Time-scale Modification using the Phase Vocoder

An approach based on deterministic/stochastic
component separation in frequency domain

Diploma Thesis

submitted by

Florian Hammer

Institute for Electronic Music and Acoustics (IEM),
Graz University of Music and Dramatic Arts
A-8010 Graz, Austria

Graz, September 2001

Advisor: o.Univ.-Prof. Mag. DI Dr. Robert Höldrich

For my parents

Abstract

The phase vocoder has been used as a time-scale-modification tool for several decades. Applying large positive modification factors to different kinds of sounds (time-stretching), the result will always sound “phasy” or “reverberant”. The sound quality can be improved by “locking” the phases. Phase-locking preserves the phase relations around a local maximum in the magnitude spectrum. For large modification factors, locking the entire phase spectrum sounds “rigid”.

In this work, the deterministic and stochastic components of a sound are separated in the frequency domain, and only the phases of sinusoids are locked, while the remaining phases are set to random numbers. The deterministic part is detected within a “reduced variance” magnitude spectrum using heuristic conditions. The reduced variance spectrum is calculated by weighting the actual magnitude spectrum, the spectrum of a frame about 11 ms in advance and the reduced variance spectrum 11 ms before the actual spectrum. In this way, the variance of the approximation can be reduced for the peak-detection yielding better results for noise-like signals. For resynthesis, the deterministic magnitudes are combined with the smoothed stochastic magnitudes, and the locked phases are combined with the random phases.

Kurzfassung

Der Phasen-Vocoder wird seit mehreren Jahrzehnten als Algorithmus zur Zeitskalenmodifikation verwendet. Wird ein Klang extrem in der Zeit “gedehnt”, klingt er “phasy” bzw. “hallig”. Eine Methode zur Verbesserung der Klangqualität ist “phase-locking”. Dabei werden die Phasenbeziehungen rund um ein lokales Maximum im Analyseamplitudenspektrum in die Berechnung der Synthesephasen übernommen. Für grosse Skalierungsfaktoren klingt das Ergebnis “starr”, wenn phase-locking auf das gesamte Phasenspektrum angewendet wird.

In dieser Arbeit werden die deterministischen und stochastischen Anteile eines Klanges im Frequenzbereich getrennt. Dadurch ist es möglich, nur für die deterministischen Phasen phase-locking zu verwenden und die stochastischen Phasen auf Zufallszahlen zu setzen. Die Detektion der deterministischen Komponente geschieht innerhalb eines Amplitudenspektrums mit verringerter Varianz und basiert auf heuristischen Bedingungen. Die Varianz der Schätzung wird dadurch verringert, da das aktuelle Amplitudenspektrum, das Spektrum ca. 11 ms danach und das Spektrum mit reduzierter Varianz ca. 11 ms davor, gewichtet werden. In diesem gewichteten Spektrum werden die spektralen Maxima gesucht. Durch die Reduktion der Varianz der Schätzung werden für rauschhafte Signale bessere Resultate bei der Detektion des deterministischen Anteils erreicht. Für die Resynthese des modifizierten Signals werden die deterministischen Amplituden mit den geglätteten stochastischen Amplituden und die deterministischen Phasen mit den zufälligen Phasen kombiniert.

Acknowledgements

This work is dedicated to my parents, who supported me and who encouraged me to study what I was interested in.

I want to thank my colleagues at the IEMA for giving me a helping hand. Thanks to Thomas Musil for all the innumberable on- and off-topic talks and to Anderl for a lot of upcheering coffee breaks and our joint exploration of the LaTeX-environment.

I especially wish to thank my advisor, Prof. Robert Höldrich, for his support, and for leading me to autonomous scientific work.

Special mention to Alberto DeCampo, Stephen T. Pope and Curtis Roads for their interest and their hospitality during my research stay at the University of California at Santa Barbara.

Florian Hammer
Graz, Austria, September 2001

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What this thesis is about

This thesis focuses on time-expansion of musical signals using the phase vocoder. The aim is to find an algorithm, which yields good resynthesis using large scaling factors. At the beginning, we give a brief general overview on time-scale modification based on an article of Moulines and Laroche [1]. Chapter 2 presents an introduction to the phase vocoder and its application to time-scale-expansion. As alternatives to the phase vocoder as a tool for sound analysis/transformation/resynthesis, three signal models are reviewed in chapter 3. Chapter 4 explores the characteristics of the phase vocoder parameters in order to find conditions usable for deterministic component detection. This is part of an enhanced phase vocoder system presented in chapter 5. Finally chapter 6 concentrates on strategies to detect transients and shows how such a detection could be included in the system.

1.2 Talking about time-scale modification

1.2.1 Definition

(From [1]:) The object of time-scale modification is to alter the signal's apparent time-evolution without affecting its spectral content. Defining an arbitrary time-scale modification amounts to specifying a mapping between the time *in the original signal* and the time *in the modified signal*. This mapping $t \rightarrow t' = T(t)$ is refered to as the *time warping function*. In the following, t refers to the time in the original signal, t' to the time in the modified signal. It is often convenient to use an integral definition of T:

$$t \rightarrow t' = T(t) = \int_0^t \beta(\tau) d\tau \tag{1.1}$$

where $\beta(\tau) > 0$ is the time-varying time-modification rate[1].

[1] $\beta(\tau) > 0$ guarantees that $T(t)$ is never decreasing and therefore that $T^{-1}(t')$ exists.

$\beta(\tau) > 0 \ldots$ time-scale modification rate:

$\beta(\tau) > 1 \Leftrightarrow$ time-scale expansion
$\beta(\tau) < 1 \Leftrightarrow$ time-scale compression

1.2.2 Mathematical model

The following definitions are based on a sinusoidal model (see chapter 3), in which the signal is represented by a sum of sinusoids. It is important to note that the sinusoids' instantaneous amplitude $A_i(t)$ and frequency $\omega_i(t)$ are allowed to vary only slowly in time. A short section of a sound can then be considered as stationary.
Signal model:

$$x(t) = \sum_{i=1}^{I(t)} A_i(t) e^{j\phi_i(t)} \tag{1.2}$$

with

$$\phi_i(t) = \int_{-\infty}^{t} \omega_i(\tau) d\tau$$

Ideal time-scaled signal:

$$x'(t') = \sum_{i=1}^{I(T^{-1}(t'))} A_i(T^{-1}(t')) e^{j\phi'_i(t')} \tag{1.3}$$

with

$$\phi'_i(t') = \int_{-\infty}^{t'} \omega_i(T^{-1}(\tau)) d\tau$$

Remarks:

- The amplitude of the i^{th} oscillation of the time-scaled signal at time t' is equivalent to the amplitude of the original signal at time T^{-1}(t')
- $\frac{d\phi'(t')}{dt'} = \omega_i(t')$ equals $\omega_i(T^{-1}(t'))$, so the temporal evolution is modified, but the frequency content remains unchanged.

1.2.3 Applications

Time-scale modification is used in

- **Synthesis by sampling.** Synthesizers based on the sampling technique typically hold a dictionary of pre-recorded sound units (e.g., musical sounds or speech segments) and generate a continous output sound by splicing together the segments with a pitch and duration corresponding to the desired melody. Because there can only be a limited number of sound segments stored in the dictionary, one cannot afford sampling all possible pitches and durations, hence the need for indepentent time-scale and pitch-scale control.

- **Post-synchronization.** Synchronizing sound an image is required when a soundtrack has been prepared independently from the image it is suppose to accompany. By modifying the time evolution of the sound track, one is able to re-synchronize sound and image. A typical example is dialogue post-synchronization in the movie industry.
- **Data compression.** Time-scale modification has also been studied for the purpose of data compression for communications or storage [2]. The basic idea consisted of shrinking the signal, transmitting it, and expanding it after reception. It was found however, that only a limited amount of data reduction could be obtained using this method.
- **Reading for the blind.** For visually impaired people, listening to speech recordings can be the only practical alternative to reading. However, one can read at a much faster rate than one can speak, so 'reading by listening' is a much slower process than sight reading. Time-scale modification makes it possible to increase this listening rate.
- **Foreign language learning.** Learning a foreign language can be significantly facilitated by listening to foreign speakers with an artificially slow rate of elocution which can be made faster as the student's comprehension improves.
- **Computer interface.** Speech-based computer interfaces suffer from the same limitation as encountered in 'reading by listening'. The pace of the interaction is controlled by the machine and not by the user. Techniques for time-scale modifications can be used to overcome the 'time bottleneck' often associated with voice interfaces.
- **Post-production Sound Editing.** In the context of sound recording, the ability to correct the pitch of an off-key musical note can help salvage a take that would otherwise be unusable. Multi-track hard-disk recording machines often offer such capabilities.
- **Musical composition.** Finally, music composers working with pre-recorded material find it interesting to be given an independent control over time and pitch. In this context, time and pitch-scale modification systems are used as composition tools and the 'quality' of the modified signal is an important issue.

1.2.4 Techniques for time-scale modification

Generally time-scale modification can be implemented in three ways, for further details refer to the bibliography.

1. Time domain:
   - Modified tape recorder [3]
   - Digital implementation of the modified tape recorder [4].
   - Improvements:

     Speech: [5], [6], [7], and [8]
     Music: [9], [10], [11], and [12]

2. Frequency domain: Short time fourier transfomation (STFT) [13]
    - Improvements:

      Speech: [33] and [14]
      Music: [31], [35] and [15]
      See also chapter 2.

3. Signal models: Modeling a signal and changing the parameters
    - Linear prediction models: [16] and [17]
    - Sinusoidal models (plus noise): see chapter 3
    - “Granular models”: [18], [19], [20] and [21]

Chapter 2

Phase Vocoder

2.1 Introduction to the phase vocoder

Homer Dudley introduced the "channel vocoder" (voice coder) in 1939 [28], which operates on the principle of deriving voice codes to re-create the speech which it analysed. In the analysis stage, the fundamental frequency is determined and spectral information is provided by ten filters. The synthesizer discriminates voiced-unvoiced speech by use of the energy of the fundamental frequency and generates a buzz for voiced sounds and random noise for the hiss. This signal gets filtered corresponding to the analysis.

Flanagan and Golden [29] have been extending this model by taking the short-time magnitude *and* phase spectra of the signal into account. Since then the now called "phase vocoder" has been developed in various ways.

In his phase vocoder tutorial [35], Mark Dolson presented two complementary viewpoints, which explain how phase vocoder calculations work: He refered to these viewpoints as the filterbank interpretation and the Fourier transform interpretation (Figure 2.1).

Filterbank interpretation

In the analysis stage, the time-varying amplitudes and frequencies of a signal are extracted by a fixed bank of bandpass filters. These parameters are then used to control a bank of sine-wave oscillators for resynthesis.

This thesis is based on the Fourier transform interpretation, so this kind of analysis/synthesis will be described in the next section.

2.2 Short-Time Fourier Transform (STFT)

A sound analysis/synthesis system based on the STFT is structured as shown in figure 2.2. The following sections will describe the main parts of the system.

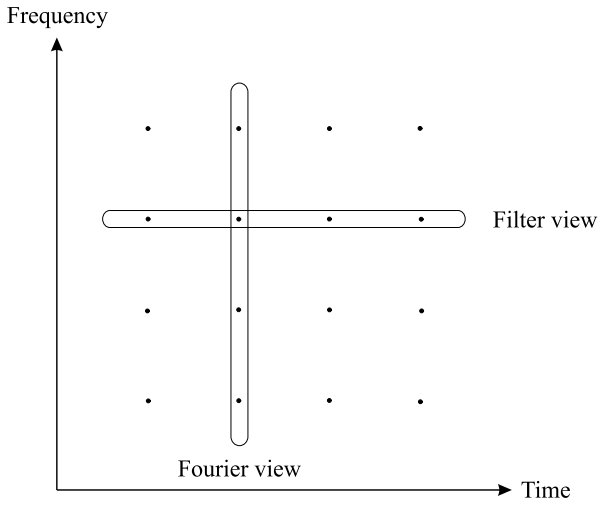

FIGURE 2.1: Filterbank vs. Fourier-transform interpretation (from: [35])

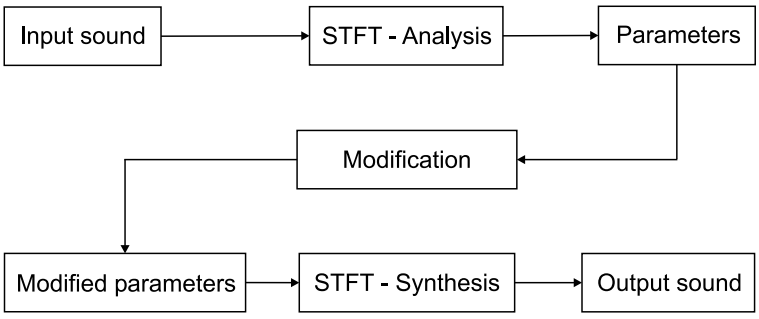

FIGURE 2.2: An analysis/synthesis system based on the short-time Fourier transform.

2.2.1 Analysis

During the analysis stage (figure 2.3), analysis time-instants t_a^u for successive values of integer u are set along the original signal, possibly uniformly: $t_a^u = uR_a$ where R_a is the so-called analysis hop factor. At each of these analysis time-instants, a Fourier transform is calculated over a windowed portion of the original signal, centered around t_a^u. The result is the *nonheterodyned* STFT representation of the signal, denoted $X(t_a^u, \Omega_k)$:

$$X(t_a^u, \Omega_k) = \sum_{n=-\infty}^{\infty} h(n)x(t_a^u + n)e^{-j\Omega_k n} \tag{2.1}$$

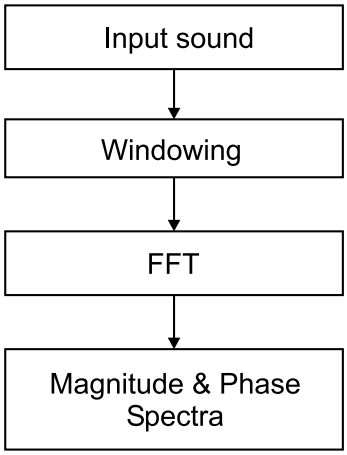

FIGURE 2.3: Short-time Fourier transform: Analysis.

where x is the original signal, $h(n)$ is the analysis window, $\Omega_k = \frac{2\pi k}{N}$ is the center frequency of the kth vocoder "channel" and N is the size of the discrete Fourier transform. In practise, $h(n)$ has a limited time span (typically N samples) and the sum above has a finite number of terms. $X(t_a^u, \Omega_k)$ is both a function of time (via variable u) and frequency (via Ω_k). (From: [41])

2.2.1.1 The analysis window - time/frequency trade-off

The well-known time/frequency trade-off is a very important issue in audio analysis. The point is that large windows provide good frequency resolution, but poor time resolution, and small windows vice versa.

2.2.2 Resynthesis

The resynthesis stage (figure 2.4) involves setting synthesis time-instants t_s^u, usually uniformly, so that $t_s^u = R_s u$, where R_s is the synthesis hop factor. At each of these synthesis time-instants, a short-time signal $y_u(n)$ is obtained by inverse-Fourier-transforming the synthesis STFT $Y(t_s^u, \Omega_k)$. Each short-time signal is then multiplied by an optional synthesis window $w(n)$, and the windowed short-time signals are all summed together, yielding the output signal $y(n)$:

$$y(n) = \sum_{u=-\infty}^{\infty} w(n - t_s^u) y_u(n - t_s^u) \tag{2.2}$$

with

$$y_u(n) = \frac{1}{N} \sum_{k=0}^{N-1} Y(t_s^u, \Omega_k) e^{j\Omega_k n}$$

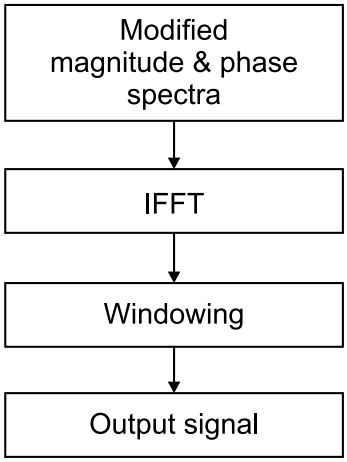

FIGURE 2.4: Short-time Fourier transform: Resynthesis.

In the absence of modifications (i.e., $R_a = R_s$ and $Y(t_s^u, \Omega_k) = X(t_a^u, \Omega_k)$), this output signal is identical to the original signal x, under mild conditions on the analysis and synthesis windows [43]. (From: [41])

2.3 Time-scale modification using the phase vocoder

The time-scale of a signal can be modified by the analysis and synthesis hopsize being different

$R_a < R_s$ . . .time-expansion
$R_a > R_s$ . . .time-compression

and calculating the output phases of the modified signal explicitly in a way described below. Note that although the phase vocoder is based on a sum of sinusoids model, no explicit estimation of sinusoidal parameters is necessary.

Modification of the magnitude

In this algorithm, the synthesis magnitude values are the same as the analysis magnitude values:

$$\mid Y(t_a^u, \Omega_k) \mid = \mid X(t_a^u, \Omega_k) \mid \tag{2.3}$$

where

$$t_s^u = R_s u$$

So there is no need for modifying the magnitude spectrum.

Modification of the phase

Phase unwrapping is required to calculate the synthesis phase. Therefore the instantaneous frequencies $\omega_i(t)$ of sinusoids near a certain vocoder channel are calculated by use of the phase increment of two consecutive frames.
At first, the heterodyned phase increment is calculated:

$$\Delta\Phi_k^u = \angle X(t_a^u, \Omega_k) - \angle X(t_a^{u-1}, \Omega_k) - R_a\Omega_k \tag{2.4}$$

After taking the principal determination $\Delta_p\Phi_k^u$ (between $\pm\pi$), the instantaneous frequency $\hat{\omega}(t_a^u)$ of the closest sinusoid is derived for each channel.
Phase unwrapping:

$$\hat{\omega}(t_a^u) = \Omega_k + \frac{1}{R_a}\Delta_p\Phi_k^u \tag{2.5}$$

Phase propagation formula:

$$\angle Y(t_s^u, \Omega_k) = \angle Y(t_s^{u-1}, \Omega_k) + R_s\hat{\omega}_k(t_a^u) \tag{2.6}$$

Phase coherence

The phase propagation formula ensures phase consistency *within* each frequency channel over time, which is denoted as *horizontal phase coherence*. On the other hand, phase consistency *across* channels within a synthesis frame has to be preserved, too. We refer to this kind of consistency as *vertical phase coherence*.
In the next section, algorithms will be presented which take care of the phase consistency.

More references on the STFT/Phase Vocoder (in chronological order):
[30], [24], [31], [32],[33], [43], [34], [36], [44], [26], [37], [45], [42], [38]

2.4 Phase-locked Vocoder

2.4.1 Loose phase-locking

Miller Puckette [39] refers to the fact that a complex exponential does not only excite one channel of the phase vocoder analysis but *all of the channels within the main lobe* of the analysis window. He points out the necessity to "lock" the phases around bins of sinusoids to reduce artefacts in the resynthesis. The new synthesis values are based on a weighted average of three neighboring synthesis phases of the previous frame (here adjacent bins are 180 degrees out of phase):

$$Y[u_i, k] = X[t_j, k](\frac{Z[u_{i-1}, k]}{X[t_{j-1}, k]}) \mid \frac{Z[u_{i-1}, k]}{X[t_{j-1}, k]} \mid^{-1} \tag{2.7}$$

with

$$Z[u_{i-1}, k] = Y[u_{i-1}, k] - Y[u_{i-1}, k-1] - Y[u_{i-1}, k+1].$$

t_i ..... actual analysis frame
t_{i-1} .... previous analysis frame
u_i ..... actual synthesis frame
u_{i-1} .... previous synthesis frame
k..... bin-number
X[t,k]... actual analysis phases
Y ..... synthesis phases

2.4.2 Rigid phase-locking

Inspired by the phase-locking method by Puckette, Laroche and Dolson ([40], [41]) developed the so called “rigid phase-locking”-algorithm. Moving a step further, the peaks of the magnitude spectrum are detected and the phases of the neighboring bins are locked to the phases of the peakbins within a “region of influence”. The borders of the region of influence can be set to the middle frequency between two peak-bins or to the channel of lowest magnitude between the two peaks.
Two kinds of (rigid) phase-locking are presented:

- Identity phase-locking
- Scaled phase-locking

2.4.2.1 Identity phase-locking

This kind of phase-locking assigns the analysis phase relations of a peak bin to the neighboring bins to the synthesis phases within the region of influence.

$$\angle Y(t_s^u, \Omega_k) = \angle Y(t_s^u, \Omega_{k_l}) + \angle X(t_a^u, \Omega_k) - \angle X(t_a^u, \Omega_{k_l}) \tag{2.8}$$

A major advantage of this method is that only peak channels require trigonometric calculations: Once the synthesis phase of a peak channel has been derived, the difference to the analysis phase Ω_a is used to create a complex exponential by which the bins in the region of influence are rotated.

$$\theta = \angle Y(t_s^u, \Omega_{k_l}) - \angle X(t_a^u, \Omega_{k_l}) \tag{2.9}$$

$$Z = e^{j\theta} \tag{2.10}$$

$$Y(t_s^u, \Omega_k) = ZX(t_a^u, \Omega_k) \tag{2.11}$$

Identity phase-locking scheme:

1. For the new STFT frame, locate prominent peaks.
2. For each peak, calculate the instantaneous frequency using horizontal phase unwrapping, and calculate the updated synthesis phase, according to (2.6).

3. Calculate rotation $\angle\theta$, according to (2.9), and phasor Z.
4. Apply rotation to all channels around and including peak channel, according to (2.11).
5. Repeat the above steps for the next peak, until all peaks have been processed.
6. Proceed to the next synthesis frame.

2.4.2.2 Scaled phase-locking

The identity phase-locking technique can be improved by recognizing a peak switching from channel k_0 at frame u-1 to channel k_1 at frame u. In this case, the calculation of the phase increment in the phase unwrapping equation changes
from: $\angle X(t_a^u, \Omega_{k_1}) - \angle X(t_a^{u-1}, \Omega_{k_1})$
to: $\angle X(t_a^u, \Omega_{k_1}) - \angle X(t_a^{u-1}, \Omega_{k_0})$
and the phase-propagation equation should be:

$$Y(t_s^u, \Omega_{k_1}) = \angle Y(t_s^{u-1}, \Omega_{k_0}) + R_s \hat{\omega}_{k_1}(t_a^u) \tag{2.12}$$

As soon as the corresponding peak in the previous frame has been found for the actual peak, we use the analysis and synthesis phases of both peaks to calculate the new synthesis phase. Now the neighboring channels can be synchronized by a generalized phase-locking equation:

$$\angle Y(t_s^u, \Omega_k) = \angle Y(t_s^u, \Omega_{k_l}) + \beta[\angle X(t_a^u, \Omega_k) - \angle X(t_a^u, \Omega_{k_l})] \tag{2.13}$$

β ..... Scaling factor
β =1... Identity phase-locking

Using the upper formula, it appears that identity phase-locking can be further improved by setting β to a value between one and the timescaling-factor α. Informal listening tests have shown that setting $\beta \approx 2/3 + \alpha/3$ helps further reduce phasiness. The phases $X(t_a^u, \Omega_k)$ must be unwrapped across channels k around the peak channel before applying (2.13), in order to avoid $2\beta\pi$ channel jumps in the synthesis phases. Since it is not possible to calculate with simple complex multiplications here, the implementation of scaled phase-locking needs more computation than identity phase-locking. On the other hand it provides consistently higher sound quality.

Scaled phase-locking scheme:

1. For the new STFT frame, locate prominent peaks
2. For each peak channel k_i, locate the corresponding peak in the preceding frame, calculate the instantaneous frequency using horizontal phase unwrapping, and compute the updated synthesis phase according to (2.12).
3. Unwrap analysis phases across all channels in the region of influence.

4. For each channel around the peak channel, calculate analysis phase difference between peak and current channel, and compute the current synthesis phase using (2.13).
5. Repeat the above steps for the next peak, until all peaks have been processed.
6. Proceed to the next synthesis frame.

Chapter 3

Signal Models

3.1 Introduction

This chapter shortly reviews signal models: The *sinusoidal model* and its developments, *Spectral Modeling Synthesis (SMS)*, which extends the sinusoidal model by including the stochastic part of a signal, and finally *Transient Modeling Synthesis*, a system, which introduces a model for transients.

3.2 Sinusoidal Modeling

Quasi-stationary sounds can be modeled by extracting the sinusoidal components of the signal:

- Magnitude
- Phase
- Frequency

McAulay and Quatieri ([57], [58], [59]) set up a speech analysis/synthesis technique based on this signal characterization. This technique and its developments will be described in the next sections.

3.2.1 McAulay-Quatieri

The McAulay/Quatieri analysis/synthesis system (further refered to as MQ-system) is based on a sum-of-sinusoids model for which the sinusoidal parameters are calculated explicitly.

The model:

$$x(n) = \sum_i A_i(n) cos[\Theta_i(n)] \tag{3.1}$$

with

$$\Theta_i(n) = \int_0^{nT} \omega_i(\tau)d\tau + \Phi_i$$

$A_i(n)$ .. time-varying envelope
$\omega_i(n)$... instantaneous frequency (frequency track of the i^{th} sine wave)
$\Theta_i(n)$... instantaneous phase

3.2.1.1 MQ-Analysis

As you can see in figure 3.1, after applying a STFT on the input signal, the magnitude spectrum is searched for spectral peaks, which indicate the sinusoidal components of the analyzed signal. For those peaks the sinusoidal parameters are derived.

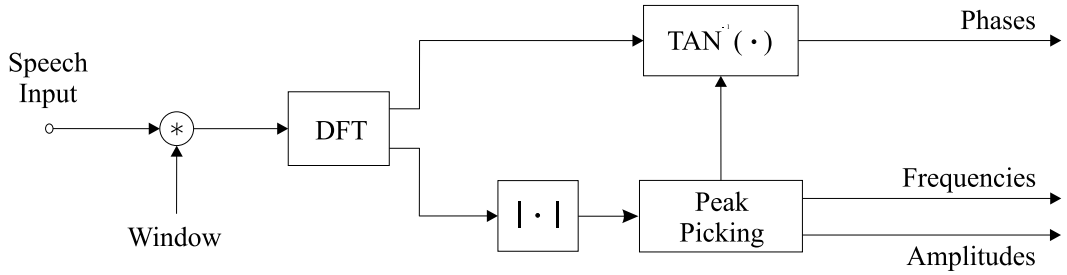

FIGURE 3.1: MQ-Analysis (from: [59])

3.2.1.2 MQ-Synthesis

Figure 3.2 shows the further processing of the peak-parameters. The peaks are linked over successive frames to form tracks. Each track represents the time-varying behavior of a single sinusoidal component in the analysed sound.

For resynthesis the magnitudes are linearly interpolated and track frequencies are modulated parabolically (cubic phase interpolation) to preserve the phase accuracy at frame boundaries.

Application to nonharmonic sounds by Smith and Serra: PARSHL [60]

Transformations: [61], [62], [63], [64], [65], [66]

3.2.2 LEMUR

Fitz and Haken [68] developed an extended version of the MQ-system called LEMUR. This tool performes an enhanced MQ-Analysis on sampled sounds and generates a data file describing a MQ-style sinusoidal model of the signal. Using Lemur's built-in editing functions

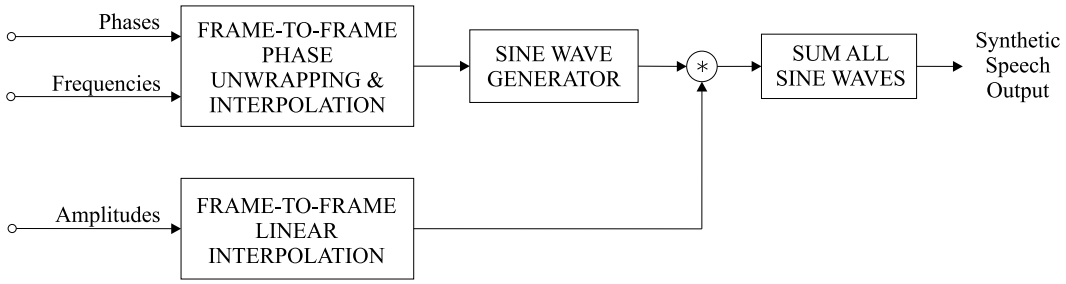

FIGURE 3.2: MQ-Synthesis (from: [59])

those files may be modified in different ways. Finally, Lemur can synthesize the modified data to create a new sampled signal.

Further references: [67], [69]

3.2.3 Waveform preservation based on relative phase delays

Di Federico [70] introduced a system for waveform invariant time-stretching and pitch-shifting of quasi-stationary sounds based on a *relative phase delay* representation of the phase. Once the sinusoidal parameters have been extracted from the signal, partial phases are transformed into phase delays:

$$\tau_{i,k} = \frac{\Theta_{i,k}}{\omega_{i,k}} \tag{3.2}$$

i ..... frame number
$\Theta_{i,k}$... Phase of the kth sinusoid
$\omega_{i,k}$ ... Frequency of the kth sinusoid
$\tau_{i,k}$ ... Phase delay of the kth sinusoid

Since phase delays are homogeneous quantities (unlike phases), the phase delays of two different partials can be compared. *Relative phase delays (rpds)* are defined as the differences between the partial phase delays and the phase delay of the fundamental.

$$\Delta\tau_{i,k} = \tau_{i,k} - \tau_{i,1} \tag{3.3}$$

The original waveform of a quasi-harmonic sound can be "built" on the phase value of the fundamental. So the sound is no longer described by magnitudes + frequencies + phases, but by a magnitude + frequencies + *rpds* + fundamental phase representation. Because analysis phases are wrapped, the relative phase delays have to be normalized for further calculations so that the *normalized relative phase delays (nrpds)* lie in the range $[0, 2\pi/\omega_i]$. For resynthesis, the synthesis phase of the fundamental is evaluated by cubic phase interpolation and the *intraframe* phase coherence of the other partials is preserved by use of the *nrpds*.

This algorithm yields high quality results for harmonic or quasi-harmonic sounds even for time-stretching factors up to 30 and more.

3.2.4 High Precision Fourier Analysis using Signal Derivatives

Introducing a k^{th}-order Fourier transform, Desainte-Catherine and Marchand ([71], [72], and [73]) have developed a system, which improves Fourier analysis precision in amplitude, frequency, and time.

Restrictions:

As for all the other systems based on a sinusoidal model, partials must be slowly varying, sufficiently spaced and must not cross each other.

3.2.4.1 k^{th}-order DFT

DFT^k denotes the amplitude spectrum of the discrete Fourier transform of the k-th signal derivative.

$$DFT^k[m] = \frac{1}{N} \mid \sum_{n=0}^{N-1} w[n] \frac{d^k a}{dt^k}[l+n] e^{-j\frac{2\pi}{N}nm} \mid \tag{3.4}$$

N... window size
l ... window location
w .. N-point analysis window

For each partial p there is a maximum in both DFT^0 and DFT^1 spectra for a certain index m$_p$.
The DFT^0 corresponds to the classic STFT analysis.

$$f_p^0 = m_p \frac{f_s}{N} \tag{3.5}$$

$$a_p^0 = DFT^0[m_p] \tag{3.6}$$

f_p^0... partial frequency (=bin frequency)
a_p^0 .. partial magnitude
f_s... sampling rate
N .. size of the analysis window

The following equations provide much more accurate frequency and magnitude values:

$$f_p^1 = \frac{1}{2\pi} \frac{DFT^1[m_p]}{DFT^0[m_p]} \tag{3.7}$$

$$a_p^1 = \frac{a_p^0}{W(\mid f_p^1 - f_p^0 \mid)} \tag{3.8}$$

W(f) ... amplitude of the continous spectrum of the analysis window w at frequency f.
The window size can be smaller, so a better time-resolution (e.g., for vibratos) can be obtained. The effects of the analysis window compensate by dividing the two DFTs, so e.g., tremolos can be represented better.
Essential drawbacks:

- Not usable for low-pitched sounds due to the small window-lengths
- Noise components have to be negligible

3.3 Spectral Modeling Synthesis - SMS

In [74], Serra introduces a very important extension to the sinusoidal model: The incorporation of the noise components of sounds into an extended signal model.
In this system, a sound is considered to be a combination of a sum of sinusiods (*deterministic component*) and a residual (*stochastic component*), which represents transients and noise.
Input sound model:

$$s(t) = \sum_{i=1}^{L} A_i(t) cos[\Theta_i(t)] + e(t) \tag{3.9}$$

$A_i(t)$ and $\Theta_i(t)$ are the instantaneous amplitude and phase of the ith sinusoid, respectively, and $e(t)$ is the noise component at time t.
The restrictions of this model are the same as for the sinusoidal model:
Slowly changing amplitude and frequency of the deterministic part of the sound.

3.3.1 SMS - Analysis

In the analysis stage, which is shown in figure 3.3, we want to find the sinusoids within the sound, resynthesize them and subtract them from the original signal in time domain to get the residual.

Deterministic analysis

Peaks are picked from the magnitude spectrum with the additional constraint that the phases of sinusoidal peaks must have a constant phase around the peak-bin, when applying *zero-phase windowing* to the analysis frame. Zero-phase windowing is necessary to obtain a phase spectrum free of the linear phase trend induced by the analysis window. Therefore the signal frame to be analysed has to be centered around the origin before applying the FFT to it.

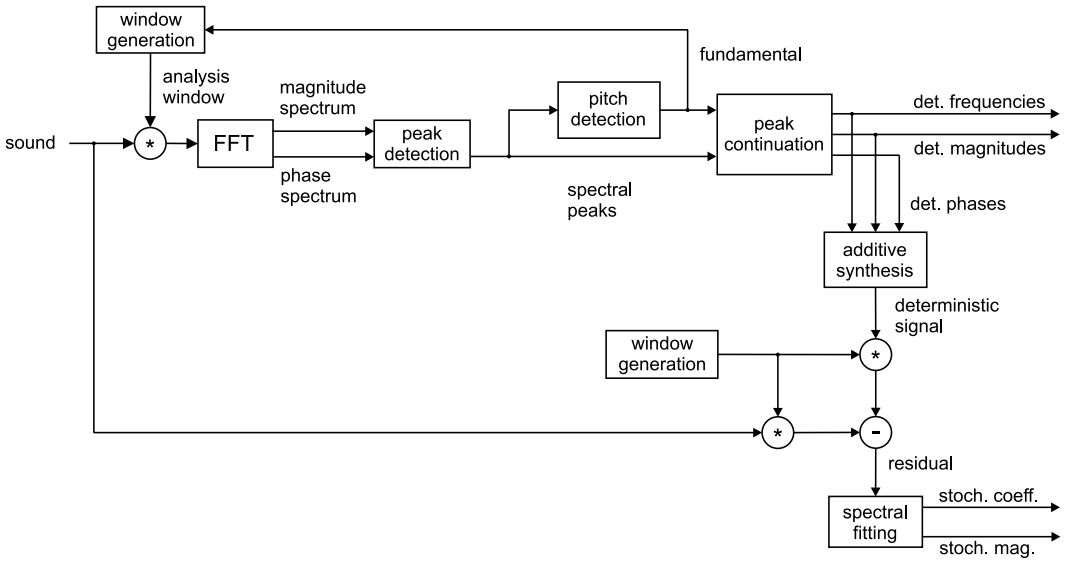

FIGURE 3.3: SMS-Analysis (from: [77]).

After the peak detection, the fundamental pitch is extracted to adapt the analysis window-size, and the peak continuation finds sinusoidal trajectories. The result is a set of parameters: the deterministic magnitudes, frequencies and phases.

Stochastic analysis

The residual is calculated by resynthesizing and windowing the deterministic component and subtracting it from the windowed original signal in time-domain. After applying a FFT to the remaining signal, it can be represented as the shape of a filter by methods like:

- spline interpolation
- the method of least squares
- straight line approximations
- Linear Prediction Coding LPC

An advantage of this system is that the synthesis is completely independent from the analysis. So the STFT parameters (window-size, window-type, FFT-size and frame-rate) can be chosen as to accomplish best performance.

3.3.2 Modification of the analysis data

The result of the analysis is a set of amplitude and frequency functions as deterministic representation and time-varying filter envelopes as stochastic representation. This amount

of data can be modified in many different ways. Time- or pitch-scale modification can be performed as well as impressive transformations like “morphing” or “hybridization” [76].

3.3.3 SMS - Synthesis

Figure 3.4 shows the SMS-Synthesis scheme.

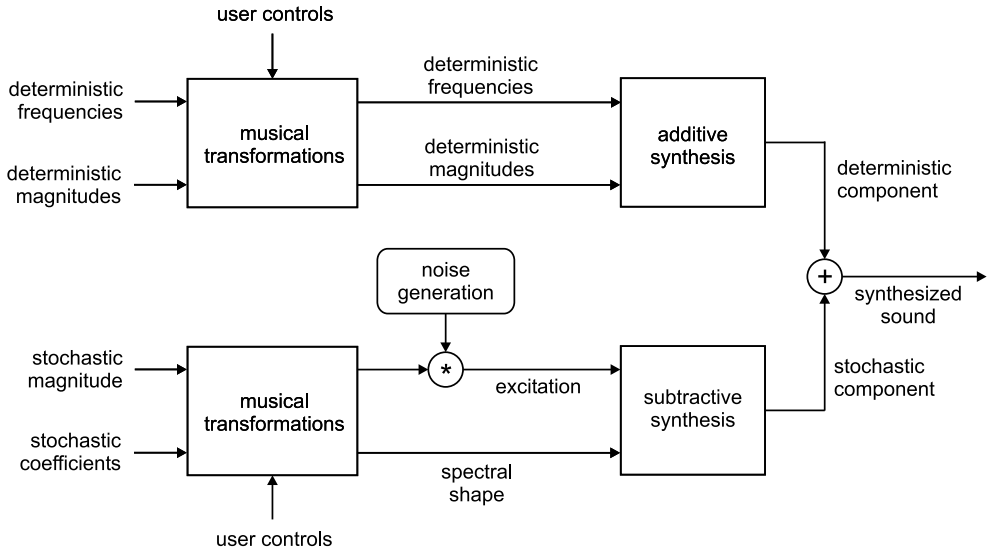

FIGURE 3.4: SMS-Synthesis (from: [77]).

Deterministic synthesis

After potential modifications, the deterministic component is generated by additive synthesis without the phase information, which makes synthesis possible either in time domain, similar to the sinusoidal synthesis, or in frequency domain based on the inverse FFT [78]. The IFFT method is computationally more efficient.

Stochastic synthesis

The synthesis of the stochastic component can be understood as time-varying filtering of white noise, which is generally implemented by the time-domain convolution of white noise with the impulse response of the filter. In practice, an IFFT is applied to a complex spectrum consisting of the spectral shape of the residual and random phases.

Both components are either added in frequency domain, which is more effective saving one IFFT, or in time domain.

3.4 Transient Modeling Synthesis

An interesting extension to the Spectral Modeling Synthesis is taking transients into account, which are not appropriately modeled up to now.

The need for an explicit low-order parametric model for transients, that allows a wide range of modifications and fits well into current sines-, and sines+noise-models, motivated the transient model introduced by Verma [79]. Here, transients are modeled as parameters, so the system is more flexible.

3.4.1 Analysis

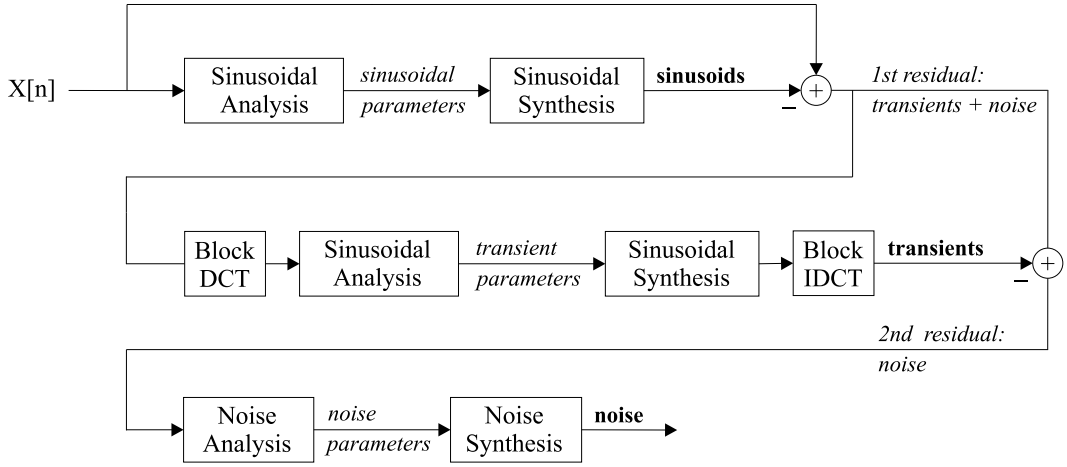

FIGURE 3.5: TMS-Analysis. (From: [79])

As you can see in figure 3.5, at first the parameters of the prominent sinusoids are extracted and the sines are removed from the original, leaving the transients and noise in the 1^{st} residual. Then transients are detected, parameterized and subtracted from this residual resulting in a 2^{nd} residual, which represents the noise component of the signal. At last, the noise parameters are determined.

Since the sinusoids + noise model has been described above, we want to focus on transient modeling here.

3.4.2 Transient modeling

This algorithm makes use of the duality between sinusoids and transients. A slowly-varying sinusoid in time domain is impulsive in frequency domain. This makes it possible to detect it in a STFT - magnitude spectrum. Impulsive signals in time domain are oscillatory in frequency domain. So the first step is to map the time domain transients to sinusoidal signals in some frequency domain. The discrete cosine transform (DCT) provides such a mapping:

Definition of the discrete cosine transform (DCT):

$$C(k) = \beta(k) \sum_{n=0}^{N-1} x(n) cos[\frac{(2n+1)k\pi}{2N}] \tag{3.10}$$

for

$$n, k \in \{0, 1, \ldots, N-1\}$$

with

$\beta(k) = \sqrt{1/N}$ for k = 1
$\beta(k) = \sqrt{2/N}$ otherwise (see [80])

We can say that an impulse at the beginning of a frame results in a low-frequency cosine, wheras an impulse occuring at the end of the frame yields a high-frequency cosine. If the frequency of a sinusoid corresponding to a transient is changed, the onset of the time domain impulse moves within the frame.
The DCT - frequency domain representation is well suited for sinusoidal modeling, which is used for parameter extraction of the transients.

Algorithm:

- Take non-overlapping blocks of the input signal
- Perform a DCT on each block
- Extract transient parameters applying sinusoidal modeling on each DCT-frame

3.4.3 Synthesis

Reconstructing the DCT domain sinusoids and using an Inverse Discrete Cosine Transform (IDCT) synthesizes the potentially modified transient in time domain.
The synthesis stage of Transient Modeling is shown in figure 3.6.

3.4.4 Time-scale modification

When time-scaling a signal, in transient modeling the onset of the actual transient is very important. Therefore, the time-scale of the DCT-sinusoid has to be modified by the same factor as the time domain sines and noise.
Further References: [83], [84]

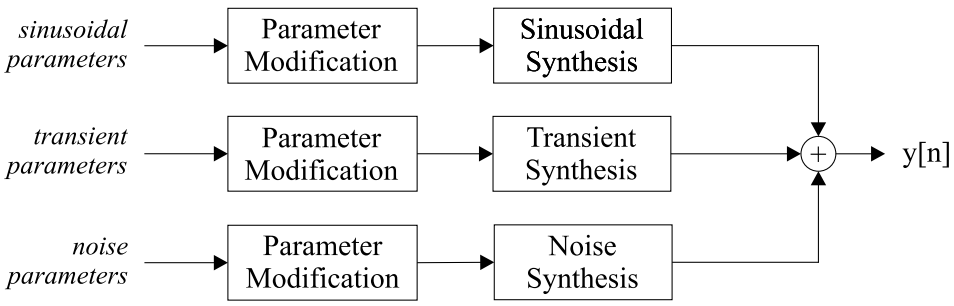

FIGURE 3.6: TMS-Synthesis. (From: [79])

Chapter 4

Characteristics of the phase vocoder parameters

4.1 Introduction

This chapter explores the characteristics of the phase vocoder parameters concerning the ability to extract deterministic components/peaks out of them.

Since we want to time-stretch sinusoidal sounds as well as noise, important constraints for deterministic component detection are to prevent peak detections in pure noise and to have a residual free of deterministic components when analysing a pure sinusoidal sound. These constraints are not easy to accomplish.

Therefore, we want to introduce criteria with regard to the parameters resulting from phase vocoder analysis:

- Magnitude spectrum
- Phase spectrum
- Instantaneous frequency[1]

The conditions figured out for each parameter form the basis for the MatLab^{TM}- implementation of a deterministic component detection, presented in section 5.2.

4.2 The magnitude spectrum

A deterministic peak detection should not only search for local maxima in the magnitude spectrum but verify certain conditions in terms of the magnitude relations of the windows' main-lobe.

We can distinguish three positions of a sinusoid in a magnitude spectrum regarding its frequency:

[1] Calculated from the phase spectra of two consecutive frames

1. A sinusoid exactly at a frequency bin
2. A sinusoid exactly between two bins
3. A sinusoid somewhere in between the upper positions

The first two cases will be explored in detail since they result in extreme magnitude relations of the corresponding main-lobe.
Figures 4.1 and 4.2 show the shapes of these peak-frequency positions while the magnitude relations around the peak are presented in tables 4.1 and 4.2.

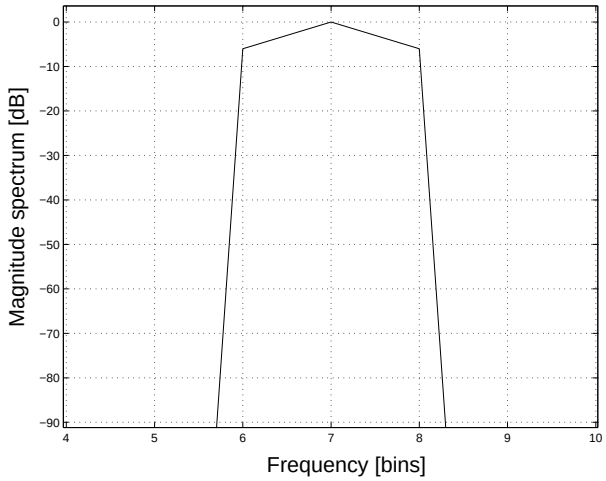

FIGURE 4.1: A sinusoidal peak, exactly at a frequency-bin (“sharp” shape)

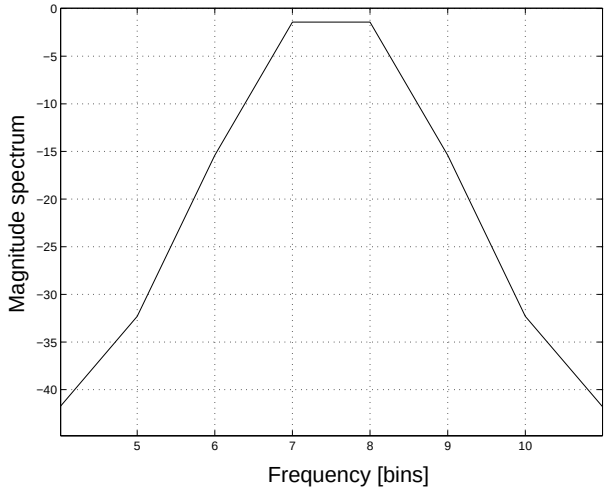

FIGURE 4.2: A sinusoidal peak, exactly between two frequency-bins (“flat” shape).

Reasonable conditions for “enhanced” peak detection would be:

Freq. bin	Mag.value [dB]
5	$-\infty$
6	-6.00
7	0.00
8	-6.00
9	$-\infty$

TABLE 4.1: Magnitude relations for a sinusoidal peak, corresponding to Figure 4.1.

Freq.bin	Mag.value [dB]
5	-32.31
6	-15.40
7	-1.42
8	-1.42
9	-15.40
10	-32.31

TABLE 4.2: Magnitude relations for a sinusoidal peak, corresponding to Figure 4.2.

- The magnitude values of the local minima around the peak (within a certain range) may not exceed a defined level below the peak magnitude.

$$\mid X(k_{p,lmin}) \mid = min \mid X(k) \mid_{k_p-\delta}^{k_p-1} \tag{4.1}$$

$$\mid X(k_{p,umin}) \mid = min \mid X(k) \mid_{k_p+1}^{k_p+\delta} \tag{4.2}$$

k_p ..... location of the actual peak p
$\mid X(k) \mid$ ... magnitude spectrum of the actual frame
$k_{p,lmin}$ .... location of the lower local minimum of peak p
$k_{p,umin}$ ... location of the upper local minimum of peak p
δ ..... range around the peak (bins)

$$\mid X(k_{p,lmin}) \mid < \mid X(k_p) \mid - M_{th} \tag{4.3}$$

$$\mid X(k_{p,umin}) \mid < \mid X(k_p) \mid - M_{th} \tag{4.4}$$

M_{th} ... Necessary minimum difference between the peak magnitude and the local minima for the peak to be detected.

The value for M_{th} has been chosen to be 14 dB, which is the theoretical maximum magnitude value of $\mid X(k_{p\pm2}) \mid$ (table 4.2).

- The magnitudes of the four nearest neighbor-bins of a peak must be lower than the peak magnitude:

$$\mid X_{k_p-2} \mid < \mid X_{k_p-1} \mid < \mid X_p \mid > \mid X_{k_p+1} \mid > \mid X_{k_p+2} \mid \tag{4.5}$$

  This can be verified by the checking the distance of the local minima to the peak location:

$$k_p - k_{p,lmin} >= 2 \tag{4.6}$$

  and

$$k_{p,umin} - k_p >= 2 \tag{4.7}$$

These conditions make it possible to also detect peaks caused by chirp signals, since this kind of signals use to have a broader main-lobe width than pure sinusoids.

4.3 The phase spectrum

Centering the windowed samples around the origin results in a constant-phase spectrum. This is called “zero-phase windowing”. In a constant-phase spectrum, the phases of deterministic peaks are equal (figure 4.3), which could be used as a criterion for peak-detection.

Detection conditions can now be stated by letting ζ be a range within the phase values may vary around a peak bin Ω_{k_p}:

$$\mid \angle X(t_a^u, \Omega_k) - \angle X(t_a^u, \Omega_{k_p}) \mid < \zeta \tag{4.8}$$

$$for \qquad k = k_p \pm 1$$

During the test phase it came out that the phase condition is not consistent, since deterministic peaks were found in *noise*, in which the phase values should be random. These deterministic peaks are caused by the inconsistency of the approximation method.

4.4 The instantaneous frequency

The instantaneous frequencies are derived from the phases of two consecutive frames (see section 2.3 for details).

$$\hat{\omega} = \Omega + \frac{\Delta_p \Phi_k^u}{R_a} \tag{4.9}$$

When slowly varying sinusoids are analysed, we can state that the instantaneous frequency values of those sinusoids have to be almost the same over the successive frames. Therefore,

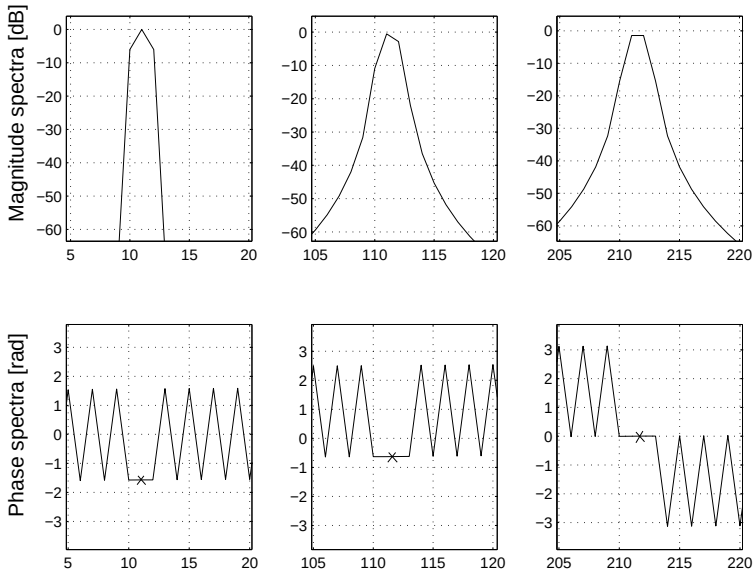

FIGURE 4.3: Magnitude and Phase spectra of a sine at different frequencies.

$\hat{\omega}$ ..... Instantaneous frequencies
Ω ..... Bin - frequencies
$\Delta_p \Phi_k^u$ ... Principal determination of the heterodyned phase increment
R_a ..... Analysis hop size

the subtraction of the instantaneous frequencies of two consecutive frames must result in values near zero for the peak bins. Figure 4.4 shows such a subtraction for a "sawtooth + noise"-like signal. The important regions of the subtraction result are marked with arrows.

As we can see on the plot, the values of the peak-bins are near zero. In this way, a detection condition could be introduced, which forces delta-values of instantaneous frequencies of peak-bins to lie within a certain threshold level ξ.

$$\hat{\omega}(t_a^u) - \hat{\omega}(t_a^{u-1}) < \xi \tag{4.10}$$

for values of $\hat{\omega}$ at peak-bins $\pm$ 1 bin

$\hat{\omega}(t_a^u)$ .... Instantaneous frequencies at analysis frame u
$\hat{\omega}(t_a^{u-1})$ ... Instantaneous frequencies at analysis frame $u-1$ (previous frame)

According to what was said about the phase criterion, short-time sinusoids hidden in noise unfortunately effect inconsistency of this condition by recognizing deterministic peaks in pure noise.

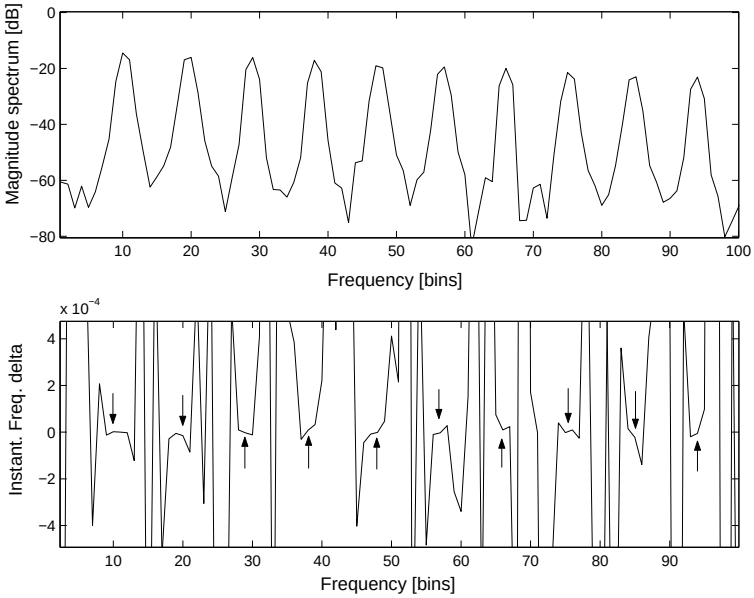

FIGURE 4.4: Delta of the instantaneous frequencies for a "sawtooth + noise"-like signal.

In the next chapter, a system is introduced as an application of some of the findings we have presented up to now.

Chapter 5

The Analysis/Transformation/Resynthesis System

5.1 Introduction

The aim of the system developed is to ensure good time-expansion of sinusoids and noise. The basic idea is to determine deterministic and stochastic bins of short-time spectra and to treat them in different ways when time-expanding a signal.

5.2 MatLab^{TM}-function *detanalysis*

The magnitude criteria described in section 4.2 have been implemented in a MatLab^{TM}-function [92]. Figure 5.1 shows an outline of the implementation.

5.2.1 Parameters

```
Usage: [detpeaklocs,roi]=detanalysis(Xm)

Xm.............Logarithmic magnitude spectrum
detpeaklocs...locations of the detected deterministic peaks
roi...........number of bins for the region of influence
```

The function is fed with the logarithmic magnitude spectrum of the actual frame. Internally a threshold level and a maximum number of peaks are set for the basic peak picking algorithm [93] ("peak-picker" - figure 5.2). In return we get the locations of the detected deterministic bins and the numbers of bins indicating the region of influence of each peak.

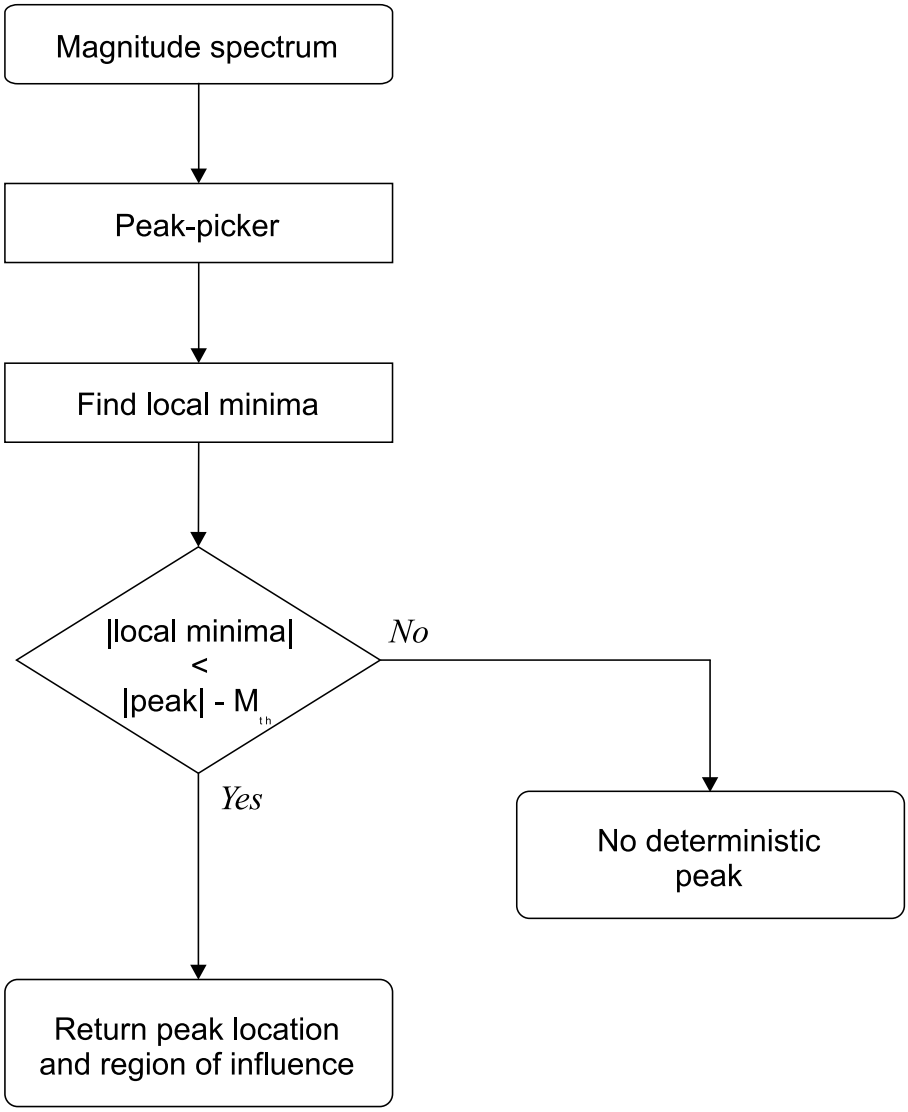

FIGURE 5.1: MatLab^{TM}-function *detanalysis*.

Peak-picking parameters:
NTH=40dB .... threshold level
maxpeaks=50 ... max. number of peaks returned by the peakpicker

5.2.2 Peak detection scheme

At the beginning of the "enhanced" peak detection method, the peak-picker finds the local maxima within the magnitude spectrum and picks a certain number of peaks lying within a defined range below the maximum peak.

After picking the magnitude peaks, the following calculations are applied to every peak:

- Find the local minima within a defined range around the peak
- Verify, if the magnitudes of the local minima lie below a certain level

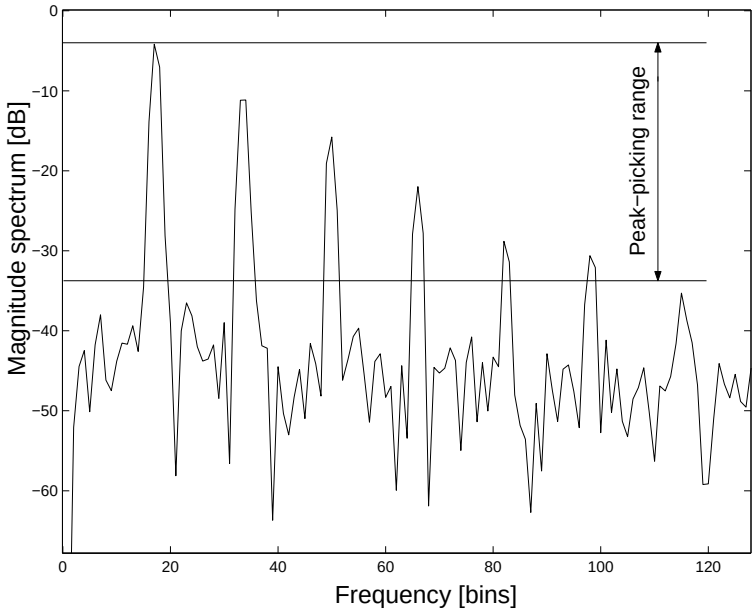

FIGURE 5.2: Picking peaks in the Magnitude spectrum within a certain peak-picking range.

- If so, save the peak location, and the distances of the local minima (as the region of influence)

Eventually, the peak locations and the corresponding regions of influence are returned.

In the next section we will introduce an analysis/transformation/resynthesis system which makes use of this algorithm.

5.3 The Analysis/Transformation/Resynthesis System

The *detanalysis*-function, presented in the previous section is embedded into a phase vocoder framework, which means that the analysis and resynthesis stages are based on the short-time Fourier transform described in chapter 2. The outline of the system is shown in figure 5.3.

5.3.1 Analysis

As a result of the short-time Fourier transformation (see section 2.2.1), we obtain the magnitude and phase spectrum of a windowed frame of the sound in examination.

In this implementation we used the following parameters:

FFT-size ..... 2048 at f_s=44,100 Hz
Window-size ..... FFT-size
Analysis hopsize ... $\lfloor \frac{window-size/4}{time-scalingfactor} \rfloor$

The equation for the analysis hopsize results from the fact, that for time-scaling, the analysis hopsize equals the synthesis hopsize divided by the time-scaling factor, while the synthesis hopsize is kept constant $\frac{windowsize}{4}$ (4x-overlap).

Note that in this implementation time-scale factors are limited to values of $\frac{synthesishopsize}{analysishopsize}$. Since the analysis hopsize is rounded off to an integer number of samples, the number of large time-scale factors is very restricted. A remedy for this problem would be an alteration of the hopsize.

5.3.2 Calculation of a reduced variance spectrum for peak-detection

To reduce the variance of the appoximated spectrum of an actual frame for the peak detection, an average magnitude spectrum is calculated including the actual frame spectrum and weighted spectra of a frame about 11 ms in advance, and the (averaged) spectrum of the frame about 11 ms before the actual frame.

The reduced variance spectrum is calculated as follows:

$$\mid X_{rv}^{u} \mid = \sqrt{\frac{a \mid X_{rv}^{u-v} \mid^2 + b \mid X^{u} \mid^2 + c \mid X^{u+v} \mid^2}{a+b+c}} \tag{5.1}$$

$\mid X_{rv}^{u} \mid$ .... actual reduced variance spectrum
$\mid X_{rv}^{u-v} \mid$ ... reduced variance spectrum v frames before the actual frame
$\mid X_{a}^{u} \mid$ .... actual analysis magnitude spectrum
$\mid X_{a}^{u+v} \mid$ ... analysis magnitude spectrum, v frames in advance
a,b,c ..... weighting factors
v ..... number of frames corresponding to about 11 ms

Setting the weighting factors a=c=0, and b=1 results in a “normal” variance, resulting from the approximation method, while a=b=c=1 minimizes the variance. In this implementation we have chosen the following values: a=c=0.5, and b=1.

Since for this calculation we need one spectrum in advance and another from the past, a buffer is introduced which carries the magnitude spectra of the v next frames and the reduced variance spectra of the v previous frames. Using the reduced variance spectra as previous ones instead of the actual magnitude spectra on the one hand further reduces variance, while on the other hand it reduces the time-resolution.

5.3.3 *detanalysis* and deterministic/stochastic seperation

The deterministic peak locations and the region of influence for each peak allow to distinguish the following parameters:

- Deterministic magnitudes
- Deterministic phases
- Stochastic magnitudes

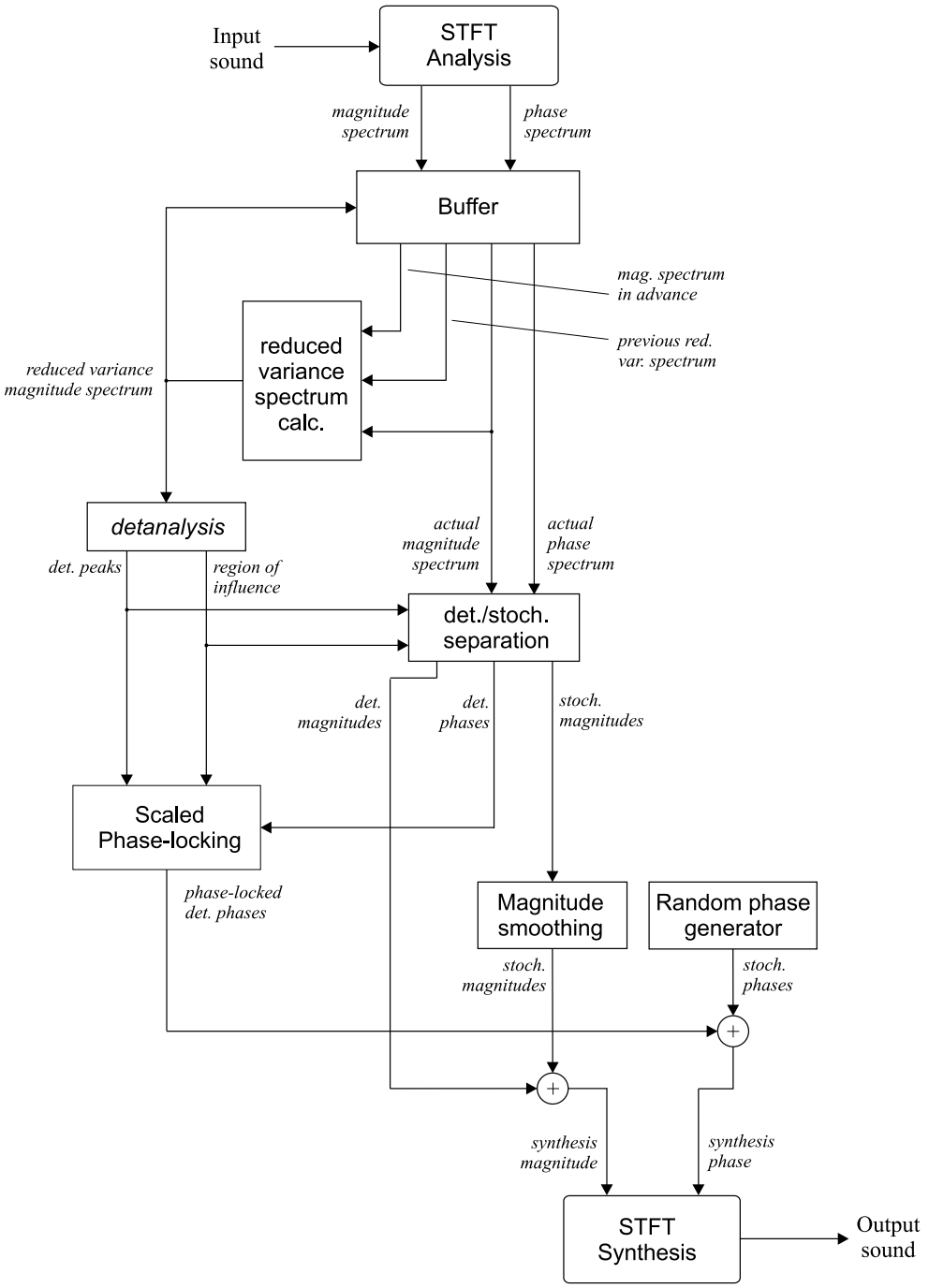

FIGURE 5.3: The Analysis/Transformation/Resynthesis system.

5.3.3.1 Deterministic magnitudes and phases

The deterministic magnitudes and phases are simply the magnitude and phase values at the peak-bins resulting from the *detanalysis*-function, plus the bins of the region of influence, which depends on the main-lobe shape.

5.3.3.2 Stochastic magnitudes

The stochastic magnitudes are calculated from the actual magnitude spectrum by setting the deterministic magnitude values to the lower value of the outer peak-bin neighbors and smoothing the resulting spectrum. As a smoothing method, 4-sample moving average filtering was chosen.

A plot of the deterministic and stochastic magnitudes of a “sawtooth + noise”-like sound (Appendix B) is shown in figure 5.4.

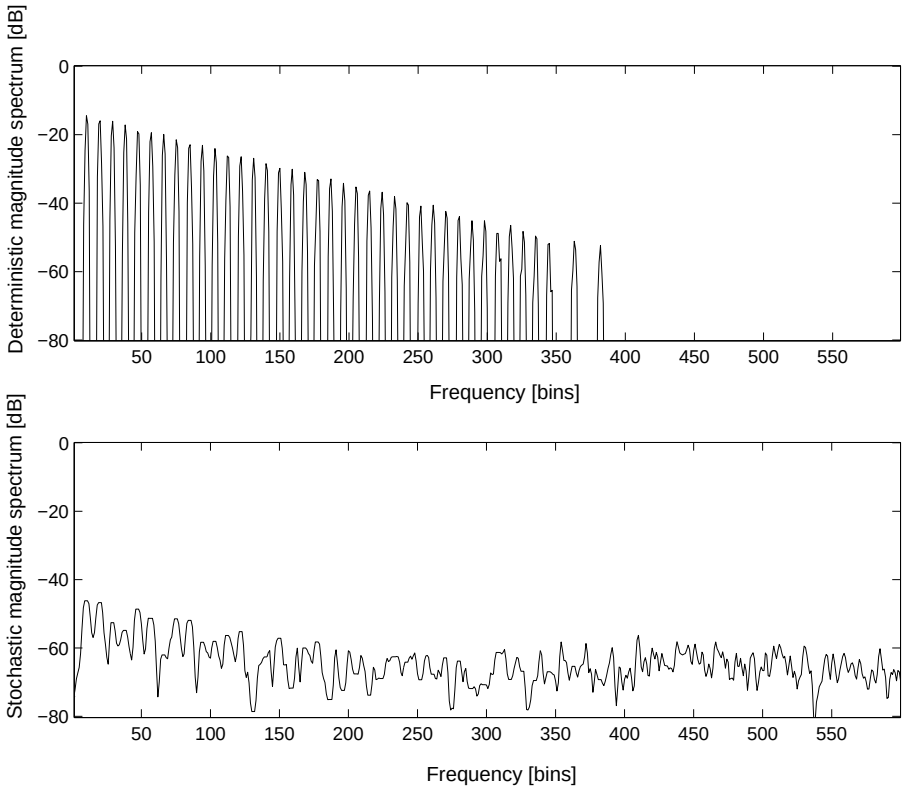

FIGURE 5.4: Deterministic and smoothed stochastic magnitude spectra (“sawtooth + noise”-like sound)

5.3.4 Resynthesis

5.3.4.1 Resynthesis magnitude spectrum

The resynthesis magnitude spectrum is a combination of two spectra: the stochastic magnitudes plus the magnitudes of the deterministic components. In this way, we maintain the original magnitudes of the sinusoids, while smoothing the stochastic signal component.

5.3.4.2 Synthesis phase spectrum

As soon as the phases have been determined, scaled phase-locking is applied to them in order to preserve vertical phase coherence. This algorithm contains two basic calculations:

- Recognition of peaks moving from one bin to another. In our system-scheme, the *peak switch detection*-block corresponds to this important feature of scaled phase-locking: The actual peak locations are compared with the ones of the previous frame. If a peak switches to a neighboring bin, the synthesis phase and phase increment of the old peak is used for deriving the synthesis phase of the actual peak (see equation 2.12).
- The phase relations of the bins around the peak in the analysis frame are preserved in the synthesis frame within the region of influence.

The *stochastic phases* are derived by a random phase generator and masked by the deterministic phases to obtain the synthesis phase spectrum.

5.3.4.3 Inverse short-time Fourier transformation

At last, the output signal is generated from the resulting magnitude and phase spectra by an IFFT/overlap-add synthesis.

Chapter 6

Strategies for transient detection

6.1 Introduction

Four algorithms for detection and location of abrupt time/spectral changes in a signal have been implemented and tested in MatLab^{TM}. Masri [81] proposed three methods which are based on spectral energy distribution, the attack envelope and spectral dissimilarity. Another algorithm uses the energy distribution in time domain. Two further methods to detect fast changes are mentioned.

6.2 Detection based on energy distribution

MatLab Implementation: *trendist.m*
This algorithm calculates the energy and high frequency content (HFC) of each frame.

$$E = \sum \mid X(k) \mid^2 \tag{6.1}$$

$$HFC = \sum \mid X(k) \mid^2 k \tag{6.2}$$

E . . . . . energy function for the current frame
HFC . . high frequency content
(weighted energy function, linearly biased toward the higher frequencies)
X(k). . . FFT-Magnitude (k...bin index)

Relating those values results in a measure of transience MoT:

$$MoT_r = \frac{HFC_r}{HFC_{r-1}} \frac{HFC_r}{E_r} \tag{6.3}$$

subscript r denotes current frame subscript r-1 denotes the previous frame

A transient is detected if

$$MoT_r > T_{D,ED} \tag{6.4}$$

$T_{D,ED}$ ... Threshold for detection using the energy distribution method

Parameters:

fftsize = 128
no overlap
threshold = 1

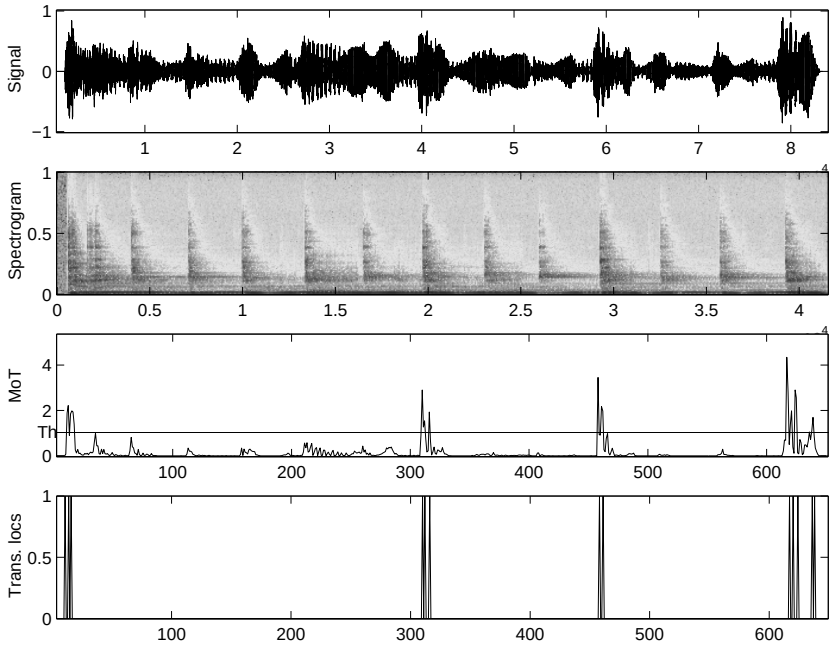

FIGURE 6.1: Detection based on energy distribution applied on a berimbao sound.

6.3 Detection by attack envelope

MatLab Implementation: *trattenv.m*
Block-based calculation of the signal envelope makes it possible to detect sudden rises within the signal. At first, the maximum values of consecutive, non overlapping blocks are evaluated:

$$y(n) = max_{m=0}^{m=M-1}\{ \mid x(nT+m) \mid \} \tag{6.5}$$

The peak follower will be updated, if the new sample-block value is higher than the preceding one multiplied by a factor of decay:

$$P(n) = max\{y(n), P(n-1) \times k_{decay}\} \tag{6.6}$$

x(T) .. sample sequence
y(n) ... max. value of the block
m .... samples within each block
M .... block size

k_{decay} ... decay factor

A detection is registered when the relation of the actual peak follow value to the last one exceeds a threshold $T_{D,AE}$ (detection threshold for the attack envelope method).

$$\frac{P(n)}{P(n-1)} > T_{D,AE} \tag{6.7}$$

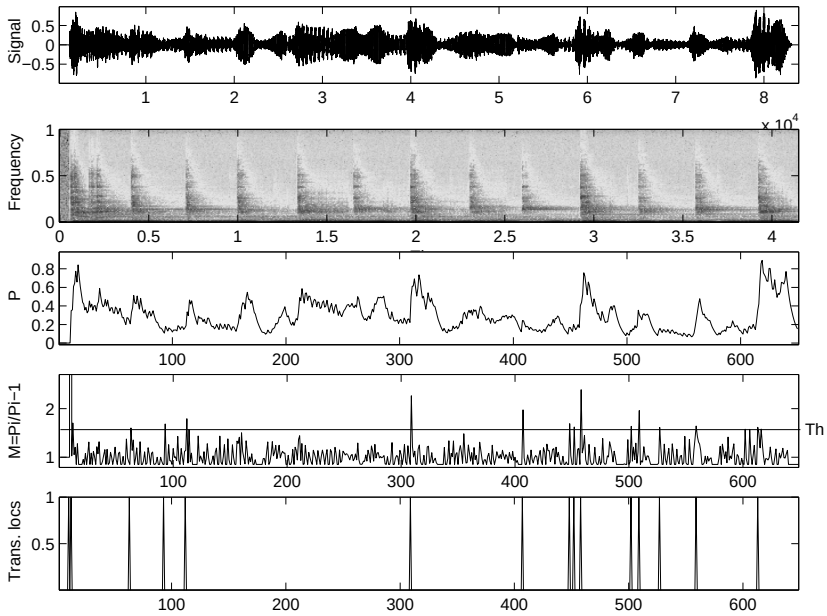

FIGURE 6.2: Detection based on the attack envelope applied on a berimbao sound.

Parameters:

block length = 128
decay = 0.85
threshold = 1.6

6.4 Detection by spectral dissimilarity

MatLab Implementation: *trspecdis.m*
A sudden change in energy and a sudden change of spectral content can be detected by

comparing the spectra of neighboring frames.

$$D_r = \sum \frac{|\ |X_r(k)| - |X_{r-2}(k)|\ |}{E_{r-2}} \tag{6.8}$$

D_r ..... dissimilarity function for frame r
$X_r(k)$ ... kth bin of FFT frame r
E_r ..... energy function (6.1)

Alternatively:

$$D_r = \sum \frac{|\ |X_r(k)|^2 - |X_{r-2}(k)|^2\ |}{|X_{r-2}(k)|^2} \tag{6.9}$$

Condition for detection:

$$D_r > T_{D,SD} \tag{6.10}$$

$T_{D,SD}$ ... threshold for detection using the spectral dissimilarity method

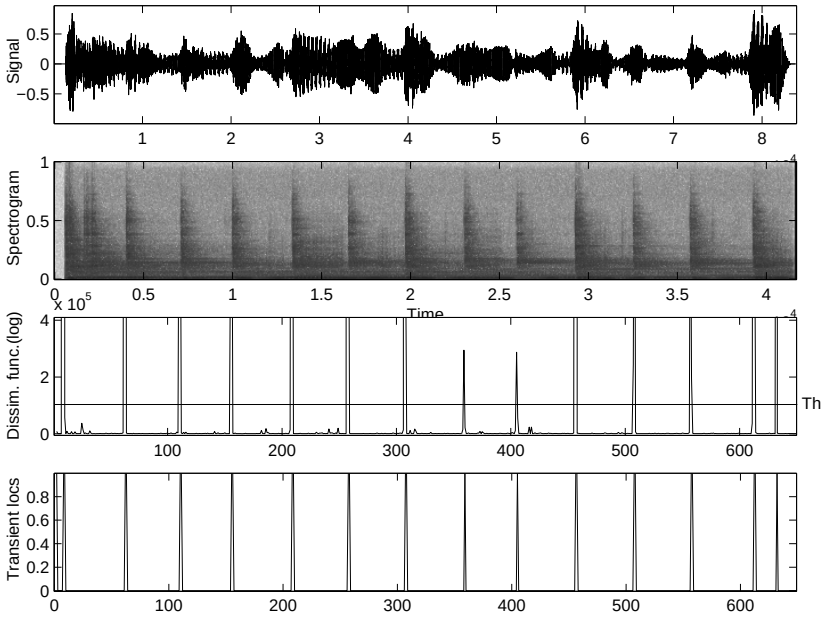

FIGURE 6.3: The spectral dissimilarity method applied on the Berimbao sound.

Parameters:

fftsize = 256
2x overlap
threshold = 100.000

6.5 Detection by energy relations in the time domain

MatLab Implementation: *trentime.m*
Similar to the energy distribution method which deals with energy relations in the frequency domain, we introduce a time domain algorithm which relates the high frequency content, calculated by high pass filtering the input signal, to the total energy of the signal. Therefore, the signal is divided into non-overlapping blocks. Appropriate filter coefficients and block size have to be found.

A block of 128 Samples s is to be filtered, squared and summed up to get a value for the high frequency content:

$$hfe = \sum s_f^2 \tag{6.11}$$

hfe ... value for high frequency energy
s_f^2 ... filtered and squared block samples

$$E = \sum s^2 \tag{6.12}$$

Now we relate the high frequency energy to the total energy E

$$\gamma = \frac{hfe}{E} \tag{6.13}$$

γ ... HFC to Energy measure
In order to find positive slopes we differentiate γ and eliminate the negative values (set to zero).

$$\gamma' = \frac{d\gamma}{dt} \tag{6.14}$$

with

$$\gamma' = 0 \qquad \forall\, \gamma' < 0$$

All values for γ' beyond a certain threshold are registered as transients:

$$\gamma' > T_{D,ERTD} \tag{6.15}$$

$T_{D,ERTD}$ ... Threshold for detection using the spectral dissimilarity method
Parameters:

Blocksize = 128
No overlap
$T_{D,ERTD}$ = 0.1
Filter coefficients:

A = [1 -1.1098 0.3678]
B = [0.9026 -1.1753 0.3981]

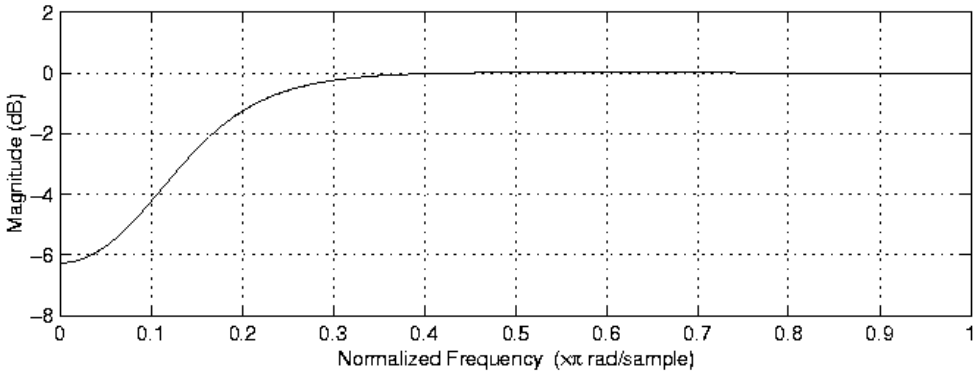

FIGURE 6.4: Magnitude response of the HP-filter.

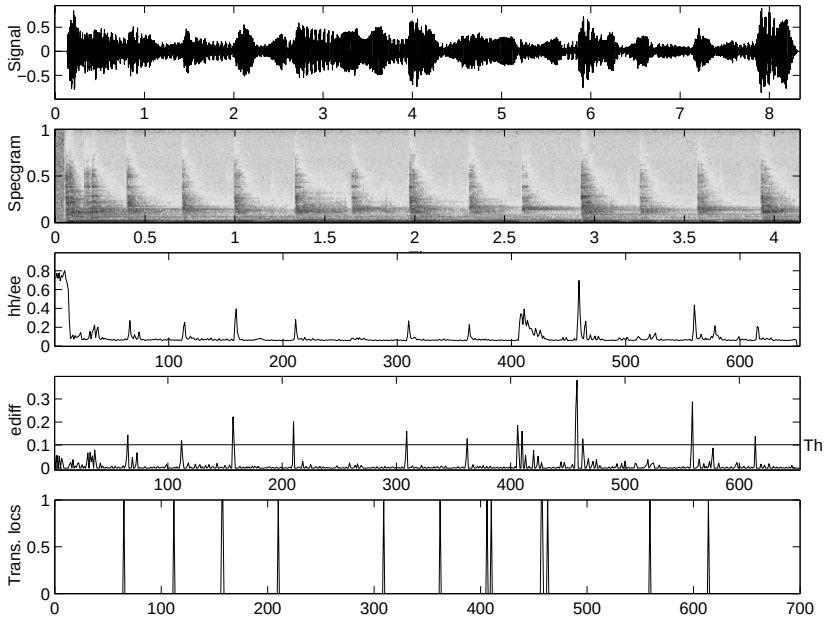

FIGURE 6.5: The time domain energy relations method applied on the Berimbao sound.

6.6 Other methods

6.6.1 Constant-Q analysis

Based on a Constant-Q Analysis [96], the bonk~-object was developed for PD and MAX/MSP by Miller Puckette et al. [99]. It is able to detect sharp relative changes in the spectrum without any accompanying large change in the overall power. The spectrum is divided into 11 constant-Q filters, of which the outputs represent spectral energies. A growth function g relates the change of power of consecutive frames for each channel. The sum of the growth estimates of all channels is compared with a threshold. If the total growth exceeds the threshold, an attack is reported. Bonk~ offers the possibility to compare a new attack with a set of

pre-recorded attacks in order to guess which of the possible instruments was responsible for it. After a certain learning phase (storing templates) bonk~ can identify some sorts of instruments

6.6.2 Detection of fast changes

This approach [100] observes bank filter energies, Mel cepstrum coefficients and its deltas. These values are combined following several rules. The main idea is to find points of maximum increasing slope.

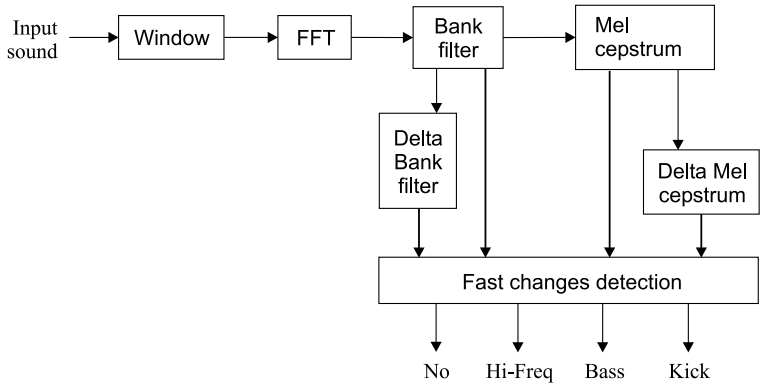

FIGURE 6.6: Fast changes detection. (From: [100])

6.7 Choosing a transient detection method

The spectral dissimilarity method has been found to be an appropriate way to detect transients with rather low computation time. This method makes it possible to register fast spectral changes which do not go along with fast changes in time domain.

6.8 Embedding transient detection into a frequency domain Analysis/Resynthesis system

Figure 6.7 shows how transient detection can be included in the system. Before processing an analysis frame, a certain time span at the end of the frame is scanned for transients using the spectral dissimilarity method described above. If a transient occurs, the analysis hopsize is set equal to the synthesis hopsize and the following four successive frames are directly taken over for resynthesis, taking care of appropriate computation of the output phase.

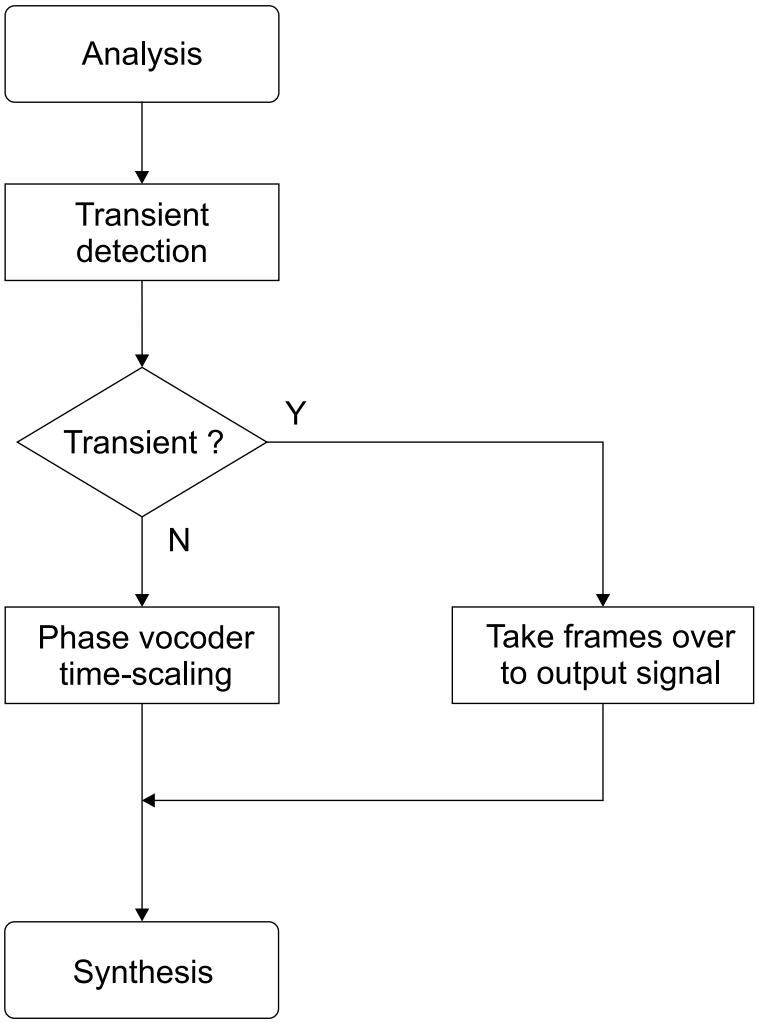

FIGURE 6.7: Transient detection integration.

Chapter 7

Results

7.1 Implementation

The following implementations[1] have been tested on various sounds, which are documented in Appendix B:

- **Standard phase vocoder.** The implementation of the standard phase vocoder was adapted from [38]. The phase vocoder analysis-function calculates the magnitudes and phases of the signal frames of the entire signal and stores the data within two vectors. This kind of implementation requires a lot of memory space for time-expansion of long samples. On this account, the analysis/synthesis functions have been combined and edited for frame-by-frame computation.
- **Phase-locked vocoder using "scaled phase-locking".** Based on the improvements presented by Laroche and Dolson (see section 2.4), the standard phase vocoder program has been upgraded by the scaled phase-locking algorithm. In this implementation, the region of influence has been set to the channel of lowest magnitude between two peaks.
- **The deterministic/stochastic phase vocoder system.** The extensions featured by this system are described in section 5.3.
- **The d/s-system including transient detection.** The transient detection add-on was applied only to some special samples (Appendix B.2).

7.2 General statements

For very large modification factors, the phasiness has been further reduced by deterministic/stochastic separation in comparison to scaled phase-locking. Smoothing stochastic magnitudes and randomizing stochastic phases seems to "relax the tension" resulting from phase-locking the entire spectrum.

[1] see appendix A for information on the MatLab^{TM}-scripts

Especially in harmonic sounds, higher partials are not detected, which results in a more dull timbre. A main reason for failing detection are the fixed conditions within the enhanced peak-detection (*detanalysis*). Low magnitude lobe-shapes require a lower peak-picking threshold NTH, and thus more accurate conditions to get recognized. Detection misses result in a degradation of the quality of the perceived sound.

Observing the computation time of the different implementations, it is clear that the algorithms which include peak-detection take much longer calculation times than the standard phase vocoder.

7.3 Timbre-dependent alterations

- **Harmonic sounds.** As we already mentioned above, the timbre of this kind of sounds changes. For large scaling factors, slowly varying frequencies result in beats when modified by the standard phase vocoder. This artefact can be prevented by phase-locking in the most cases.

- **Noise.** As the analysis of white noise shows, spectral impulses cause short-term “partials” due to the inconsistency of the approximation method. The reduced variance mechanism helps suppressing such spurious peaks in the peak-picking stage.

- **Transients.** The well-known “transient smearing” comes up for the same reason as the spurious peaks resulting from noise. Very steep attacks get weakened. Some sound examples illustrate the *transient detection extension*. As you can hear in the samples, the natural time evolution of the sound toddles due to the variation of the frame rate in the analysis during the transient period. Thence it is destroyed. A further drawback of this method is that the decay envelope of the transients cannot be modeled.

Chapter 8

Summary & Outlook

8.1 Summary

In this thesis, an enhanced phase vocoder system for time-scale modification was presented, which is based on the separation of the deterministic and stochastic components of a sound signal. As the heart of the system, a MatLab^{TM}-function identifies sinusoids by screening reduced variance magnitude spectra, so deterministic and stochastic magnitudes and phases can be distinguished. While the phases of the deterministic main-lobes are locked to the peak-bin phase, the stochastic phases are set to random numbers. In addition, the stochastic magnitudes are slightly smoothed in order to obtain better results when modifying noisy sounds. All these features have been implemented in a MatLab^{TM}-script.

8.2 Outlook

As an outlook, some improvements are proposed, which could be the basis for further work on this matter:

- Improving the detection algorithm to avoid missing partials.
- Thus, extending the detection function by distinguishing peaks of high and low magnitudes (related to the neighboring magnitudes) and verifying them using different parameter sets.
- Optimizing the script in terms of computational efficency and implementing the system in a C/C++ programming environment.
- Including a kind of peak continuation system. For a peak which obviously belongs to a partial track, the detection constraints could be relaxed.
- The relative phase delay representation of quasi-harmonic signals (section 3.2.3) could ensure phase consistency for either new partials, or partials, of which the phase consistency got lost for whatever reason.

- Within the transient region, other time-stretching factors should be used to warrant a minimum of amplitude decay preservation. We propose to scale the transient frames following an e^{-t}-curve.
- The range of time-scaling factors could be extended by alternating the hopsize.

Appendix A

Software

For each of the implementations, a ‘stand-alone’-script and a function has been created. The functions allow easy use of the algorithms by just setting the input parameters as documented below. Another advantage of the functions is that a series of sound examples can comfortably be generated by a script containing function calls with different input parameters.

A.1 Standard phase vocoder

```
Script:    pvoc1_std.m
Function:  pvoc1_std_function(infile,outfile,factor)

Input parameters:
 infile....input filename
 outfile...output filename
 factor....time-scaling factor
```

A.2 Phase-locked vocoder

```
Script:    pvoc2_lock.m
Function:  pvoc2_lock_function(infile,outfile,factor)

Input parameters:
 infile....input filename
 outfile...output filename
 factor....time-scaling factor
```

A.3 d/s phase vocoder

```
Script:    pvoc3_ds.m
Function:  pvoc3_ds_function(infile,outfile,factor)
```

```
Input parameters:
 infile....input filename
 outfile...output filename
 factor....time-scaling factor
```

A.3.1 Extension to the d/s-pv: Transient detection

```
Script:    pvoc4_td.m
Function:  pvoc4_td_function(infile,outfile,factor)

Input parameters:
 infile....input filename
 outfile...output filename
 factor....time-scaling factor
```

A.4 *detanalysis*

As described in section 5.2, this function returns the deterministic peak locations and regions of influence for a given magnitude spectrum.

```
Function:  [detpeakloc,roi]=detanalysis(Xm)

Input parameters:
 Xm..........Magnitude spectrum (logarithmic)

Output parameters:
 detpeakloc...locations of the detected peaks
 roi.........region of influence for each peak [l,u]
             l...distance to the lower limit
             u...distance to the upper limit
```

Appendix B

Sound examples

B.1 Analysis sounds

The algorithms have been tested on the following sound samples:
From the ICMC2000 Analysis/Synthesis comparison session:
(`http://cnmat.cnmat.berkeley.edu/SDIF/ICMC2000/sounds.html`)

- Harmonic monophonic phrase: Clarinet phrase "deplus". *"Harris_trim"* (James Beauchamp)
- Harmonic monophonic phrase on a polyphonic instrument: Acoustic guitar playing a monophonic line. *"Brokendownengine"* (Adrian Freed)
- Noisy reattacked string: Berimbao. *"Berimbao"* (Xavier Rodet)
- Singing into and playing a flute at the same time. *"Flute_voice"* (Adrian Freed)
- Speech: Shafqat Ali Khan saying "research". *"Research"* (Matt Wright)

From Jean Laroche (private communication):

- Fragment from a Mozart piano piece. *"Mozart"*
- Fragment from Susan Vega's "Tom's Diner". *"Vega"*

Others:

- Average White Band: "Pick up the pieces" (fragment). *"Pick"*
- White noise. *"Whitenoise"*

B.2 Time-scaled sound examples

For each of the sound examples, 15 time-scaled versions have been generated: five using the standard phase vocoder, five using the phase-locked vocoder and five using the deterministic/stochastic time-scaling system.

The sounds have been time-expanded by the following factors:

1.1, 1.5, 2, 4.1, and 10.2

Each example is put together in the following way for each example and each time-scaling factor:

- Original sound
- Sound synthesized using the standard phase vocoder algorithm
- Sound synthesized using the scaled phase-locked vocoder algorithm
- Sound synthesized using the det./stoch. phase vocoder algorithm

For the demonstration of an extremly large time-scale factor, the *"Vega"* sample has been time-expanded by the factor of 256 (which is the maximum limit[1]).

Additionally, the transient detection was applied to "Mozart" and "Pick".

All sound examples are compiled on the accompanying Compact Disc. Table B.1 presents the content of the CD.

[1]This factor corresponds to an analysis hopsize of 1.

Track	Sound example	Factor
01	“Harris trim”	1.1
02	“Harris trim”	1.5
03	“Harris trim”	2.0
04	“Harris trim”	4.1
05	“Harris trim”	10.2
06	“Brokendownengine”	1.1
07	“Brokendownengine”	1.5
08	“Brokendownengine”	2.0
09	“Brokendownengine”	4.1
10	“Brokendownengine”	10.2
11	“Berimbao”	1.1
12	“Berimbao”	1.5
13	“Berimbao”	2.0
14	“Berimbao”	4.1
15	“Berimbao”	10.2
16	“Flute voice”	1.1
17	“Flute voice”	1.5
18	“Flute voice”	2
19	“Flute voice”	4.1
20	“Flute voice”	10.2
21	“Research”	1.1
22	“Research”	1.5
23	“Research”	2.0
24	“Research”	4.1
25	“Research”	10.2
26	“Mozart”	1.1
27	“Mozart”	1.5
28	“Mozart”	2.0
29	“Mozart”	4.1
30	“Mozart” (+ transient detection)	4.1
31	“Mozart”	10.2
32	“Vega”	1.1
33	“Vega”	1.5
34	“Vega”	2.0
35	“Vega”	4.1
36	“Vega”	10.2
37	“Vega” (standard pvoc)	256
38	“Vega” (phase-locking pvoc)	256
39	“Vega” (d/s pvoc)	256
40	“Pick”	1.1
41	“Pick”	1.5
42	“Pick”	2.0
43	“Pick”	4.1
44	“Pick” (+ transient detection)	4.1
45	“Pick”	10.2
46	“Whitenoise”	10.2

TABLE B.1: CD-Tracklist

BIBLIOGRAPHY

The bibliography contains papers which were added for the sake of completeness, and which are not cited in the thesis.

CHAPTER 1. INTRODUCTION

[1] Laroche, J. "Time and pitch scale modification of audio signals". *Applications of digital signal processing to audio and acoustics*, M. Kahrs and K. Brandenburg, eds., Kluwer Academic Publishers, 1998.

[2] Makhoul, J. and El-Jaroudi, A. "Time-scale modification in medium to low rate speech coding". *IEEE Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 1705–1708, 1986.

[3] Fairbanks, G., Everitt, W. and Jaeger, R. "Method for time or frequency compression-expansion of speech". *IEEE Trans. Audio and Electroacoustics*, AU-2, pp. 7–12, 1954.

[4] Lee, F. "Time compression and expansion of speech by the sampling method". *J. Audio Eng. Soc.*, Vol. 20:3, pp. 738–742, 1972.

[5] Scott, R. and Gerber, S. "Pitch-synchronous time-compression of speech". *Proceedings of the Conference for Speech Communication Processing*, pp. 63–65, 1972.

[6] Malah, D. "Time-domain algorithms for harmonic bandwidth reduction and time scaling of speech signals". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-27:2, pp. 113-120, 1979.

[7] Roucos, S. and Wilgus, A. M. "High quality time-scale modification of speech". *IEEE Proc. Int. Conf. Acoust., Speech and Signal Processing*, Tampa, pp. 493–496, 1985.

[8] Moulines, E. and Charpentier, F. "Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones". *Speech communication*, Vol. 9:5/6, pp. 453–467, 1990.

[9] Dattorro, J. "Using digital signal processor chips in a stereo audio time compressor/expander". *Proc. 83rd AES Convention, New York*, preprint 2500 (M-6), 1987.

[10] Roehrig, C. "Time and pitch scaling of audio signals". *Proc. 89th AES Convention, Los Angeles*, preprint 2954 (E-1), 1990.

[11] Laroche, J. “Autocorrelation method for high quality time/pitch scaling”. *IEEE Proc. Workshop Appl. of Signal Processing to Audio and Acoustics*, Mohonk, NY., 1993.

[12] Truax, B. “Discovering inner complexity: Time shifting and transposition with a real-time granulation technique”. *Computer Music Journal*, Vol. 18:2, pp. 38–48, 1994.

[13] Schroeder, M.R., Flanagan, J. and Lundry, E. “Bandwidth compression of speech by analytic-signal rooting”. *Proceedings of the IEEE*, Vol. 55, pp. 396–401, 1967.

[14] Seneff, S. “System to independently modify excitation and/or spectrum of speech waveform without explicit pitch extraction”. *IEEE Trans. Acoust., Speech, and Signal Processing*, Vol. ASSP-24, pp. 358–365, 1982.

[15] Dembo, A. and Malah, D. “Signal synthesis from modified discrete short-time transform”. *IEEE Trans. Acoust., Speech, and Signal Processing*, Vol. ASSP-36:2, pp. 168–181, 1988.

[16] Makhoul, J. “Linear prediction: A tutorial review”. *Proceedings of the IEEE*, Vol. 63:4, pp. 561–580, 1975.

[17] Griffin, D.W. and Lim, J.S. “Multiband-excitation vocoder”. *IEEE Trans. Acoust., Speech, and Signal Processing*, Vol. ASSP-36:2, pp. 236–243, 1988.

[18] Poirot, G., Rodet, X. and Depalle, P. “Diphone sound synthesis based on spectral envelopes and harmonic/noise excitation functions”. *Proceedings of the International Computer Music Conference*, Kln 1988.

[19] Arfib, D. “Analysis, transformation, and resynthesis of musical sounds with the help of a time-frequency representation”. *Representation of musical signals*, de Poli, G., Piccialli, A., and Roads, C., eds., pp. 87–118, M.I.T Press, 1991.

[20] Arfib, D. and Delprat, N. “Musical transformations using the modification of time-frequency images”. *Computer Music Journal*, Vol. 17:2, pp. 66–72, 1993.

[21] Jones, D. and Parks, T. “On the generation and combination of grains for music synthesis”. *Computer Music Journal*, Vol. 12:2, pp. 27–34, 1988.

[22] Moulines, E. and Laroche, J. “Non-parametric techniques for pitch-scale and time-scale modification of speech”. *Speech communication 16*, pp. 175–205, 1995.

[23] Suzuki, R. and Misaki, M. “Time-scale modification of speech signals using cross-correlation functions”. *IEEE Transactions on Consumer Electronics*, Vol. 38:3, pp. 357–363, 1992.

CHAPTER 2. PHASE VOCODER

[24] Rabiner, L.R. and Schafer, R.W. *Digital Processing of Speech Signals.* Prentice-Hall, 1978.

[25] Nawab, S.H. and Quatieri, T.F. “Short-time Fourier Transform”. *Advanced Topics in Signal Processing*, Lim, J.S. and Oppenheim, A.V., eds., pp. 289–337, Prentice Hall, 1988.

[26] Roads, C. *The Computer Music Tutorial.* M.I.T Press, 1996.

[27] Harris, F. J. “On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform”. *Proceedings of the IEEE*, Vol. 66, pp. 51–83, 1978.

[28] Dudley, H. “The Vocoder”. *Bell Labs*, Record 18, pp. 122–126, December 1939. Reprinted in: Schafer, R. W. and Markel, J. D. *Speech Analysis.* IEEE Press, 1979.

[29] Flanagan, J. L. and Golden, R. M. “Phase Vocoder”. *Bell Syst. Tech.*, vol. 45, pp. 1493–1509, Nov. 1966. Reprinted in: Schafer, R. W. and Markel, J. D. *Speech Analysis.* IEEE Press, 1979.

[30] Portnoff, M. R. “Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform”. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-24:3, pp. 243–248, 1976.

[31] Moorer J. A. “The Use of the Phase vocoder in computer Music Applications”. *J. Audio Eng. Soc.*, Vol. 24:9, pp. 717-727, 1978.

[32] Portnoff, M. R. “Time-Frequency Representation of Digital Signals and Systems Based on Short-Time Fourier Transform”. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-28:1, pp. 55–69, February 1980.

[33] Portnoff, M.R. “Time-Scale Modification of Speech Based on Short-Time Fourier Analysis”. *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-29:3, June 1981.

[34] Gordon, J. W. and Strawn, J. “An Introduction to the Phase Vocoder”. *Digital Audio Signal Processing: An Anthology.* John Strawn, ed., Los Altos, CA. W. Kaufmann, Inc., 1985.

[35] Dolson, M. “The Phase Vocoder: A Tutorial”. *Computer Music Journal*, Vol. 10:4, pp. 14–27, Winter 1986.

[36] Moore R.F. “The Phase Vocoder”. *Elements of Computer Music*, Prentice-Hall, 1990.

[37] Serra, M.-H. “An Introduction to the Phase Vocoder”. *Musical Signal Processing*, Roads, C. et al. eds., Swets & Zeitlinger, 1997.

[38] De Goetzen, A., Bernardini, N. and Arfib, D. “Traditional (?) Implementations of a Phase-Vocoder: The tricks of the trade”. *Proceedings DAFX-00*, Verona, pp. 37–43, 2000.

[39] Puckette M. S. “Phase-locked Vocoder”. *Proc. IEEE Conf. on Applications of Signal Processing to Audio and Acoustics*, Mohonk, 1995.

[40] Laroche, J. and Dolson, M. “About this phasiness business”. *Proceedings of the International Computer Music Conference*, 1997.

[41] Laroche, J. and Dolson, M. “Improved Phase Vocoder Time-Scale Modification of Audio”. *IEEE Transactions on Speech and Audio Processing*, Vol. 7:3, pp. 223–232, May 1999.

[42] Puckette, M.S. and Brown, J.C. “Accuracy of Frequency Estimates Using the Phase Vocoder”. *IEEE Transactions on Speech and Audio Processing*, Vol. 6:2, March 1998.

[43] Griffin, D.W. and Lim, J.S. “Signal Estimation from Modified Short-Time Fourier Transform”. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-32:2, pp. 236–243, April 1984.

[44] Depalle, Ph. and Poirot, G. “SVP: A modular system for Analysis, Processing and Synthesis of Sound Signals”. *Proceedings of the International Computer Music Conference*, 1991.

[45] Fischman, R. “The Phase Vocoder: Theory & Practise”. *Organised Sound*, Vol. 2:2, 127–145, 1997.

[46] Ferreira, A.J.S. “An Odd-DFT Based Approach to Time-Scale Expansion of Audio Signals”. *IEEE Transactions on Speech and Audio Processing*, Vol. 7:4, July 1999.

[47] Oppenheim, A.V. and Schafer R.W. *Discrete-Time Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[48] Allen, J.B. “Short term spectral analysis, synthesis, and modification by discrete Fourier transform”. *IEEE Trans. Acoutics, Speech, and Signal Processing*, Vol. ASSP-25, pp. 235–238, June 1977.

[49] Allen, J.B. and Rabiner, L.R. “A Unified Approach to Short-Time Fourier Analysis and Synthesis”. *Proceedings of the IEEE*, Vol. 65:11, pp. 1558–1564, November 1977.

[50] Jaffe, D. “Spectrum Analysis Tutorial, Part 1: The Discrete Fourier Transform”. *Computer Music Journal*, Vol. 11:2, Summer 1987.

[51] Jaffe, D. “Spectrum Analysis Tutorial, Part 2: Properties and Applications of the Discrete Fourier Transform”. *Computer Music Journal*, Vol. 11:3, Fall 1987.

[52] Picinbono, B. “On instantaneous amplitude and phase of signals”. *IEEE Transactions on Signal Processing*, Vol. 45:3, March 1997.

[53] Loughlin, P. J. and Tacer, B. “Comments on the instantaneous frequency”. *IEEE Signal Processing Letters*, Vol. 4:5, May 1997.

[54] Crochiere, R.E. and Rabiner, L.R. *Multirate Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, 1983.

[55] Settel, Z. and Lippe, C. “Real-time Musical Applications using Frequency Domain Signal Processing”. *IEEE Workshop Appl. of Signal Processing to Audio and Acoustics*, 1995.

[56] Jones, G. and Boashash, B. “Generalized Instantaneous Parameters and Window Matching in the Time frequency plane”. *IEEE Transactions on Signal Processing*, Vol. 45:5, pp. 1264–1275, May 1997.

CHAPTER 3. SIGNAL MODELS

[57] McAulay, R. J. and Quatieri, T. F. “Magnitude-only reconstruction using a sinusoidal speech model”. *IEEE Proc. Int. Conf. Acoust., Speech and Signal Processing*, San Diego, CA, p. 27.6.1., 1984.

[58] McAulay, R. J. and Quatieri, T. F. “Speech analysis/synthesis based on a sinusoidal representation”. *M.I.T., Lincoln Lab., Rep. TR-693*, AD-A157023, May 1985.

[59] McAulay, R. J. and Quatieri, T. F. “Speech Analysis/Synthesis Based on a Sinusoidal Representation”. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34:4, pp. 744–754, August 1986.

[60] Smith, J.O. and Serra, X. “PARSHL: An Analysis/Synthesis Program for Nonharmonic Sounds based on a Sinusoidal Representation”. *Proceedings of the International Computer Music Conference*, pp. 290–297, 1987.

[61] McAulay, R. J. and Quatieri, T. F. “Speech Transformations Based on a Sinusoidal Representation”. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34:6, pp. 1449–1464, August 1986.

[62] Quatieri, T. F. and McAulay, R.J. “Shape Invariant time-Scale and Pitch Modification of Speech”. *IEEE Transactions on Signal Processing*, Vol. 40:3, March 1992.

[63] Quatieri, T. F., Dunn, R.B. and Hanna, T. E. “A subband Approach to Time-Scale Expansion of Complex Acoustic Signals”. *IEEE Transactions on Speech and Audio Processing*, Vol. 3:6, November 1995.

[64] Quatieri, T.F. and Hanna, T.E. “Time-Scale Modification with Inconsistent Constraints”. *IEEE Workshop Appl. of Signal Processing to Audio and Acoustics*, 1995.

[65] Pollard, M.P., Cheetham, B.M.G., Goodyear, C.C. and Edgington, M.D. “Shape-invariant Pitch and Time-Scale Modification of Speech by Variable Order Phase Interpolation”. *IEEE Proceedings ICASSP-97*, pp. 919–922, 1997.

[66] O’Brian, D. and Monaghan, A. “Shape invariant time-scale modification of speech using a harmonic model”. *IEEE Proc. Int. Conf. Acoust., Speech and Signal Processing*, Vol. 1, pp. 381–384, 1999.

[67] “Time and Frequency Scale Modification of Audio Signals Using an Extended Sinusoidal Model”. *MSc Thesis.* University of Illinois, 1992.

[68] Fitz, K. and Haken, L. “Sinusoidal Modeling and Manipulation Using Lemur”. *Computer Music Journal*, Vol. 20:4, pp. 44–59, Winter 1996.

[69] Fitz, K. and Haken, L. “Bandwidth Enhanced Sinusoidal Modeling in Lemur”. *Proceedings of the International Computer Music Conference*, Banff 1995.

[70] Di Federico, R. “Waveform Preserving Time Stretching and Pitch Shifting for Sinusoidal Models of Sound”. *Proceedings DAFX-98*, pp. 44–48, Barcelona, 1998.

[71] Desainte-Catherine, M. and Marchand, S. “High-Precision Fourier Analysis of Sounds using Signal Derivatives”. *J. Audio Eng. Soc.*, Vol. 48:7/8, July/August 2000.

[72] Desainte-Catherine M. and Marchand S. “High Precision Fourier Analysis of Sounds using Signal Derivatives”. *LaBRI Research Report No. 120498*, University of Bordeaux, 1998.

[73] Marchand, Sylvain “Improving Spectral Analysis Precision with an Enhanced Phase Vocoder using Signal Derivatives”. *Proceedings DAFX-98*, 1998.

[74] Serra, X. “A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition”. *PhD Thesis.* CCRMA, Dept. of Music, Stanford University, 1989.

[75] Serra, X. and Smith, J. O. “Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition”. *Computer Music Journal*, Vol. 14:4, pp. 14–24, Winter 1990.

[76] Serra, X. “Sound hybridization techniques based on a deterministic plus stochastic decomposition model”. *Proceedings of the International Computer Music Conference*, Aarhus, 1994.

[77] Serra, X. “Musical sound modeling with sinusoids plus noise”. *Musical Signal Processing*, Roads, C. et al. eds., Swets & Zeitlinger, 1997.

[78] Rodet X. and Depalle Ph. “Spectral envelopes and inverse FFT synthesis”. *93rd Convention of the Audio Engineering Society*, New York, 1992.

[79] Verma, T. S. and Meng, T. H. Y. “Extending Spectral Modeling Synthesis with Transient Modeling Synthesis”. *Computer Music Journal*, Vol. 24:4, pp. 47–59, Summer 2000.

[80] Rao, K. and Yip, P. *Discrete Cosine Transform: Algorithms, Advantages, Applications.* Boston: Academic Press, 1990.

[81] Masri, P. “Computer Modelling of Sound for Transformation and Synthesis of Musical Signals”. *PhD Thesis.* University of Bristol, 1996.

[82] Masri, P. “Improved Modeling of Attack Transients in Music Analysis-Resynthesis”. *Proceedings International Computer Music Conference*, Hong Kong, 1996.

[83] Strawn, J. “Analysis and Synthesis of Musical Transitions Using the Discrete Short-Time Fourier Transform”. *J. Audio Eng. Soc.*, Vol. 35:1-2, Jan/Feb 1987.

[84] Rodet, X. “Musical Sound Signal Analysis/Synthesis: Sinusoidal+Residual and Elementary Waveform Models”. *IEEE Time-Frequency and Time-scale Workshop 97*, Coventry, August 1997.

[85] George, E.B. and Smith M.J.T. “Analysis-by-synthesis/overlap-add sinusoidal modeling applied to the analysis and synthesis of musical tones”. *J. Audio Eng. Soc.*, Vol. 40:6, pp. 497–515, June 1992.

[86] Marques, J. and Almeida, L. “Frequency-varying sinusoidal modeling of speech”. *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-37:5, pp. 763–765, 1989.

[87] Depalle, Ph. and Hélie, T. “Extraction of spectral peak parameters using a short-time Fourier transform modeling and no sidelobe windows”. *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk, October 1997.

[88] Goodwin, M. and Vetterli, M. “Time-frequency signal models for music analysis, transformation, and synthesis”. *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 133–136, June 1996.

[89] Hamdy, K. N., Tewfik, A. H., Chen, T., and Takagi, S. “Time-scale modification of audio signals with combined harmonic and wavlet representations”. *IEEE Proc. Int. Conf. Acoust., Speech and Signal Processing*, April 1997.

CHAPTER 4. CHARACTERISTICS OF THE PHASE VOCODER PARAMETERS

[90] Masri, P. and Bateman, A. “Identification of Nonstationary Audio Signals using the FFT, with Application to Analysis-based Synthesis of sound”. *IEE Audio Engineering Colloquium Digest*, pp. 11/1–11/6, 1995.

[91] Peeters, G. and Rodet, X. “Signal Characterization in terms of Sinusoidal and Non-Sinusoidal Compontents”. *Proceedings DAFX-98*, 1998.

CHAPTER 5. THE ANALYSIS/TRANSFORMATION/RESYNTHESIS SYSTEM

[92] MATLAB is a trademark of The Mathworks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA. `http://www.mathworks.com/`

[93] Burrus, C.S. et al. *pkpicker.m*. For use with the book *Computer-Based Exercises for Signal Processing Using MATLAB*, Prentice-Hall, 1994.

[94] Goodwin, M. “Residual Modeling in Music Analysis/Synthesis”. *IEEE Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 1005–1008, 1996.

[95] Depalle, Ph., Garćia, G. and Rodet, X. “Analysis of Sound for Additive Synthesis: Tracking of Partials Using Hidden Markov Models”. *Proceedings of the International Computer Music Conference*, 1993.

CHAPTER 6. STRATEGIES FOR TRANSIENT DETECTION

[96] Brown, J.C. and Puckette, M.S. “An Efficient Algorithm for the Calculation of a Constant Q Transform”. *J. Acoust. Soc. Am. 92*, pp. 2698-2701, 1992.

[97] Brown, J.C. “Calculation of a constant Q spectral transform”. *J. Acoust. Soc. Am. 89*, pp. 425–434, 1991.

[98] Brown, J.C. “Musical fundamental frequency tracking using a pattern recognition method”. *J. Acoust. Soc. Am. 92*, pp. 1394–1402, 1992.

[99] Puckette, M.S., Apel, T. and Zicarelli, D.D. “Real-time audio analysis for Pd and MSP”. *Proceedings of the International Computer Music Conference*, 1998.

[100] Bonada, J. “Automatic technique in frequency domain for near-lossles time-scale modification of audio”. *Proceedings of the International Computer Music Conference*, Berlin, 2000.

[101] Fitz, K., Haken, L. and Christensen, P. “Transient Preservation under Transformation in an Additive Sound Model”. *Proceedings International Computer Music Conference*, Berlin 2000.

[102] Thornburg, H. and Gouyon, F. “A Flexible Analysis-Synthesis Method for Transients”. *Proceedings International Computer Music Conference*, Berlin 2000.

APPENDIX B. SOUND EXAMPLES

[103] Fitz, K., et al. *ICMC2000 Analysis/Synthesis Comparison Session*
`http://cnmat.cnmat.berkeley.edu/SDIF/ICMC2000/sounds.html`