitudes of colliding frequency partials, and separation of sounds that have the same onset time.

7.6 Separation using a multipitch estimation

Since the separation of signals using only the sinusoidal model becomes difficult for more than one sound, a system was built that uses also estimates of the fundamental frequencies and their harmonic partials. This work has been originally presented in [Klapuri et al. 2000]. The applied system differs from the standard sinusoidal model in a few ways. The frequencies of the harmonic components are obtained from a multipitch estimator (MPE),

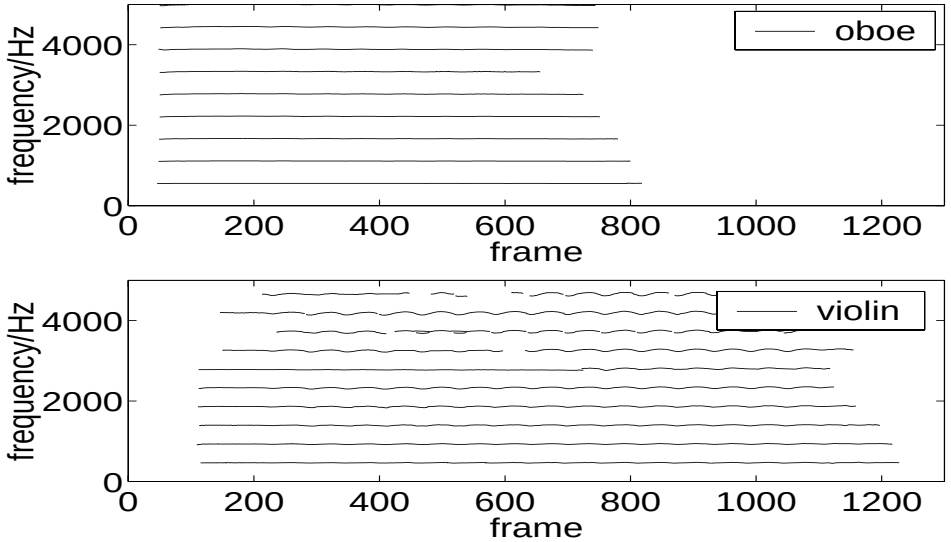

Figure 21: Separated trajectories. The 6th harmonic of the violin which was colliding with the 5th harmonic of the oboe, doesn't have vibrato.

which also deduces which component belong to which sound. Therefore, peak detection is not needed and the amplitudes and phases of the components can be solved using the LSQ algorithm. Peak continuation is not needed since the frequencies of the harmonic components are assumed constant inside one MPE window, which is much longer than one sinusoidal modeling frame. Unfortunately, this method fails to detect small changes in the fundamental frequency, such as vibrato.

After the parameter estimation, the sinusoids that are result of more than one harmonic partial are deduced from their sum. If the frequencies of two components are not exactly the same, the amplitude envelope of the sum of the components modulates at the rate which the difference between the frequencies of the components. Assuming that the original amplitude envelopes are slowly-varying, we can solve the mixed components as follows. The first amplitude envelope is obtained by lowpass filtering the envelope of the mixed components, and the other by subtracting the first from the original, and then half-wave rectifying and lowpass filtering the difference. Association of the two separated amplitude curves to their due sources of production is done by comparing the curves to other, already solved amplitude envelopes that were not overlapping. This comparing can be done using the perceptual measures presented in previous chapters. If more than two harmonic components are overlapping, their amplitudes are simply interpolated using the other, already solved components of each sound.

Some demonstration signals generated with this method are available at http://www.cs.tut.fi/~klap/iiro/dafx2000/.

7.7 Pitch and time-scale modifications

The sinusoidal and stochastic models allow modifications of the pitch without affecting the time scale and modifications of the time scale without affecting the pitch. The modifications are done for the parametric data so that we analyze the original audio signal, make desired modification for the parameters and then synthesize the signal. The quality of the modified signal is same as the quality of the synthesized signal without modifications. Also, the modifications are very simple: they do not require any FFTs or windowing, only a couple of multiplications and summations.

Let us have the frequencies $\omega(t, i)$, amplitudes $a(t,i)$ and phases $\varphi(t, i)$ of the deterministic part, and Bark-band energies $S(t,i)$ of the non-deterministic part. For modifications, we also need the hop size S. We stretch the time scale by factor ρ_t, which means that our original signal of length T becomes length $\rho_t T$. Also, we shift the pitch of the signal by factor ρ_ω, or, multiply the fundamental frequencies of the sounds by factor ρ_ω. In musical terms, a shift of s semitones is obtained using $\rho_\omega = 2^{(s/12)}$.

We assume that the non-deterministic part of the signal does not change when the pitch is changed, so the Bark-band energies do not require pitch-shift. For sinusoids, new frequencies $\omega'(t, i)$ are simply the old frequencies multiplied by the pitch-shift factor:

$$\omega'(t, i) = \rho_{\omega}\omega(t, i) \tag{60}$$

This modification technique does not preserve the formant structure of the signal. The synthesis of sinusoids and stochastic components allows us to modify the time scale simply by multiplying the hop size S by the time-stretch factor:

$$S' = \rho_t S \tag{61}$$

However, if the pitch or time-scale is modified, the exact waveform can not be preserved, so we have to generate phases that are an integral of the modified frequencies:

$$\varphi'(t, i) = S' \sum_{n=0}^{t} \omega'(t, i). \tag{62}$$

The modification abilities of the model were not examined very much, but some synthesized signals are available at http://www.cs.tut.fi/~tuomasv/demopage.html.

8. Conclusions

In this work, the sinusoids+noise model was studied, with the aim of applying it as a middle-level presentation for computational auditory scene analysis. The purpose of this work was to study the existing analysis and synthesis algorithms, and to try some original improvements. The usability of the model was verified in sound separation and manipulation experiments.

The whole sinusoids+noise model has some features taken from the human sound perception, but especially the sinusoidal model can be considered as a physical rather than a psychoacoustic model. For complex real-world signals, it is very difficult to detect the meaningful peaks and estimate their parameters in a single analysis frame. In this thesis, we have used the approach that is used in almost all sinusoidal models: at first the meaningful peaks are detected and then tracked into trajectories independently. For human sound perception, the process is somewhat different: the amplitude and length of a sound, and also other interfering sounds present affect how strong the perception is. The trajectory filtering is the only part of the system which tries to take into account this phenomenom.

Even though several combinations of advanced sinusoidal analysis algorithms were tested, the experimental results show that none of them alone is the ultimate answer to the sinusoidal analysis. The are many fundamental problems in the estimation of the parameters, mostly relating to the limited time- and frequency resolution.

The perceptual quality of the synthesized sounds is not good enough for high-quality audio coding, but the model fulfills the properties desired for a mid-level representation: it reduces the amount of data in the representation significantly without making too much high-level deductions that can not be guaranteed to be correct.

The experiments done show that the system is applicable in sound separation. With the sinusoidal model alone the separation is very limited and produces good results only if the number of mixed sounds is small. With the multipitch estimator the separation becomes more reliable. However, a lot of work has to be done before a good-quality sound separation can be achieved with rich-polyphony real-world signals.

During the development and implementation of this analysis/synthesis system a lot of knowledge of many audio signal processing ares was gained. The next step is to utilize this system further in the sound separation and other areas of computational auditory scene analysis.

References

[Ali 1996] Ali, M. “Adaptive Signal Representation with Applications in Audio Coding”. Ph.D. thesis, University of Minnesota.

[Bregman 1990] Bregman, Albert. S. “Auditory Scene Analysis”. MIT Press, 1990.

[Brown 1991] Brown, Judith C. “Calculation of a constant Q Spectral transform”. Journal of Acoustic Society of America, Vol 89(1), January 1991.

[Brown&Puckette 1992] Brown, Judith C. & Puckette, Miller S. “An efficient algorithm for the calculation of a constant Q transform”. Journal of Acoustic Society of America, Vol 92(5), November 1992.

[Colomes et al. 1995]. Colomes, C. & Lever, M. & Rault J.B. & Dehery Y.F. “A Perceptual Model Applied to Audio Bit-Rate Reduction”. Journal of Audio Engineering Society, Vol. 43(4), New York, October 1995.

[Depalle&Helie 1997] Depalle, Ph. & Hélie, T. “Extraction of Spectral Peak Parameters Using a Short-Time Fourier Transform And No Sidelobe Windows”. IEEE 1997 Workshop on Applications of Signal Processing to Audio and Acoustics”. Mohonk, New York, 1997.

[Depalle et. al 1993] Depalle, Ph. & Garcia, G. & Rodet, X. “Tracking of Partials for Additive Sound Synthesis Using Hidden Markov Models”. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Minneapolis, Minnesota, USA 1993.

[Desaint-Catherine&Marchand 2000] Desainte-Catherine, M. & Marchand, S. “High-Presicion Fourier Analysis of Sounds Using Signal Derivatives”. Journal of Acoustic Engineering Society, Vol 48(7), July/August 2000.

[Dolson 1986] Dolson, M. “The phase vocoder: a tutorial”, Computer Music Journal 10(4), 1986.

[Ellis&Rosenthal 1995] Ellis, D. & Rosenthal, D. “Mid-level representations for Computational Auditory Scene Analysis”. International Joint Conference on Artificial Intelligence - Workshop on Computational Auditory Scene Analysis, Montreal, Quebec, August 1995.

[Griffin&Lim 1985] Griffin, D. & Lim, J. “A New Model-Based Speech Analysis/Synthesis System”. IEEE International Conference on Acoustics, Speech, and Signal Processing, Tampa, Florida 1985.

[Hartmann 1997] Hartmann, William M. “Signals, Sound, and Sensation”. Springler-Verlag New York Inc., 1997.

[Kay 1993] Kay, M. “Fundamentals of Statistical Signal Processing: Estimation Theory”. PTR Prentice-Hall, Englewood Cliffs, New Jersey 1993.

[Klapuri 1998] Klapuri, Anssi. “Number Theoretical Means of Resolving a Mixture of Several Harmonic Sounds”. Proceedings of the European Signal Processing Conference, 1998.

[Klapuri et al. 2000] Klapuri, A., Virtanen, T., Holm, J.-A. “Robust Multipitch Estimation for the Analysis and Manipulation of Polyphonic Musical Signals”. Proceedings of the COST G-6 Conference on Digital Audio Effects, Verona, Italy, December 2000.

[Laroche 1998] Laroche, Jean. “Time and Pitch Scale Modifications of Audio Signals”. Kahrs, M. & Branderburg, K. (eds). “Applications of Digital Signal Processing to Audio and Acoustics”. Kluwer Academic Publishers, Boston / Dordrecht / London.

[Levine 1998] Levine, Scott. “Audio Representation for Data Compression and Compressed Domain Processing”. Ph.D. thesis. Stanford University.

[Meddis&Hewitt 1991] Meddis, R., Hewitt, J. “Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification”. Journal of Acoustic Society of America, June 1991.

[McAulay&Quatieri 1986] McAulay, Robert J., Quatieri, Thomas F. “Speech Analysis/Synthesis Based on a Sinusoidal Representation”. IEEE Transactions on Acoustics, Speech, And Signal Processing, Vol 34(4), August 1986.

[Moore 1997] Moore, Brian C.J. “An Introduction to the Psychology of Hearing”. Academic Press, 1997.

[Moorer 1985] Moorer, J.A. “Signal processing aspects of computer music: a survey”. Strawn, J. (ed.), “Digital Audio Signal Processing: An Anthology”, William Kauffman, Inc. 1985.

[Roads 1995] Roads, C. “The Computer Music Tutorial”, MIT Press, Cambridge, Massachusetts, USA, 1995.

[Rodet 1997] Rodet, Xavier. “Musical Sound Signal Analysis/Synthesis: Sinusoidal+Residual and Elementary Waveform Models”. IEEE Time-Frequency and Time-Scale Workshop 1997, Coventry, Grande Bretagne.

[Serra 1989] Serra, Xavier. “A system for analysis/transformation/synthesis based on a deterministic plus stochastic decomposition”. Ph.D. thesis, Stanford University.

[Serra 1997] Serra, Xavier. “Musical Sound Modeling with Sinusoids plus Noise”. Roads C. & Pope S. & Picialli G. & De Poli G. (eds). “Musical Signal Processing”. Swets & Zeitlinger Publishers.

[Sillanpää et al. 2000] Sillanpää, J., Klapuri, A., Seppänen, J., Virtanen T. “Recognition of Acoustic Noise Mixtures by Combined Bottom-up and Top-down Processing”. European Signal Processing Conference, Tampere, Finland 2000.

[Smith&Serra 1987] Smith, J.O., Serra, X. “PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation”, Proceedings of the International Computer Music Conference, 1987.

[Thomson 1982] Thomson, David J. “Spectrum Estimation and Harmonic Analysis”. Proceedings of the IEEE, 70(9), 1982.

[Tolonen 1999] Tolonen, Tero. “Methods for Separation of Harmonic Sound Sources using Sinusoidal Modeling”. AES 106th Convention, Munich, Germany, May 1999.

[Tolonen et al. 1998] Tolonen, T., Välimäki, V., Karjalainen, M. “Evaluation of Modern Sound Synthesis Methods. Helsinki University of Technology, Department of Electrical and Communications Engineering, Report 48., Espoo 1998.

[Virtanen&Klapuri 2000] Virtanen, T. & Klapuri, A. “Separation of Harmonic Sound Sources Using Sinusoidal Modeling”. IEEE International Conference on Acoustics, Speecg and Signal Processing, Istanbul, Turkey 2000.

[Virtanen 2001] Virtanen, Tuomas. “Accurate Sinusoidal Model Analysis and Parameter Reduction by Fusion of Components”, to be presented in AES 110th convention, Amsterdam, Netherlands, May 2001.

[Verma 1999] Verma, Tony S. “A Perceptually Based Audio Signal Model with Application to Scalable Audio Compression”. Ph.D. thesis. Stanford University, October 1999.

[Zwicker&Fastl 1999] Zwicker, E., Fastl, H. “Psychoacoustics: Facts and Models”. Springer-Verlag Berlin Heidelberg, 1999.

Appendix A: Fusion of Two Sinusoids: Derivation of the Equations

Starting from a sum of two sinusoids which have amplitudes, frequencies and phases a_1, a_2, ω_1, ω_2, φ_1 and φ_2, we represent the sum of the sinusoids with one sinusoid, which amplitude and phase are time-varying. Let the sum of the sinusoids at time t be denoted by $x(t)$:

$$x(t) = a_1 \sin(\omega_1 t + \varphi_1) + a_2 \sin(\omega_2 t + \varphi_2) \,. \tag{63}$$

Let $\operatorname{atan}\left(\frac{a_1}{a_2}\right) = \Phi$, $a_1 \neq 0$, $a_2 \neq 0$. The amplitudes are then

$$a_1 = \sqrt{a_1^2 + a_2^2} \sin(\Phi)$$

and

$$a_2 = \sqrt{a_1^2 + a_2^2} \cos(\Phi),$$

and $x(t)$ becomes

$$x(t) = \sqrt{a_1^2 + a_2^2} [\sin(\Phi)\sin(\omega_1 t + \varphi_1) + \cos(\Phi)\sin(\omega_2 t + \varphi_2)] .$$

Using basic trigonometric formulas, we get

$$x(t) = \frac{\sqrt{a_1^2 + a_2^2}}{2} [\cos(\Phi - \omega_1 t - \varphi_1) - \cos(\Phi + \omega_1 t + \varphi_1) +$$

$$\sin(\Phi + \omega_2 t + \varphi_2) - \sin(\Phi - \omega_2 t + \varphi_2)]$$

$$= \frac{\sqrt{a_1^2 + a_2^2}}{2} \left[\sin\left(\Phi - \omega_1 t - \varphi_1 + \frac{\pi}{2}\right) - \sin\left(\Phi + \omega_1 t + \varphi_1 + \frac{\pi}{2}\right) + \right.$$

$$\left. \sin(\Phi + \omega_2 t + \varphi_2) - \sin(\Phi - \omega_2 t + \varphi_2) \right]$$

$$= \sqrt{a_1^2 + a_2^2}\left[\cos\left(\Phi + \frac{(\omega_1 + \omega_2)t + \varphi_1 + \varphi_2}{2} + \frac{\pi}{4}\right)\sin\left(\frac{(\omega_2 - \omega_1)t + \varphi_2 - \varphi_1}{2} - \frac{\pi}{4}\right)\right.$$
$$\left. + \cos\left(\Phi - \frac{(\omega_1 + \omega_2)t + \varphi_1 + \varphi_2}{2} + \frac{\pi}{4}\right)\sin\left(\frac{(\omega_2 - \omega_1)t + \varphi_2 - \varphi_1}{2} + \frac{\pi}{4}\right)\right]$$

$$= \frac{\sqrt{a_1^2 + a_2^2}}{\sqrt{2}}\left[\sin\left(\frac{(\omega_2 - \omega_1)t + \varphi_2 - \varphi_1}{2} - \frac{\pi}{4}\right)\cos\left(\Phi + \frac{\pi}{4}\right)\cos\left(\frac{(\omega_1 + \omega_2)t + \varphi_1 + \varphi_2}{2}\right)\right.$$
$$\left. + \cos\left(\frac{(\omega_2 - \omega_1)t + \varphi_2 - \varphi_1}{2} - \frac{\pi}{4}\right)\sin\left(\Phi + \frac{\pi}{4}\right)\sin\left(\frac{(\omega_1 + \omega_2)t + \varphi_1 + \varphi_2}{2}\right)\right]$$

The sine and cosine of an inverse tangent can be simplified by:

$$\cos\left(\Phi + \frac{\pi}{4}\right) = \cos\left(\operatorname{atan}\frac{a_1}{a_2} + \frac{\pi}{4}\right) = \frac{1}{\sqrt{2}\sqrt{\frac{a_1^2}{a_2^2} + 1}} - \frac{\frac{a_1}{a_2}}{\sqrt{2}\sqrt{\frac{a_1^2}{a_2^2} + 1}} = \frac{a_2 - a_1}{\sqrt{2}\sqrt{a_1^2 + a_2^2}}$$

and

$$\sin\left(\Phi + \frac{\pi}{4}\right) = \frac{a_2 + a_1}{\sqrt{2}\sqrt{a_1^2 + a_2^2}}.$$

resulting to an expression of $x(t)$ which contains a sine and a cosine with equal frequencies and time-varying amplitudes:

$$x(t) = \left[\sin\left(\frac{(\omega_2 - \omega_1)t + \varphi_2 - \varphi_1}{2}\right)(a_2 - a_1)\cos\left(\frac{(\omega_1 + \omega_2)t + \varphi_1 + \varphi_2}{2}\right) + \right. \quad . \quad (64)$$
$$\left.\cos\left(\frac{(\omega_2 - \omega_1)t + \varphi_2 - \varphi_1}{2}\right)(a_2 + a_1)\sin\left(\frac{(\omega_1 + \omega_2)t + \varphi_1 + \varphi_2}{2}\right)\right]$$

By setting this expression of $x(t)$ equal to expression which has only one sinusoid of equal frequency and amplitude is a_3 and phase $\varphi_3(t)$, we get:

$$x(t) = a_3(t)\sin\left(\frac{(\omega_1 + \omega_2)t + \varphi_1 + \varphi_2}{2} + \varphi_3(t)\right) \quad (65)$$

$$= a_3(t)\left[\sin\left(\frac{(\omega_1+\omega_2)t+\varphi_1+\varphi_2}{2}\right)\cos(\varphi_3(t)) + \cos\left(\frac{(\omega_1+\omega_2)t+\varphi_1+\varphi_2}{2}\right)\sin(\varphi_3(t))\right], \quad (66)$$

From Equations (64) and (66) we get

$$a_3(t)\cos\varphi_3(t) = \cos\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)(a_2+a_1) \tag{67}$$

$$a_3(t)\sin\varphi_3(t) = \sin\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)(a_2-a_1) \tag{68}$$

Negative values are taken into account in the phase φ_3 so that we can get solution for a_3 by taking power of two and summing up the equations:

$$a_3(t) = \sqrt{\left[\sin\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)(a_2-a_1)\right]^2+\left[\cos\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)(a_2+a_1)\right]^2}$$

$$= \sqrt{a_1^2+a_2^2+a_1a_2\left[\cos^2\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)-\sin\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\right]}$$

$$= \sqrt{a_1^2+a_2^2+2a_1a_2\cos((\omega_2-\omega_1)t+\varphi_2-\varphi_1)}. \tag{69}$$

Dividing the Equation (68) by Equation (67) we get:

$$\tan(\varphi_3(t)) = \tan\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\frac{(a_2-a_1)}{(a_2+a_1)}, \tag{70}$$

from which the phase φ_3 can be solved using an inverse tangent. Since the inverse tangent is limited to interval $[-\pi/2, \pi/2]$, negative amplitudes are taken into account with a correction term ϕ:

$$\phi = \begin{cases} \pi & \frac{\pi}{2} < \frac{\varphi_2-\varphi_1}{2} \bmod 2 < \frac{3\pi}{2} \\ 0 & \text{otherwise} \end{cases}.$$

The equation for the phase becomes:

$$\varphi_3(t) = \operatorname{atan}\left(\tan\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\frac{(a_2-a_1)}{(a_2+a_1)}\right)+\phi \tag{71}$$

Using time-varying phase, we can represent the sum of the sinusoids with one sinusoids which amplitude and phase are time-varying. Now we want to express the phase $\varphi_3(t)$ at time instant *t* as a sum of initial phase $\varphi_3(0)$ and time-dependent term, or, frequency. The initial phase is solved by setting *t*=0, which results:

$$\varphi_3(0) = \operatorname{atan}\left(\tan\left(\frac{\varphi_2-\varphi_1}{2}\right)\frac{(a_2-a_1)}{(a_2+a_1)}\right)+\phi \tag{72}$$

Because instantaneous frequency is the derivative of the phase, it can be obtained by differentiating the Equation 71. The instantaneous frequency $\omega_3(t)$, or the derivative of phase $\varphi_3(t)$ becomes:

$$\omega_3(t) = \frac{d}{dt}(\varphi_3(t)) = \frac{d}{dt}\left(\operatorname{atan}\left(\tan\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\frac{(a_2-a_1)}{(a_2+a_1)}\right)+\phi\right)$$

$$= \frac{\frac{d}{dt}\tan\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\frac{(a_2-a_1)}{(a_2+a_1)}}{1+\tan^2\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\left[\frac{(a_2-a_1)}{(a_2+a_1)}\right]^2}$$

$$= \frac{\left[1+\tan^2\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\right]\frac{(a_2-a_1)}{(a_2+a_1)}\frac{d}{dt}\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}}{1+\tan^2\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\left[\frac{(a_2-a_1)}{(a_2+a_1)}\right]^2}$$

$$= \frac{\left[1+\tan^2\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\right]}{1+\tan^2\left(\frac{(\omega_2-\omega_1)t+\varphi_2-\varphi_1}{2}\right)\left[\frac{(a_2-a_1)}{(a_2+a_1)}\right]^2}\left(\frac{\omega_2-\omega_1}{2}\right)\frac{(a_2-a_1)}{(a_2+a_1)} \tag{73}$$

Now we can represent the sum of two sinusoids with one sinusoid the amplitude and frequency of which are time-varying:

$$x(t) = a_3(t)\sin\left(\frac{(\omega_2+\omega_1)t+\varphi_2+\varphi_1}{2}+\int_0^t\omega_3(u)\mathrm{d}u+\varphi_3(0)\right), \tag{74}$$

where $a_3(t)$, $\omega_3(t)$ and $\varphi_3(0)$ are described in Equations 69, 72 and 73. This equation is analogous to Equation 65 which has time-varying amplitude and phase.

Appendix B: Numerical Comparison of Algorithm Sets

The algorithm sets used in tables in this appendix are explained in the Chapter 6.2. The sets were used to analyze the generated test signal described in the same chapter in Table 3. Also the sections used in this appendix refer to that table.

Table 9: Peak interpolation: average frequency error / Hz

	signal section									
algorithm set	1	2	3	4	5	6	7	8	9	10
1	1.4	3.0	2.4	4.5	2.1	4.8	3.6	6.4	1.7	3.3
2	0.3	1.9	0.3	3.9	1.2	3.9	2.8	6.2	0.5	2.7
3	0.5	1.1	0.8	3.7	1.6	0.3	2.2	4.1	0.7	1.7
4	0.5	1.1	0.8	3.7	1.5	0.3	2.3	4.1	0.7	1.7
5	0.5	1.3	0.8	3.6	1.5	0.3	2.4	5.1	0.7	1.8
6	0.4	1.4	0.7	3.8	1.2	0.3	2.5	5.2	0.9	1.8
7	0.5	0.4	0.8	1.2	0.4	0.6	0.7	5.0	0.6	0.9
8	0.5	1.3	0.8	3.6	1.6	0.4	2.3	5.4	0.7	1.8

Table 10: Parameter estimation: average amplitude- and phase errors (distance to correct point in imaginary space)

	signal section									
algorithm set	1	2	3	4	5	6	7	8	9	10
1	0.2	1.6	0.2	0.9	1.2	1.0	2.0	0.8	0.6	1.6
2	0.2	1.6	0.2	0.9	1.2	1.0	2.0	0.8	0.5	1.6
3	0.2	1.1	0.2	0.7	1.1	0.2	1.9	0.3	0.4	1.7
4	0.7	0.6	0.2	0.4	1.3	0.2	0.7	0.2	0.3	0.9
5	0.7	0.7	0.2	0.4	1.3	0.2	0.8	0.2	0.3	1.0
6	0.7	0.7	0.2	0.4	1.3	0.2	0.8	0.2	0.3	1.0
7	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.2	0.3	0.7
8	0.7	0.7	0.2	0.4	1.7	0.3	0.8	0.3	0.3	1.0

Table 11: Peak detection: percentage of missed and extra peaks

	signal section									
algorithm set	1	2	3	4	5	6	7	8	9	10
1	0	6	77	4	18	23	10	0	40	17
2	0	6	77	4	18	23	10	0	40	17
3	0	7	28	0	27	67	78	68	33	89
4	0	7	28	0	27	67	78	68	33	89
5	0	4	28	1	23	64	64	21	31	74
6	0	2	28	1	23	64	64	21	31	74
7	25	49	25	22	53	47	82	32	33	63
8	0	5	28	1	18	62	57	10	31	67

Table 12: Peak continuation: percentage of false continuations and breaks in trajectories

	signal section									
algorithm set	1	2	3	4	5	6	7	8	9	10
1	0	6	0	8	11	8	11	3	30	18
2	0	4	0	8	7	9	10	3	30	17
3	0	2	2	1	11	0	3	4	30	2
4	0	2	2	1	10	0	2	4	30	2
5	0	2	2	2	11	0	4	8	30	4
6	0	0	2	2	11	0	4	8	30	4
7	0	1	2	1	2	13	1	6	30	6
8	0	1	2	2	9	0	5	8	30	7