Audio Time-Scale Modification
in the Context of Professional Audio Post-production

by

Jordi Bonada Sanjaume

Research work

for

PhD Program

Informàtica i Comunicació digital

in the

GRADUATE DIVISION

of the

UNIVERSITAT POMPEU FABRA, BARCELONA

Fall 2002

Audio Time-Scale Modification

in the Context of Professional Audio Post-production

By

Jordi Bonada Sanjaume

Abstract

Time-Scale Modification of Audio

in the Context of Professional Audio Post-production

by

Jordi Bonada Sanjaume

In this research work we review most techniques that have been used for time-scale modification of audio. These techniques are grouped in three categories: time-domain algorithms, phase-vocoder and variants, and signal models. Considering the context of professional audio post-production, we choose one category as the base technique to work from and finally we define the research to be carried out for the Doctoral Thesis.

Acknowledgements

I thank Professor Xavier Serra, who introduced me into this magic world of audio digital signal processing and provided funding for me to work at the Music Technology Group. Thank you to Yamaha Corp. for funding my research at MTG. Thank you to all the people working at MTG for helping me to spend a really nice time at work, usually a difficult task, and also for being my friends, which is even more impressive. Special thanks to Perfe, Ramon, Àlex, Pedro, Maarten, Xavier, Eduard, Mark, Tue, Lars, Oscar & Oscar, Jaume, Claudia and Enric who have worked next to me and have stood my background music.

Contents

1 Introduction

1.1 Context

The Music Technology Group (MTG) is a research group that belongs to the Audio Visual Institute (IUA) of the Universitat Pompeu Fabra (UPF), in Barcelona. It was founded in 1994 by his current director, Xavier Serra, and it has more than 30 researchers.

From the initial work on spectral modeling, the MTG is dedicated to sound synthesis, audio identification, audio content analysis, description and transformations, interactive systems, and other topics related to Music Technology research and experimentation.

The research at the MTG is funded by a number private companies and various public institutions (Generalitat de Catalunya, Ministerio Español de Ciencia y Tecnología and European Commission).

MTG researchers are also giving classes in different teaching programmes within and outside the UPF: Diploma in Computer Systems, Degree in Computer Engineering, Degree in Audiovisual Communication, Doctorate in Computer Science and Digital Communication, Doctorate in Social Communication, and at the *Escola Superior de Música de Catalunya* (ESMUC).

1.2 Author's background

I studied Telecommunication Engineering at the Catalunya Polytechnic University of Barcelona (Spain) and graduated in 1997 [P15]. In 1996, I joined the Music Technology Group (MTG) as a researcher and developer in digital audio analysis and synthesis. I was initially involved in several projects [P1, P3, P7, P11-P14] related to Spectral Modeling Synthesis (SMS), a signal model developed by Xavier Serra [80, 81], director of the MTG. From 1998 to 2000, I was in charge of a project intended to develop a *voice morphing system for impersonating in karaoke* [P5, P6, P9, P10, Pat1-Pat9], in cooperation with YAMAHA. Since 1999 I have been a lecturer at the same university where I have been teaching Audio Signal Processing, and I am also a PhD candidate in Informatics and Digital Communication. Since 2000, I am in charge of a project aiming at developing a *singing voice synthesizer* [P2, P4, Pat10-Pat12], in cooperation with YAMAHA. I am currently involved in research in the fields of spectral signal processing, especially in audio time-scaling [P8] and voice synthesis and modeling.

1.3 What this research work is about

This research work focuses on time-scale modification of general audio signals. First of all, in this chapter, we will define what time-scale is and what it can be used for. In the context of professional audio post-production, we will define the ideal product and briefly review some of the best available software packages in the market.

Then, in the following chapters we will review most of the techniques used for time-scaling purposes trying to understand how they work. In chapter 2, time domain techniques will be reviewed, including variable speed replay and time-segment processing methods (OLA, SOLA, TD-PSOLA and WSOLA). In chapter 3 we will talk about the short-time Fourier transform and the traditional phase-vocoder, some improvements made to deal with its main drawback (the loss of vertical phase coherence), and a recent promising enhancement to deal with the time-frequency resolution compromise. To conclude, some algorithms based on wavelet representations will be discussed. In chapter 4 some signal models will be covered, starting from sinusoidal modeling, then adding residual (SMS), transients (TMS) and at last a multiresolution approach.

Finally, in chapter 5, we will focus on the proposed research thesis to be done. First we will define the target of the research. Then we will choose the best existing approach from which to start, and finally propose the needed improvements to be done.

1.4 About time-scale modification

1.4.1 Definition

In the context of music, we could think of time-scaling as something similar to tempo change. If a musical performance is time-scaled to a different tempo, we should expect to listen to the same notes starting at a scaled time pattern, but with durations modified linearly according to the tempo change. The pitch of the notes should however remain unchanged, as well as the perceived expression. Thus, for example, vibratos should not change their depth, tremolo or rate characteristics. And of course, the audio quality should be preserved in such a way that if we had never listened to that musical piece, we wouldn't be able to know if we were listening to the original recording or to a transformed one. In a more rigorous way, time-scaling could be considered as a low level concept associated to the higher level musical concept of tempo change.

Another way to define it would be to say that the time scaled version of an acoustic signal should be perceived as the same sequence of acoustic events as the original signal being reproduced according to a scaled time pattern.

1.4.2 Applications

Time-scale modification can be used for several different applications. Here we list most of them:

- **Post-synchronization**. Often a soundtrack has been prepared independently from the image it is supposed to accompany and therefore they are not synchronized. Time-scale modification of the sound track is a way to synchronize both sound and image. A typical example in the movie industry is dialogue post-synchronization.
- **Broadcasting applications.** Conversion between video (25 or 30 fps) and cinema (24fps) format preserving the quality of the soundtrack according to broadcasting standards requires a high-quality time-scale modification [66].
- **Data compression**. Time-stretching of audio has been used to compress data for communications or storage [2]. Basically, the audio is compressed, transmitted (or stored) and finally expanded after reception. However, only a limited amount of data reduction can be obtained using this method.
- **Synthesis by sampling**. Sampler synthesizers usually hold a dictionary of prerecorded sound units (samples) and produce a continuous output by joining together these units with the desired pitch and duration. Since the dictionary has memory limitations, it is not possible to record all possible pitches and durations for each sample, thus independent time-scale and pitch-scale modifications are needed.
- **Musical composition**. Music composers that often work with pre-recorded material like to independently control time and pitch. Thus, time and pitch-scale modifications can be understood as composition tools.
- **Real-time music performance**. Time-stretch can be very useful in real-time music performances as a control.
- **Orchestra conductor**. A user can interact with an electronic orchestra at a high level of realism controlling an original audio and video recording of a real orchestra. The interaction controls needed are not only volume and instrumentation but also tempo of the orchestra. A high–fidelity time-scale algorithm working in real-time can be used to control the tempo. *Personal Orchestra* is a recent system being used as virtual conductor and exhibited in the House of Music of Vienna [11, 12].
- **Concatenate different music pieces with a very smooth tempo transition**. In techno music, the disc jockey plays different pieces of music one after the other as a continuous stream. Often these musical excerpts don’t have the same tempo,

although the stream is supposed to have only very smooth tempo transitions. Using time-scale modifications the disc jockey could change smoothly the tempo from one piece to each other without undesired pitch modifications.

- **Computer interface**. The speed of speech-based computer interfaces could be controlled by the user using time-scale modifications to adapt the speed of the interactions to the user requirements.
- **Teaching**. Studies have indicated that listening twice to teaching materials that have been speeded up by a factor of two is more effective than listening to them once at normal speed [83] .
- **Foreign language learning**. Slowing down by time-scale modification the rate of foreign speaker recordings could be a good way to significantly facilitate learning a foreign language. Then, as the student improves, the speaker rate could be gradually increased.
- **Reading for the blind**. Speech recordings can be an alternative to reading for blind people, although usually one can read at a faster rate than one can speak. With the proper time-scale modification, speech recordings rate could be increased while preserving the intelligibility.
- **Voice mail systems**. Time-compressed speech has been used to speed up message presentation in voice mail systems [42, 59].
- **Adaptive layout scheduling in packet voice communications**. Time-scale transformation is useful to dynamically adjust the playout time of voice packets in packet voice communications, thus modifying the rate of playout while preserving voice pitch. This way buffering delay and loss rate can be significantly reduced [25].
- **Media browsing**. Time-scale modification can be useful to speed up trough boring material and slow down for interesting portions [1].
- **Speech recognition**. Time compression techniques have also been used in speech recognition systems to time normalize input utterances to a standard length [53].
- **Watermarking**. Audio watermarking can be achieved by time-scale modification if it is used to change the length of the intervals between relevant points of the audio signal to embed data [54]. This algorithm has been shown to be robust to common audio processing operations like mp3 compression, low pass filtering and time-scale modification. With an appropriate time-scale algorithm the watermarked signal can be indistinguishable from the original signal.

1.4.3 Requirements in the context of professional audio post-production

What are the requirements of time-stretch software in the context of professional audio post-production?

- **Sound Quality**. It should be the outstanding. The transformed signal should sound as the original one, sounding like a tempo change. No phasing or reverberation should be added (typical problem of the phase-vocoder, see §3). Transients should sound as clear and sharp as in the original audio. No granularity should be added. Timbre should not be colored. No artifacts should be perceived. Singer's voice should preserve its timbre characteristics and voice quality (breathiness, hoarseness, sweetness …). Vibrato of instruments and voices should keep its characteristics (rate, tremolo, depth).

- **Time-scale factor**. For broadcasting applications, usually the needed scale factor goes from 24/30 to 30/24 for the conversions between video and cinema formats. However, for musical applications, the scale factor should be ideally between 50 and 200%, but a practical range could be 70-130%. The factor should be introduced as tempo change, length change, target length or target BPM. Besides, it should be possible to apply a variable time-stretch factor. Tempo mapping for synchronization purposes should also be a requirement.

- **Timing accuracy**. Ideally it should have sample accuracy, so that rhythm would not become irregular (typical problem of time domain techniques, see §2) and the synchronization between audio channels would be kept.

- **Bit depth**. It should be unlimited.

- **DSP processing**. At least 32 bit.

- **Sampling rate**. It should be able to work at any sampling rate. However, at least 22.5, 32, 44.1, 48, 88.2, 96 and 192 Khz should be accepted.

- **Multichannel processing with phase coherence and aural image preservation**. At least stereo pairs should be able to be time-scaled while preserving their phase coherence and aural image. Moreover, ideally it should be able to work with Dolby Pro Logic, Dolby Pro Logic II and Dolby Digital tracks preserving the surrounding effect.

- **Interface**. The system should be easy to use, with no complex parameters to choose from and not many presets for different type of input signals (the least, the better). However, it should be as powerful as possible while keeping the simplicity.

- **Computational cost**. Ideally with a low computational cost. It should be able to work in real-time, at least in preview mode with low quality.
- **Plug-in**. It should be implemented as a plug-in for most popular audio post-production software (TDM, DirectX …)
- **Streaming**. This could be considered as an add-on feature. The system could work in streaming mode, so it wouldn't be needed to preprocess the entire input before listening to some results.
- **Hardware integration.** Another interesting add-on feature would be the possibility to be integrated in hardware platforms.

1.4.4 Commercial products

There exist several commercial products in the market that use time-stretch as one of their most attractive feature.

Maybe the best three software systems are the ones from Prosoniq [3], Serato [5] and Wave Mechanics [4].

- **Prosoniq's *TimeFactory***

  Prosoniq Products is a German software company that does research and development for audio editing and digital signal processing. *Time Factory* is a standalone software for MAC and PC platforms dedicated to perform near-lossless time and pitch scale modifications on monophonic and polyphonic music signals. It is based on Prosoniq's proprietary MPEX (Minimum Perceived Loss Time Compression/Expansion) algorithm that uses an artificial neural network for time series prediction in the scale space domain. The signal is represented in terms of basis functions that have a good localization in both the time and frequency domain (like certain types of wavelets have). The signal is transformed on the basis of the Prosoniq's proprietary MCFE (Multiple Component Feature Extraction), which is said to "*provide relatively unlimited access to distinct spectral properties as well as to the phase relation and exact frequency of harmonics, formants and temporal sound developments*". Features can be found in appendix A.

- **Serato's *Pitch'n Time 2***

  Serato Audio Research is a software company from New Zealand. Its proprietary *Intelligent Sound* technology is the base of its award winning Pro Tools plug-in *Pitch'n Time*. Under this technology there is a sophisticated model

Prosoniq TimeFactory
File Setup Algorithm Help
PROSONIQ TimeFactory™ Batch List
TimeFactory
File Properties
Semitones
Cents
Duration / Speed
lower
44100
upper
44100
Processing
Processing upper ...
0:32
Cancel
MPEX Advanced Polyphonic Time / Pitch Scaling
© 1998, 99 PROSONIQ PRODUCTS SOFTWARE GmbH

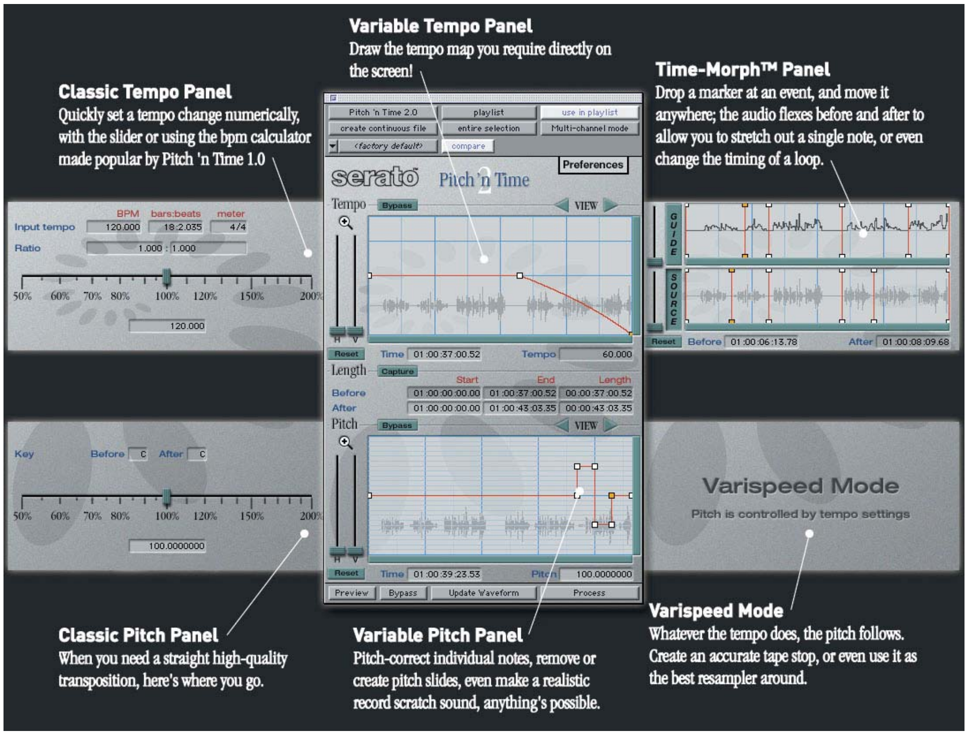
Figure 1.1 Prosoniq's *Time Factory*

of the human auditory system which "*allows the software to listen to the music, performing a sophisticated auditory scene analysis. Only by listening can it determine what 'sounds' the same, but faster or slower. Normally using such a sophisticated model of the human auditory system would be computationally prohibitive, so novel mathematical methods had to be developed by Serato to speed up this process in software*". Serato have US and international patents pending on this technique [43] (see §3.6 for details). Features can be found in appendix A.

- **Wave Mechanics's *Speed***

  Wave Mechanics Inc is an american company dedicated to building professional-quality digital audio processing tools. Its algorithms can be found in high-end processors, digital consoles, digital audio workstations and in multimedia products. Speed is a plug-in software for the ProTools platform. "*With SPEED, the power to transform tempo and pitch of almost any source imaginable is now at your fingertips, with unprecedented ease-of-use and unparalleled audio quality. SPEED uses an entirely new approach to time and pitch modification that now makes it possible to transform single instruments and even entire mixes without the distortion that is usually introduced by these processes. Even stereo mixes can be processed with perfect phase alignment*".

In Table 1 we can find a comparison between the three commented software packages and the ideal one. The sound quality comparison has been done after some informal listening tests of several time-scaled audio signals by the author and other people from the MTG. As a conclusion, it seems that Serato's *Pitch'n Time* is the best time-scale plug-in in terms of sound quality and potential, and the most suitable for professional audio post-production.



Variable Tempo Panel
Draw the tempo map you require directly on the screen!

Classic Tempo Panel
Quickly set a tempo change numerically, with the slider or using the bpm calculator made popular by Pitch 'n Time 1.0

Time-Morph™ Panel
Drop a marker at an event, and move it anywhere; the audio flexes before and after to allow you to stretch out a single note, or even change the timing of a loop.

Classic Pitch Panel
When you need a straight high-quality transposition, here's where you go.

Variable Pitch Panel
Pitch-correct individual notes, remove or create pitch slides, even make a realistic record scratch sound, anything's possible.

Varispeed Mode
Whatever the tempo does, the pitch follows. Create an accurate tape stop, or even use it as the best resampler around.

Varispeed Mode
Pitch is controlled by tempo settings

Figure 1.3 Serato's *Pitch'n Time*

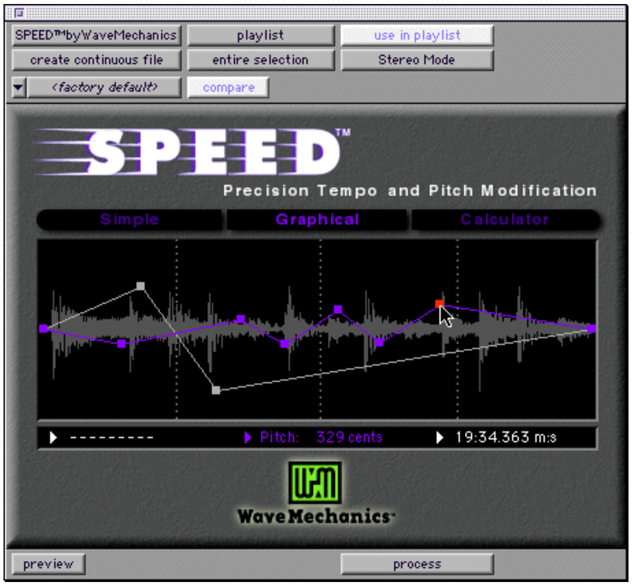


Figure 1.3 Wave Mechanics's *Speed*

On the other hand, *Harmo* from Genesis [2] is one of the best hardware used for broadcasting format conversion from cinema to video and vice versa. It is a real-time multi-channel pitch-shifter hardware using a wavelet approach (see §3.7 for more details).

1.4.5 Techniques for time-scale modification

Time-scale modification has been implemented in several different ways. Generally the algorithms are grouped in three different categories:

1. Time domain
2. Phase-vocoder and variants
3. Signal models

In the next chapters we will explain in detail the basics of these approaches and how they can be used to achieve time-scale modifications.

	ideal commercial product	**Wave Mechanic's *Speed***	**Prosoniq's *Time Factory***	**Serato's *Pitch'n Time***
sound quality	Outstanding for all stretching factors	artifacts, flanging, dirty, little reverberation, phasiness	artifacts, sounds dirty, little amplitude modulation. Smoothing of low frequencies	close to outstanding for 70-130% stretching factors. Smoothing of low frequencies.
transients	preserved all transients	Smoothed bass attacks. Flanging transients for high stretching factors.	It does not preserve high frequency attacks (hi-hats for example). Highly smoothed bass attacks.	It does not preserve high frequency attacks (hi-hats for example). Little smoothed bass attacks. At high time-scale ratios, sometimes transients are doubled for high stretching factors. Smoothed transients for low stretching factors.
vibrato	preserved with same characteristics (rate, tremolo, depth)	time-scaled rate and tremolo	time-scaled rate and tremolo	time-scaled rate and tremolo
time-scale factor	50-200%	50-200%	50-200%	50-200%
timing accuracy	sample	??	??	sample timing accuracy
bit depth	anyone	any bit depth available trough Digidesign AudioSuite	8,16,24	any bit depth available trough Digidesign AudioSuite
sampling rate	22.5, 32, 44.1, 48, 88.2, 96 and 192 Khz	22.5, 32, 44.1, 48, 88.2, 96 and 192 Khz	22.5, 44.1, 48, 96Khz	22.5, 32, 44.1, 48, 88.2, 96 and 192 Khz
Multi-channel phase coherence & aural image	yes, for unlimited number of channels	not perfect, for stereo	yes, for stereo	yes, up to 48 channels
interface	easy and powerful	easy	easy, but it's not possible a time-varying	easy and powerful
computational cost	low	normal	high	high
real-time preview	yes	yes	no	yes
streaming mode	yes	no	no	no
plug-in	TDM and DirectX	TDM	standalone	TDM

Table 1 Comparison of three software packages that time-scale audio with the ideal one

TECHNIQUES
FOR
TIME-SCALE MODIFICATION

2 Time domain processing

2.1 Introduction

In this section we will explain the basics of different time domain techniques. We will concentrate on how they can be used for time-scale purposes. Finally, we will talk about general benefits and drawbacks of time domain techniques for time-stretching.

2.2 Variable speed replay

Analog audio tape recorders allow replay at different speeds. During faster playback, the pitch of the sound is raised while the duration is shortened. On the other hand, during slower playback, the pitch of the sound is lowered while the duration is lengthened. This was historically the first technique used to time-scale the audio.

If we define v as the relative replay speed, $T_{\mathrm{s},recording}$ and $T_{s,replay}$ as the recording and replay sampling periods, then we could say that what was happening during the time period $nT_{s,recording}$ is now happening in the time period

$$nT_{s,replay} = nvT_{s,recording} \qquad \textbf{(2.1)}$$

Therefore, if $v > 1$ the time scale is expanded, and if $v < 1$ the time scale is compressed. A straightforward method of implementing the variable speed replay is hence to modify the sampling frequency while playing back the sound in the following way

$$f_{s,replay} = \frac{f_{s,recording}}{v} \qquad \textbf{(2.2)}$$

In the case of an **analog audio tape**, if we just modify the output DAC sampling frequency then the spectrum of the signal will be scaled by factor $1/v$ and the analog reconstruction filter should be tuned in order to avoid aliasing [35, 58].

In the case of a **digital audio tape**, the desired replay sampling frequency $f_{s,replay}$ has to be converted to the output sampling frequency (usually equal to the input sampling frequency $f_{s,recording}$). If the time is expanded ($v > 1$), then the output signal should be interpolated (over-sampled). Otherwise, if the time is compressed ($v < 1$) then the output signal should be filtered and decimated [62]. The quality of this sampling rate conversion depends very much on the interpolation filter used. A good review of interpolation methods can be found in [16, 58].

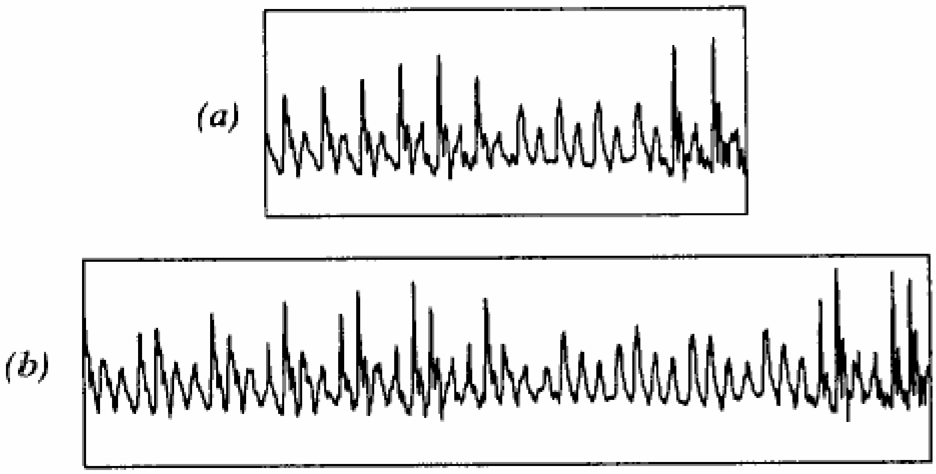


Figure 2.1 OLA time-scale (from [89])

Historical methods – Phonogène

Special machines based on a modified tape recorder allowed alteration of the duration and pitch of sounds. We can find several in the literature but maybe the historically most important was the *Phonogène universel* by Pierre Schaeffer [77]. This modified tape recorder had several playback heads mounted on a rotating head drum. The duration of the sound was determined by the absolute speed of the tape at the capstan. This machine also allowed a transposition controlled by the relative speed of the heads to that of the tape.

2.3 Time-scale modification using time-segment processing

The intended time scaling does not correspond to the mathematical time scaling as realized by vary-speed, but we rather require scaling the perceived timing attributes without scaling the perceived frequency attributes, such as pitch. In time segment processing, the basic idea of time stretching is to divide the input sound into segments. Then if the sound is to be shortened, some segments are discarded. Otherwise, if the sound is to be lengthened, some segments are repeated. One source of artifacts is the amplitude and phase discontinuity at the boundary of the segments. Some techniques that have been developed to deal with this problem will be presented in next subsections.

2.3.1 Granulation - OLA

Time-compression or expansion by granulation is achieved by extraction of windowed grains from sampled real sound and re-ordering in time [73]. For simple time-scale compression by granulation, the grain segments are excised from the source at time

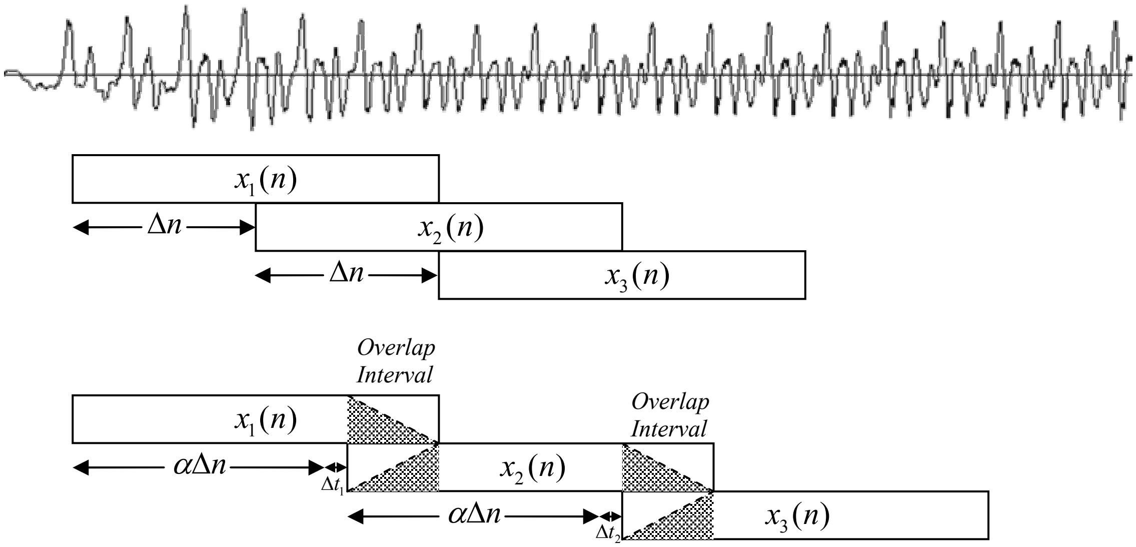


Figure 2.2 SOLA time-scale

instants t_i and added to the output at time instants $\tilde{t}_i = \alpha t_i$, where α is the time scale factor. More generally, we could say that individual output segments should ideally correspond to input segments that have been repositioned according to the desired time-warp function $\tilde{t} = \tau(t)$.

As a result, OLA repositions the individual input segments with respect to each other, destroying the original phase relationships, and constructs the output signal by interpolating between these misaligned segments. This cause pitch period discontinuities and distortions that are detrimental for signal quality (see Figure 2.1).

2.3.2 Synchronous Overlap and Add (SOLA)

The synchronized overlap and add method (SOLA) first described by Roucos and Wilgus in [76] is a simple algorithm for time stretching based on correlation techniques. It became popular in computer-based systems at the beginning, because “of all time scale modification methods proposed, SOLA appears to be the simplest computationally, and therefore most appropriate for real-time applications" [40]. The input signal is divided into overlapping blocks of a fixed length and each block is shifted according to the time scaling factor α. Then the discrete-time lag Δt_n of highest cross-correlation is searched in the area of the overlap interval. At this point of maximum similarity the overlapping blocks are weighted by a fade-in and fade-out function, as shown in Figure 2.2, producing significantly less artifacts than traditional sampling techniques. This technique tends to preserve the pitch, magnitude and phase of a signal.

Makhoul used both linear and raised cosine functions as fade-in and fade-out functions, and found the simpler linear function sufficient [52]. The SOLA algorithm is robust in the presence of correlated or uncorrelated noise, and can improve the signal to noise ratio of noisy speech [92, 93].

2.3.3 Time-domain Pitch-synchronous Overlap and Add (TD-PSOLA)

Moulines *et al.* proposed in [64, 65] a variation of the SOLA algorithm based on the hypothesis that the signal is characterized by a pitch (for example, human voice and monophonic musical instruments). This technique, Time-Domain Pitch Synchronous Overlap and Add (TD-PSOLA), uses the pitch detected to correctly synchronize the time segments, thus avoiding pitch discontinuities. The overlapping procedure is performed pitch-synchronously in order to retain high quality time-scale modification. One of the main problems of this technique is estimating the basic pitch period of the signal, especially in cases where the actual fundamental frequency is missing.

The algorithm can be decomposed in two phases. The first one analyzes and segments the input signal, and the second phase overlaps and adds the extracted segments to synthesize the time stretched signal. The whole procedure is shown in Figure 2.3.

Analysis phase

The fist step is to determine the time instants (or pitch marks) t_i corresponding to the maximum amplitude (or glottal pulses for voice) during the periodic parts of the sound, and to a synchronous rate during the unvoiced portions. The pitch period $P(t)$ can be obtained out of the time instants by $P(t_i) = t_{i+1} - t_i$ if it is considered constant on the time interval (t_i, t_{i+1}).

Then the input sound is segmented and the segments are centered at every pitch mark t_i. Each segment is windowed with a Hanning window of two pitch periods length that will fade in and out the overlapping segments.

Synthesis phase

The stretching factor α will determine the pitch periods of the output signal $\tilde{P}(\tilde{t})$ by $\tilde{P}(\tilde{t}) = \tilde{P}(\alpha t) = P(t)$. The output time instants $\tilde{t}_k$ can be calculated from the pitch periods by $\tilde{t}_{k+1} = \tilde{t}_k + \tilde{P}(\tilde{t}_k) = \tilde{t}_k + P(t_i)$, where i is the index of the analysis segment that minimizes the time distance $\left|\alpha t_i - \tilde{t}_k\right|$. For each synthesis pitch mark $\tilde{t}_k$, the corresponding analysis segment is overlapped and added.

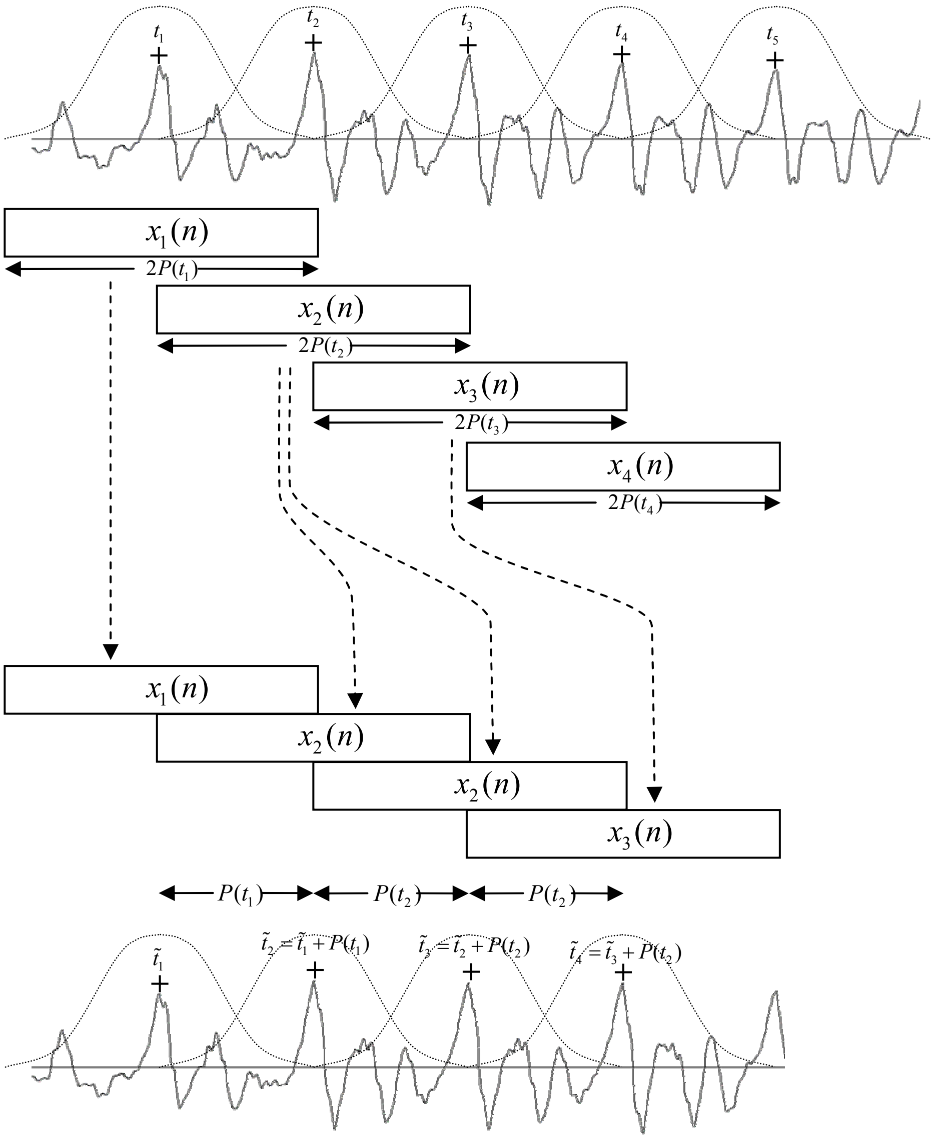


Figure 2.3 TD-PSOLA time-scale expansion ($\alpha > 1$)

It has been supposed that the scaling factor α is constant along time, but this could be generalized. In that case, we could define and use $\tilde{t} = \int_0^t \alpha(\tau)d\tau$ instead of $\tilde{t} = \alpha t$. Notice that when the signal is expanded ($\alpha > 1$) some segments are repeated twice or more, and when the signal is compressed ($\alpha < 1$) some segments are discarded.

There are several improvements on this technique that can increase the quality of the output signal [22]. On one hand, for large values of scaling, some buzziness appears in unvoiced parts due to the regular repetition of identical overlapped segments. This can somehow be reduced by reversing the time axis for every repeated version of an unvoiced

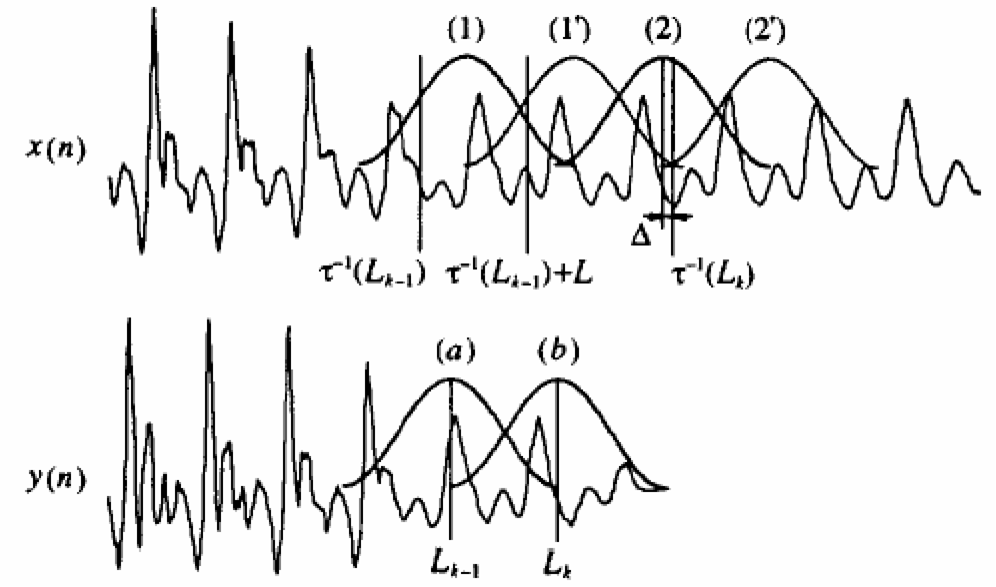


Figure 2.5 WSOLA algorithm (from [89])

segment. On the other hand, uniformly applied time stretching can produce some artifacts on fast transitions (or transients). Here one possible solution could be to identify this transitions and don't time-scale them (no segment would be repeated).

2.3.4 Waveform-similarity-based Synchronous Overlap and Add (WSOLA)

Verhelst and Roelands proposed in [89] a variant of the SOLA tradition techniques called WSOLA. It ensures sufficient signal continuity at segment joints by requiring maximal similarity to the natural continuity in the original signal via cross-correlation.

Basic operation

Let's assume (a) is the last segment added to the output at time instant $L_{k-1} = (k-1)L$ and (a) corresponds to the segment (1) in the input signal. At this point, a segment (b) must be found in the input signal around the input time instant $\tau^{-1}(L_k)$, so that it resembles as closely as possible the segment $(1')$. This segment $(1')$ is the one that follows (1) in the input sound, thus the one that overlaps (1) in a natural way. The position of this best segment $(b) = (2)$ is found by maximizing a similarity measure, for example cross-correlation or cross-AMDF.

2.4 Conclusions

Maybe time domain techniques are the simplest methods for performing time-scale modification. Many papers show good results segmenting the input signal into several windowed sections and then placing these sections in new time locations and overlapping them to get the time scaled version of the input signal. This set of algorithms is referred to as Overlap-Add (OLA). To avoid phase discontinuities between segments, the synchronized OLA algorithm (SOLA) uses a cross-correlation approach to determine where to place the segment boundaries. In TD-PSOLA, the overlapping operation is performed pitch-synchronously to achieve high quality time-scale modification. It works well with signals having a prominent basic frequency and can be used with all kinds of signals consisting of a single signal source. When it comes to a mix of signals, this method will produce satisfactory results only if the size of the overlapping segments is increased to include a multiple of cycles thus averaging the phase error over a longer segment making it less audible. More recently, WSOLA uses the concept of waveform similarity to ensure signal continuity at segment joints, providing high quality output with high algorithmic and computational efficiency and robustness. We can find a comparison of these time-domain techniques in Table 2.

All the above mentioned techniques consider equally transient and steady state parts of the input signal, thus time-scale them both in the same way. To get better results, it would be preferable to detect the transients regions and don't time-scale them, just translate them into a new time position, while time-scaling the rests parts of the input signal (non-transient segments). The earliest found mention of this technique is the Lexicon 2400 time compressor/expander from 1986. This model detected transients, and only time-scaled the remaining audio using TD-PSOLA style algorithm. In [13, 48] it is shown that using time-scale modification on only non-transient parts of speech improves the intelligibility and quality of the resulting time-scaled speech. We can find in [8] a good review of the literature on methods for time-compressing speech, including related perceptual studies of intelligibility and comprehension.

	SOLA	TD-PSOLA	WSOLA
Synchronizing method	output similarity	pitch epochs	input similarity
Effective window length	fixed (>4pitch)	pitch adaptive	fixed
Algorithmic & computational efficiency	high	low	very high
Robustness	high	low	high
Speech quality	high	high	high
Pitch modification	no	yes	No

Table 2 Comparison of three time-domain techniques

3 Phase-Vocoder and variants

3.1 Introdution

In this section we will introduce the basics of several frequency domain techniques, starting with the phase-vocoder based on the short-time Fourier transform. Then different improvements will be introduced, basically related to the vertical phase coherence problem. Finally, different approaches that deal with the time-frequency resolution problem will be presented, including wavelet representations. We will concentrate on how these techniques can be used for time-scale purposes.

3.2 Short-Time Fourier Transform (STFT)

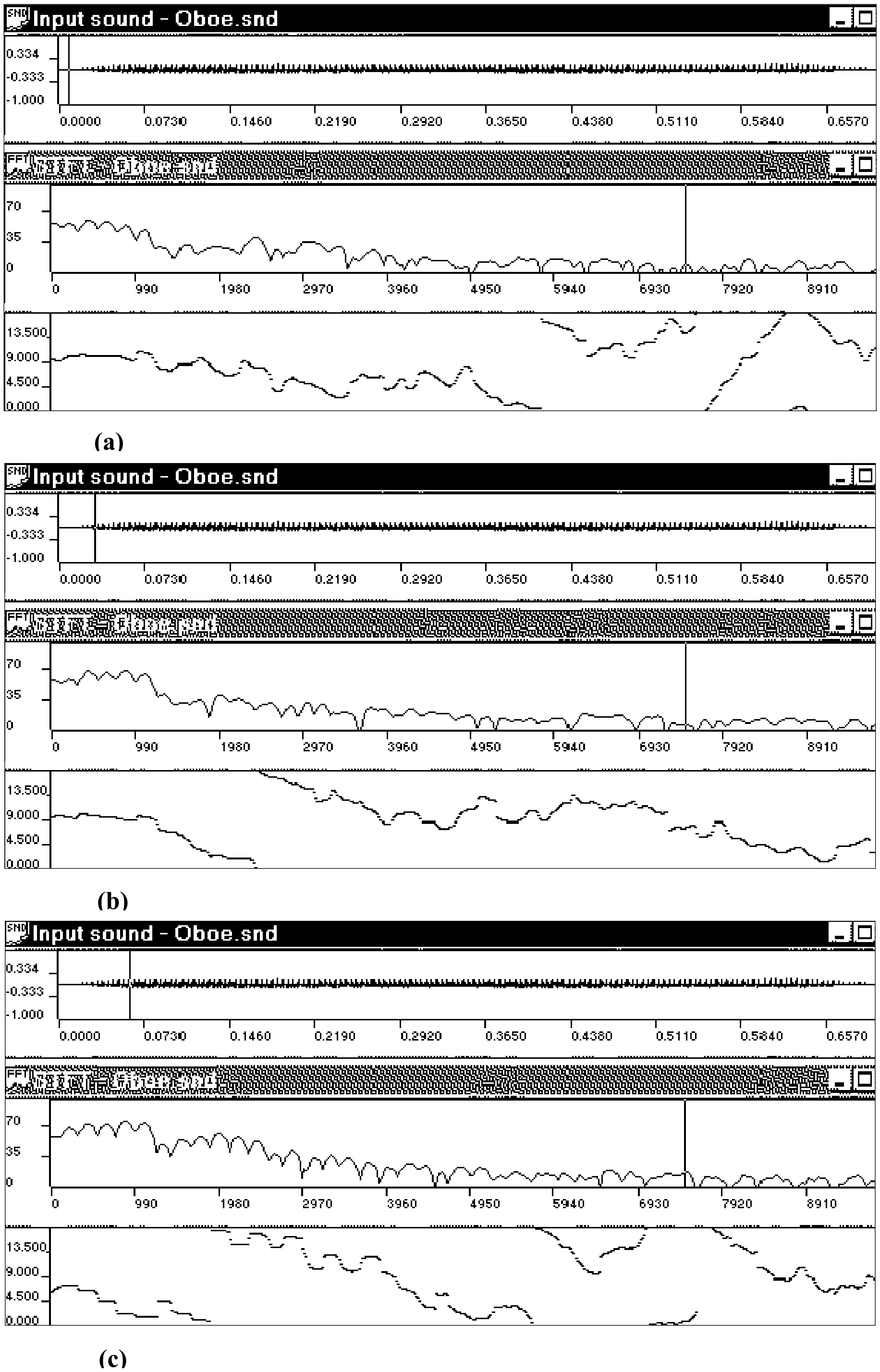
The concepts of short-time Fourier analysis and synthesis are well-known and have been widely described in the literature [15, 16, 69]. We can express the short-time Fourier transform (STFT) of the input signal $x(n)$ as

$$X(n,k) = \sum_{m=-\infty}^{\infty} x(m)h(n-m)e^{-j\frac{2\pi mk}{N}}, \quad k = 0,1,...,N-1 \tag{3.1}$$

where $X(n,k)$ is the time-varying spectrum with the frequency bin k and time index n. At each time index, the input signal $x(m)$ is weighted by a finite length window $h(n-m)$. $X(n,k)$ is a complex number and can be expressed as magnitude $|X(n,k)|$ and phase $\varphi(n,k)$. Figure 3.1 shows an example of an input signal $x(m)$ and its short-time spectra at three successive time indexes.

3.3 Phase-vocoder basics

This technique was introduced by Flanagan and Golden in 1966 [31] and digitally implemented by Portnoff [68] ten years later. There are two traditional ways to understand the phase-vocoder: as a filter bank or as a block-by-block model. These two models can be sawn respectively as horizontal and vertical lines in the time-frequency plane



(a)

(b)

(c)

Figure 3.1 Example of a STFT. We can see the magnitudes and phases of three different windowed segments of the input signal, in this case an oboe. The vertical line on the waveform shows the window center. It's easy to see how the harmonics appear as the window advances in time.

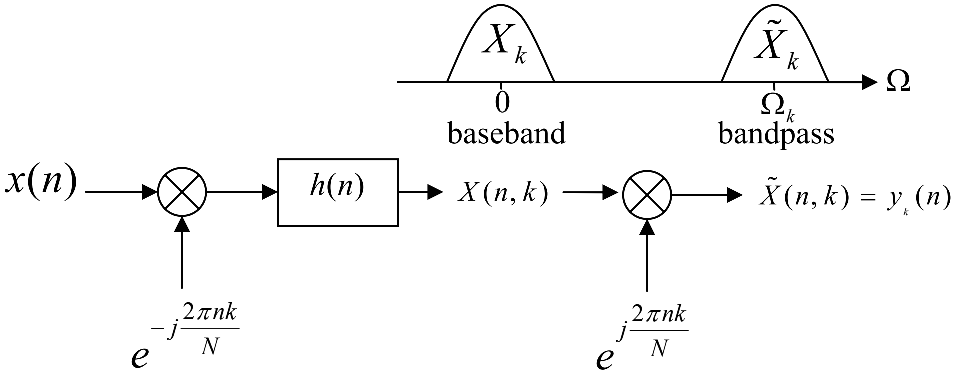


Figure 3.2 Filter bank description of the short-time Fourier transform. The frequency bands Y_k corresponding to each bandpass signal $y_k(n)$ are shown on the top.

3.3.1 Filter bank summation model

The computation of the time-varying spectrum of the input signal can be interpreted as a parallel bank of N bandpass filters with impulse response and transform given by

$$h_k(n) = h(n)e^{-j\frac{2\pi nk}{N}}, \quad k = 0,1,...,N-1 \tag{3.2}$$

$$H_k\left(e^{j\Omega}\right) = H\left(e^{j(\Omega-\Omega_k)}\right), \quad \Omega_\kappa = \frac{2\pi k}{N} \tag{3.3}$$

The bandpass signal of band k is obtained after filtering the input signal with the bandpass filter $h_k(n)$, as denoted by

$$y_k(n) = \sum_{m=-\infty}^{\infty} x(m)h_k(n-m) = \sum_{m=-\infty}^{\infty} x(m)h(n-m)e^{j\frac{2\pi(n-m)k}{N}} \tag{3.4}$$

$$= e^{j\frac{2\pi nk}{N}} \sum_{m=-\infty}^{\infty} x(m)e^{-j\frac{2\pi mk}{N}} h(n-m) = e^{j\frac{2\pi nk}{N}} X(n,k) \tag{3.5}$$

The bandpass signal $y_k(n)$ will be complex-valued, since the filters are complex-valued. Therefore we can write it as

$$y_k(n) = \tilde{X}(n,k) = \left|X(n,k)\right| e^{j\tilde{\varphi}(n,k)} \tag{3.6}$$

and we can derive the relation between the output bandpass signal and the STFT

$$\tilde{X}(n,k) = e^{j\frac{2\pi nk}{N}} X(n,k) \tag{3.7}$$

$$\tilde{\varphi}(n,k) = \frac{2\pi k}{N} n + \varphi(n,k) \tag{3.8}$$

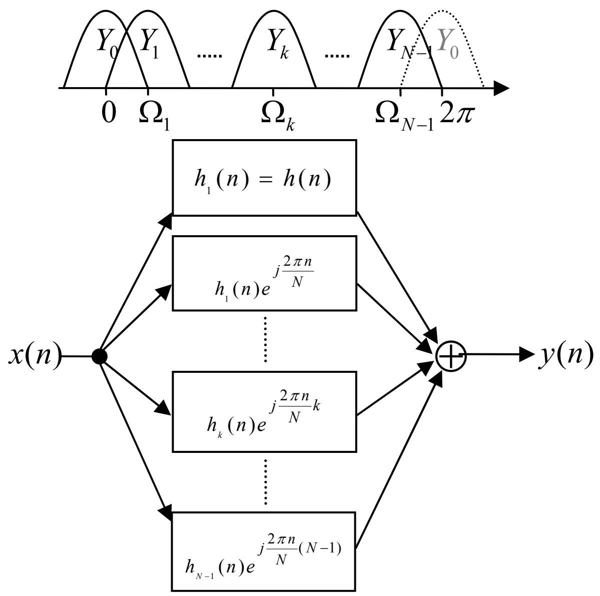


Figure 3.3 Filter bank description of the short-time Fourier transform. The frequency bands Y_k corresponding to each bandpass signal $y_n(n)$ are shown on the top.

We can understand $\tilde{X}_k$ as the bandpass signal obtained from the modulation of the baseband signal $X(n,k)$ with $e^{j\frac{2\pi nk}{N}}$, and the baseband signal $X(n,k)$ as the result of lowpass filtering by $h(n)$ the modulation of $x(n)$ with $e^{-j\frac{2\pi nk}{N}}$ (see Figure 3.2).

The output sequence $y(n)$ will be the sum of all the bandpass signals, as shown in Figure 3.3.

$$y(n) = \sum_{k=0}^{N-1} y_k(n) = \sum_{k=0}^{N-1} \tilde{X}(n,k) = \sum_{k=0}^{N-1} X(n,k)e^{j\frac{2\pi nk}{N}} \tag{3.9}$$

In the case of a real-valued input signal $x(n)$, the bandpass signals will satisfy

$$y_k(n) = \tilde{X}(n,k) = \tilde{X}^*(n,N-k) = y^*{}_{N-k}(n) \tag{3.10}$$

and therefore each bandpass will be also real-valued

$$\begin{aligned} \hat{y}_k(n) &= \tilde{X}(n,k) + \tilde{X}(n,N-k) = \tilde{X}(n,k) + \tilde{X}^*(n,k) \\ &= \left|X(n,k)\right|\left[e^{j\tilde{\varphi}(n,k)} + e^{-j\tilde{\varphi}(n,k)}\right] \\ &= 2\left|X(n,k)\right|\cos\left[\tilde{\varphi}(n,k)\right], \quad k = 1,...,\mathrm{N/2}-1 \end{aligned} \tag{3.11}$$

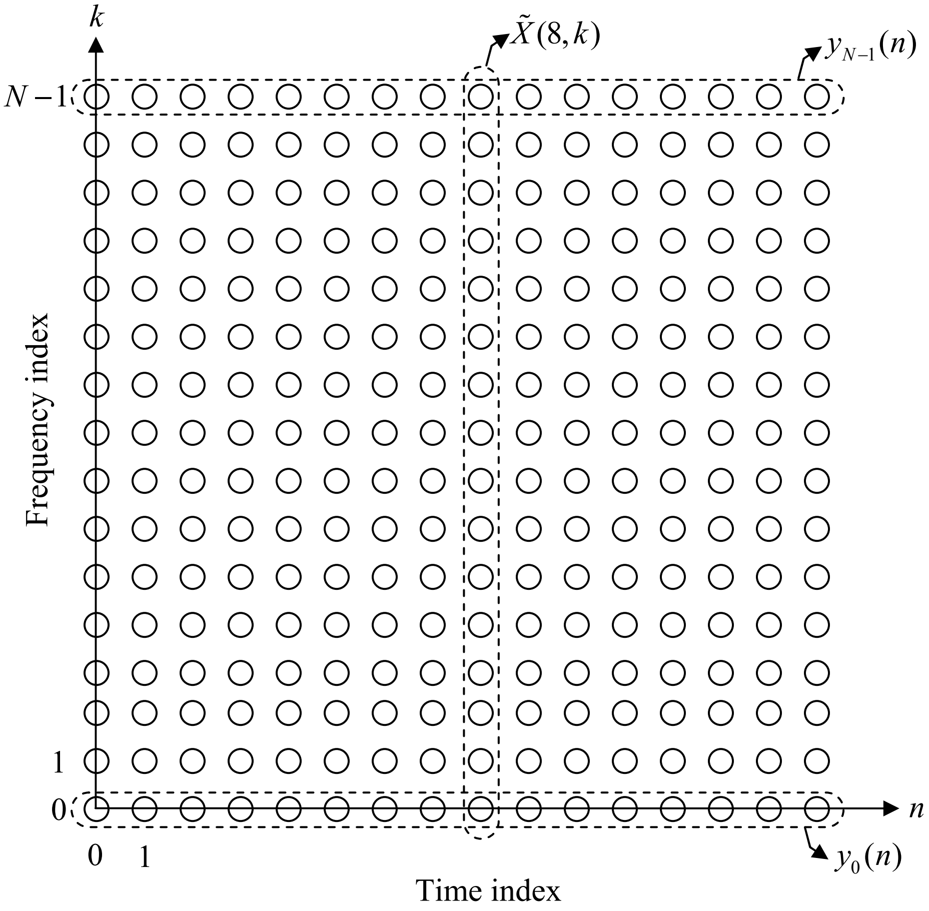


Figure 3.4 Discrete time-frequency plane

For the dc and highpass channels we have $\hat{y}_0(n) = y_0(n)$ and $\hat{y}_{N/2}(n) = y_{N/2}(n)$. This way we have a dc channel, a higpass channel and $N/2$ cosine signals with fixed frequencies Ω_k and time-varying amplitude and phase. We can find a detailed implementation in [94].

In the discrete time-frequency plane, each bandpass signal $y_k(n)$ corresponds to a horizontal line, as shown in Figure 3.4, and each sample of $y(n)$ corresponds to the sum of a vertical line, i.e. $y(n) = \sum_{k=0}^{N-1} \tilde{X}(n,k)$.

3.3.2 Block-by-block analysis/Synthesis model using FFT

The filter bank interpretation of the phase vocoder shows an analysis of a signal by a filter bank, a modification of the time-varying spectrum on a sample-by-sample basis for each bandpass signal, and a synthesis by summation of all the bandpass signals. The baseband signals are bandlimited by the lowpass filter $h(n)$, and this allows a sampling

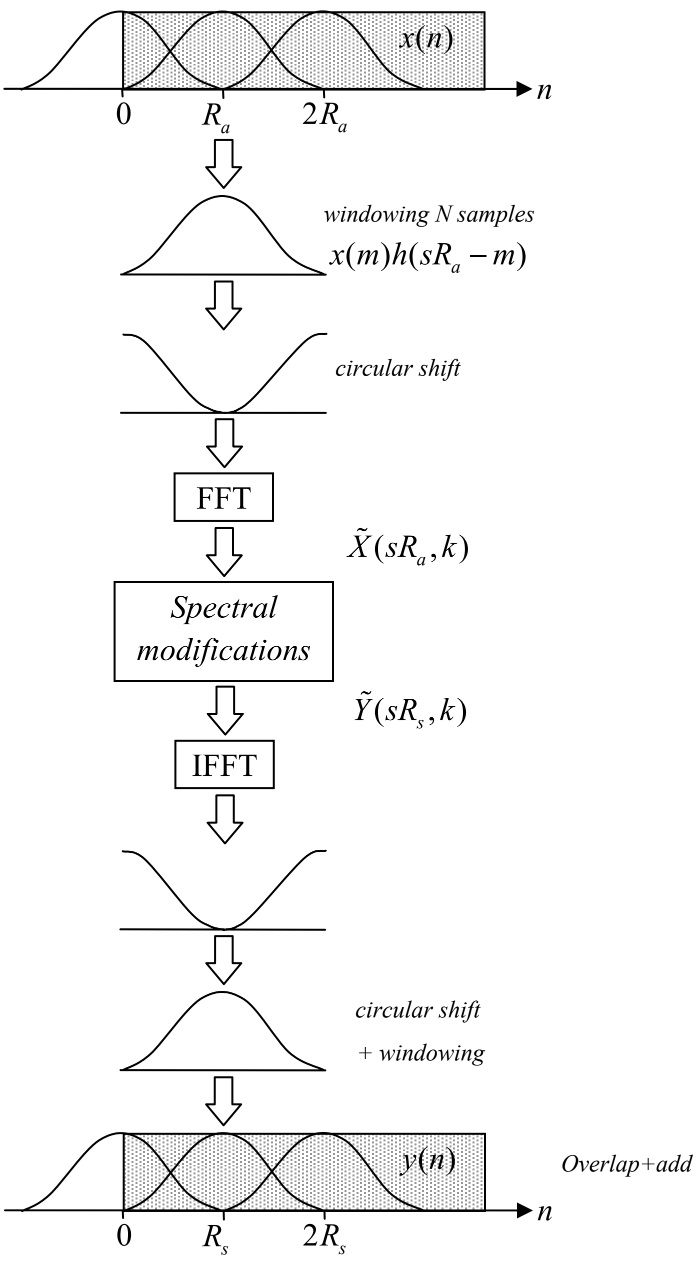


Figure 3.5 Block-by-block phase-vocoder using FFT/IFFT

rate reduction that can be performed in each channel to get $X(sR,k)$, where s denotes the time index and only every Rth sample is taken. At this point, it's easy to see that we could use a short-time transform $X(sR,k)$ with a hop size of R samples. However we should do upsampling and interpolation filtering before the synthesis [16].

We can find a description of the block-by-block phase-vocoder using FFT in [15, 16, 69] and the analysis and synthesis implementations in [16, 94]. The analysis gives as

$$\begin{aligned}X(sR_a,k) &= \sum_{m=-\infty}^{\infty} x(m)h(sR_a-m)e^{-j\frac{2\pi mk}{N}} \\ &= e^{-j\frac{2\pi sR_ak}{N}} \sum_{m=-\infty}^{\infty} x(m)h(sR_a-m)e^{j\frac{2\pi(sR_a-m)k}{N}} \\ &= e^{-j\frac{2\pi sR_ak}{N}} \tilde{X}(sR_a,k) \\ &= X_r(sR_a,k)+jX_i(sR_a,k)=\left|X(sR_a,k)\right|e^{j\varphi(sR_a,k)}\end{aligned} \tag{3.12}$$

where $X(sR_a,k)$ is the short-time Fourier transform sampled every R_a samples in time, s is the time index, k the frequency index and $h(n)$ is the analysis window. It's important to notice that we yet find $X(n,k)$ and $\tilde{X}(n,k)$ as in the filter bank approach.

After spectral modifications we get $Y(sR_s,k)$, where R_s denotes the synthesis hop size. The output signal $y(n)$ is given by

$$y(n)=\sum_{s=-\infty}^{\infty} f(n-sR_s)y_s(n-R_s) \tag{3.13}$$

where $f(n)$ is the synthesis window and $y_s(n)$ is obtained from inverse transforms of the short-time spectra $Y(sR_s,k)$ as follows

$$y_s(n)=\frac{1}{N}\sum_{k=0}^{N-1}\left[\mathrm{e}^{\mathrm{j}\frac{2\pi \mathrm{sR_s}k}{N}}Y(sR_s,k)\right]e^{j\frac{2\pi nk}{N}} \tag{3.14}$$

As shown in (3.13), each $y_s(n)$ is weighted by the synthesis window and then added to $y(n)$ by overlap-add.

In [16] is described a zero-phase analysis and synthesis regarding the center of the window by applying a circular shift of the windowed segment before the FFT and after the IFFT. In Figure 3.5 is shown the overall analysis/synthesis block-by-block procedure.

Modifying the phase of the spectrum before the IFFT is equivalent to using an all-pass filter whose Fourier transform shows the phase correction applied. If we do not apply a synthesis window we will have discontinuities at the edges of the signal buffer because of the circular convolution aspect of the filtering operation. Therefore, it is needed to apply a synthesis window. But the shape of the analysis and synthesis windows should meet some conditions. It's better to use windows whose Fourier transform presents small side lobes. Besides, the sum of the result of the multiplication of the analysis and synthesis windows, regularly spaced at the synthesis hop size, should be one for a perfect reconstruction with no modifications, or approximately one with oscillations below the level of perception.

3.4 Time-scale modification using the phase-vocoder

There are two implementations of the *traditional* phase vocoder that have been historically used for time-stretching. The first one used the filter bank approach: a bank of oscillators whose amplitudes and frequencies vary over time. The scale modification was achieved by scaling the amplitude and frequency functions (see section 3.4.2). The second approach used the sliding STFT. Here the idea was to spread the image of the STFT over time, keep the magnitude unchanged and calculate new phases for each bin in such a way that the instantaneous frequencies were preserved (see section 3.4.3).

3.4.1 Phase unwrapping and instantaneous frequency

Both filter bank and block-by-block approaches rely on phase interpolation, which need an unwrapping algorithm at the analysis stage or an instantaneous frequency calculation. We start from (3.8)

$$\tilde{\varphi}(n,k) = \frac{2\pi k}{N} n + \varphi(n,k) = \Omega_k n + \varphi(n,k) \tag{3.15}$$

From an analysis with hop size R_a , we get the phase values $\tilde{\varphi}(sR_a,k)$ and $\tilde{\varphi}\left(\left(s+1\right)R_a,k\right)$ from consecutives frames. Both values belong to the same range $]-\pi,\pi]$, regardless of the frequency index k . If we assume that a stable sinusoid with frequency Ω_k exists, we can compute the target phase $\tilde{\varphi}_t\left(\left(s+1\right)R_a,k\right)$ from the previous phase value $\tilde{\varphi}(sR_a)$ as

$$\tilde{\varphi}_t\left(\left(s+1\right)R_a,k\right) = \tilde{\varphi}(sR_a,k) + \Omega_k R_a \tag{3.16}$$

And the unwrapped phase can be calculated adding a deviation phase to the target phase according to

$$\tilde{\varphi}_u\left(\left(s+1\right)R_a,k\right) = \tilde{\varphi}_t\left(\left(s+1\right)R_a,k\right) + \text{princarg}\left[\tilde{\varphi}\left(\left(s+1\right)R_a,k\right) - \tilde{\varphi}_t\left(\left(s+1\right)R_a,k\right)\right] \tag{3.17}$$

where princarg is a function called principle argument [17] that puts an arbitrary phase value into the range $]-\pi,\pi]$ by adding/subtracting multiples of 2π . From the previous equation we can formulate the unwrapped phase in terms of the output analysis phase values as

$$\tilde{\varphi}_u\left(\left(s+1\right)R_a,k\right) = \tilde{\varphi}(sR_a,k) + \Omega_k R_a + \text{princarg}\left[\tilde{\varphi}\left(\left(s+1\right)R_a,k\right) - \tilde{\varphi}\left(sR_a,k\right) - \Omega_k R_a\right] \tag{3.18}$$

From the unwrapped phase difference between consecutive frames we can calculate the instantaneous frequency for frequency bin k at time instant $\left(s+1\right)R_a$ by

$$
\begin{aligned}
f_i\left(\left(s+1\right)R_a,k\right) &= \frac{\Delta\varphi\left(\left(s+1\right)R_a,k\right)}{2\pi R_a} f_s \\
&= \frac{\Omega_k R_a + \operatorname{princarg}\left[\tilde{\varphi}\left(\left(s+1\right)R_a,k\right) - \tilde{\varphi}\left(sR_a,k\right) - \Omega_k R_a\right]}{2\pi R_a} f_s
\end{aligned}
\tag{3.19}
$$

3.4.2 Filter bank approach

In this approach the hop size factor R_s / R_a is changed in order to get the desired time-scale modification. Since the synthesis algorithms consists of a sum of sinusoids (see eq. (3.9) and (3.11)), we calculate for each bin the corresponding phase increment per sample at time index sR_a as

$$
d\varphi\left(k\right) = \frac{\Delta\varphi\left(\left(s+1\right)R_a,k\right)}{R_a} \tag{3.20}
$$

and integrate it sample by sample to get the synthesis phases (denoted by $\tilde{\phi}$)

$$
\tilde{\phi}(n+1,k) = \tilde{\phi}(n,k) + d\varphi(k) \tag{3.21}
$$

Notice that the synthesis unwrapped phase difference will depend on the hop size relation

$$
\Delta\theta\left(sR_s,k\right) = \frac{R_s}{R_a}\Delta\varphi\left(sR_a,k\right) \tag{3.22}
$$

3.4.3 Block-by-block approach

As in 3.4.2, in this approach the synthesis hop size is modified in order to get the stretched output audio signal. Using the FFT algorithm, this approach is much faster than the previous one. For each bin, we calculate the new phase value as

$$
\tilde{\theta}(sR_s,k) = \tilde{\varphi}(sR_a,k) + \frac{R_s}{R_a}\Delta\varphi\left(\left(s+1\right)R_a,k\right) \tag{3.23}
$$

The amplitude of the spectrum is unchanged. It's important to notice that the synthesis hop size should at least allow a minimal overlap of windows.

3.5 Phase-locked vocoder

An important problem of the phase-vocoder is the difference of phase unwrapping between different bins. The phase unwrapping algorithm (see §3.4.1) gives a phase equal to the measured phase plus a term that is a multiple of 2π , but this second term is not the same for all bins. Once multiplied by the time stretching ratio, see equation (3.23), this produces dispersion of the phases. Maybe this is the main drawback of the phase-vocoder and its removal is still a matter of research. Only for integer time stretching ratios this phase dispersion is avoided, because the 2π modulo relation is still preserved when the phase is multiplied by an integer. This phase consistency across channels (or bins) within a synthesis frame is referred to as *vertical phase coherence*.

3.5.1 Loose phase-locking

Miller Puckette [70] pointed out 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. In fact, a constant-amplitude, constant-frequency sinusoid in channel k should have identical analysis phases in all nearby channels. He proposed a simple way to loosely constrain synthesis phase values: for each channel, the synthesis phase is calculated with the standard phase-propagation formula, but the final synthesis phase is that of the complex number

$$\tilde{\theta}(sR_s, k-1) + \tilde{\theta}(sR_s, k) + \tilde{\theta}(sR_s, k+1) \tag{3.24}$$

As a result, if bin k is next to a maximum, its phase will be approximately that of the maximum $\tilde{\theta}(sR_s, k_{\max})$. Otherwise, if channel k is a maximum, its phase will be basically unchanged, because the neighbor bins have much loser amplitude (thus, much lower contribution to the average phase). This technique is especially attractive because its simplicity. However, informal listening tests show that the reduction in phasiness is very signal-dependent and never dramatic [47].

3.5.2 Rigid phase-locking

The main limitation of the loose phase-locking algorithm is that it avoids any explicit determination of the signal structure. The result is that the synthesis phase values around a given sinusoid only gradually and approximately show vertical phase coherence. To really restore vertical phase coherence we should get closer to an actual sum-of-sinusoids model. Laroche and Dolson [47] proposed a new phase-locking technique inspired on the loose phase locking but based on explicit identification of peaks in the spectrum and the assumption that they are sinusoids.

The first step is a simple peak detection algorithm, where local maximums are considered as peaks. Then the series of peaks subdivides the frequency axis into *regions of influence* located around each peak. The region boundary can be set to the middle frequency between adjacent peaks or to the lowest amplitude between the two peals. The basic idea is to calculate the phases only for the peak channels (local maximums) and *lock* the phase of the remaining channels within each region to their peak channel phase. This kind of phase-locking is called *rigid phase locking*. Laroche and Dolson proposed two ways of rigid phase-locking: identity and scaled phase locking.

3.5.2.1 Identity phase-locking

In identity phase-locking, the synthesis phases around the peak are constrained to be related in the same way as the analysis phases. Thus, the phase differences between successive channels around a peak in the analysis and synthesis Fourier transform are identical. If k_{peak} is the channel index of the dominant peak, we set

$$\begin{gathered}\tilde{\theta}(sR_s,k)=\tilde{\theta}(sR_s,k_{peak})+\tilde{\varphi}(sR_a,k)-\tilde{\varphi}(sR_a,k_{peak})\\ k\in\text{Region of influence}(k_{peak})\end{gathered} \tag{3.25}$$

for all channel k in the peak's region of influence. This idea of preserving the phase-relations between nearby bins had been independently proposed by Quatieri et al in [72] as a way to reduce transient smearing, and by Ferreira [26] in the context of integer-factor time-scale modifications.

A major advantage of this method is that only peak channels require trigonometric calculations (for phase unwrapping). And the rest of channels within the area of influence only require one complex multiply, as shown below

$$\begin{gathered}Y(sR_s,k)=ZX(sR_a,k)\\ Z=e^{j\left[\tilde{\theta}(sR_s,k_{peak})-\tilde{\varphi}(sR_a,k_{peak})\right]}\end{gathered} \tag{3.26}$$

3.5.2.2 Scaled phase-locking

The identity phase-locking can be improved by detecting peaks switching from one channel to another one. For example, if a peak switches from channel k_0 at time index s to channel k_1 at time index $s+1$, the unwrapping equation (3.18) should be

$$\begin{aligned}\tilde{\varphi}_u\left(\left(s+1\right)R_a,k_1\right)=\tilde{\varphi}(sR_a,k_0)+\Omega_{k_1}R_a+&\\ \operatorname{princarg}\left[\tilde{\varphi}\left(\left(s+1\right)R_a,k_1\right)-\tilde{\varphi}\left(sR_a,k_0\right)-\Omega_{k_1}R_a\right]&\end{aligned} \tag{3.27}$$

and the instantaneous frequency

$$f_i\left(\left(s+1\right)R_a,k_1\right)=\frac{\Omega_{k_1}R_a+\operatorname{princarg}\left[\tilde{\varphi}\left(\left(s+1\right)R_a,k_1\right)-\tilde{\varphi}\left(sR_a,k_0\right)-\Omega_{k_1}R_a\right]}{2\pi R_a}f_s \tag{3.28}$$

The identity phase-locking equation can be generalized to

$$\begin{gathered}\tilde{\theta}(sR_s,k)=\tilde{\theta}(sR_s,k_{peak})+\beta\left(\tilde{\varphi}(sR_a,k)-\tilde{\varphi}(sR_a,k_{peak})\right)\\ k\in\text{Region of influence}(k_{peak})\end{gathered} \tag{3.29}$$

where β is a phase scaling factor. If $\beta=1$ then we would have identity phase-locking. It appears that identity phase-locking can be further improved by setting β to a value between one and the time scale factor α. Informal listening tests [47] showed that phasiness is further reduced by setting $\beta \approx 2/3 + \alpha/3$. The quality of the time-scaled signal has been found to be consistently higher with scaled phase-locking than with identity phase-locking. However, the phases of the channels around the peaks must be unwrapped before applying (3.29) in order to avoid $2\beta\pi$ channel jumps in the synthesis phases, and therefore scaled phase-locking requires more computation than identity phase-locking.

3.6 Multiresolution phase-locked vocoder

Hoek points out in his patent [43] a solution for the compromise between frequency and time resolution. As known, big windows achieve a good frequency resolution but poor time resolution, in opposite to short windows. Instead of the typical multiband or wavelet approach, he proposes to convolve the spectrum with a variable kernel function in order to get a variable windowed spectrum. The variable kernel function can be understood as a filter parameterized by a function $s(f)$, which specifies how the behavior of the filter varies with frequency. This filter can be described by the recursive relation

$$y_{out}(f) = \left[1 - s(f)\right] y_{in}(f) + s(f) y_{out}(f-1) \tag{3.30}$$

In fact, a different convolution kernel is used for each frequency bin, and this technique can be considered as an efficient way to process a signal trough a large bank of filters where the bandwidth of each filter is individually controllable by the control function $s(f)$.

He proposes a control function $s(f)$ that approximates the excitation response of human cilia located on the basilar membrane in the human ear, thus approximates the time/frequency response of the human ear:

$$s(f) = 0.4 + 0.26 \operatorname{arctg}\left(4 \ln(0.1 f) - 18\right) \tag{3.31}$$

where f is the frequency expressed in hertz.

Since the signal was analyzed with different temporal resolutions at different frequencies, the synthesized time domain signal is only valid in the region equivalent to the highest temporal analysis resolution used (shortest window). In consequence, the output of the IFFT should be windowed with a small window before the overlap and add procedure.

This technique has been implemented by Hoek in the commercial plug-in software *Pitch'n Time* from Serato [5].

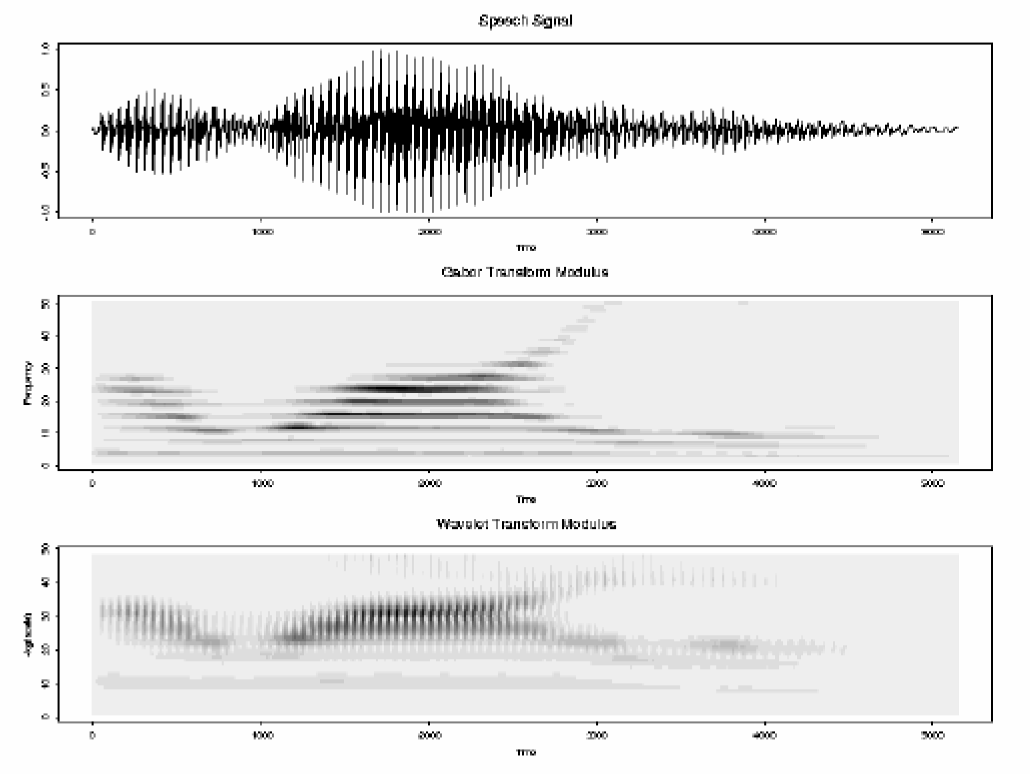


Figure 3.6 Fixed window size and CQFB spectrograms of speech. Graphics from [87]

3.7 Wavelet transform

Wavelet transform [87] deals with the time-frequency resolution limitations of fixed-size windowing. The wavelet transform uses analysis windows (wavelets) that dilate according to the frequency being analyzed. Ellis proposed in [24] a constant-Q analysis, with a filterbank where the bandwidth of individual elements increased with frequency, trying to emulate the response of the human auditory system (CQFB, Constant-Q filterbank). In Figure 3.6 we can see two spectrograms of a speech signal, one with a fixed window size and the other with a window length decreasing with frequency. In the first one we can distinct the harmonics at low and high frequencies. However, on the second one, the analysis varies from narrowband behavior at low frequencies, resolving the first few harmonics, to wideband behavior at high frequencies, resolving the individual pitch pulses (high time resolution) but not the harmonic frequencies. One interesting property of the CQFB is that the point at which harmonics begin to fuse into pitch pulses happens always at the same harmonic index, regardless of the pitch.

Pallone et al [66, 67] recently implemented a real-time wavelet based method for time-scale modification of audio. They implemented a filterbank based on the Bark scale

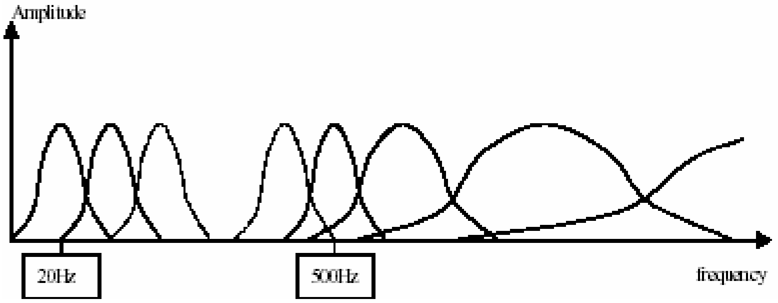


Figure 3.7 Filter bank implemented by Pallone et al. Illustration from [32]

[95] with a constant bandwidth of 20Hz from 20Hz to approximately 500Hz (similar to the phase vocoder analysis), and bandwidth of 1/10 octave for higher frequencies (similar to the standard wavelet analysis [45]). The characteristics of this filter bank are shown in Figure 3.7. The goodness of this filterbank is that it preserves the good frequency resolution of the phase vocoder at low frequencies while achieving a good time resolution at high frequencies (important for a good representation of transient signals).

The final goal of their system was to synchronize audio and video in broadcasting applications avoiding the pitch change on the audio due to the change of frame rate when converting between video and cinema formats. Thus, the ratio of desired time-stretching is within 24/25 to 25/24 (-4% to +4%). In fact, a pitch shifting is equivalent to a time stretching without change of pitch followed by a downsampling. Thus, with a resampling it is equivalent to consider a pitch shifting or a time scaling problem [46]. When the film is projected at a different rate, a pitch shifting of ratio α occurs. If $X(\omega)$ is the Fourier transform of the input signal $x(t)$, then the transformed signal $y(t)$ will be

$$y(t) = \int_{-\infty}^{\infty} X(\alpha\omega)e^{j\omega t}d\omega = \frac{1}{\alpha}x\left(\frac{t}{\alpha}\right) \tag{3.32}$$

Therefore, a pitch shifting of a ratio α implies a time scaling of a factor $1/\alpha$. Using the filterbank above mentioned, the modulus and phase of each sub-band analytic signal are extracted. We can denote these signals as

$$x_i(t) = M_i(t)e^{j\varphi_i(t)} \tag{3.33}$$

where $M_i(t)$ and $\varphi_i(t)$ are respectively the modulus and phase of the i^{th} sub-band. To achieve a pitch shifting of β, the modulus of each sub-band is kept and the phase is multiplied by the transposition ratio.

$$\begin{aligned} M_i^{'}(t) &= M_i(t) \\ \varphi_i^{'}(t) &= \beta\varphi_i(t) \end{aligned} \tag{3.34}$$

The transposed output signal will be the sum of the real parts of each sub-band

$$y(t) = \sum_i M_i^{'}(t)\cos\left(\varphi_i^{'}(t)\right) = \sum_i M_i(t)\cos\left(\varphi_i^{'}(t)\right) \tag{3.35}$$

When the output signal $y(t)$ is played at a different rate $r' = \alpha r$, it will be transposed by the rate factor α, as shown in equation (3.32). Thus, we get

$$y' = \frac{1}{\alpha} y\left(\frac{t}{\alpha}\right) \tag{3.36}$$

If $\beta = 1/\alpha$, the transposition factors will be compensated, and the final output will be a time scaled version of the original signal $x(t)$ without pitch change. This system has been implemented as a real-time multichannel pitch-shifter hardware called “Harmo” by the company genesis [2], that is used in post-production studios, improving the previous popular system Lexicon 2400, which used a TD-PSOLA like technique with transient handling.

3.8 Combined harmonic and wavelet representations

Hamdy et al proposed in [41] a time-scale modification method for high quality audio signals based on a combination of harmonic and wavelet representations. The algorithm block diagram is shown in Figure 3.9. The input signal is initially modeled as a sum of sinusoids (*Harmonic analysis* block) which are synthesized (*Reconstruct Harmonics* block) and subtracted from the original sound to get the residual. This residual is analyzed with the wavelet transform (*WT* block) and decomposed into edges and noise. Both harmonic and residual components are time scaled, synthesized and finally added to get the output sound.

Harmonic component time scaling

Harmonics are detected frame by frame using a technique developed by Thomson [86] and then incorporated to frequency tracks. As shown in Figure 3.9, each harmonic component is demodulated down to DC, and then lowpass filtered. The lowpass filter bandwidth is selected to correspond to the lower limits on the just noticeable frequency difference that human auditory system can perceive. Thus, it is increased with frequency. The in-phase and quadrature signals obtained from the output of the filter are interpolated and decimated to achieve the desired time-scaling $\alpha = M / N$, where M is the interpolation factor and N is the decimation factor. Finally, these signals are modulated back to their original frequency.

Residual component time scaling

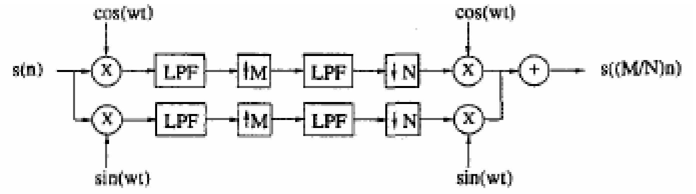


Figure 3.9 Harmonic component time-scaling block diagram. Illustration from [41]

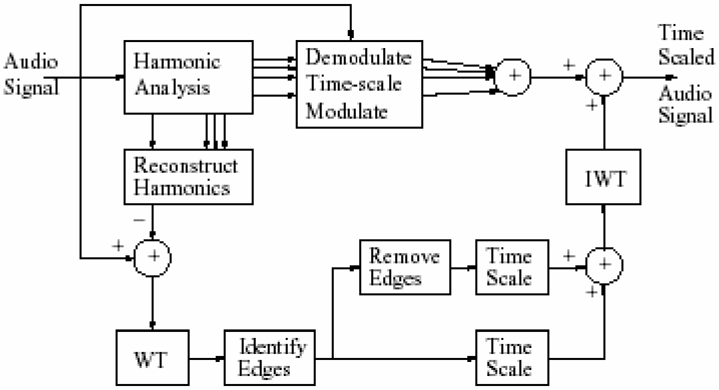


Figure 3.9 Block diagram of the time scale using combined harmonic and wavelet representations. Illustration from [41]

The residual component contains transients (edges) and noise, which need to be treated separately. It is analyzed with a wavelet transform whose frequency bands are designed to correspond to the critical band structure of the human auditory system. Edges are detected by comparing the relative energy within each band. High energy regions are considered edge regions and tracked frame by frame. They are time-scaled using a coherent time-scale modification that relies on Hadamard transforms. The noisy regions of the residual are time-scaled by interpolation and decimation. Finally, the residual is resynthesized with the inverse wavelet transform.

3.9 Constant-Q Phase Vocoder by exponential sampling

Garas and Sommen proposed in [33] an improvement to the phase-vocoder based on warping techniques that transform the constant-bandwidth spectral resolution into a constant-Q resolution, with low increase of complexity. A way to get a constant-Q resolution is to warp the frequency co-ordinate f to $\gamma(f) = f_0 \cdot a^f$, where f_0 is the smallest frequency of interest, and a controls the frequency bin spacing and bandwidth of the warped frequency co-ordinate. This frequency warping is equivalent to a time warping of $\gamma(t) = t_0 \cdot a^t$, as demonstrated in [34]. Thus, the input signal is first sampled nonuniformly and then a short-time Fourier transform is applied. The resulting transform in this case is given by

$$s(f) = \sqrt{\log(a)} \int_{0}^{\infty} \frac{s(t)}{\sqrt{t}} e^{j2\pi f \log_a(t)} dt \tag{3.37}$$

which performs a constant-Q analysis similar to the analysis made by the human auditory system. The variable a can be adjusted for better quality or higher time scaling ratios.

3.10 Conclusions

The phase-vocoder is a relative old technique that dates from 70's. It is a frequency domain algorithm computationally quite more expensive than time domain algorithms introduced in chapter 2. However it can achieve high quality results even with high time-scale factors. Basically, the input signal is splitted into many frequency channels, uniformly spaced, usually using FFT. Each frequency band (bin) is decomposed into magnitude and phase parameters, which are modified and resynthesized by the IFFT or a bank of oscillators. With no transformations, the system allows a perfect reconstruction of the original signal. In the case of time-scale modification, the synthesis hop size is changed according to the desired time-scale factor. Magnitudes are linearly interpolated and phases are modified in such a way that maintains phase consistency across the new frame boundaries.

The phase-vocoder introduces signal smearing for impulsive signals due to the loss of phase alignment of the partials. For example, the impulsive characteristic of speech is given by the synchronization of harmonics' phase at the beginning of each glottal period. Since each spectral peak is processed independently by the phase-vocoder algorithm this synchronization is lost and it results into the above commented smearing effect.

A typical drawback of the phase vocoder is the loss of vertical phase coherence that produces reverberation or loss of presence in the output. This effect is also referred to as phasiness. Recently, the synthesis quality has been improved applying phase-locking techniques (Puckette, Laroche and Dolson) among bins around spectral peaks. Adding peak tracking to the spectral peaks, the phase-vocoder seems to get close to the sinusoidal modeling algorithms, which will be introduced in next chapter 4.

Another traditional drawback of the phase vocoder is the bin resolution dilemma: the phase estimates are incorrect if more than one sinusoidal peak reside within a single spectral bin. Increasing the window may solve the phase estimation problem, but it implies a poor time resolution and smoothes the fast frequency changes. And the situation gets worse in the case of polyphonic music sources because then the probability is higher that sinusoidal peaks from different sources will reside in the same spectrum bin.

Recently, Hoek patented a promising technique that allows different temporal resolutions at different frequencies by a convolution of the spectrum with a variable kernel function. Thus, long windows are used to calculate low frequencies, while short windows are used to calculate high frequencies. Finally, Garas and Sommen presented a different approach to get a constant-Q phase-vocoder by nonuniform sampling.

4 Signal models

4.1 Introduction

In this section we will briefly review several signal modeling techniques, starting with the sinusoidal model and then adding residual and transient models. Special interest will be put about partial estimation algorithms. Finally a multiresolution approach built on top of a sinusoidal+residual+transient model will be presented.

4.2 Sinusoidal Modeling

4.2.1 McAulay-Quatieri

The technique developed by McAulay and Quatieri [60, 61] is based on modeling the time-varying spectral characteristics of a sound as sums of time-varying sinusoids. The input sound $x(n)$ is modeled by

$$x(n)=\sum_{i=1}^{N} A_i(n)\cos\left[\theta_i(n)\right] \tag{4.1}$$

where $A_i(n)$ and $\theta_i(n)$ are the instantaneous amplitude and phase of the i^{th} sinusoid. The instantaneous phase is taken to be the integral of the instantaneous frequency $\omega_i(n)$ as denoted by

$$\theta_i(n)=\int_{0}^{nT}\omega_i(\tau)d\tau \tag{4.2}$$

To obtain a sinusoidal representation from a sound, an analysis is performed in order to estimate the instantaneous amplitudes and phases of the sinusoids. This estimation is generally done by first computing the STFT of the sound, then detecting the spectral peaks and measuring their magnitude, frequency and phase, as shown in Figure 4.2. The peaks are then linked over successive frames to form tracks. Each track represents the time/varying behavior of a single sinusoidal component in the analyzed sound. The synthesis is done through a bank of sine wave oscillators. For each track, the measured magnitudes are linearly interpolated and the frequencies are interpolated with a cubic phase interpolation, in order to preserve the phase accuracy at frame boundaries. In fact, this phase interpolation is based on the minimization of the mean square of the second derivative of the analysis phase. McAulay and Quatieri used this technique for

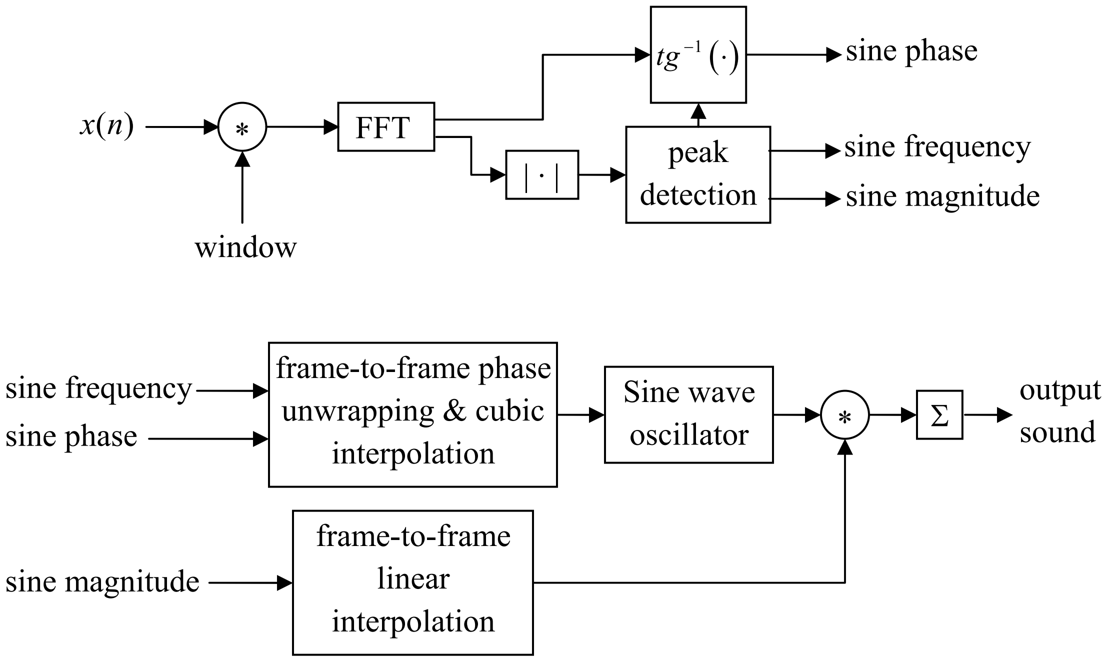


Figure 4.2 Block diagram of the sinusoidal analysis (top) and synthesis (bottom), from [60]

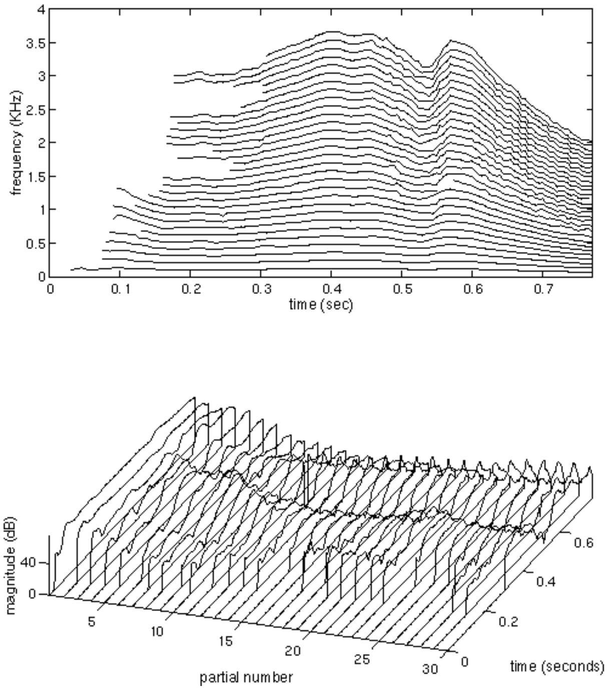


Figure 4.2 Sinusoidal trajectories resulting from the analuysis of a vocal sound.

harmonic sounds (basically speech), and Smith and Serra [82] extended it to nonharmonics sounds. The traditional way of performing time scale modifications using a sinusoidal model is to resample the frequency and amplitude tracks at higher or lower

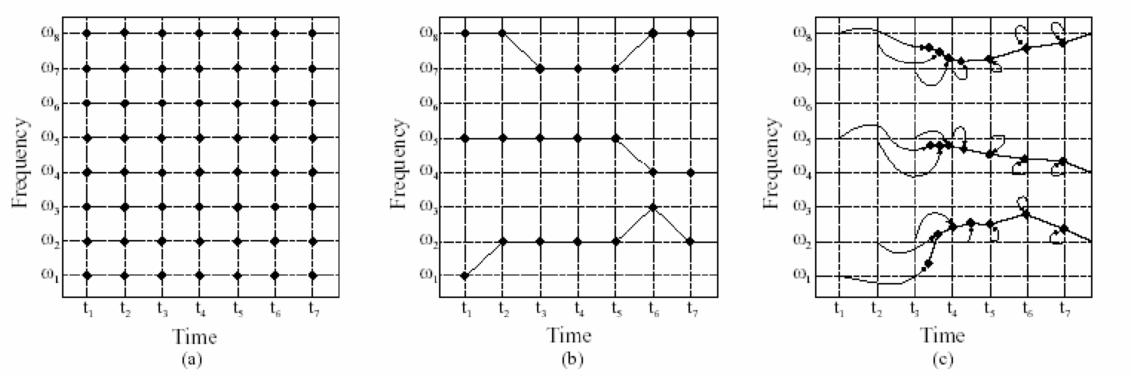


Figure 4.3 Comparison of time-frequency data included in common representations. The STFT (a) retains data at every point of the time-frequency plane. The McAulay-Quatieri method (b) retains data at selected time and frequency samples. The reassigned bandwidth-enhanced model (c) distributes data continuously in the time-frequency plane, and retains data only at time-frequency ridges. The arrows in Figure show the mapping of short-time spectral samples to time-frequency ridges, due to the method of reassignment. Illustration from [27], p.84.

rates, without affecting the spectral envelope. This method makes no difference between monophonic signals and a mix, but fails to handle the transients correctly.

4.2.2 Reassigned Bandwidth-enhanced sinusoidal modeling: Lemur & Loris

Fitz and Haken [30] developed and extended version of the McAulay and Quatieri system called LEMUR. This tool analyzed sampled sounds and generated a data file with the output of the analysis. With Lemur's built-in editing functions was possible to modify these files in different ways to finally synthesize the modified data and create a new sampled signal.

Recently, Fitz and Haken expanded LEMUR [29] and developed Loris [28], an Open Source C++ class library that implements the Reassigned Bandwidth-Enhanced Additive Model, an extension to the sinusoidal model. Bandwidth-Enhanced expands the notion of a partial to include the representation of both sinusoidal and noise energy by a single component type. Each partial is defined by three breakpoint envelopes that specify the time-varying amplitude, center frequency and noise content (represented by its bandwidth).

This technique shares with traditional sinusoidal methods the notion of temporally connected partial parameter estimates, but by contrast, the reassigned estimations are non-uniformly distributed in both time and frequency. This yields a greater resolution in

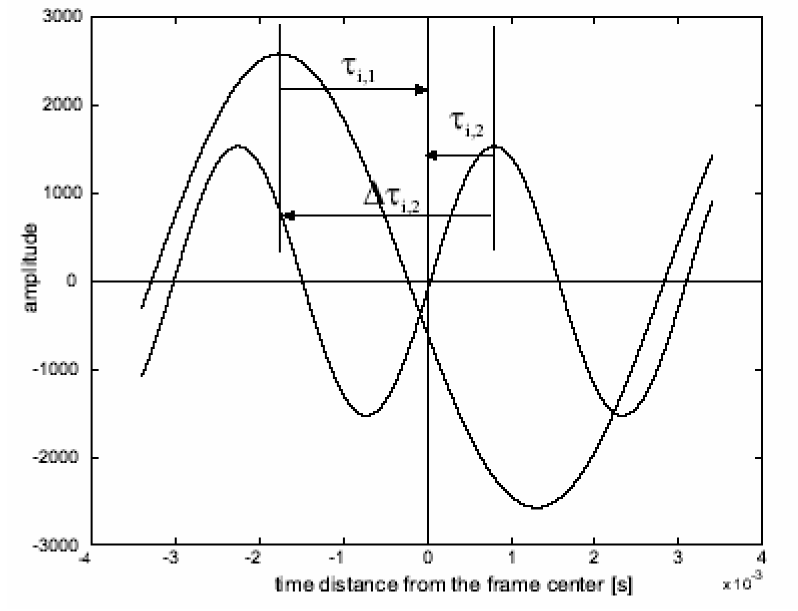


Figure 4.4 Relation between phase delays τ_k and relative phase delays $\Delta\tau_k$. The center of the analysis frame matches the origin of the time axis. Illustration from [21]

time and frequency than the one possible using conventional additive techniques (see Figure 4.3). This method of reassignment was first introduced by Auger and Flandrin [9].

4.2.3 Waveform preservation based on relative delays

Di Federico [21] 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, defined as the difference between the phase delay of the partials and the phase delay of the fundamental. This representation allows the waveform to be independent from the phase of the first partial.

Starting from the sinusoidal model equations (4.1) and (4.2), partial phases are transformed into phase delays

$$\tau_k = \frac{\theta_k}{\omega_k} \tag{4.3}$$

These phase delays can be interpreted as the temporal distance between the frame center and the nearest partial maximum, as shown in Figure 4.4. The waveform can be locally characterized by referring each phase delay to the phase delay of the first partial (fundamental). This can be done by defining the relative phase delays (*rpds*) as

$$\Delta\tau_k = \tau_k - \tau_1 \tag{4.4}$$

where τ_1 is the phase delay of the fundamental. The vector

$$\Delta\tau = \left\{\Delta\tau_k\right\} \qquad k = 2,...,N \tag{4.5}$$

describes completely the waveform without any dependence on the phase of the first partial. Therefore, the original waveform can be rebuilt by giving an arbitrary value for the phase of the fundamental and adding the relative phase delays for each partial.

This method extends the traditional way of performing time stretching by sinusoidal models, because besides amplitude and frequency, also the phase (rpd) can be interpolated or decimated. The basic idea is to consider the fundamental as a carrier for the upper partials. Thus, the fundamental phase is time-scaled using the unwrapping procedure proposed by McAulay and Quatieri (see §4.2.1), and the phases of the other partials are obtained by adding the *rpds* to the fundamental phase. This method is oriented to quasi harmonic monophonic signals and requires a pitch detection algorithm. It yields high quality results even for time-stretching factors up to 30 and more.

4.2.4 High Precision Fourier Analysis using Signal Derivatives

Desainte-Catherine and Marchand [19, 20, 55] improved the Fourier analysis precision using a k^{th}-order Fourier transform. Starting from the sinusoidal model, the input signal can be expressed as

$$s(t) = \sum_{p=1}^{p} a_p(t)\cos\left(\varphi_p\left(t\right)\right) \tag{4.6}$$

and the relation between frequency and phase is given by

$$\begin{gathered} \frac{d\varphi_p}{dt} = 2\pi f_p(t) \\ \varphi_p(t) = \varphi_p(0) + 2\pi\int_0^t f_p(\tau)d\tau \end{gathered} \tag{4.7}$$

where t is the time in seconds, p is the partial index, P the number of partials, f_p, a_p and φ_p respectively the frequency, amplitude and phase of the p^{th} partial. Assuming that frequency and amplitude are slow time-varying parameters, we could suppose that during a single analysis window of the STFT the frequency and amplitude derivatives are close to zero. In other words, derivating a sine gives a sine with a different phase but the same frequency.

$$\frac{ds}{dt}(t) = \sum_{p=1}^{P} 2\pi f_p(t)a_p(t)\cos\left(\varphi_p(t) - \frac{\pi}{2}\right) \tag{4.8}$$

Let note DFT^k the amplitude spectrum of the Discrete Fourier Transform of the *k*-th signal derivative

$$DFT^k(m) = \frac{1}{N}\left|\sum_{n=0}^{N-1} w(n)\frac{d^k s}{dt^k}[l+n]e^{-j\frac{2\pi m}{N}}\right| \tag{4.9}$$

where w is the 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. Approximate frequency and amplitude values for the partial p^{th} are

$$\begin{aligned} f_p^0 &= m_p\frac{f_s}{N} \\ a_p^0 &= DFT^0(m_p) \end{aligned} \tag{4.10}$$

where f_s is the sampling frequency. From equation (4.8) we can get much more accurate frequency and amplitude values:

$$\begin{aligned} f_p^1 &= \frac{1}{2\pi}\frac{DFT^1(m_p)}{DFT^0(m_p)} \\ a_p^1 &= \frac{a_p^0}{W\left(\left|f_p^1 - f_p^0\right|\right)} \end{aligned} \tag{4.11}$$

where $W(f)$ is the amplitude of the continuous spectrum of the analysis window w at frequency f. The window should be chosen as small as possible, with the only restriction that partials must lie in two different Fourier transform bins. Thus, with the DFT^1 method the window can be smaller than with the standard STFT, and a better time-resolution can be achieved, especially important for example when dealing with vibratos.

4.2.5 High Precision Fourier Analysis using the triangle algorithm

Recently, Althoff et al [6] proposed a new method to improve the estimation of sinusoid parameters. The idea is to use an analysis window function with a triangular Fourier transform. Thus, the absolute value of the wanted zero-phase frequency response would be

$$A(e^{j\Omega}) = \begin{cases} 1-\left|\dfrac{\Omega}{\Omega_c}\right| & , \quad |\Omega| < \Omega_c \\ 0 & , \quad \text{otherwise} \end{cases} \tag{4.12}$$

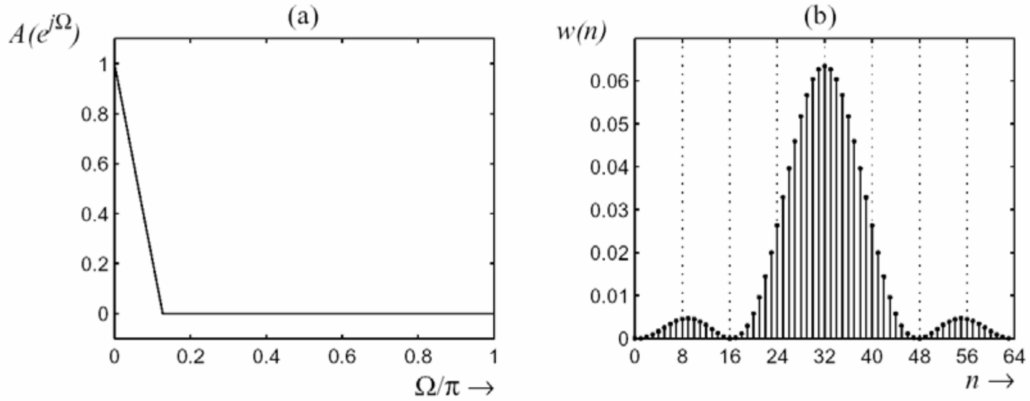


Figure 4.5 Triangular amplitude response and corresponding causal window for N=64. Illustration from [6]

for $|\Omega| < \pi$. In Figure 4.5 it is shown $A(e^{j\Omega})$ and the corresponding causal window. Frequency resolution is higher for small values of Ω_c while noisy sinusoids can be better detected with greater values of Ω_c. A good compromise is to choose $\Omega_c = 8\pi / N$ which results into a triangle length of 8 bins without zero-padding. If D is the length of the triangle, then it can be described by two lines, $h_1(k)$ on the left and $h_2(k)$ on the right.

$$\begin{aligned} h_1(k) &= ak + b \\ h_2(k) &= -a(k - D) + b \end{aligned} \tag{4.13}$$

with $a > 0$.

If k_m is a local maximum, the six closest spectral values surrounding $X_m(k_m)$ are used to calculate the parameters a and b by minimizing the squared error. And finally the peak is determined by

$$k_0 = 2 - \frac{b}{a} \tag{4.14}$$

Compared with the derivative algorithm (see §4.2.4), the triangle algorithm performs better at noisy situations while the derivative algorithm is superior at low noise levels.

4.3 Spectral Modeling Synthesis (SMS)

Serra [80] incorporated the noise component of the sound into an extended sinusoidal model: Spectral Modeling Synthesis (SMS). In this approach, the sinusoids model only the stable partials of a sound, and the residual models what is left, which should ideally be a stochastic component. The input sound $s(t)$ is decomposed as

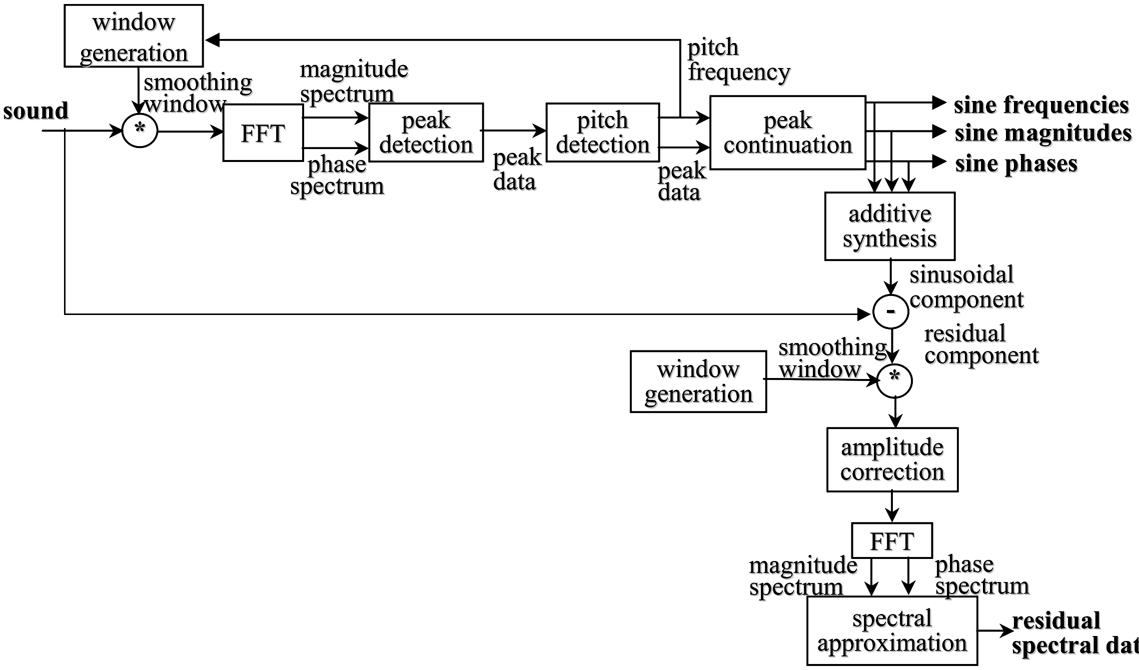


Figure 4.6 SMS analysis block diagram

$$s(t) = \sum_{p=1}^{P} a_p(t)\cos\left(\varphi_p(t)\right) + e(t) \qquad (4.15)$$

where $A_p(t)$ and $\varphi_p(t)$ are respectively the instantaneous amplitude and phase of the p^{th} partial, and $e(t)$ is the noise component. When the sinusoids are used to model only the stable partials of they sound, they are referred to as the deterministic component. The residual $e(t)$ is assumed to be a stochastic signal and it can be described as filtered white noise

$$e(t) = \int_0^t h(t,\tau)u(\tau)d\tau \qquad (4.16)$$

where $u(t)$ is white noise and $h(t,\tau)$ is the response of a time varying filter to an impulse at time t.

SMS Analysis

The SMS analysis (see Figure 4.6) starts with the STFT calculation. In the case of a monophonic single-pitched input source, the window length is adapted to the estimated pitch, to better identify and estimate the partials (deterministic component). A peak detection algorithm is applied to the STFT spectra and, if it is needed, the pitch is calculated from the estimated peaks. The peak continuation algorithm identifies the peaks

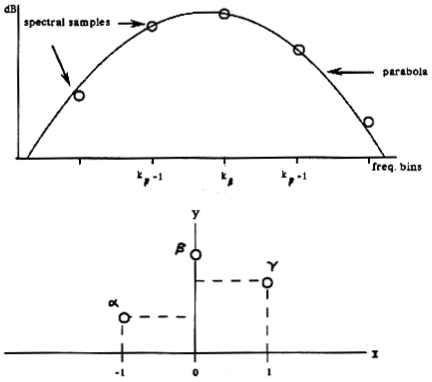


Figure 4.7 Parabolic interpolation in the peak detection process

corresponding to stable partials and discards the rest. Then the residual component is obtained by subtracting the stable partials from the original sound, and its spectral approximation is calculated.

peak detection

A peak is defined as a local maximum in the magnitude spectrum. Due to the sampled nature of the spectrum, each peak is accurate only to within half a sample. A spectral sample represents a frequency interval of f_s / N Hz, where f_s is the sampling rate and N the size of the FFT. Zero-padding in the time domain increases the number of spectral samples per Hz and therefore the accuracy of the peak estimation. However, a zero-padding factor of 1000 is required to obtain frequency accuracy on the level of 0.1 percent of the distance from the top of an ideal peak to its first zero crossing (in the case of a Rectangular window).

A more efficient way to increase the frequency accuracy is to zero-pad such that quadratic spectral interpolation refines the estimate to 0.1 percent accuracy, if only samples surrounding the peak are used, as shown in Figure 4.7. The frequency and magnitude of the peak are the ones of the maximum of the parabola. The phase value of the peak is calculated by linear interpolation of the unwrapped phase spectrum.

peak continuation

In order to have a good partial/residual decomposition the peak continuation algorithm must be able to identify the stable partials of the sound. Several strategies can be used to accomplish this. Maybe the simplest case is when the sound is monophonic and single-pitched. In this case, the fundamental frequency can be incorporated to the peak continuation algorithm to easily identify the harmonic partials. A good overview of pitch detection algorithms can be found in [36]

McAulay and Quatieri proposed in [60] a simple peak continuation algorithm based on finding, for each peak, the closest one in frequency in the following frame. Serra

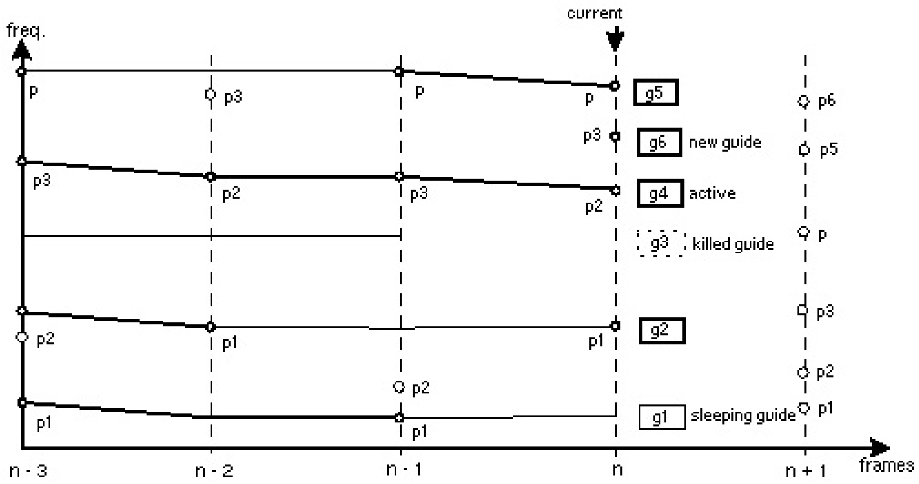


Figure 4.8 Peak continuation algorithm. *g* represents the guides and *p* the spectral epaks.

included in the peak continuation algorithm a set of *frequency guides* that are used to create sinusoidal trajectories, as shown in Figure 4.8. The frequency guide values are obtained from the peak values and their context, such as surrounding peaks and fundamental frequency. In the case of harmonic sounds, these guides are initialized according to the harmonic series of the estimated fundamental frequency. Each peak is assigned to the guide that is closest to it.

There are other continuation methods, like continuation based on derivatives or based on Hidden Markov models [18]. The last one seems to be very valuable for tracking partials in polyphonic sounds and complex inharmonic tones.

Residual analysis

The residual is obtained by subtracting the deterministic component from the original sound, as shown in Figure 4.10. The deterministic component can be obtained in time domain by using additive synthesis, where each sine wave oscillator is controlled by the estimated parameters (amplitude, frequency, phase) out from the analysis, smoothly interpolated frame to frame.

Since SMS assumes that the residual is a stochastic signal, it should be fully described by its amplitude and its general frequency characteristics. Discarding its phase information, it could be reproduced as white noise filtered trough a time-varying filter defined by the approximated envelope of the residual amplitude spectra (see Figure 4.10). This envelope approximation can be done by performing, for each frame, some sort of curve fitting in the magnitude spectrum [79, 84]. Standard techniques are: spline interpolation [14], the method of least squares [79], or straight-line approximation. Another way to approximate the envelope is to step trough the magnitude spectrum, find local maxima in each of several defined sections, and interpolate linearly between them.

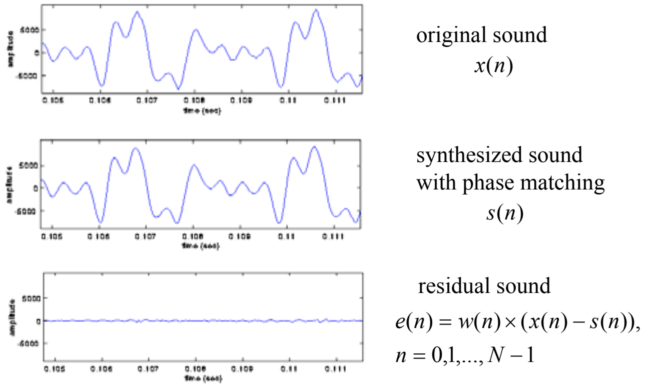


Figure 4.10 Deterministic component subtraction in time domain

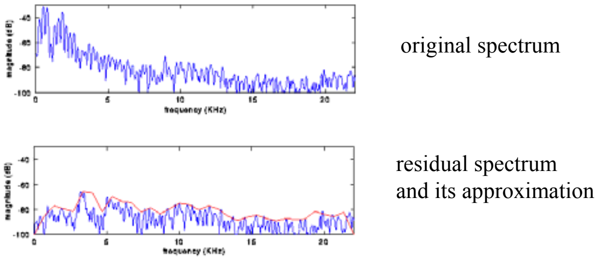


Figure 4.10 Approximated residual envelope

Another alternative, is to use linear predictive coding, LPC [51, 56], a popular technique used in speech research for fitting an *n*th-order polynomial to a magnitude spectrum. A comprehensive collection of different approximation techniques for the residual component can be found in [38].

SMS Synthesis

There are several ways to synthesize using SMS. The most efficient one is to add both sinusoidal and residual components into a single spectrum and then use a single IFFT to get the time domain signal to be overlapped, as shown in Figure 4.11.

Sinusoidal synthesis

The additive synthesis based on the inverse FFT was proposed by Rodet in [74]. Compared to the traditional oscillator bank implementation, it loses some of its flexibility (instantaneous control of frequency and magnitude) but the gain in speed is significant. With a synthesis window that concentrates most of its energy in the main lobe, like for example Blackman-Harris 92dB, it is enough to calculate the complex samples of the main lobe of the window transform for each partial and add them to the same spectrum. The problem comes when such window does not overlap perfectly to a constant in time domain. A way to avoid this is to divide, after the IFFT, the time domain segment by the

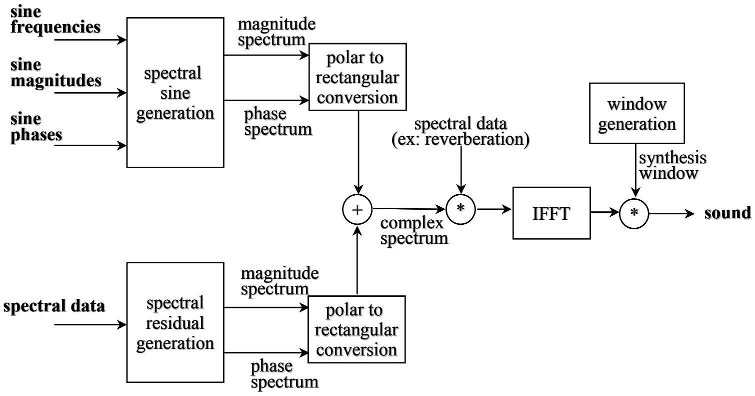


Figure 4.11 SMS synthesis block diagram

window (to undo its effect) and multiply it by a triangular window before the overlap-add procedure.

Residual synthesis

The synthesis of a stochastic signal from the residual approximation can be understood as the generation of noise that has the frequency and amplitude characteristics described by the approximated spectral magnitude envelopes. Thus, the amplitude spectrum is generated from the approximated envelopes, and random phases are calculated at each frame in order to avoid periodicity.

If we want to add both sinusoidal and residual components to the same spectrum, they should have been convolved by the same analysis window. As said above, the sinusoids are convolved by the main lobe of the synthesis window. Therefore, we should also convolve the noise spectrum with the same window. This can be approximated with discrete convolution by the main lobe of the window.

SMS Time-scale modifications

When applied to time-stretching an audio input, SMS allows modified sinusoidal components of the signal to preserve pitch, and noise-like components to remain noisy [81]. For both sinusoidal and residual component, the ratio between synthesis and analysis hop size determines the factor of time-scale modification. This method fails to handle the transients correctly because they are modeled as sinusoids and/or stochastic noise.

4.4 Transient modeling synthesis (TMS)

Verma et al [90] extended the Spectral Modeling Synthesis (SMS) into a new model called Transient modeling synthesis (TMS) that includes an explicit and flexible model for transient signals in addition to the sinusoidal plus residual model. The explicit handling of transients provides a more realistic and robust signal model, compared to SMS, while maintaining its spirit as a flexible signal representation.

In the SMS framework, transient+noise residual is modeled as slowly varying filtered white noise. As suggested by [37, 80], transients need to be considered separately from noise. The system block diagram is shown in Figure 4.12. TMS analysis block is used on the first residual (r1) that contains both noise and transients. It first detects the transients and then subtracts them from r1 to create the second residual, r2, which ideally consists of slowly varying white noise.

The basic idea underlying the transient analysis is the duality between time and frequency. While sinusoidal analysis find sinusoids by tracking the spectral peaks of a time domain signal, TMS finds transients by tracking the spectral peaks of a frequency domain signal. The Discrete Cosine Transform (DCT) provides an appropriate mapping from the time domain to frequency domain so that transients in the time domain become sinusoidal in the frequency domain. DCT is defined as

$$C(k) = \alpha(k)\sum_{n=0}^{N-1} x(n)\cos\left[\frac{(2n+1)k\pi}{2N}\right] \;, \qquad \text{for } n,k \in 0,1,...,N$$
$$\alpha = \begin{cases} \sqrt{\dfrac{1}{N}} & \text{for } k = 0 \\ \sqrt{\dfrac{2}{N}} & \text{for } k = 1,2,...,N \end{cases} \tag{4.17}$$

If $x(n)$ is a Kronecker delta, then $C(k)$ is a cosine whose frequency depends on the time location of the impulse. Thus, an impulse toward the beginning of the frame results in a low frequency cosine in the DCT domain, and on the other hand, an impulse near the end of the frame results in a high frequency cosine. On the DCT domain a sinusoidal analysis is performed with a window that has to be shorter than the DCT size. The *magnitude*, *frequency* and *phase* of the output spectral peaks parameterize the detected transients. But here the *frequency* value corresponds to where a transient occurs. Since *frequency* corresponds to a time location, the amount of frequency resolution should be greater than the number of time samples used in computing the block DCT in order to avoid quantization of transients.

The transient detection block that appears in Figure 4.12 is used to determine where possible transients occur, and therefore it restricts TMS to model transients only in those areas. However, this step is optional because TMS could be used without restrictions with proper control parameters.

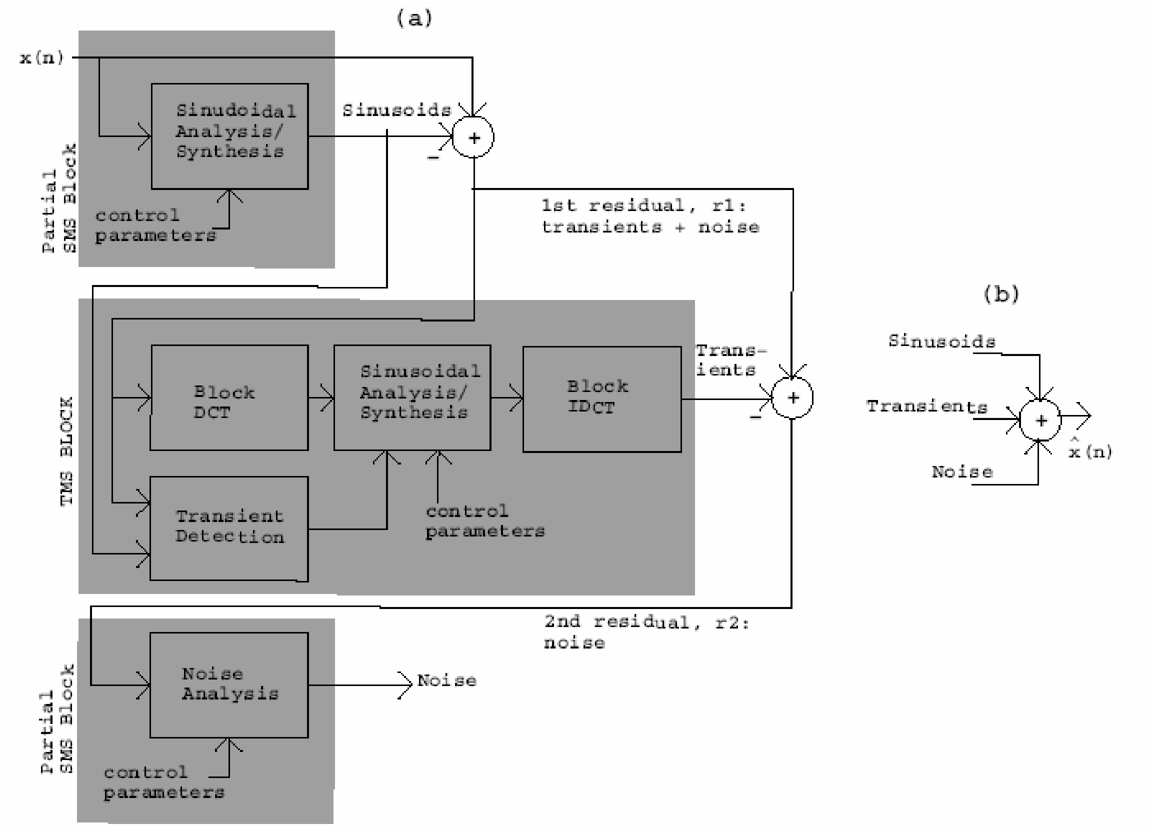


Figure 4.12 Analysis and synthesis block diagrams of TMS

When applied to time-stretching an audio input, this method allows modified tonal components of the signal to preserve pitch, transient components to preserve edges and noise-like components to remain noisy [91]. The modifications required for the sinusoidal and residual part of the model are the same as in SMS: during synthesis, it is used a different set of points in time than during analysis. Transient modification also requires a time-scale modification. To do so, the DCT block length is changed by the same factor as sines and noise. Thus, more/fewer cycles of each sinewave appear in the DCT block which effectively increases/decreases the frequencies in the DCT domain and translates the transients to their proper onset location.

4.5 Multiresolution TMS modeling for wideband polyphonic audio source

Levine et all [50] developed recently a computationally efficient method to get more accurate sinusoidal parameters (amplitude, frequency and phase) from a wideband polyphonic audio source in a multiresolution, non-aliased fashion.

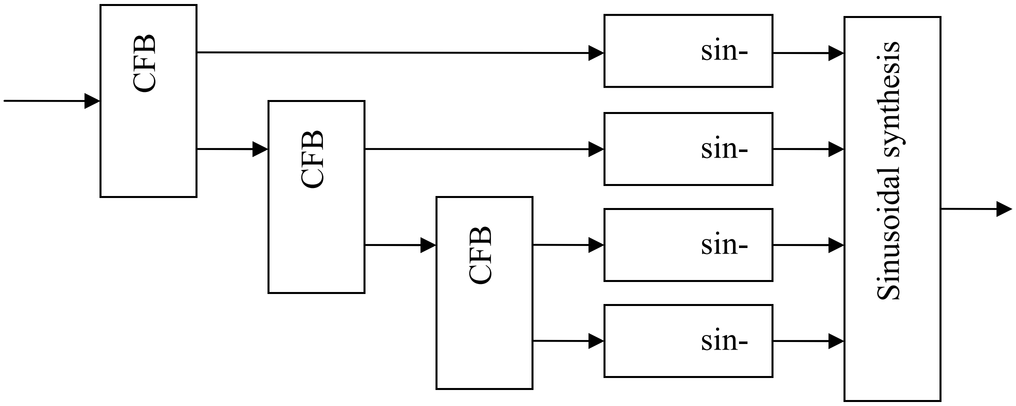


Figure 4.13 Multiresolution sinusoidal modeling block diagram

The trade of between time and frequency resolution is a challenging problem for polyphonic sinusoidal analysis. We get good frequency resolution with long windows, but poor time resolution. On the other hand, we get good time resolution with short windows, but poor frequency resolution. For a good sinusoidal analysis, very long windows are needed in order to accurately estimate the low frequency partials, and the higher the frequency of the partial the less frequency resolution is needed.

In the case of a monophonic signal and single-pitched, a good approach is to adapt the window size to the estimated fundamental pitch: the lower the fundamental pitch, the longer the window. This guarantees the correct estimation of the fundamental pitch and all the partials above it.

Unfortunately, for polyphonic audio it is impractical to attempt to discern multiple pitches. To solve this problem, the input signal is splitted into several bandlimited frequency channels with different window lengths. Previously, there had been several approaches to solve the parameter estimation problem in a multiresolution manner. One method was an octave-spaced, critically sampled, wavelet filter bank [7, 39, 75]. This method has the problem that there is no known way to eliminate all aliasing between channels in the filter bank, and therefore each channel contains the actual bandpassed-octave signal, in addition to aliased octaves adjacent in frequency. These cross-talk aliasing terms can be reduced, but the complexity increases and the results are not good enough for high quality wideband sinusoidal modeling [7, 23, 85]. Besides, there are other alternative methods, that are discussed in [49].

Levine et al proposed an octave-spaced filter bank where each channel output goes into a separate sinusoidal modeling block, with its own analysis parameters. The parameters extracted from each band analysis goes to a single sinusoidal synthesizer, as shown in Figure 4.13, which can be either a bank of oscillators or a block IFFT.

The filter bank is designed to assure that the subband signals are alias-free, and downsampling is introduced in the filter bank to decrease memory and computational cost. Some partials may exist in both channels because there is overlap in frequency ranges between channels, but once synthesized with the correct phase, the two partials will constructively sum to the single original partial.

The residual is obtained by subtracting all the synthesized sinusoids from the original signal, and then it is processed trough the TMS algorithm (see §4.4). The TMS algorithm parametrically models the transients and generates a second residual without transients that is modeled as a time-varying filtered noise.

This method significantly improves previous work of sinusoidal modeling that supposed a pitched monophonic source, and allows high-quality time-stretching modifications on polyphonic audio with ease. The time-scaling factor determines the ratio of synthesis window length to analysis window length in each octave-spaced channel, thus the partial parameters are simply interpolated over a different hop size length.

4.6 Conclusions

In this chapter we have reviewed several signal models that include sinusoidal modeling, introduced by McAulay and Quatieri, as starting point. In sinusoidal modeling, the input signal is represented as a sum of sinusoids with time-varying amplitude, phase and frequency. Di Federico introduced a method based on relative delays to preserve the vertical phase coherence after transformations. Additionally, Fitz and Haken added bandwidth as an extra parameter to each sinusoid to include noise information. Several papers have been dedicated to improve the parameter estimation using interpolation methods, signal derivatives and special windows. Time-scale using sinusoidal modeling achieves good results with harmonic signals, especially when keeping the vertical phase coherence. However it fails to successfully represent and transform noise and transient signals. Attacks are smoothed and noise sounds artificial.

Serra introduced the idea of subtracting the estimated sinusoids from the original sound to get a residual, and modeled it as a stochastic signal (SMS). This method allowed splitting for example a flute signal into the air flow and the harmonics, and transforming both parts independently. This technique successfully improves the quality of time-scale transformations but fails to handle transients.

Verma et al noticed that transients where included in the SMS residual and not well represented as a stochastic signal, so they proposed to add a transient model (TMS) which could be synthesized and subtracted from this first residual to obtain a second noisy residual free of transients. Then, all three components (sinusoidal, noise and transient) could be modified independently and resynthesized. When time-scaling an input signal, transients can successfully be translated to new onset location, preserving their perceptual characteristics.

Finally, Levine et al introduced a multiresolution method to deal with the time and frequency resolution problem, which is especially challenging for wideband polyphonic audio sources. They stated that this method allows high-quality time-scale modifications.

THESIS RESEARCH SELECTED APPROACH AND CONCLUSIONS

5 Selected approach and conclusions

In this chapter we define the research to be carried out for the Doctoral Thesis. After the review that has been done in previous chapters, the first question that arises is what technique to use. And the second is how to improve it in order to justify a PhD Thesis.

5.1 What technique to use?

The answer depends on the constraints imposed to the system. If speed was the main issue, time domain techniques have shown to have the lowest computational cost. Nevertheless, nowadays most computers can run frequency domain techniques in real-time. Thus, speed seems not to be the main factor in the decision. On the other hand, if we knew a priori the input signals to be processed, maybe some algorithms would be more suitable than others and could be optimized accordingly. However, in the context of professional audio post-production, the input sources can go from monophonic speech to movie soundtracks, or from classical to techno music. Thus, the audio material is not the main factor in the decision. Leaving aside technical aspects as sampling rate, bit depth or interface, the main issue is then sound quality.

In first chapter we saw that sound quality must be outstanding for time-scale factors from 70 to 130%. **Time-domain techniques** reviewed in chapter 2 give best results for small modification factors with single signal sources, even in noisy situations. However, they fail with polyphonic material.

Phase-vocoder gives smoother results for large time-scale factors and works well with polyphonic material. However it introduces smearing for impulsive signals and smoothes transients. On the other hand, recent improvements have successfully minimized the reverberance or phasiness introduced by the loss of vertical phase coherence. And some multiresolution approaches on top of the phase-vocoder perform spectral analysis closer to that performed by the human auditory system.

Signal models have shown their ability to split the input signal into different components which can be processed independently. Basically these components are sinusoids, transients and noise. This decomposition gives a lot of flexibility when thinking on transformations. *Sinusoids* are good to model quasi-stationary components (slow-varying partials), and can deal with impulsive signals with a waveform preservation technique based on phase delays, thus preserving the synchronization of the harmonics' phase, but only in the case of single signal sources. Several advances have

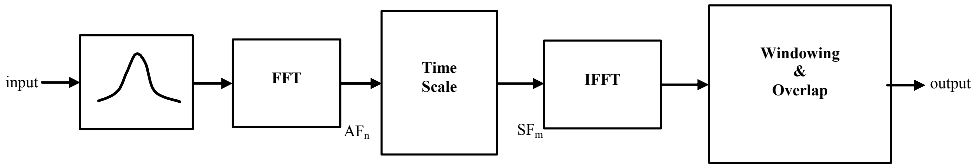


Figure 5.1 General diagram of the proposed time-scale algorithm

been achieved in estimating the sinusoid parameters, including multiresolution approaches, which results in higher sound quality and less pre-echo, but there is much work yet to do. *Transients* are good to model attacks and fast changes. In fact, time-scale of transients means nothing more than translating them into a new onset position. They are detected and parameterized from the residual obtained by subtracting the sinusoids from the original sound. The *noisy residual* is obtained from this first residual by subtracting the transients. The noise is modeled as slowly varying filtered white noise, which is a good model for time-scaling breath or air flow of instruments for example. We can say that these signal models are very powerful and give much more flexibility than time-domain or phase-vocoder techniques. Very extreme time-stretch ratios can be applied producing high quality results; even the signal can be slowed down to a frozen state. However, with no transformations, they don't allow a perfect reconstruction of the original signal, unless the noisy residual is not modeled but just kept as it is. Thus, the synthesized output signal does not sound exactly as the original one, but very close. This is a drawback for professional post-production applications.

In the professional context, we may expect to find mainly polyphonic material. Therefore, time-domain algorithms are discarded. Signal models have a great potential and flexibility, but they are not able to resynthesize a signal that sounds exactly like the original. Therefore, it seems that phase-vocoder techniques should be the best option for post-production. They allow a perfect reconstruction when no transformation is applied and work well with polyphonic sources. However, some enhancements should be done, especially to deal with transients and impulsive signals.

5.2 What to improve?

Once we have established the base technique to work from, we have to point out the improvements to be done and how they can be achieved. At first, we chose the block-by-block approach for the phase-vocoder because it's much faster than the filter bank model. In Figure 5.1 we find a general block diagram of the proposed time-scale system.

5.2.1 Constant hop size

Traditionally, the time-scale modification is achieved playing with the ratio between the synthesis and the analysis hop size. However, is that better than the time-

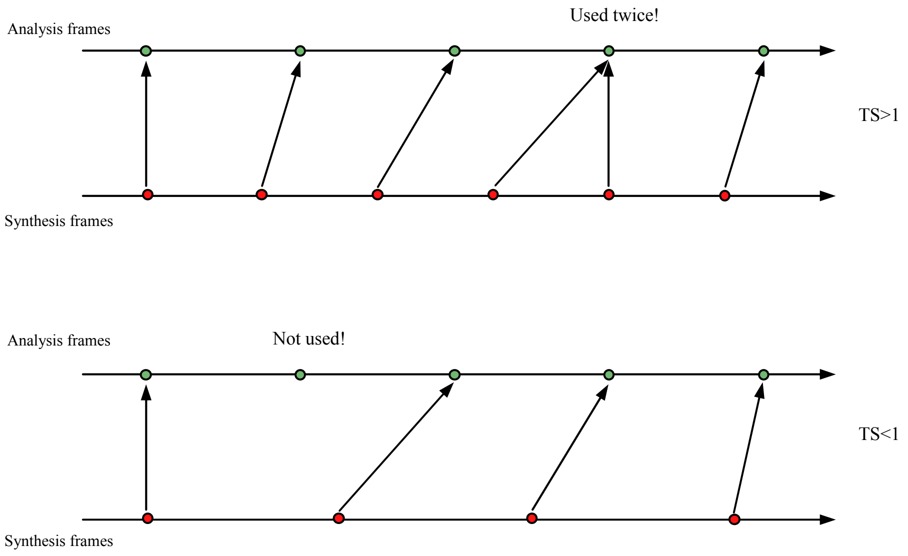


Figure 5.2 Time-scale achieved by repeating and dropping small segments

segment processing idea of repeating or dropping small frames? If we play with the hop size ratio there is a limit of time-stretch ratio because a minimal window overlap is needed in synthesis. This does not happen when repeating or dropping small frames. Besides, in the last case there is no need to compute phase unwrapping since the synthesis and the analysis hop size are equal.

In Figure 5.2 we can see what happens in the case of a time-scale factor TS that makes the audio length larger than the original $(\mathrm{TS} > 1)$ and in the opposite case $(\mathrm{TS} < 1)$. The horizontal axis corresponds to the center time of the analysis frames. The points in the lower time arrow are the time position (referred to the input signal) of each successive synthesis frame considering the time-stretch factor applied. A simple algorithm is proposed for choosing the analysis frame: to pick the nearest analysis frame looking to the right, i.e. for each synthesis point select the nearest analysis point looking to the right.

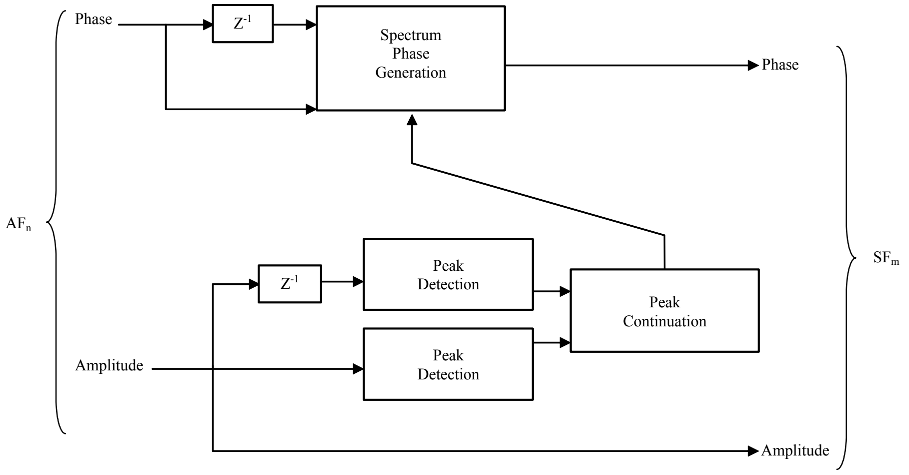


Figure 5.3 Time-scale module block diagram

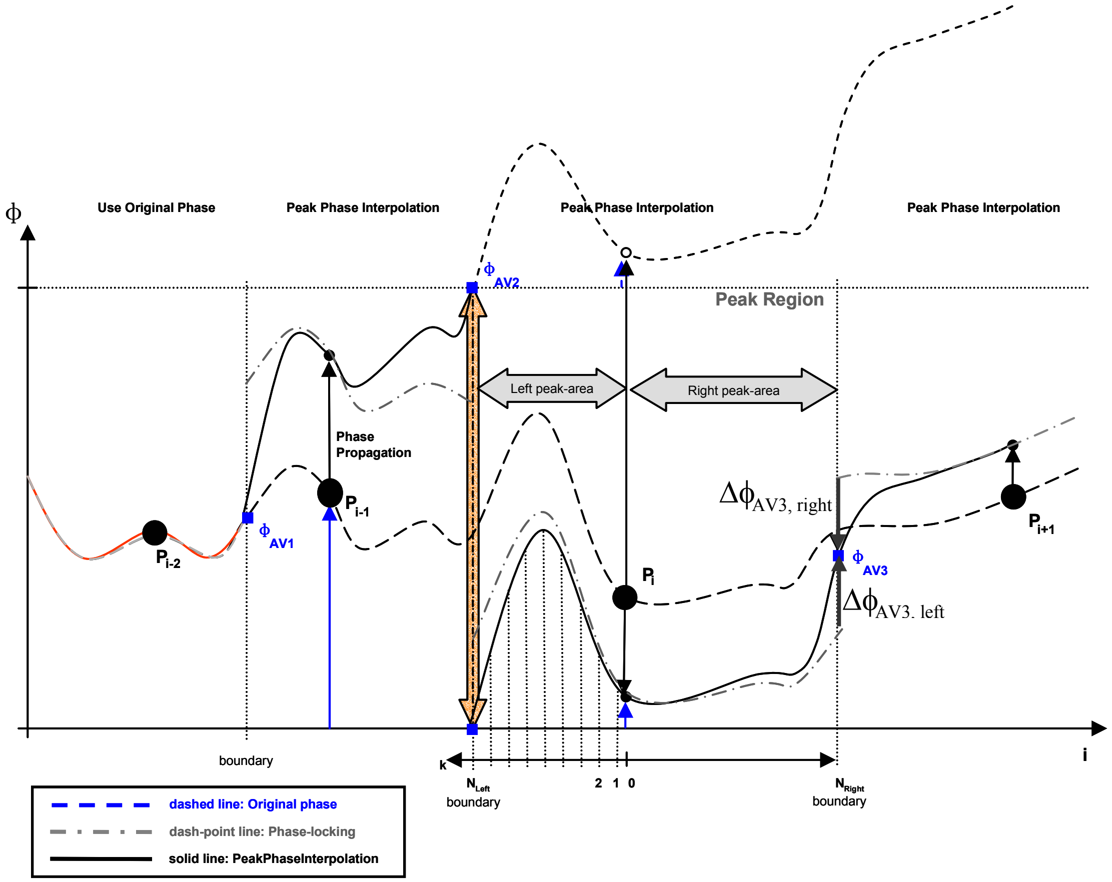


Figure 5.4 Interpolated phase locking

Thus, when the input signal is slowed down (TS>1) one analysis frame is used twice. On the other hand, for the opposite case ($TS < 1$) one analysis frame is not used. Constant and variable hop size should be properly compared during the thesis research to come to a conclusion of which is the best one in terms of sound quality, possibilities, complexity and computational cost.

5.2.2 Interpolated phase-locking

Scaled phase-locking should be added in order to minimize phasiness. This technique requires peak detection and continuation, as shown in Figure 5.3. As we saw in chapter 3, only the spectral phase is modified. However, in the case of constant hop size, using the factor β (see §3.5.2.2) makes no sense because the time distance between consecutives frames is not stretched. In this case, using scaled phase-locking with $\beta = 1$ is the same as using identity phase-locking with the addition of peak tracking. However, some phase discontinuities may be introduced in the peak's region boundaries, as seen in Figure 5.4. An interpolation method is proposed to avoid phase jumps while keeping the behavior of the scaled phase-locking around the peaks. If we look around the boundary

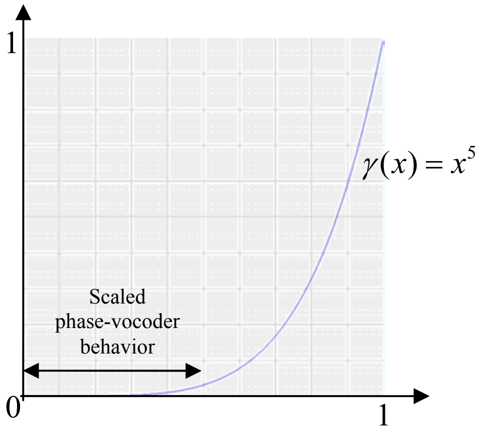


Figure 5.5 Interpolating function

between peaks P_i and P_{i+1}, the dashed line shows the original continuous spectrum phase. The time-scale modification propagates the phase of both peaks by adding different phase amounts. The dash-point line shows the resulting discontinuous spectral phase after scaled phase-locking is applied. ϕ_{AV3} is the average between the phase values around the boundary. For both sides, the phase differences between the phase-locking and the average values are calculated ($\Delta\phi_{\mathrm{AV3,\ left}}$ and $\Delta\phi_{\mathrm{AV3,\ right}}$). Let's now define the interpolating function $\gamma(x)$ as

$$\gamma(x) = x^m \tag{5.1}$$

The left delta phase multiplied by the interpolating function is added to the region between the peak P_i and the boundary (*Right peak-area*) like this

$$\phi_{\text{interpolated phase-locking}} = \phi_{\text{scaled phase-locking}} + \Delta\phi_{\text{AV3,left}} \cdot \gamma\left(\frac{k - k_{P_i}}{k_{boundary} - k_{P_i}}\right) \tag{5.2}$$

where k is the frequency index. The value of γ will go from 0 at the peak to 1 at the boundary. For high values of m the phase behaves like the scaled phase-locking. $m = 5$ could be a good choice, as shown in Figure 5.5. This proposed variant of the scaled phase-locking technique should be tested and compared with the original one during the thesis research.

5.2.3 Transient processing

We saw in chapter 4 a transient model that detected and parameterized transients from the residual obtained by subtracting sinusoids to the input signal. Although this is powerful technique, in our context this parameterization is not really needed. Transients should just be translated to a new time onset. First of all we need a way to detect the transients in an unsupervised manner. There are many different approaches and

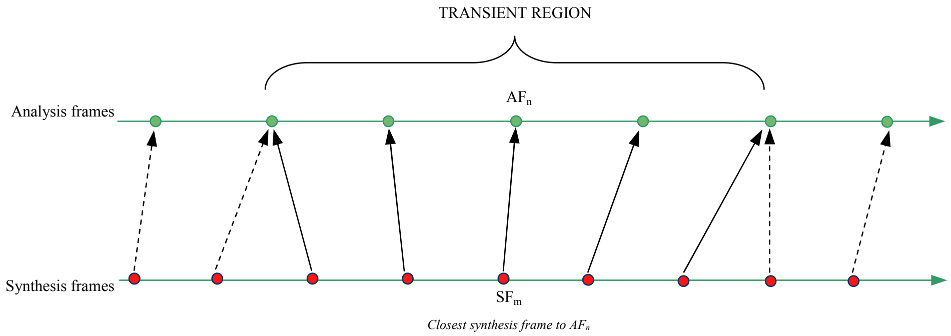


Figure 5.6 Transient region

algorithms in the literature to deal with this problem [10, 44, 57, 63, 71, 78, 90]. As starting point, a simple approximation is proposed based on the computation of relative changes of energy along several frequency bands. A low frequency band could, for example, detect sharp bass notes, while a high frequency band would detect hits of a crash cymbal.

The spectrum of the input signal is given by

$$X(sR_a,k) = \left|X(sR_a,k)\right| \cdot e^{j\varphi(sR_a,k)} \tag{5.3}$$

where the short-time Fourier transform has been calculated every R_a samples, and s is the time index of the short-term transform. If we define a set of frequency bands $B_i(k)$, then the energy of the i^{th} band can be computed as

$$E(s,i) = \sum_{i=0}^{N-1} B_i(k) \cdot X^2(sR_a,k) \tag{5.4}$$

and the relative change of energy $C(s,i)$ at frame s as

$$C(s,i) = \frac{-2E(s-2,i) - E(s-1,i) + E(s+1,i) + 2E(s+2,i)}{E(s,i)} \tag{5.5}$$

The maximums of $C(s,i)$ over some threshold should then indicate the transients or attacks of the input signal at the desired band.

Transient should not be time-scaled, but just translated to a new time onset position. Considering the constant hop size approach, one possible solution is to not time-scale transient regions of the input signal, as shown in Figure 5.6. In that case, a greater amount of scale modification should be applied to surrounding regions in order to preserve the overall scaling factor.

To keep the output signal as close as possible to the original one, the phase synchronization between peaks should be kept. On way could be to try to use as much as possible of the original phase during a transient region. Taking advantage of the

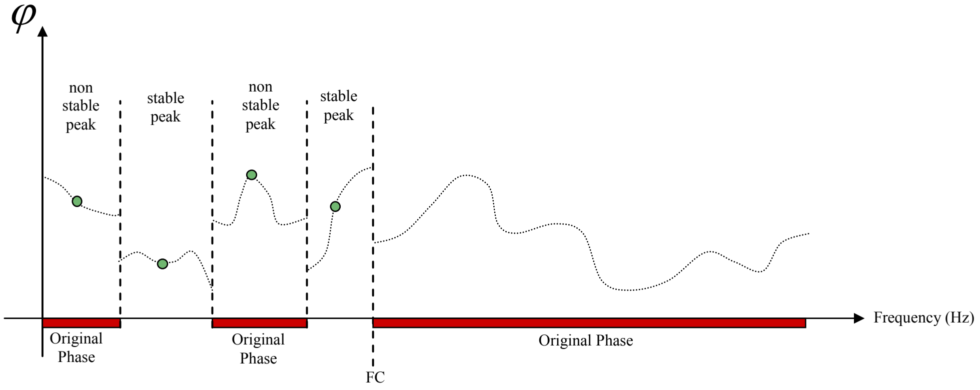


Figure 5.7 Using original phase at transient regions

perceptual masking effect of a transient plus the poor frequency resolution at high frequencies of the human auditory system, we could use the original spectrum phase above a certain frequency cut FC along the transient region. Preliminary experiments have shown that FC=2500Hz is a good choice that preserves the perceptual behaviour without perceived artifacts. For frequencies lower than this frequency cut, we have to continue the phase of the quasi-stable peaks, but we can use the original phase in the non stable peaks' regions, as shown in Figure 5.7.

This transient processing proposal should be fully tested, improved and discussed during the thesis research. Other onset detection methods should be also considered as well.

5.2.4 Multiresolution approach

As we saw in previous chapters, short-time Fourier transform performs a uniform frequency analysis of the input signal. Thus, the time and frequency resolution are constant for each input segment processed. High time resolution implies poor frequency resolution and vice versa. On the other hand, human auditory system performs a nonuniform frequency analysis with good frequency resolution at low frequencies (i.e. poor time resolution) and poor frequency resolution at high frequencies (i.e. good time resolution). This can be understood as if we were using long analysis windows at low frequencies and short ones at high frequencies. Several approaches based on the phase vocoder that perform nonuniform frequency analysis were presented in chapter 3 (see §3.6-§3.9).

A multiband analysis is proposed as starting point, where each band is the result of a short-time Fourier analysis with a specific window size, window type and zero padding, as shown in Figure 5.9. Inside the time-scale module, peak detection is applied to each band and a peak continuation module takes care of the desired band frequency cuts, so it

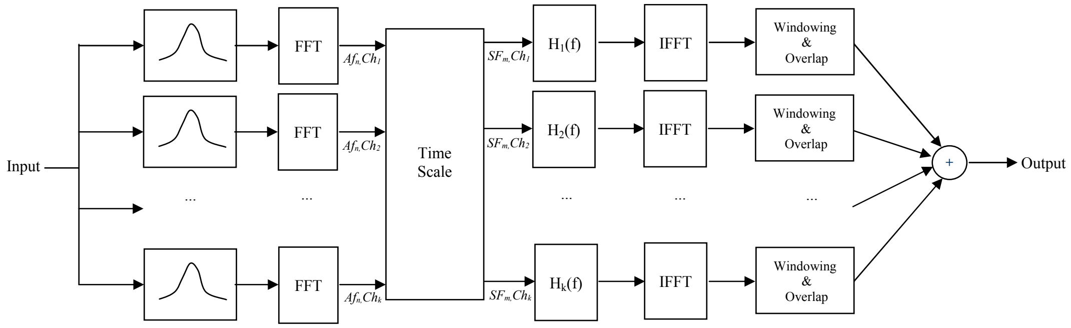


Figure 5.9 Multiband phase-vocoder

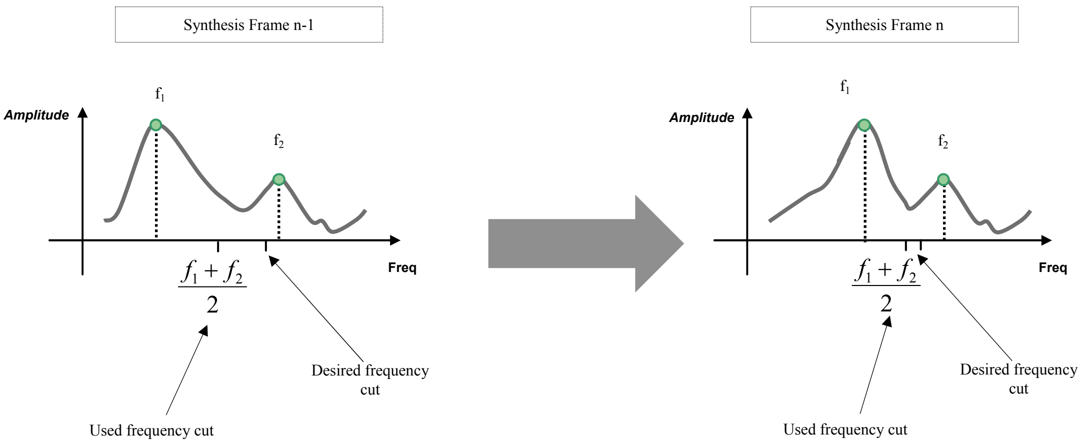


Figure 5.9 time-varying frequency cut

can connect peaks of different bands. Finally, the spectrum of each band is filled and a set of parallel filters $H_n(f)$ is applied. This set of bank filter should be equivalent to an all pass filter. Time-varying band frequency cuts are required because if a peak is very close to a frequency cut, then breaking it into different bands could produce some artifacts. It is proposed to set the frequency cut to the middle frequency between the two closest peaks to the desired frequency cut (see Figure 5.9). In this way we can guarantee the amplitude and phase envelopes around the peak to be the right ones. In order to lower the computational cost it is proposed to use a multirate system with polyphase filter banks [88].

During the thesis research, this proposed multiresolution algorithm should be implemented, tested, and compared to other possible approaches. Then an optimal solution should be discussed.

5.2.5 Phase coherence and aural image

The aural image should be preserved after the time-scale transformation. To do so, both amplitude and phase relations between audio channels should be preserved. Since

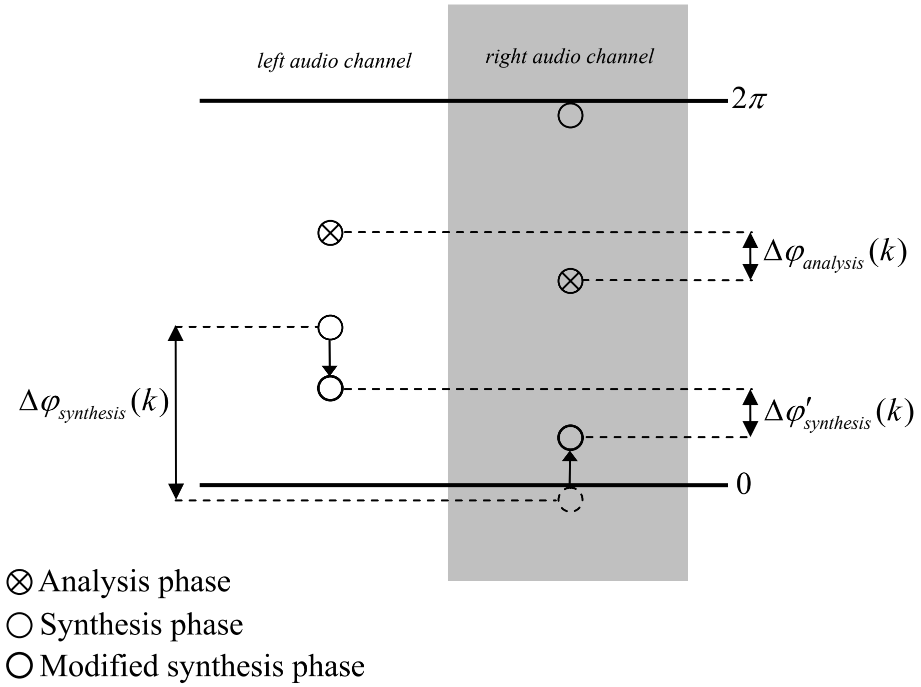

left audio channel
right audio channel
2π
$\Delta\varphi_{analysis}(k)$
$\Delta\varphi_{synthesis}(k)$
$\Delta\varphi'_{synthesis}(k)$
0
⊗ Analysis phase
○ Synthesis phase
○ Modified synthesis phase

Figure 5.10 Phase modification algorithm for preserving the aural image

spectral amplitude is not modified, then the amplitude relation is already kept if same analysis frame center times are used for all channels. This means that detected transient onsets should be synchronized, because previously it has been proposed to change the analysis frame selection procedure along transient regions (see §5.2.3).

In order to preserve the phase relation, it is proposed to compare the synthesis and the analysis phase relations bin by bin. Considering only two audio channels (stereo signal), we can write these phase relations as

$$\begin{aligned}\Delta\varphi_{analysis}(k) &= \varphi_{R,analysis}(k) - \varphi_{L,analysis}(k) \\ \Delta\varphi_{synthesis}(k) &= \varphi_{R,synthesis}(k) - \varphi_{L,synthesis}(k)\end{aligned} \tag{5.6}$$

where k is the frequency index (bin), φ_R and φ_L the wrapped phase value of right and left audio channels, and $\Delta\varphi$ the phase relation between channels. Synthesis phase values should be modified to match the analysis phase relation, but in such a way phase mismatches are not introduced at frame boundaries. As shown in Figure 5.10, it must be taken into account that phases are wrapped to the range $[-\pi, \pi[$. Thus, 2π is subtracted to the right channel synthesis phase before calculating $\Delta\varphi_{synthesis}(k)$. The same absolute phase amount is added to each synthesis phase to finally match the desired phase relation. Preliminary experiments have shown that the phase coherence and aural image are preserved. However, during the thesis research this algorithm should be carefully studied, tested and extended to a general case with n audio channels. Also other approaches should be reviewed and discussed.

5.2.6 Other improvements and related

There are some tasks related to the implementation of the proposed algorithm that will be carried out during the development of the PhD thesis. Some of these tasks are the following:

1. Database construction: the algorithm should be tested with diverse types of audio material with different features. One important task then is to build a representative database covering the whole range of audio signals that are used in the context of professional audio post-production.

2. It is necessary to define and implement a procedure for evaluating the performance of the technique developed in the Doctoral Thesis and comparing it with existing algorithms. Thus, a user evaluation method should be defined.

3. The system will be implemented in C++, and optimized so to run in real-time with workstation computers. It would be nice as well to implement it as a DirectX or TDM plug-in and build an appropriate interface.

Apart from this, other questions and improvements will arise during the thesis research. For example, we think of adapting the sinusoidal method of waveform preservation for single pitched signal sources. Also, we plan to implement a peak masking technique to both reduce the computational cost and increase the sound quality.

Appendix A

Commercial products features

Prosoniq's *Time Factory*

- Batch Processing
- supports WAVE, SDII (on Mac), AIFF file formats
- supports split stereo files
- 8, 16 and 24bit file support, 22.05, 44.1, 48 and 96 kHz Sample Rate
- 24bit/96kHz compatible
- double click to edit any soundfile in your favorite editing program
- Transcribe Mode for crystal clear 200% time scaling of complete songs
- Pitch Shifting including formant correction
- High quality resampling with anti-aliasing for pitch shifting
- no sound coloration, no phasing with stereo files
- no timing inconsistencies
- input speed in new length, BPM or %
- new MPEX Time Scaling technology for best results at the first mouse click
- works with any material of any genre, be it classic, pop, rock, techno or other
- easy 'Save as...' capability for archiving files
- both Mac and PC version in a single box - no more worrying about switching platforms
- Mac version comes with sonicWORX Essential audio editing software included

Minimum system requirements

Apple PowerMacintosh with at least PPC 603e processor running @ 120MHz, 604e or G3 preferred, 256k 2nd level cache, 16 MBytes of free RAM, Audio In/Out, System 8.0 or higher. Apple Sound Manager 3.2 or higher.

Windows PC running a Pentium-2 @ 266 MHz or better, Windows 95, 98, 2000, NT or ME with DirectX 5.0 or higher installed, 32 MB RAM, Audio In/Out.

All trademarks are property of their respective holder

Serato's *Pitch'n Time*

- Perfect Pro-Tools integration.
- Modify tempo from 50% to 200% of original and simultaneously pitch shift by ±12 semitones.
- Unrivalled and unprecedented processing quality.

- Unique patent-pending time compression/expansion and pitch-shifting algorithm.
- No loss of timing accuracy.
- Process stereo tracks without phasing.
- Process Dolby matrix encoded tracks without losing surround information.
- Time stretch by tempo change, length change, target length, or target BPM.
- Select pitch shift by frequency change or semitone shift.
- Preview changes in real time.
- Fully Mac and Windows compatible
- Multi-Channel Mode - allows you to process up to 48 tracks together while maintaining their original phase coherency.
- Time-Morph - allows you to drop a marker at an event and move it anywhere; the audio flexes before and after to allow you to stretch out a single note, or even change the timing of a loop.
- Variable Pitch Mapping - allows you to pitch-correct individual notes, remove or create pitch slides, or even make a realistic record scratch sound.
- Variable Tempo Mapping - gives you the power to create variable tempo maps and changes over the length of your sample.
- Waveform Overview - means you can actually see the effects of your tempo map as it is applied directly to the waveform.

Genesis's *Harmo*

Stereo AES3 input	32 to 48 kHz
Stereo AES3 output	48 kHz internal (other frequency possible on demand), totally independent of the input frequency or WordClock synchronisation from 32 to 48 kHz or frequency of the AES input
Transposition ratio	-20% to +20% with a step of 0.1% (401 values)
Algorithm pilots	Master, Slave, Independant
Algorithm modes	Left+Right, Left, Right
Word format	24 bits (internal processing : 32 bits)
Racks connection	Proprietary wires link the different units; all the slave racks are **synchronised** and driven by the master rack.
Processing latency	1 second (AES in -> AES out)
Duration before starting	about 3 seconds after power on
Power	85 to 260 VAC, 47 to 440 Hz
Consumption	inferior to 20 VA
Standard 19" Rack mount (1U)	418 x 310 x 43 mm

Appendix B

Publications

[P1] Amatriain, X. Bonada, J. Loscos, A. Serra, X. *Spectral Processing,* Chapter 10, Udo Zölzer Ed*.,* ***DAFX: Digital Audio Effects****, p.554 John Wiley & Sons Publishers.* 2002

[P2] Bonada, J. Celma, O. Loscos, A. Ortolà, J. Serra, X. *Singing Voice Synthesis Combining Excitation plus Resonance and Sinusoidal plus Residual Models,* Proceedings of International Computer Music Conference, Havana, Cuba, 2001

[P3] Amatriain, X. Bonada, J. Loscos, A. Serra, X. *Spectral Modeling for Higher-level Sound Transformation,* Proceedings of MOSART Workshop on Current Research Directions in Computer Music. Barcelona, 2001

[P4] Bonada, J. Loscos, A. Cano, P. Serra, X. *Spectral Approach to the Modeling of the Singing Voice,* Proceedings of 111th AES Convention, New York, USA, 2001

[P5] Cano, P. Loscos, A. Bonada, J. de Boer, M. Serra, X. *Voice Morphing System for Impersonating in Karaoke Applications,* Proceedings of International Computer Music Conference, Berlin, Germany, 2000

[P6] de Boer, M. Bonada, J. Cano, P. Loscos, A. Serra, X. *Singing Voice Impersonator Application for PC,* Proceedings of International Computer Music Conference, Berlin, Germany, 2000

[P7] de Boer, M. Bonada, J. Serra, X. *Using the Sound Description Interchange Format within the SMS Applications,* Proceedings of International Computer Music Conference, Berlin, Germany, 2000

[P8] Bonada, J. *Automatic Technique in Frequency Domain for Near-Lossless Time-Scale Modification of Audio,* Proceedings of International Computer Music Conference, Berlin, Germany, 2000

[P9] Cano, P. Loscos, A. Bonada, J. *Score-Performance Matching using HMMs,* Proceedings of International Computer Music Conference. Beijing, China, 1999

[P10] Loscos, A. Cano, P. Bonada, J. *Low-Delay Singing Voice Alignment to Text,* Proceedings of International Computer Music Conference, Beijing, China, 1999

[P11] Amatriain, X. Bonada, J. Serra, X. *METRIX: A Musical Data Definition Language and Data Structure for a Spectral Modeling Based Synthesizer*, Proceedings of COST G6 Conference on Digital Audio Effects, Barcelona, 1998

[P12] Herrera, P. Bonada, J. *Vibrato Extraction and Parameterization in the Spectral Modeling Synthesis framework,* Proceedings of COST G6 Conference on Digital Audio Effects, Barcelona, 1998

[P13] Serra, X. Bonada, J. *Sound Transformations Based on the SMS High Level Attributes*, Proceedings of COST G6 Conference on Digital Audio Effects, Barcelona, 1998

[P14] Serra, X. Bonada, J. Herrera, P. Loureiro, R. *Integrating Complementary Spectral Models in the Design of a Musical Synthesizer*, Proceedings of International Computer Music Conference, Thessaloniki, Greece, 1997

[P15] Bonada, J. *Desenvolupament d`un entorn gráfic per a l`análisi, ransformació i síntesi de sons mitjanant models espectrals*, Master thesis, Universitat Politècnica de Catalunya (UPC), Barcelona, 1997

Appendix C

Patents

[Pat1] EP0982713 *Voice converter with extraction and modification of attribute data.* Inventors: Bonada, J., Kayama, H., Yoshioka, Y., Serra, X. Applicant: Yamaha Corp. Publication date: 2000-03-01.

[Pat2] JP2000003197 *Voice transforming device, voice transforming method and storage medium which records voice transforming program.* Inventors: Bonada, J. and Yoshioka, Y. Applicant: Yamaha Corp. Publication date: 2000-01-07.

[Pat3] JP2001117564 *Device and method for processing musical sound.* Inventors: Bonada, J., Kawashima, T., and Serra, X. Applicant: Yamaha Corp. Publication date: 2001-04-27.

[Pat4] JP2001117597 *Device and method for voice conversion and method of generating dictionary for voice conversion.* Inventors: Bonada, J., Yoshioka, Y., Serra, X. and Shiimentsu, M. Applicant: YAMAHA Corp. and Univ. Pompeu Fabra. Publication date: 2000-03-01.

[Pat5] JP2001116780 *Signal analyzer and signal analysis method.* Inventors: Bonada, J. and Yoshioka, Y. Applicant: YAMAHA Corp. and Univ. Pompeu Fabra. Publication date: 2001-04-27.

[Pat6] JP2001117578 *Device and method for adding harmony sound.* Inventors: Bonada, J., Cano, P., Kondo, T. and Loscos, A. Applicant: YAMAHA Corp. and Univ. Pompeu Fabra. Publication date: 2001-04-27.

[Pat7] JP2001117600 *Device and method for aural signal processing.* Inventors: Bonada, J., Kayama, H. and Serra, X. Applicant: YAMAHA Corp. and Univ. Pompeu Fabra. Publication date: 2001-04-27.

[Pat8] JP2001184099 *Device and method for voice conversion.* Inventors: Bonada, J., Shiimentsu, M. and Kawashima, T. Applicant: YAMAHA Corp. and Univ. Pompeu Fabra. Publication date: 2001-07-06.

[Pat9] EP0982713 *Voice converter with extraction and modification of attribute data.* Inventors: Bonada, J., Kayama, H., Serra, X. and Yoshioka, Y. Applicant: YAMAHA Corp. Publication date: 2000-03-01.

[Pat10] EP1220195 *Singing voice synthesizing apparatus, singing voice synthesizing method, and program for realizing singing voice synthesizing method.* Inventors: Bonada, J., Kenmochi, H. and Serra, X. Applicant: YAMAHA Corp. Publication date: 2002-07-03.

[Pat11] EP1239457 *Voice synthesizing apparatus.* Inventors: Bonada, J. and Hisaminato, Y. Applicant: YAMAHA Corp. Publication date: 2002-09-11.

[Pat12] EP1239463 *Voice analyzing and synthesizing apparatus and method, and program* Inventors: Bonada, J. and Yoshioka, Y. Applicant: YAMAHA Corp. Publication date: 2002-09-11.

Bibliography

[1] Enounce 2xAV plug-in for RealPlayer and Windows Media Player, http://www.enounce.com/products.shtml.

[2] Genesis's Harmo, http://www.genesis.fr/.

[3] Prosoniq's Time Factory, http://www.prosoniq.com.

[4] Wave Mechanics's Speed, http://www.wavemechanics.com.

[5] Serato's Pitch and Time v2, http://www.serato.com/products/pnt/.

[6] Althoff, R., Keiler, F., and Zölzer, U., *Extracting sinusoids from harmonic signals*, Proceedigns of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx99), Trondheim 1999.

[7] Anderson, D., *Speech analysis and coding using a multiresolution sinusoidal transform*, Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 1037-1040, Atlanta 1996.

[8] Arons, B., *Techniques, Perception and Applications of Time-Compressed Speech*, Proceedings of 1992 Converence, American Voice I/O Society, 169-177.

[9] Auger, F. and Flandrin, P., *Improving the Readability of Time Frequency and Time Scale Representations by the Reassignment Method*, IEEE Transactions on Signal Processing, vol. 43 pp. 1068-1089, 1995.

[10] Bilmes, J., *Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning, and Reproducing Expressive Timing in Percussive Thytm*. PhD thesis, MSc thesis, Massachusetts Institute of Technology, 1993.

[11] Borchers, J. O., Samminger, W., and Mühlhäuser, M., *Conducting a realistic electronic orchestra*, UIST 2001 14th Annual Symposium on User Interface Software and Technology, Orlando, Florida, USA 2001.

[12] Borchers, J. O., Samminger, W., and Mühlhäuser, M., *Personal Orchestra: Conducting the Vienna Philharmonic*, Proceedings of the ACM CHI 2002 Conference on Human Factors in Computing Systems, Minneapolis, Minnesota, USA 2002.

[13] Covell, M., Withgott, M., and Slaney, M., *Nonuniform time-scale modification of speech*, Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Seattle 1998.

[14] Cox, M. G., *An algorithm for approximating convex functions by means of first-degree splines*, Computer Music Journal, vol. 14 pp. 272-275, 1971.

[15] Crochiere, R. E., *A weighted overlap-add method of short-time fourier analysis/synthesis*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 7, no. 4, pp. 441-453, 2002.

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

[17] De Götzen, A., Bernadini, N., and Arfib, D., *Traditional (?) implementations of a phase vocoder: The tricks of the trade*, 37-43, Verona.

[18] Depalle, Ph., Garcia, G., and Rodet, X., *Analysis of sound for additive synthesis: tracking of partials using hidden markov models*, Proceedigns of the International Computer Music Conference, 94-97, 1993.

[19] Desainte-Catherine, M. and Marchand, S. *High Precision Fourier Analysis of Sounds using Signal Derivatives*. LaBRI Research Report 120498. University of Bordeaux, 1998.

[20] Desainte-Catherine, M. and Marchand, S., *High-Precision Fourier Analysis of Sounds using Signal Derivatives*, J.Audio Eng.Soc., vol. 48, no. 7/8, 2002.

[21] Di Federico, R., *Waveform Preserving Time Stretching and Pitch Shifting for Sinusoidal Models of Sound*, Proceedings DAFX, 44-48, Barcelona.

[22] Dutilleux, P., De Poli, G., and Zölzer, U., "Time-segment Processing," in John Wiley & Sons (ed.) *Digital Audio Effects* 2002.

[23] Edler, B., *Aliasing reduction in sub-bands of cascaded filter banks with decimation*, Electronic Letters, vol. 28, no. 12, pp. 1104-1106, 1992.

[24] Ellis, D. P., *Timescale modifications and wavelet representations*, Proceedigns of the International Computer Music Conference, 6-9, San Jose, California USA 1992.

[25] Färber, N and Girod, B., *Adaptive playout scheduling using time-scale modification in packet voice communications*, Proceedings of the IEEE International Conference on Acoustics,Speech,and Signal Processing, ICASSP-2001, vol. 3, 1445-1448, Salt Lake City, UT 2001.

[26] 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, no. 4, pp. 441-453.

[27] Fitz, K., *The reassigned bandwidth-enhanced method of additive synthesis*. PhD thesis, PhD thesis, University of Illinois at Urbana-Champaign, 1999.

[28] Fitz, K., Haken, L, Lefvert, S, and O'Donnell, M., *Sound morphing using Loris and the Reassigned Bandwidth-Enhanced Additive Sound Model: Practice and Applications*, Proceedings of the International Computer Music Conference, 2002.

[29] Fitz, K. and Haken, L., *Bandwidth Enhanced Sinusoidal Modeling in Lemur*, Proceeding sof the International Computer Music Conference, 154-157, Banff, Canada.

[30] Fitz, K. and Haken, L., *Sinusoidal Modeling and Manipulation Using Lemur*, Computer Music Journal, vol. 20, no. 4, pp. 44-59, Winter1996.

[31] Flanaga, J. L. and Golden, R. M., *Phase Vocoder*, Bell System Technical Journal, pp. 1493-1509, 1966.

[32] Gallone, G., Boussard, P., Daudet, L., Guillemain, P., and Kronland-Martinet, R., *A wavelet based method for audio-video synchronization in broadcasting applications*, Proceedings of the Digital Audio Effects Workshop (DAFx99), Trondheim 1999.

[33] Garas, J. and Sommen, P. C. W., *Time/pitch scaling using the constant-Q phase vocoder*, Proceedings CSSP-98, ProRISC/IEEE Worrkshop on Circuits, Systems and Signal Processing, 173-176, Mierlo, The Netherlands 1998.

[34] Garas, J. and Sommen, P. C. W., *Warped Linear Time Invariant Systems and their applications in audio Signal Processing*, Proceedings of the IEEE 1999 international Conference on Acoustics, Speech and Signal Processing, ICASSP-99, Phoenix, Arizona 1999.

[35] Gaskell, P. S., *A hybrid approach to the variable speed replay of digital audio*, Journal of the Audio Engineering Society, vol. 35 pp. 230-238, 2002.

[36] Gómez, E., Klapuri, A., and Meudic, B., *Melody description and extraction in the context of music content processing*, Journal of New Music Research, to appear in 2003.

[37] Goodwin, M., *Residual modeling in music analysis/synthesis*, Proceedings of ICASSP-96, vol. 2, 1005-1008, 1996.

[38] Goodwin, M., *Adaptative Signal Models: Theory, Algorithms and Audio Applications*. PhD thesis, PhD thesis, University of California, Berkeley, 1997.

[39] Goodwin, M. and Vetterli, M., *Time-frequency signal models for music analysis, transformation and synthesis*, IEEE SP International Symposium on Time-Frequency and Time-Scale Analysis, 1996.

[40] Griffin, D. W. and Lim, J. S., *Signal estimation from modified short-time fourier transform.*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, no. 2, pp. 236-243.

[41] Hamdy, K., Tewfik, A., Chen, T., and Takagi, S., *Time-scale modification of audio signals with combined harmonic and wavelet representations*, Proceedings of ICASSP, 1997.

[42] Hejna, D. J., *Real-time time-scale modification of speech via the synchronized overlap-add algorithm*. PhD thesis, Master thesis, Massachusetts Institute of Technology. Department of Electrical Engineering and Computer Science., 1990.

[43] Hoek, S. M. J. *Method and apparatus for signal processing for time-scale and/or pitch modification of audio signals*. Sigma Audio Research Limited. US 6266003. A, 24-7-2001. 9-3-1999.

[44] Klapuri, A., *Sound onset detection by applying psychoacoustic knowledge*, Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 1999.

[45] Kronland-Martinet, R., *The wavelet transform for analysis, synthesis and processing of speech and music sound*, Computer Music Journal, vol. 12, no. 4, pp. 11-20, 1988.

[46] Laroche, J., *Time and pitch scale modification of audio signals*, Kahrs, M. and Brandenburg, K., eds., Kluwer Academic Publishers, 1998.

[47] Laroche, J. and Dolson, M., *Improved phase-vocoder. Time-Scale Modification of Audio*, IEEE Transactions on Speech and Audio Processing, vol. 7, no. 3, pp. 323-332, May1999.

[48] Lee, S., Kin, H. D., and Kim, H. S., *Variable time-scale modification of speech using transient information*, Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Munich 1997.

[49] Levine, S. N., Verma, T. S., and Smith, J. O., *Alias-free, multiresolution sinusoidal modeling for polyphonic, wedeband audio*, Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY 1997.

[50] Levine, S. N., Verma, T. S., and Smith, J. O., *Multiresolution Sinusoidal Modeling for Wideband Audio with Modifications*, Proceedigns of the International Conference on Acoustics, Speech, and Signal Processing, Seattle 1998.

[51] Makhoul, J., *Linear prediction: a tutorial review*, Proceedigns of the IEEE, vol. 63(4), 561-580, 1975.

[52] Makhoul, J. and El-Jaroudi, J., *Time-scale modification in medium to low rate coding*, Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 1705-1708, IEEE, 1986.

[53] Malah, D., *Time-domain algorithms for harmonic bandwidth reduction and time scaling of speech signals*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 27, no. 2, pp. 121-133, 1979.

[54] Mansour, M. and Tewfik, A., *Audio watermarking by time-scale modification*, Proceedings of the IEEE International Conference on Multimedia and Expo, ICME, Japan 2001.

[55] Marchand, S., *Improving Spectral Analysis Precision with an Enhanced Phase Vocoder using Signal Derivatives*, Proceedings DAFX-98.

[56] Markel, J. D. and Gray, A. H., *Linear Prediction of Speech*, Springer-Verlag, 1975.

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

[58] Massie, D. C., "Wavetable sampling synthesis," in Kahrs, M. and Brandenburg, K. (eds.) *Applications of Digital Signal Processing to Audio and Acoustics* Kluwer: 2002, pp. 311-341.

[59] Maxemchuk, N., *An experimental speech storage and editing facility*, Computer Music Journal, vol. 59, no. 8, pp. 1383-1395, 1980.

[60] McAulay, R. J. and Quatieri, T. F., *Speech analysis/synthesis based on a sinusoidal representation*, IEEE Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, vol. ASSP-34:4, 744-754, 1986.

[61] McAulay, R. J. and Quatieri, T. F., *Magnitude-only reconstruction using a sinusoidal speech model*, IEEE Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 27.6.1, San Diego, CA.

[62] McNally, G. W., *Variable speed replay of digital audio with constant output sampling rate*, Proceedings 76th AES Convention, 1984.

[63] Moelants, D. and Rampazzo, C., "KANSEI, The Technology of Emotion," in Camurri, A. (ed.) *A computer system for the automatic detection of perceptual onsets in a musical signal* Genova: 1997, pp. 140-146.

[64] Moulines, E. and Charpentier, F., *Pitch synchronous waveform processing techniques for text to speech synthesis using diphones*, Speech Communication, 9(5/6), 453-467, 1990.

[65] Moulines, E., Charpentier, F., and Hamon, C., *A diphone synthesis system based on time-domain prosodic modifications of speech*, Proceedings ICASSP, 238-241, 1989.

[66] Pallone, G., Boussard, P., Daudet, L., Guillemain, P., and Kronland-Martinet, R., *A wavelet based method for audio-video synchronization in broadcasting applications*, Proceedings of the Digital Audio Effects Workshop (DAFx99), Trondheim 1999.

[67] Pallone, G., Boussard, P., and Kronland-Martinet, R., *Transposition fréquentielle pour des applications de post-produiction cinématographique*, Actes du 5ème congrès francais d'acoustique, 595-598, Lausanne, Suisse 2000.

[68] Portnoff, M. R., *Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 24, no. 3, pp. 243-248, 1976.

[69] Portnoff, M. R., *Implementation of the digital phase vocoder using the fast fourier transform*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 24, no. 3, pp. 243-248, June1976.

[70] Puckette, M. S., *Phase-locked vocoder*, Proceedings of IEEE Conference on Applications of Signal Processing to Audio and Acoustics, Mohonk 1995.

[71] 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.

[72] Quatiere, 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 pp. 515-519.

[73] Roads, C., *Granular Synthesis of Sound*, Computer Music Journal, vol. 2, no. 2, pp. 61-62, 1978.

[74] Rodet, X. and Depalle, Ph., *Spectral envelopes and inverse FFT synthesis*, Proceedings of the 93rd AES Convention, San Francisco 1992.

[75] Rodriguez-Hernandez, M. and Casajus-Quiros, F., *Improving time-scale modification of audio signals using wavelets*, IC-SPAT, vol. 2, 1573-1577, 1994.

[76] Roucos, S. and Wilgus, A. M., *High quality time-scale modification for speech*, Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 493-496.

[77] Schaeffer, P., *La musique concrète*, QSJ No 1287, PUF 1973.

[78] Schloss, A., *On the Automatic Transcription of Percussive Music -- From Acoustic Signal to High-Level Analysis*. PhD thesis, Ph.D. thesis, Stanford University, 1985.

[79] Sedgewick, R., *Algoriths*, Addison-Wesley, eds., 1998.

[80] Serra, X, *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decompostion*. PhD thesis, PhD thesis, CCRMA, Dept. of Music, Standord University, 1989.

[81] Serra, X. and Smith, J. O., *Spectral modeling synthesis: a sound analysis/synthesis system based on a deterministic plus stochastic decomposition*, ICMJ, vol. 14, no. 4, pp. 14-24, 1990.

[82] Smith, J. O. and Serra, X., *PARSHL: An Analysis/Synthesis Program for Nonharmonic Sounds based on s Ainusoidal Representation*, Proceedings of the Internationtal Computer Music Conference., 290-297, 1987.

[83] Sticht, T. G., *Comprehension of repeated time-compressed recordings*, The Journal of Experimental Education, vol. 37, no. 4, 1969.

[84] Strawn, J., *Approximation and syntactic analysis of amplitude and frequency functions for digital sound synthesis*, Computer Music Journal, vol. 4, no. 3, pp. 678-689, 1995.

[85] Tang, B., Shen, A., Pottie, G., and Alwan, A., *Spectral analysis of subband filtered signals*, Detroit 1995.

[86] Thomson, D. J., *Spectrum estimation and harmonic analysis*, Proceedings of the IEEE, vol. 70, 1055-1096, 1982.

[87] Torrésani, B., *An overview of wavelet analysis and time-frequency analysis*, Proceedigns of the International Workshop, Dubna, Russia 1998.

[88] Vaidyanathan, P., *Multirate systems and filter banks*, Prentice-Hall, 1993.

[89] Verhelst, W. and Roelands, M., *An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech*, Proceedings for the International Conference of Acoustics, Speech, and Signal Processing, Minneapolis 1993.

[90] Verma, T. S., Levine, S. N., and Meng, T., *Transient Modeling Synthesis: a flexible analysis/synthesis tool for transient signals*, Proceedings of the International Computer Music Conference, Greece 1997.

[91] Verma, T. S. and Meng, T., *Time scale modification usign a Sines+Transients+Noise signal model*, Proceedigns of the Digital Audio Effects Workshop DAFX98, 49-52, Barcelona 1998.

[92] Wayman, J. L., Reinke, R. E., and Wilson, D. L., *High quality speech expansion, compression, and noise filtering using the SOLA method of time scale modification*, 23d Asilomar Conference on Signals, Systems and Computers, vol. 2, 714-717, 1989.

[93] Wayman, J. L. and Wilson, D. L., *Some improvements on the synchronized-overlap-add method of time-scale modification for use in real-time speech compression and noise filtering*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 36, no. 1, pp. 139-140, Jan.1988.

[94] Zölzer, U. e. a., *DAFX Digital Audio Effects*, John Wiley & Sons, eds., 2002.

[95] Zwicker, E. and Fastl, H., *Psychoacoustics Facts and Models*, Springer-Verlag Berlin Heidelberg, 1990.