University of Crete
Department of Computer Science

FO.R.T.H.
Institute of Computer Science

Multichannel Audio Modeling and Coding for Immersive Audio Based on the Sinusoidal Model

MSc. Thesis

Christos Tzagkarakis

Heraklion

November 2007

Department of Computer Science
University of Crete

Multichannel Audio Modeling and Coding for Immersive Audio Based on the Sinusoidal Model

Submitted to the Department of Computer Science
in partial fulfillment of the requirements for the degree of Master of Science

November 28, 2007

Author: _____
Christos Tzagkarakis
Department of Computer Science

Board
of enquiry:

Supervisor _____
Panagiotis Tsakalides
Associate Professor

Member _____
Yannis Stylianou
Associate Professor

Member _____
Apostolos Traganitis
Professor

Accepted by:

Chairman of the
Graduate Studies Committee _____
Panos Trahanias
Professor

Heraklion, November 2007

Abstract

In the last few years, multichannel audio began gradually to displace stereophonic audio systems because it offers significant advantages to audio reproduction when compared to stereo audio. The large number of channels gives the listener the sensation of being "surrounded" by sound and immerses him with a realistic acoustic scene. The main problem with the increased number of channels is the demand of higher datarates for storage and transmission purposes. Consequently, multichannel audio compression algorithms have been developed in order to further reduce the datarate requirements by exploiting the similarities among the multiple channels. These compression algorithms achieve a significant coding gain, but they still remain demanding for many practical low-bandwidth applications.

Our objective is to propose a modeling and coding method for achieving as low as possible bitrate requiremets for multichannel and furthermore for immersive audio applications such as remote mixing of the multichannel recording and remote collaboration of geographically distributed musicians. This translates into deriving a model which can take advantage of the similarities among the various microphone signals of a given multichannel recording.

In this thesis, we propose encoding one audio channel, which can be one of the multiple microphone signals of a multichannel recording or a downmix sum signal, while for the remaining microphones we retain only the parameters that allow for resynthesis of the content at the decoder. This scheme is implemented via an enhanced adaptation of the sinusoids plus noise model. According to this model, an audio signal can be decomposed into a deterministic (sinusoidal) part and a stochastic (noise) part. The proposed approach is based on the observation that the noise part for each microphone signal can be obtained by transforming the noise part of one of the signals (reference), using the noise envelope of each of the remaining (side) multiple microphone signals.

The coding process can be divided into coding of the sinusoidal parameters and coding of the noise spectral envelopes. Coding of the sinusoidal parameters is based on a high-rate quantization scheme, while the encoding process of the noise spectral envelope is based on the vector quantization method for speech coding. The coding performance is evaluated

using subjective listening tests. The results show that a reproduction of good quality can be achieved using the proposed approach, by fully encoding a single audio channel only, with side information for each microphone signal in the order of 18 kbps. In this thesis the sinusoidal model is applied for high-quality audio coding for the first time in the multichannel audio domain.

Acknowledgements

First of all I would like to thank Panagiotis Tsakalides, my supervisor, for his continuous guidance and encouragement during this work. I am also grateful to Athanasios Mouchtaris, my co-supervisor, for his great support and help, specially for the various technical assistances.

I sincerely thank ALL the members of the Telecommunications & Networks Laboratory at the Institute of Computer Science (ICS) of the Foundation for Research and Technology Hellas (FO.R.T.H.) AND ALL the members of the Media Informatics Lab at the Computer Science Department of the University of Crete, for their friendship and the pleasant moments that we spent.

I would like to thank all the volunteers who took part in the subjective listening tests. It is also important to thank professor Y. Stylianou and his graduate student, M. Vasilakis, for his help with the implementation of the sinusoidal model algorithm.

I am sincerely indebted to my parents and my brother for all the encouragement they have given to me. Their constant support and encouragement have really brought me here.

Contents

List of Tables

List of Figures

Chapter 1

Background Information

These last years, in the area of *multimedia*, the way that audio is perceived has changed and new directions towards better audio reproduction have appeared, in terms of more realistic audio recordings. Taking these new directions into consideration, *multichannel* sound began gradually to displace stereophonic sound because it offers significant advantages to audio reproduction when compared to stereo sound, since the large number of loudspeakers gives the listener the sensation of being "surrounded" by sound and offers a more realistic acoustic scene compared to 2-channel stereo.

Current multichannel audio systems place 5 or 7 loudspeakers of full frequency sound around the listener in predefined positions, and a loudspeaker for low-frequency effects (LFE) in 5.1 and 7.1 multichannel audio systems, respectively, and are utilized not only for film but also for audio-only content. The setup of the loudspeakers that reproduce the channels' signals are shown in Figure 1.1.

Multichannel audio offers the advantage of improved realistic acoustic scene compared to 2-channel stereo sound at the expense of increased information concerning the storage and transmission of this medium. This is important in many network-based applications, such as Digital Radio and Internet audio. Consequently, many compression techniques have been proposed in order to give efficient solutions in bitrate constrained applications. Multichannel audio coding methods, such as Dolby AC-3 [9], MPEG-2 Advanced Audio Coding (AAC) [3], modified AAC with Karhunen-Loève transform (MAACKLT) [55], achieve a significant coding gain but remain demanding for many low-bandwidth applications, such

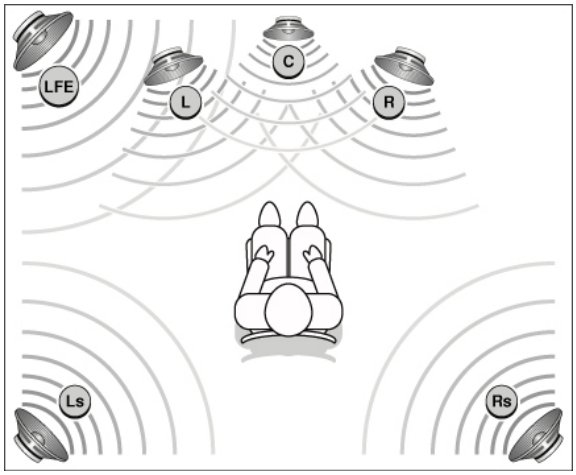

Figure 1.1: The setup of the loudspeakers in "5.1 channels" surround system.

as streaming through the Internet and wireless channels. For the sake of clarity, in the "5.1 channels" case of Dolby AC-3 the minimum achieved datarate for high-quality audio is 382 kbps, while for the case of MPEG-2 AAC the datarate is 320 kbps. MAACKLT method results in audio signals with better quality than AAC at the datarate of 64 kbps per channel. In addition, a further reduction of the bitrate requirements can be achieved by taking advantage of the inter-channel redundancy. Many multichannel audio coding methods that exploit interchannel redundancy have been proposed, such as Mid/Side Coding [21] (for frequencies below 2 kHz), Intensity Stereo Coding [19] (for frequencies above 2 kHz), and MAACKLT, where Mid/Side Coding and Intensity Stereo Coding are utilized in AAC and Dolby AC-3.

In order to further reduce the bitrate in low-bandwidth applications, MPEG Surround [6] has been recently introduced, achieving significant compression of multichannel audio recordings. MPEG Surround is based on the Spatial Audio Coding (SAC) concept. In SAC scheme, the *spatial image* of a multichannel audio signal is captured with a compact set of parameters; by encoding only one channel of audio, called reference channel which can be a downmix signal, and these parameters as side information, the objective is to resynthesize the original multichannel spatial image at the decoder with as low as possible bitrate requirements. At the decoder, the original spatial image of the multichannel recording can be recreated, by applying the extracted spatial cues to the reference channel.

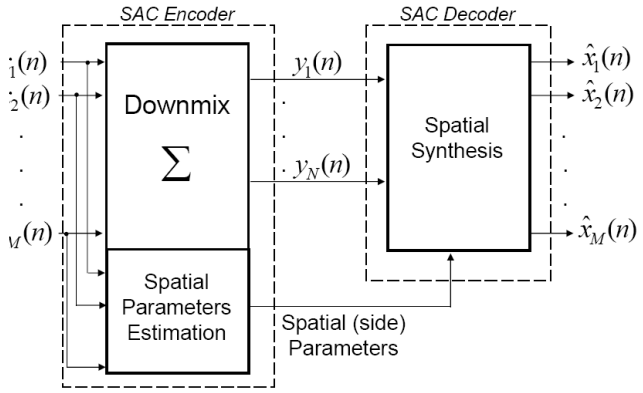

Figure 1.2: Spatial audio coding scheme.

For each channel (excluding the reference), these spatial cues can be encoded with rates as low as 5 kbps. The framework of SAC is shown in Figure 1.2. The two main methods for multichannel audio coding based on the SAC philosophy are Binaural Cue Coding (BCC) [1, 10] and Parametric Stereo (PS) [5]. BCC encodes as additional information the subband interchannel level difference, time difference, and correlation of each channel with respect to the reference audio channel. The test results in [10] give an indication that for bitrates in the range of about 24-64 kbps the BCC-based coding scheme has better quality than conventional perceptual transform audio coders for stereo. For instance, the MP3 [4] stereo encoder of 40 kbps is slightly worst in quality compared to the BCC stereo encoder of 34 kbps (where a 32 kbps MP3 encoder is used for encoding the reference channel). In addition, Parametric Stereo (PS) [5] also operates based on a similar SAC philosophy.

In the future, multichannel audio systems are going to be displaced by systems that *immerse* the listener into a virtual acoustic scene, which implies that the listener's acoustic environment will be seamlessly transformed into the environment of his/her desire, and that the listener will be able to interact and modify the content according to his/her will in a real-time manner. *Immersive audio* [32] is largely based on enhanced audio content, which translates into using a large number of microphones for obtaining an audio recording, containing as many sound sources as possible. These sound sources offer increased sound directions around the listener during reproduction, but are also useful for providing interactivity between the user and the audio environment.

Immersive audio applications include the collaboration of geographically distributed musicians [41], as well as tele-presence in a concert hall performance where the user can "move" around the venue. A mixing process is needed in such systems because the number of microphone signals is larger that the available loudspeakers. The interaction between the user and the audio environment can be accomplished only if the decoder has access to the multiple microphone signals before those are mixed to the final multichannel mix (remote mixing). These multiple microphone signals, also referred to as spot signals [1], are the signals that are captured, *e.g.*, by the various microphones that are placed inside a concert hall. Remote mixing is important for immersive audio applications since it provides the appropriate audio content needed for interactivity.

1.1 Scope of the thesis

In our work, the sinusoids plus noise model (SNM), which has been used extensively for monophonic audio signals, is introduced in the context of low-bitrate coding for immersive audio. This would allow for transmission of multichannel immersive audio environments through low bandwidth channels such as the current Internet infrastructure, and for broadcasting over wireless networks. As in the SAC method for low bitrate multichannel audio coding, our approach is to encode one audio channel only, which can be one of the spot microphone signals or a downmix sum signal, while for the remaining spot microphone signals we retain only the parameters that allow for resynthesis of the content at the decoder.

The main difference between the SAC and the proposed method is that SAC encodes the audio channels after the mixing process and the side information is used to recreate the spatial rendering of the channels, while the proposed approach focuses on the encoding of the multiple microphone signals before the mixing process of the multichannel recording. This is very important, since the actual content of each microphone recording has to be encoded (and not only its spatial image) in order to give to the audio engineer the ability of controlling the content of the recording. In addition, our goal is to achieve high-quality audio coding at bitrates lower than the standard multichannel compression algorithms of

[1] For further detail about spot microphone signals see Section 7.1.

AAC and Dolby AC-3.

The SNM model is considered to be a model that decomposes an audio signal into two parts; the sinusoidal part, which is a sum of sinusoidal functions, captures the harmonic nature of the signal and the noise part which constitutes the approximation error of the model. The parameters that allow for resynthesis of the microphone signals at the decoder are the sinusoidal parameters (sinusoidal part) of each spot microphone signal, as well as the short-time spectral envelope (estimated using Linear Predictive LP analysis) of the noise part of each spot microphone signal. These parameters, which are used as *side information* (see in Figure 1.2), are not as demanding in coding rates, as the true noise part of the SNM model. For this reason, the noise part of only the reference signal is retained.

For resynthesis of each spot microphone signal, we add the harmonic part that was fully encoded, to the noise part which is recreated by using the corresponding noise envelope with the noise LP residual obtained from the reference signal. This procedure, has been described in our recent work [50, 49] as *noise transplantation*, and is based on the observation that the noise parts of the various spot microphone signals of the same multichannel recording have similar audio content when the sinusoidal part has been captured with an appropriate number of sinusoidal functions.

A similar idea has been proposed previously in [22], where a multiresolution source/filter model is applied for coding of the spot recordings. In particular, the proposed method is based on a multiscale source/filter representation of the multiple spot microphone signals. The filter part corresponds to the specifics of each spot microphone information while the source part contains mostly the interchannel similarities. Using the appropriate filter for each channel and the source part of only one of the microphone signals, we can resynthesize a high quality approximation of each channel. The method consists of coding one audio signal only (reference channel), which can be a downmix of spot microphone recordings, along with side information consisting of the subband LPC envelopes of all the short-time frames for all microphone signals. The subjective results (average grades around 4.0 in a 5-grade scale are reported) of this work indicate that high audio quality both for modeling and coding can be achieved for bitrates as low as 10 kbps for the side information of each spot signal. However, *crosstalk* (*i.e.*, the "main" group of instruments that is captured by

a particular microphone remains the prominent part of the microphone signal, while other parts of the orchestra might be more audible in the resynthesized signal than in the original microphone signal) is introduced to the modeled signals. Alternatively, the sinusoids plus noise model used in this thesis can be employed for *alleviating the crosstalk problem, at the expense of the need for higher bitrates for coding.*

1.2 Thesis outline

The thesis is organized as follows. In Chapter 2 we give a description of the sinusoidal model. First, we give an overview of the sinusoidal parameters' estimation method. Then, we describe the synthesis stage of the sinusoidal model that produces the reconstructed audio signal. Finally, the sinusoids plus noise model is presented as an approach of more accurate representation of an arbitrary audio signal, since it takes into account the noise components of the signal.

Chapter 3 presents the theory of autocorrelation analysis that lies behind the technique of Linear Prediction analysis, which is considered to be a way of spectral estimation. The method of Linear Prediction is based on the approximation of each signal's sample with a linear combination of past samples. Furthermore, a description of the decorrelation method of stochastic processes, namely the Karhunen-Loève transform (KLT), is given.

Chapter 4 describes the different methods used for modeling the noise part of the sinusoids plus noise model. Firstly, an overview of the main principles of psychoacoustics is given. In the first approach, the noise is modeled using a filterbank based on the human auditory system. In the second noise modeling method, the noise is modeled by applying Linear Predictive analysis in the perceptual domain and representing only noise's frequency components that are of perceptual relevance. Finally, the third method of noise modeling is based on the ordinary LP analysis method.

In Chapter 5, the concept of statistical modelling using a mixture of Gaussian functions is introduced. The Gaussian mixture model (GMM) is used as a statistical tool for estimating the density of a stochastic process. The Expectation-Maximization (EM) algorithm is also presented, which constitutes the most popular iterative procedure for estimating the

parameter set of a GMM. In this thesis, the GMM and the Karhunen-Loève transform are used in a combined way in order to model and decorrelate the Line Spectral Frequencies used in the coding of the spectral envelopes of the multiple microphone signals. In Chapter 6, we set out the basic theory concerning the rate-distortion theory, as well as the theory of high-rate quantization. High-rate quantization theory comprises the main tool for coding the sinusoidal parameters of the multichannel signals.

In Chapter 7, we describe our proposed approach for modeling the multiple microphone signals using the noise transplantation method, while in Chapter 8 the algorithms for coding of the sinusoidal and the noise part are described. Finally, in Chapter 9 the main conclusions are summarized and future research directions are proposed.

Chapter 2

Sinusoidal Modeling

In the majority of speech applications such as time-scale and pitch modifications [37, 46], speech coding [23] *etc.*, the signal's accurate representation is an important task. The sinusoidal model was firstly applied for the analysis/synthesis of speech signals [30] as an attempt of representing the speech signal in a compact and efficient form. Since then, the sinusoidal model has been successfully used in all the areas of speech signal processing, leading to the use of the model in the more broad area of audio processing, which includes the signals of speech, music *etc.* In general, the sinusoidal model can be used for representing an audio signal which is harmonic or quasi-harmonic in nature. It can be considered as a spectral modeling technique, which models time-varying spectra as a collection of sinusoidal functions. Section 2.1 analyzes the main parts of the sinusoidal model.

The sinusoidal model is not accurate for the manipulation of non-harmonic parts of audio signals, such as breathy sound in speech or the attack of a drum stroke in music. A more accurate representation of such signals is achieved with the sinusoids plus noise model. The signal representation is obtained by restricting the sinusoids to modeling only the deterministic part of the audio signal, leaving the rest of the spectral information in the sinusoidal noise component. The sinusoids plus noise model is described in section 2.2.

2.1 Description of the sinusoidal model

The motivation for the sine-wave representation is that audio signals, when perfectly periodic, can be represented by a Fourier series decomposition in which each harmonic component corresponds to a single sinusoidal wave. As the signal becomes more periodic in nature the sinusoidal representation will be more accurate.

In the sinusoidal framework a discrete-time audio signal $s(n)$ is modeled as the sum of a predefined number of evolving sinusoids, called partials

$$s(n) = \sum_{l=1}^{L(n)} \alpha_l(n)\ \cos(\theta_l(n)), \tag{2.1}$$

where $L(n)$ is the number of partials at time n. The l^{th} partial $\alpha_l(n)\cos(\theta_l(n))$ has time-varying amplitude $\alpha_l(n)$ and total phase $\theta_l(n)$, which describes both its frequency evolution and phase offset. The additive components in the model are thus simply parameterized by amplitude and frequency functions or tracks.

2.1.1 Estimation of the sinusoidal parameters

To estimate the parameters of the sinusoidal model, one needs to segment the signal $s(n)$ into a number of short-time frames and compute the short-time Fourier transform (STFT) of the signal at each frame. In specific, the (discrete) time axis is subdivided into a sequence of overlapping frames analyzed with a time window of length N. The window $w(n)$ slides over the time axis with the hop size (time advance) defined at the beginning of the estimation process. The k^{th} frame of the signal $s(n)$, denoted as $s_k(n)$, is multiplied with the window $w(n)$ of length N, resulting in the short-time signal $x_k(n) = s_k(n) \cdot w(n)$, where $\sum_n w(n) = 1$. Typically, the window $w(n)$ is a Hamming, Hanning or Kaiser window.

After obtaining the frame $x_k(n)$, Fast Fourier Transform (FFT) of length M is applied to compute the magnitude and phase spectrum of the frame. The placement of the window $w(n)$ relative to the time origin is important for phase's computation. In specific, the window $w(n)$ takes values in the interval $0 \leq n < N$ and is symmetric about $(N-1)/2$.

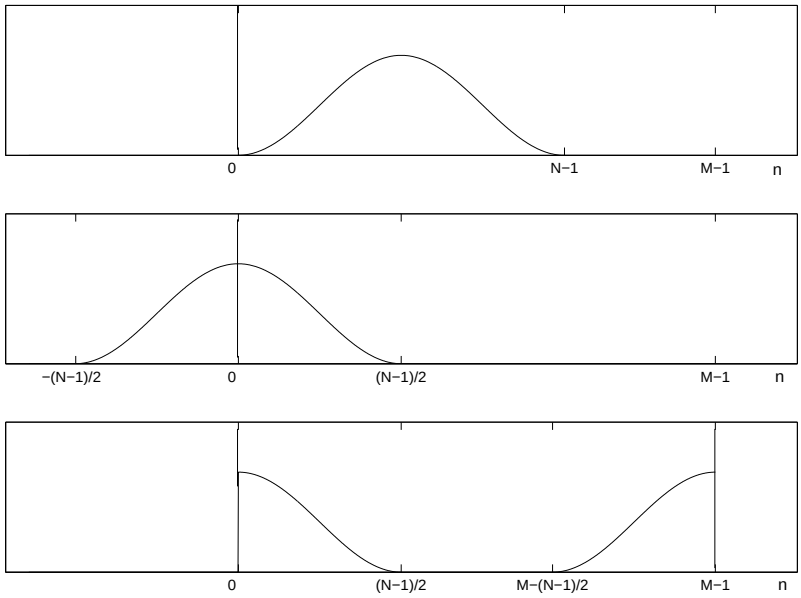

Figure 2.1: Circularly shifting of the windowed data of length N for reducing linear phases error. FFT size is denoted by M.

This placement of $w(n)$ gives a linear Fourier transform phase equal to $-\omega(N-1)/2$. Because of the fact that N is commonly on the order of up to 300 samples, any error in the estimated frequencies results in a large random phase error and thus, distortion in the reconstruction procedure. In order to diminish the linear-phase term, the frame $x_k(n)$ must be circularly shifted before before applying the FFT. The circularly shifting is shown in Figure 2.1.

Once the magnitude and phase spectrum are computed with the FFT, then the amplitudes are estimated by identifying the prominent spectral peaks of the magnitude spectrum using a peak detection algorithm. A peak is defined as the local maximum in the magnitude spectrum $|X_k(m)|$, of the frame $x_k(n)$, where m denotes the frequency bin (sample in the frequency spectrum). Thus, if m' is a bin number in the spectrum $|X_k(m)|$, then its value is a maximum when the following relation is satisfied

$$|X_k(m'+1)| \leq |X_k(m')| \ \ and \ \ |X_k(m')| \geq |X_k(m'-1)|. \tag{2.2}$$

Each local maximum is accurate only to within the half of a bin, because of the sampled nature of the spectra returned by the FFT, where a bin represents a frequency interval

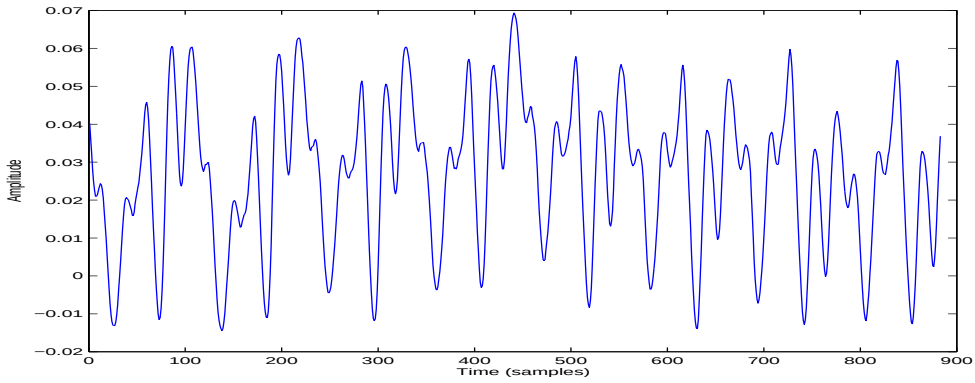

Figure 2.2: Time-domain waveform of a 20msec. segment of a violin sound.

of f_s/M Hz (f_s is the sampling frequency and M is the FFT size). The accuracy of the spectral peak detection can be increased by increasing the number of FFT bins per Hz using a larger amount of zero-padding in the time domain. However, to obtain good frequency accuracy, the zero-padding factor required is very large. A more practical solution is to use parabolic spectral interpolation [44], which fits a parabola through the three bins immediately surrounding the local maximum bin.

The general form of the parabola is $y - y_0 = a(x - x_0)^2$, where y_0 is the offset, a is a measure of concavity and x_0 is the center of parabola. Let us assume that the local maximum is y_2 at the frequency bin location x_2, (x_1, y_1) are the point's coordinates at the left of the maximum and (x_3, y_3) are the point's coordinates at the right of the local maximum. Solving for the parabola center location x_0 we get the true peak location (in bins)

$$x_0 = \frac{((y_3 - y_2)(x_1 + x_2) - (y_2 - y_1)(x_2 + x_3))}{(2(y_3 - 2y_2 + y_1))}, \tag{2.3}$$

and the value of a is given by the relation

$$a = \frac{(y_2 - y_1)}{(x_1 + x_2 - 2x_0)}. \tag{2.4}$$

Finally, the estimate of the true peak amplitude is

$$y_0 = y_1 - a(x_1 - x_0)^2. \tag{2.5}$$

The peak location in terms of Hz is given by $f_s\, x_0/M$.

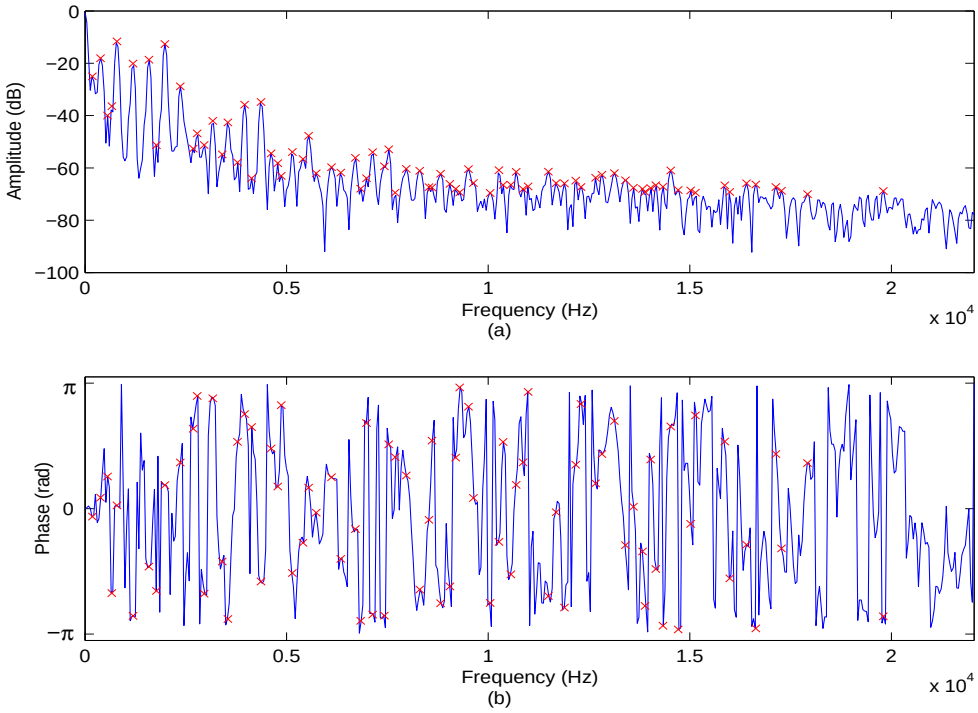

Figure 2.3: Peak detection on a spectrum of a violin sound's segment of 20msec. using 80 sinusoids in total: (a) magnitude spectrum, (b) wrapped phase spectrum

Figure 2.2 shows the time-domain waveform of a violin sound's segment with 20msec. duration. This segment is analyzed using FFT and the peak peaking algorithm described above is applied. Figure 2.3 shows the result of the peak detection on the spectrum, assuming that 80 sinusoids model the segment. In specific, Figure 2.3(a) shows the peak detection on the magnitude spectrum expressed in decibels (dB), where the spectral peaks, whose locations are denoted by the crosses, determine which frequencies are selected to represent the waveform, while in Figure 2.3(b) we can notice the corresponding phases.

2.1.2 Peak continuation

The peak detection process returns the sinusoidal parameters, but does not indicate which parameter sets correspond to a given partial. Thus, to build a signal model in terms of evolving partials that persist in time, it is necessary to form connections between the parameter sets in adjacent frames. The crucial problem is to decide how to connect the parameter sets in adjacent frames to establish continuity for the partials of the signal model.

A peak continuation algorithm [30] can be applied in a simple successive manner by

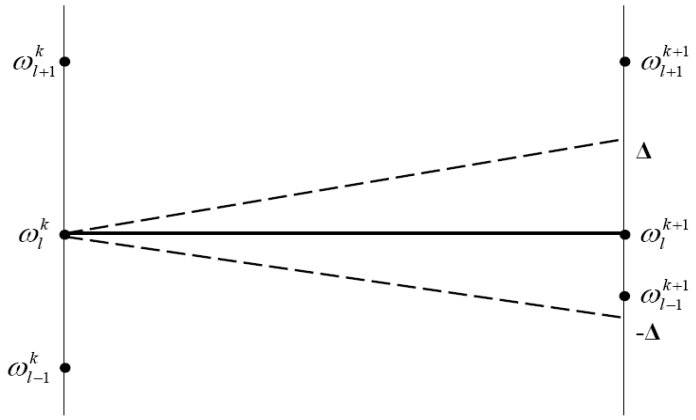

Figure 2.4: The matching interval condition used in nearest-neighbor sinewave frequency matching. [30]

associating the l^{th} parameter set in frame k, namely $\{\alpha_l^k, \omega_l^k, \phi_l^k\}$ (amplitude, frequency, phase), to the set in frame $k+1$ with frequency closest to ω_l^k. In particular, let us assume that at frame k the frequency values for the p tracks are $c_1, c_2, \ldots, c_p$ and that we want to match them to the r frequencies $d_1, d_2, \ldots, d_r$ of frame $k+1$. Each track looks for its peak in frame $k+1$ by finding the one which is more close in frequency to its current value. In other words, if the amount $|c_i - d_j|$ takes the minimum value, then the i^{th} track claims frequency d_j, where the change in frequency must fall within a matching interval $[-\Delta, \Delta]$ (see Figure 2.4). The following cases summarize the peak continuation process:

(a) If a frequency match is found inside the matching interval, then the sinewave track is continued, unless there is a conflict to resolve (a situation described below in (c)).

(b) If there is no frequency match inside the interval, the track with frequency c_i that enters frame $k+1$ is characterized as "dead", and c_i is matched to itself with the amplitude set to zero. The terminating sinewave track ramps to zero over the duration of one hop size, because of the linear ramp of the track amplitude from one frame to the next.

(c) If a track finds a frequency match that has already been claimed by another track,

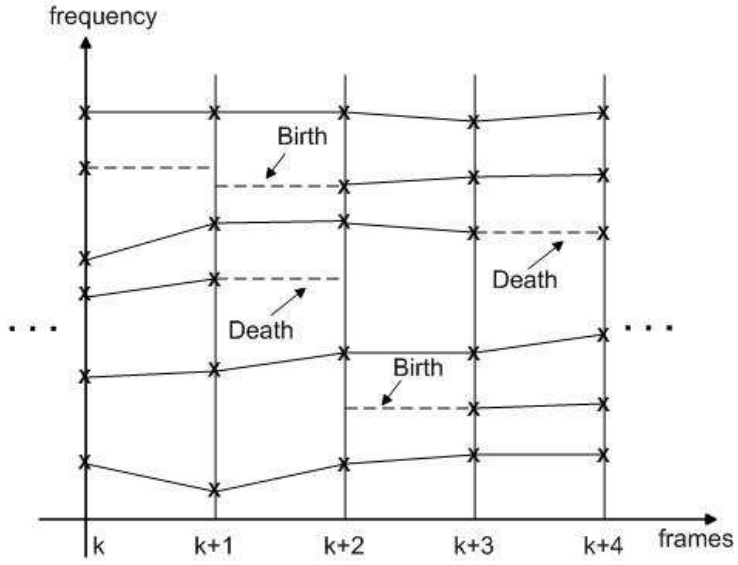

Figure 2.5: Different stages of the peak continuation procedure for determining frequency tracks. [30]

then the peak is given to the track which is closest in frequency, and the track that "loses" this conflict continues for searching another match. There are two cases concerning the current track: (i) if it "loses" the conflict, it is matched with the best available non-conflicting peak that falls inside the matching interval, while (ii) if it "wins" the conflict, it calls the assignment process on behalf of the dislodged track. This procedure is repeated for each track until all existing tracks are matched or "killed".

(d) The peaks of frame $k+1$ that have not been assigned to any track, are considered to be new tracks and a new track is "born" for each one of them. We assume the new tracks started from frame k with zero amplitude and ramped to the actual amplitude of frame $k+1$.

An illustration of the peak continuation algorithm showing the birth/death process is given in Figure 2.5

In Figure 2.6 an example is shown of typical frequency tracks for a signal that corresponds to a violin sound.

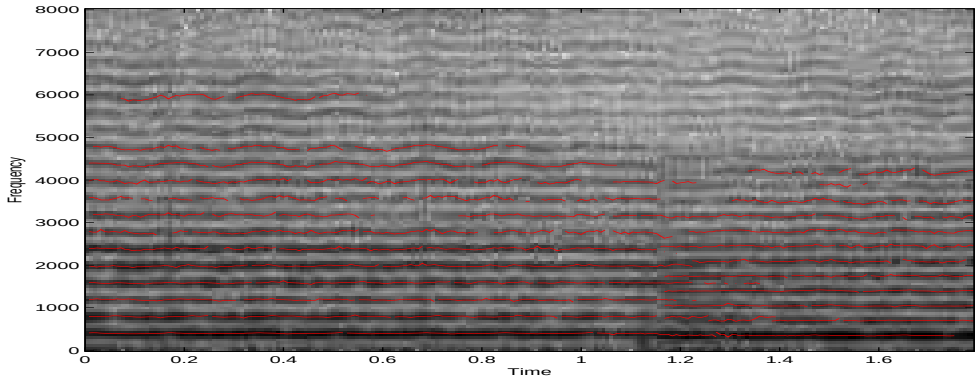

Figure 2.6: Frequency tracks for a signal of a violin sound.

2.1.3 Sinusoidal Synthesis

Since the peak continuation algorithm returns the values of the prominent peaks organized into frequency tracks, it might seem reasonable to estimate the original audio waveform on the $k+1^{th}$ frame by generating synthetic audio using the relation

$$\hat{s}_{k+1}(n) = \sum_{l=1}^{L} \alpha_l^{k+1} \cos(n\omega_l^{k+1} + \phi_l^{k+1}), \;\; n = 0, 1, \ldots S-1 \tag{2.6}$$

where S is the length of the synthesis frame. It should be noted that a synthesis frame of length S does not correspond to an analysis frame (in our case of length N). The synthesis frame $k+1$ goes from the middle of the analysis frame k to the middle of the analysis frame $k+1$, *i.e.*, corresponds to the analysis hop size. Thus, there is no overlap-add process at the synthesis stage, and the final reconstructed signal $\hat{s}(n)$ results from the juxtaposition of the synthesis frame.

Due to the time-varying nature of the sinusoidal parameters, discontinuities are produced at the frame boundaries, which is a factor of quality degradation of the synthetic audio signal. Therefore, some provision must be made for smoothly interpolating the sinusoidal parameters computed from one frame to the other. Let us assume that $\{\alpha_l^k, \omega_l^k, \phi_l^k\}$ and $\{\alpha_l^{k+1}, \omega_l^{k+1}, \phi_l^{k+1}\}$ denote the sets of sinusoidal parameters at frames k and $k+1$ for the l^{th} frequency track. A solution to the amplitude interpolation problem is to take the linear interpolation

$$\alpha_l^{k+1}(n) = \alpha_l^k + \frac{(\alpha_l^{k+1} - \alpha_l^k)n}{S}, \tag{2.7}$$

where $n = 0, 1, \ldots S-1$ is the time sample into the $k+1^{th}$ frame. The instantaneous phase

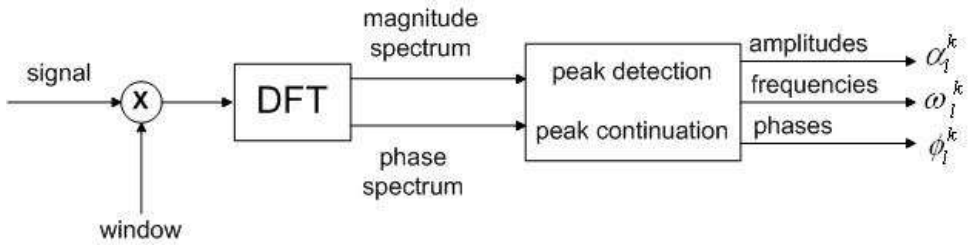

Figure 2.7: Block diagram of the sinusoidal analysis system. [43]

$\theta_l^{k+1}(n)$ presented in equation 2.1 is controlled from frequency and phase as follows

$$\theta_l^{k+1}(n) = n\omega_l^{k+1} + \phi_l^{k+1}\,. \tag{2.8}$$

The instantaneous phase is affected by four quantities: $\omega_l^k, \phi_l^k, \omega_l^{k+1}, \phi_l^{k+1}$. Thus, we need an interpolation function of three degrees

$$\theta_l^{k+1}(n) = \zeta + \gamma n + \lambda n^2 + \mu n^3\,. \tag{2.9}$$

The solution to this problem is mentioned in [30] and is given by the relation

$$\theta_l^{k+1}(n) = \phi_l^k + \omega_l^k n + \lambda n^2 + \mu n^3\,, \tag{2.10}$$

where the variables λ, μ are

$$\lambda = \frac{3}{S^2}(\phi_l^{k+1} - \phi_l^k - \omega_l^k S + 2\pi M) - \frac{1}{S}(\omega_l^{k+1} - \omega_l^k) \tag{2.11}$$

$$\mu = -\frac{2}{S^3}(\phi_l^{k+1} - \phi_l^k - \omega_l^k L + 2\pi M) + \frac{1}{S^2}(\omega_l^{k+1} - \omega_l^k)\,. \tag{2.12}$$

The aforementioned set of interpolating functions depend on the value of M. The maximal smoothness of the instantaneous function 2.10 is achieved by choosing M to be the integer closest to x, where

$$x = \frac{1}{2\pi}\left[(\phi_l^k + \omega_l^k S - \phi_l^{k+1}) + \frac{S}{2}(\omega_l^{k+1} - \omega_l^k)\right]. \tag{2.13}$$

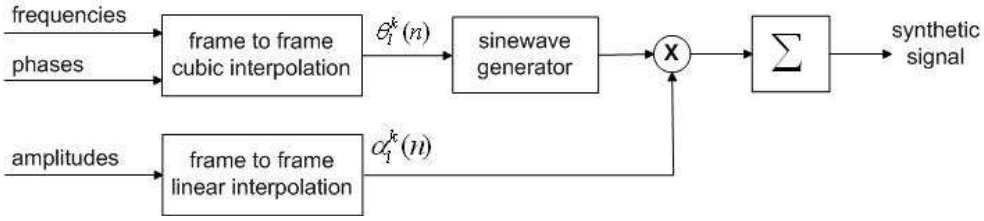

Figure 2.8: Block diagram of the sinusoidal synthesis system. [30]

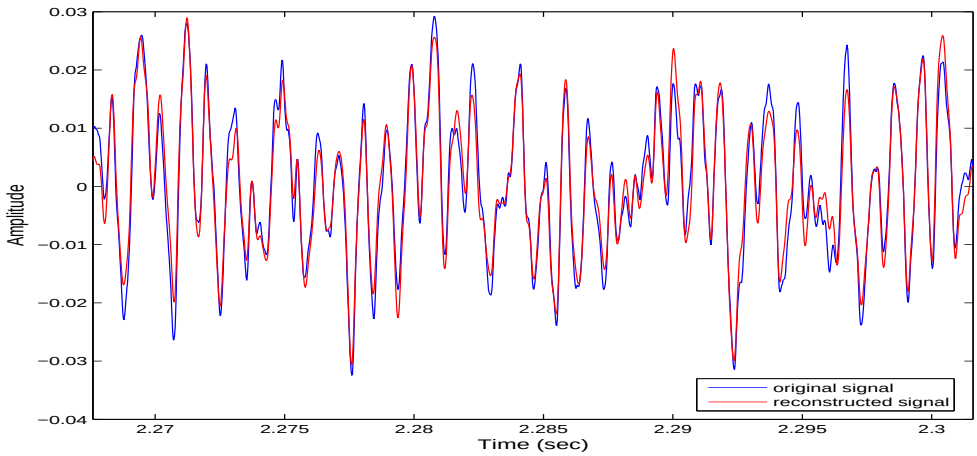

Figure 2.9: Reconstruction of a classical music signal's frame.

Finally, the final synthetic waveform for the $k+1^{th}$ frame is given by

$$\hat{s}_{k+1}(n) = \sum_{l=1}^{L} \alpha_l^{k+1}(n) \cos(\theta_l^{k+1}(n)) \,, \tag{2.14}$$

where for the l^{th} sinewave, $\alpha_l^{k+1}(n)$ is given by equation 2.7 and $\theta_l^{k+1}(n)$ is given by equation 2.10. A schematic representation of the sinusoidal analysis system is depicted in Figure 2.7, while in Figure 2.8 is shown the sinusoidal synthesis system.

Figure 2.9 shows a portion from a reconstructed signal of 1500 samples in length, obtained from a classical music audio file, using 25 sinusoids for the analysis and 20 msec. analysis/synthesis window with 50% overlapping.

2.2 Description of the sinusoids plus noise model

The sinusoidal model presented in the previous section does not provide an exact reconstruction of audio signals because of its trait for not modeling impulsive signals or highly

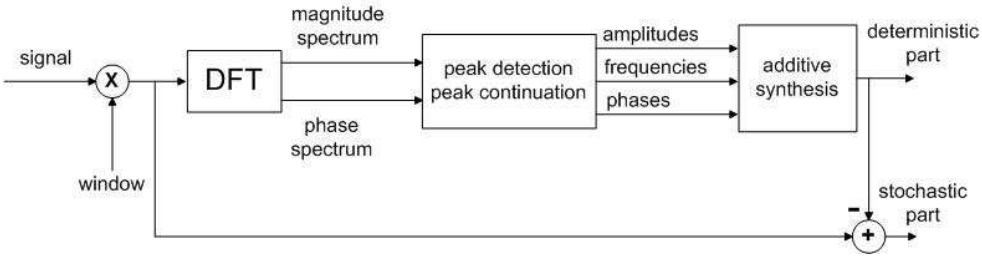

Figure 2.10: Block diagram of the sinusoids plus noise model. [43]

uncorrelated noise. In general, audio signals contain noise components (*e.g.* breathy part of a flute sound and aspiration at the glottis). Since these components are very important for high-quality reconstruction, an additional part must be considered in the signal, $s(n)$, *i.e.*, $s(n) = d(n) + e(n)$, where $d(n)$ corresponds to the deterministic part of the signal model which is harmonic, while $e(n)$ is the stochastic part.

Several variations of the sinusoids plus noise model have been proposed for applications such as signal modifications and low bitrate coding, focusing on three different problems: (1) accurately estimating the sinusoidal parameters from the original spectrum (presented in section 2.1), (2) representing the modeling error (stochastic component) whose modeling procedure will be analyzed in Chapter 4, and (3) representing signal transients. In particular, problem (1) has been extensively treated for speech signals, *e.g.*, [30, 47], and variations of these approaches have been extended to wideband audio. For addressing problem (3) use of damped sinusoids and AM modulated sinusoids (instead of constant amplitude sinusoids) have been proposed (*e.g.*, [20, 7]). In this thesis, we focus on the problem of noise representation. In music, a harmonic plus noise model was first proposed in [44], where the noise part was modeled based on a piecewise-linear approximation of its short-time spectral envelope or alternatively its Linear Predictive Coding (LPC) envelope (assuming white noise excitation during synthesis).

The audio signal representation is obtained by restricting the sinusoids to modeling only the deterministic part of the audio signal, leaving the rest of the spectral information in the stochastic component $e(n)$, *i.e.*, for each short-time frame the signal can be represented

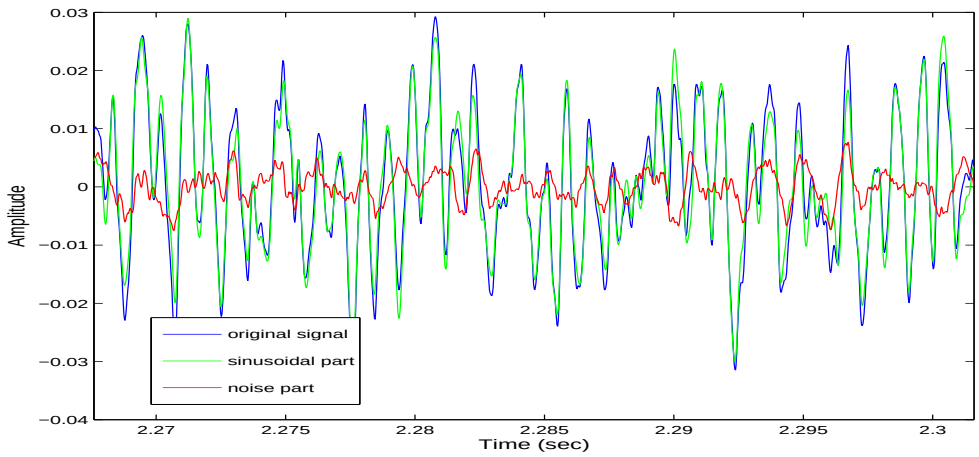

Figure 2.11: Decomposition of a music signal into a sinusoidal part and a noise part.

as

$$s(n) = d(n) + e(n) = \sum_{l=1}^{L} \alpha_l(n)\ \cos(\theta_l(n)) + e(n), \tag{2.15}$$

where L is the number of sinusoids, $\alpha_l(n)$ is the instantaneous amplitude and $\theta_l(n)$ the instantaneous phase. The stochastic component is defined as the difference between the original signal and the deterministic part, *i.e.*, $e(n) = s(n) - d(n)$. Figure 2.10 shows the block diagram of the sinusoids plus noise model. In Figure 2.11, the decomposition of the signal shown in Figure 2.9 into a sinusoidal part (green solid line) and a noise part (red solid line) is plotted.

Chapter 3

Linear Prediction Analysis and Low-Rank Approximation

Linear prediction (LP) is a fundamental tool in many areas of signal processing such as spectral estimation, filtering, system identification and speech. The basic idea behind LP analysis is that each sample of the signal can be approximated as a linear combination of past samples of the signal. In the next sections, first we introduce the notion of autoregressive process and the Yule-Walker equations and then the LP analysis method is described. Finally, we give an overview of the low-rank approximation method through the Karhunen-Loève transform.

3.1 Autoregressive processes

Let us assume the vector of $M+1$ values of a discrete-time random process $x(n)$

$$\boldsymbol{x} = [x(0), x(1), \ldots, x(M)]. \tag{3.1}$$

Definition 3.1.1 ***Wide sense stationarity****: a random process $x(n)$ is said to be wide sense stationary (WSS) if the following conditions are satisfied:*

1. *The mean of the process is a constant, $E\{x(n)\} = c$.*

2. *The autocorrelation $r_x(n, n-k) = E\{x(n)x^*(n-k)\}$ depends only on the time-*

difference $n-(n-k)=k$.

3. *The variance of the process is finite,*

$$E\{|x(n)|^2\} - E\{x(n)\} < \infty$$

The *mean ergodic theorem* [17] may be used to estimate the autocorrelation function of a WSS process with the time average

$$\hat{r}_x(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n-k)\,,\; 0 \le k \le N-1 \tag{3.2}$$

The outer product

$$\mathbf{x}\mathbf{x}^H = \begin{bmatrix} x(0)x^*(0) & x(0)x^*(1) & \dots & x(0)x^*(M) \\ x(1)x^*(0) & x(1)x^*(1) & \dots & x(1)x^*(M) \\ \vdots & \vdots & & \vdots \\ x(M)x^*(0) & x(M)x^*(1) & \dots & x(M)x^*(M) \end{bmatrix}$$

is a $(M+1)$x$(M+1)$ rectangular matrix. If $x(n)$ is wide sense stationary, then by taking the expected value and using the Hermitian property of the autocorrelation sequence, $r_x(k) = r_x^*(-k)$, leads to the $(M+1)$x$(M+1)$ autocorrelation matrix

$$\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^H\} \begin{bmatrix} r_x(0) & r_x^*(1) & r_x^*(2) & \dots & r_x^*(M) \\ r_x(1) & r_x(0) & r_x^*(1) & \dots & r_x^*(M-1) \\ r_x(2) & r_x(1) & r_x(0) & \dots & r_x^*(M-2) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(M) & r_x(M-1) & r_x(M-2) & \dots & r_x(0) \end{bmatrix}$$

Below are some of the important properties of the autocorrelation matrix:

Property 1: The autocorrelation matrix of a WSS random process $x(n)$ is a Hermitian Toeplitz matrix, *i.e.*, all of the elements along each of the diagonals have the same value.

Property 2: The autocorrelation matrix of WSS random process is nonnegative def-

inite, $\mathbf{R}_x > 0$.

Let us assume the time series $x(n), x(n-1), \ldots, x(n-M)$. We say that it represents the realization of an autoregressive process (AR) of order M if it satisfies the following equation

$$x(n) + a_1^* x(n-1) + \ldots + a_M^* x(n-M) = w(n), \tag{3.3}$$

where the constants $a_1, a_2, \ldots a_M$ are called AR coefficients and $w(n)$ is a white-noise process. The use of term *autoregressive* can be explained by rewriting equation 3.3 in the form

$$x(n) = b_1^* x(n-1) + \ldots + b_M^* x(n-M) + w(n), \tag{3.4}$$

where $b_k = -a_k$. Thus, we can notice that the present value of, $x(n)$, of the process is a linear combination of the past samples $x(n-1), \ldots, x(n-M)$ of the process, plus an error term $w(n)$. Generally, the linear model of the form

$$x = \sum_{i=1}^{M} b_i y_i + e, \tag{3.5}$$

where the variable x is a linear combination of the independent variables y_i plus an error term e, is called *regression model*. Equation 3.5 is called autoregressive because $x(n)$ is regressed on past values of itself. Equation 3.3 can be written in a more compact form as follows

$$\sum_{i=0}^{M} a_i^* x(n-i) = w(n), \tag{3.6}$$

where $a_0^* = 1$ and the left-hand side of the relation is the convolution of $x(n)$ and the sequence of parameters a_n^*. Let $A(z)$ denote the z-transform of the sequence a_n^*, $A(z) = \sum_{n=0}^{M} a_n^* z^{-n}$, $X(z)$ denote the z-transform of the sequence $x(n)$, $X(z) = \sum_{n=0}^{\infty} x(n) z^{-n}$ and $W(z)$ denote the the z-transform of $w(n)$, $W(z) = \sum_{n=0}^{M} v(n) z^{-n}$. By applying the

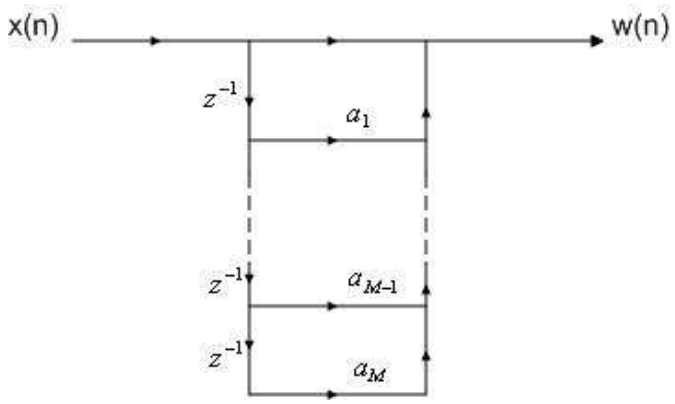

Figure 3.1: Direct form realization of the AR process analyzer filter.

z-transform to the equation 3.6 we obtain

$$A(z)X(z) = W(z) \Rightarrow A(z) = \frac{W(z)}{X(z)} = \sum_{n=0}^{M} a_n^* z^{-n}, \tag{3.7}$$

where $A(z)$ is a filter (called AR analyzer) that takes $x(n)$ as input and produces $w(n)$ as its output. Thus, the AR analyzer transforms an AR process at its input to white noise at its output. Figure 3.1 shows the direct from realization [34] of the AR analyzer, which is an Finite Impulse Response (FIR) all-zero filter.

Inversely, if the white noise $w(n)$ act as input, we have the filter for synthesizing the AR process $x(n)$

$$A(z)X(z) = W(z) \Rightarrow \frac{1}{A(z)} = \frac{X(z)}{W(z)} = \frac{1}{\sum_{n=0}^{M} a_n^* z^{-n}}. \tag{3.8}$$

Direct form realization of the synthesis filter is shown in Figure 3.2. Synthesis filter is an all-pole filter whose impulse response length is infinite (IIR).

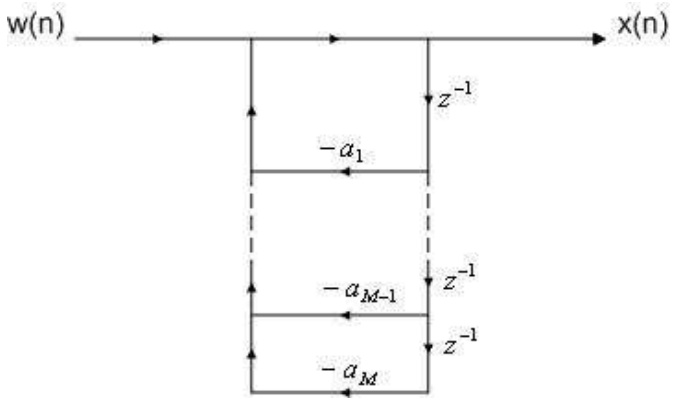

Figure 3.2: Direct form realization of the AR process synthesizer filter.

3.2 Yule-Walker equations

This section describes the method of estimating the AR parameters $a_1, a_2, \ldots a_M$ defined in the previous section. From equation 3.3 we obtain the following

$$x(n) + a_1^* x(n-1) + \ldots + a_M^* x(n-M) = w(n) \Rightarrow \sum_{i=0}^{M} a_i^* x(n-i) = w(n)$$

$$\Rightarrow \sum_{i=0}^{M} a_i^* x(n-i)x^*(n-l) = w(n)x^*(n-l)$$

$$\Rightarrow E\Big\{ \sum_{i=0}^{M} a_i^* x(n-i)x^*(n-l) \Big\} = E\{w(n)x^*(n-l)\} \text{ (w(n) and x(n) are uncorrelated)}$$

$$\Rightarrow \sum_{i=0}^{M} a_i^* E\Big\{ x(n-i)x^*(n-l) \Big\} = 0$$

$$\Rightarrow \sum_{i=0}^{M} a_i^* r_x(l-i) = 0\,, \;\; l > 0. \tag{3.9}$$

Thus, the autocorrelation function of the AR process satisfies the difference equation

$$r_x(l) = -a_1^* r_x(l-1) - a_2^* r_x(l-2) - \ldots - a_M^* r_x(l-M)\,, \;\; l > 0. \tag{3.10}$$

For $l = 1, 2, \ldots, M$ we get a set of M simultaneous equations which can be expressed in the following matrix from

$$\begin{bmatrix} r_x(0) & r_x(1) & \ldots & r_x(M-1) \\ r_x^*(1) & r_x(0) & \ldots & r_x(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_x^*(M-1) & r_x^*(M-2) & \ldots & r_x(0) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} r_x^*(1) \\ r_x^*(2) \\ \vdots \\ r_x^*(M) \end{bmatrix}$$
$$\Rightarrow \quad \mathbf{R}_x \mathbf{w} = \mathbf{r} \tag{3.11}$$

where $w_k = -a_k^*$, $k = 1, \ldots, M$ are the unknown parameters. If the rectangular matrix $\mathbf{R}_x$ is invertible, *i.e.*, its determinant is nonzero, the parameters w_i can be computed from the relation $\mathbf{w} = \mathbf{R}_x^{-1}\mathbf{r}_x$.

3.3 Power spectral density

The power spectral density (PSD), also referred to as the power spectrum, of a stochastic process gives a description of the frequency behavior of the process itself. Recall that the autocorrelation sequence of a WSS process provides a time domain description of the second-order moment of the process. Since $r_x(k)$ is a deterministic sequence, the discrete-time Fourier transform can be computed

$$S_x(e^{j\,\omega}) = \sum_{k=-\infty}^{\infty} r_x(k) e^{-j\,k\,\omega}, \tag{3.12}$$

which is called the power spectral density or power spectrum of the process. It should be pointed out that for nonzero mean random processes, the PSD is normally defined to be the discrete-time Fourier transform of the autocovariance $v_x(n, n-k) = E\{[x(n) - E\{x(n)\}][x(n-k) - E\{x(n-k)\}]^*\}$. Given the PSD, the autocorrelation sequence may be determined by taking the inverse discrete-time Fourier transform of $S_x(e^{j\,\omega})$

$$r_x(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_x(e^{j\,\omega}) e^{j\,k\,\omega} d\omega. \tag{3.13}$$

Some of the main properties of the PSD are the following:

Property 1: The PSD of a WSS random process is a real-valued function of ω. This follows from the fact that the autocorrelation function of the process is conjugate symmetric.

Property 2: The PSD of a WSS random process is nonnegative, $S_x(e^{j\,\omega}) \geq 0$.

Property 3: The power in a zero-mean WSS random process is given by

$$E\left\{|x(n)^2|\right\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_x(e^{j\,\omega})d\omega. \tag{3.14}$$

This property follows from Equation 3.13 with $k = 0$ and the fact that $r_x(0) = E\{|x(n)^2|\}$.

Theorem 3.3.1 ***Correlation relations between the input and output of a linear time-invariant (LTI) system.*** *Let us assume that the input to an LTI system with impulse response $h(n)$ is a WSS process $x(n)$ with output $y(n)$. Then*

$$r_{yx}(k) = h(k) * r_x(k), \tag{3.15}$$

$$r_{xy}(k) = h(-k) * r_x(k), \tag{3.16}$$

$$r_y(k) = h(k) * r_{xy}(k), \tag{3.17}$$

$$r_y(k) = h(k) * h(-k) * r_x(k), \tag{3.18}$$

where $r_{xy}(p,q) = E\{x(p)y^*(q)\}$ is the cross-correlation between $x(n)$ and $y(n)$. When $x(n)$ and $y(n)$ are jointly WSS, the cross-correlation depends only the time difference $k = p - q$, hence, $r_{xy}(k) = E\{x(n)y^*(n-k)\}$. Equation 3.18 indicates that the autocorrelation function of the output process is a twofold convolution of the input autocorrelation function with the system's impulse response.

Theorem 3.3.2 ***Power spectral density of the output of an LTI system.*** *Let us assume that the frequency response of an LTI system is $H(e^{j\,\omega})$, the system input is the WSS random process $x(n)$ and the output is the process $y(n)$. Then*

$$S_y(e^{j\,\omega}) = |H(e^{j\,\omega})|^2 S_x(e^{j\,\omega}), \tag{3.19}$$

is the PSD of the output process $y(n)$. This relation is obtained by applying the Fourier transform to Equation 3.18.

3.4 Linear prediction model

As it has already mentioned, LP analysis is a very important part of a large set of signal processing algorithms. It is based on the idea of estimating the sample of a discrete-time random process at the time instant n using a linear combination of the past samples of the process. Within the core of the LP scheme lies the AR model analyzed in the previous sections. In fact, LP analysis is a procedure for estimating the values of AR parameters given signal's samples. Thus, LP is an identification method where parameters of a system are found from the observation. Besides, LP can be viewed as a spectrum estimation method, allowing the computation of the AR parameters, which define the Power Spectral Density (PSD) of the signal itself.

Consider a time series $x(n), x(n-1), \ldots, x(n-M)$. The sample $x(n)$ can be estimated using the M past samples $x(n-1), x(n-2), \ldots, x(n-M)$, where M is often called the order of the linear predictor. Thus, an FIR linear predictor of order M has the following form

$$\hat{x}(n) = \sum_{i=1}^{M} w_i^* x(n-i), \tag{3.20}$$

where w_i^* for $i = 1, \ldots, M$ are the coefficients of the prediction filter. The linear predictor (Figure 3.3) consists of M unit delay elements and M (tap) weights $w_1^*, \ldots, w_M^*$ that are fed with the respective samples $x(n-1), \ldots, x(n-M)$ as inputs. The predicted value $\hat{x}(n)$ defined in Equation 3.20 constitutes the output. Hence, the prediction error is given by

$$e(n) = x(n) - \hat{x}(n) = x(n) - \sum_{i=1}^{M} w_i^* x(n-i) = \sum_{i=0}^{M} c_i x(n-i), \tag{3.21}$$

where

$$c_i = \begin{cases} 1\ , & i = 0 \\ -w_i^*\ , & i = 1, \ldots, M. \end{cases}$$

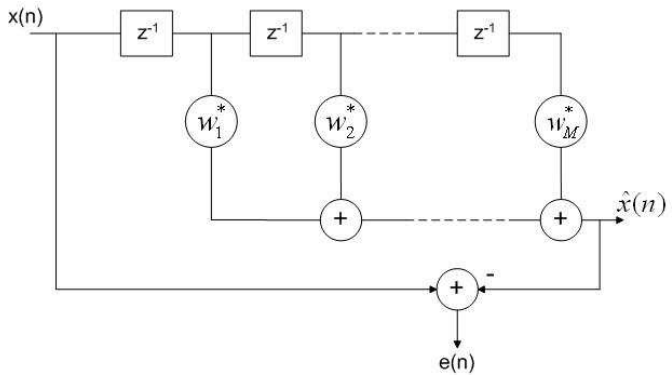

Figure 3.3: Linear prediction filter. [17]

From Equation 3.19 and the relation $x(n) = \sum_{i=1}^{M} w_i^* x(n-i) + e(n)$, we have the following pair of relations

$$\begin{aligned} S_x(e^{j\omega}) &= \left|1 + \sum_{i=1}^{M} w_i^* e^{-j\omega k}\right|^2 S_e(e^{j\omega}) \\ S_e(e^{j\omega}) &= \left|\frac{1}{1 + \sum_{i=1}^{M} w_i^* e^{-j\omega k}}\right|^2 S_x(e^{j\omega}), \end{aligned}$$

which means that if $x(n)$ is used as an input to the all-zero filter $1 + \sum_{i=1}^{M} w_i^* e^{-j\omega k}$ then the output is the prediction error $e(n)$, while if $e(n)$ is used as an input to the all-pole filter $1/(1 + \sum_{i=1}^{M} w_i^* e^{-j\omega k})$ then $x(n)$ is the output of the filter. Thus, we can see that there is an analysis/synthesis relation between $x(n)$ and $e(n)$.

The linear predictor can be considered as Wiener filtering problem (see Figure 3.4), by setting the desired response equal to $x(n)$, which is the sample we want to predict. The Wiener filter design problem requires that we find the filter coefficients, w_i^* , $i = 1, \ldots, M$ that minimize the mean-square error

$$\varepsilon = E\left\{|e(n)|^2\right\} = E\left\{|x(n) - \hat{x}(n)|^2\right\} \tag{3.22}$$

by selecting the appropriate coefficients w_i^*. The cost function ε is precisely a second-order function of the coefficients w_i^*. Consequently, we may visualize the dependence of the cost function ε on the coefficients as bowl-shaped $(M+1)$-dimensional surface, which is

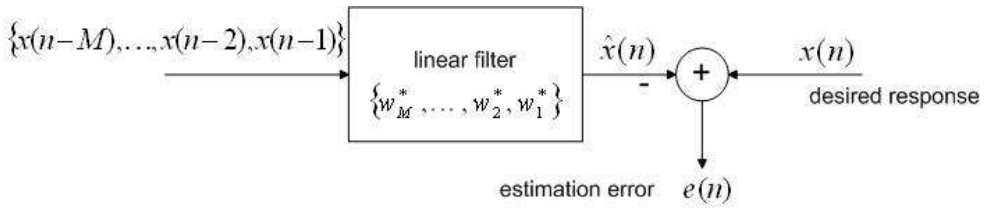

Figure 3.4: Block diagram of the Wiener filtering. [17]

characterized by a unique minimum point. The necessary and sufficient condition for ε to be minimized is that the derivative of ε with respect to w_i^* be equal to zero for $i = 1, \ldots, M$. Equation 3.22 can be rewritten using matrix form notation

$$\begin{aligned} \varepsilon &= E\Big\{|e(n)|^2\Big\} = E\Big\{|x(n) - \hat{x}(n)|^2\Big\} \\ &= E\Big\{\Big|x(n) - \sum_{i=1}^{M} w_i^* x(n-i)\Big|^2\Big\} = E\Big\{\Big|x(n) - \mathbf{w}^H \mathbf{x}(n)\Big|^2\Big\}, \end{aligned} \tag{3.23}$$

where

$$\mathbf{w} = \begin{bmatrix} w_1^* \\ w_2^* \\ \vdots \\ w_M^* \end{bmatrix}, \ \mathbf{x}(n) = \begin{bmatrix} x(n-1) \\ x(n-2) \\ \vdots \\ x(n-M) \end{bmatrix}.$$

From Equation 3.23 we have

$$\begin{aligned} \varepsilon &= E\Big\{\big|x(n) - \mathbf{w}^H \mathbf{x}(n)\big| \cdot \big|x^*(n) - \mathbf{x}^H(n)\mathbf{w}\big|\Big\} \\ &= E\Big\{|x(n)|^2\Big\} + \mathbf{w}^H E\Big\{\mathbf{x}(n)\mathbf{x}^H(n)\Big\}\mathbf{w} - \mathbf{w}^H E\Big\{\mathbf{x}(n)x^*(n)\Big\} - E\Big\{x(n)\mathbf{x}^H(n)\Big\}\mathbf{w} \\ &= \sigma_x^2 + \mathbf{w}^H \mathbf{R}_x^H \mathbf{w} - \mathbf{w}^H \mathbf{p} - \mathbf{p}^H \mathbf{w}, \end{aligned} \tag{3.24}$$

where

$$\mathbf{p} = E\{\mathbf{x}(n)x^*(n)\} = \begin{bmatrix} E\{x(n-1)x^*(n)\} \\ E\{x(n-2)x^*(n)\} \\ \vdots \\ E\{x(n-M)x^*(n)\} \end{bmatrix}$$

with $E\{x(n-k)x^*(n)\}$ be the cross-correlation between the filter input and the desired response $x^*(n)$ for a time-lag of $-k$. Hence, the derivative of ε with respect to w_i^* [17] gives the following relation

$$\mathbf{R}_x\mathbf{w} = \mathbf{p}, \tag{3.25}$$

which is called the Wiener-Hopf equations. If the correlation matrix $\mathbf{R}_x$ is nonsingular, the final solution of computing the coefficients w_i^* is given by

$$\mathbf{w} = \mathbf{R}_x^{-1}\mathbf{p}. \tag{3.26}$$

The linear system 3.25 can be solved using the Levinson-Durbin algorithm [17]. This is a computationally efficient recursive algorithm and takes advantage of the Toeplitz structure of the correlation MxM matrix $\mathbf{R}_x$. The number of multiplications and divisions is proportional to M^2 for the Levinson-Durbin algorithm compared with M^3 for Gaussian elimination. Besides, the Levinson-Durbin algorithm requires less memory for data storage. Specifically, Gaussian elimination requires M^2 memory locations, while the Levinson-Durbin recursion requires $2(M+1)$ locations.

In the many of audio coding applications, such as speech coding, the LP coefficients of the signal's autoregressive modeling error are transformed to Line Spectral Frequencies (LSF's) [45] which constitutes an alternative LP spectral representation. In this study, we use the LSF's for coding the noise part of the microphone spot signals.

3.5 Decorrelation of a stochastic process: Karhunen-Loève transform

Let us assume the Nx1 observation vector $\mathbf{x}(n)$, of a WSS random process, with correlation matrix $\mathbf{R}_x$. We are interested in finding a non-zero vector $\mathbf{c}$ such that is satisfies the equation

$$\mathbf{R}_x\mathbf{c} = \lambda\mathbf{c}, \tag{3.27}$$

where the vector $\mathbf{c}$ is called the eigenvector of the Equation 3.27, and the scalar quantity λ is called the eigenvalue of the Equation 3.27. The eigenvectors of Equation 3.27 can be computed by solving the equation

$$det(\mathbf{R}_x - \lambda\mathbf{I}) = 0, \tag{3.28}$$

which is called the characteristic polynomial of $\mathbf{R}_x$. The roots of the characteristic polynomial give the eigenvalues of $\mathbf{R}_x$. The set of roots $\{\lambda_1, \ldots, \lambda_N\}$ of the characteristic polynomial is called the spectrum of $\mathbf{R}_x$ and is denoted by $\lambda(\mathbf{R}_x)$. Thus, the i^{th} eigenvalue λ_i and the i^{th} eigenvector $\mathbf{c}_i$ satisfy the relation

$$\mathbf{R}_x\mathbf{c}_i = \lambda_i\mathbf{c}_i. \tag{3.29}$$

Below the properties of the eigenvalues and eigenvectors of the correlation matrix $\mathbf{R}_x$ are mentioned

Property 1: If $\lambda_1, \ldots, \lambda_N$ denote the eigenvalues of the correlation matrix $\mathbf{R}_x$, then the eigenvalues of the matrix $\mathbf{R}_x^k$ equal to $\lambda_1^k, \ldots, \lambda_N^k$ for $k \in \mathbb{Z}$, $k > 0$.

Property 2: The eigenvectors $\mathbf{c}_1, \ldots, \mathbf{c}_N$ are linearly independent. The eigenvectors $\{\mathbf{c}_i\}_{i=1}^N$ are linearly independent if the scalars $\mu_1, \ldots, \mu_N$ do not exist that are not all zeros, such that $\sum_{i=1}^N \mu_i\mathbf{c}_i = 0$.

Property 3: The eigenvalues $\lambda_1, \ldots, \lambda_N$ of the correlation matrix $\mathbf{R}_x$ are real and nonnegative.

Property 4: The eigenvectors $\mathbf{c}_1, \ldots, \mathbf{c}_N$ of the correlation matrix $\mathbf{R}_x$ are orthogonal to each other, *i.e.*, $\mathbf{c}_i^H\mathbf{c}_j = 0$, $\forall i, j$, $i \neq j$.

Property 5: The sum of the eigenvalues $\lambda_1, \ldots, \lambda_N$ equals the trace of matrix $\mathbf{R}_x$, *i.e.*, $tr(\mathbf{R}_x) = \sum_{i=1}^N \lambda_i$.

Property 6: Each eigenvalue λ_i, $i = 1, \ldots, N$ of the correlation matrix $\mathbf{R}_x$ is bounded by the minimum and maximum values of the power spectral density of $\mathbf{x}(n)$.

Property 7: The correlation matrix $\mathbf{R}_x$ can be diagonalized as $\mathbf{C}^H\mathbf{R}_x\mathbf{C} = \mathbf{\Lambda}$, where

$$\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_N], \text{ with } \mathbf{c}_i^H\mathbf{c}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{3.30}$$

and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$. Due to the orthonormal nature of the eigenvectors, as defined in Equation 3.30, we have that (unitary matrix property)

$$\mathbf{C}^H\mathbf{C} = \mathbf{I} \Rightarrow \mathbf{C}^{-1} = \mathbf{C}^H.$$

Thus, the relation $\mathbf{C}^H\mathbf{R}_x\mathbf{C} = \mathbf{\Lambda}$ can be rewritten as

$$\begin{aligned} \mathbf{R}_x &= \mathbf{C}\mathbf{\Lambda}\mathbf{C}^H = \sum_{i=1}^{N} \lambda_i \mathbf{c}_i \mathbf{c}_i^H \\ &\Rightarrow \mathbf{\Lambda} = \mathbf{C}^H\mathbf{R}_x\mathbf{C}, \end{aligned}$$

where $\lambda_1, \ldots, \lambda_N$ are the eigenvalues of the correlation matrix $\mathbf{R}_x$ and the matrix $\mathbf{Q}$ is called the KLT matrix.

Now, let us assume that the Nx1 random vector $\mathbf{x}(n)$ has a zero mean. Let $\mathbf{y}(n) = \mathbf{C}^{-1}\mathbf{x}(n) = \mathbf{C}^H\mathbf{x}(n)$. Then, $\mathbf{y}(n)$ is a random vector with uncorrelated components, since $E\{\mathbf{y}(n)\mathbf{y}^H(n)\} = E\{\mathbf{C}^H\mathbf{x}(n)\mathbf{x}^H(n)\mathbf{C}\} = \mathbf{C}^H E\{\mathbf{x}(n)\mathbf{x}^H(n)\}\mathbf{C} = \mathbf{C}^H\mathbf{R}_x\mathbf{C} = \mathbf{\Lambda}$. Therefore, the matrix $\mathbf{C}$ can be viewed as a *decorrelation filter*, since the correlation matrix of vector $\mathbf{y}(n)$ is diagonal. Thus, we can conclude that the cross-correlation has been removed and the vector $\mathbf{x}(n)$ has been transformed into a decorrelated vector $\mathbf{y}(n)$, with the use of the KLT matrix $\mathbf{C}$.

The Karhunen-Loève transform have theoretical application to transform coding for data compression [13]. KLT is also used in many other applications of signal processing such as speech enhancement [38], image coding [27] and multichannel audio coding [55].

Chapter 4

Residual modeling

In Section 2.1, sinusoidal model presented as a useful method for parametric representation of audio signals. However, as it is mentioned in Section 2.2, sinusoids cannot be used alone for high-quality audio modeling because they do not represent all the audible information of an audio signal. Thus, it is necessary to separately model the noise component, which mainly captures the noisy part of the signal, and incorporate it into the reconstruction to achieve higher audio quality leading to the sinusoids plus noise model proposed in [44]. This approach is depicted in Figure 4.1. The choice of the appropriate method for modeling the noise component is crucial in the sinusoids plus noise model approach. In the pioneer work of Serra [44], the noise component is modeled using a piecewise-linear spectral estimation, where a random phase is applied to this spectrum, and an inverse discrete Fourier transform (IDFT) followed by overlap-add (OLA) is used for synthesis. In the next sections, three

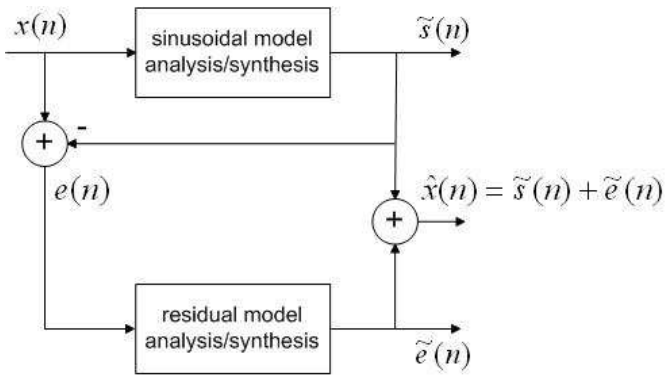

Figure 4.1: Sinusoids plus noise modeling.

different methods used for noise modeling are presented.

Specifically, in section 4.2 we focus in the noise part modeling using a filterbank based on the human auditory system [14]. In section 4.3 the noise component is modeled by applying Linear Predictive (LP) analysis in the perceptual domain and representing only noise components that are of perceptual relevance [18]. The third method of noise modeling is based on the ordinary LP analysis method [44].

4.1 Principles of psychoacoustics

Before proceeding to the next sections of noise modeling, it is necessary to provide a short description of the fundamental psychoacoustic principles. In particular, sound is generated through the mechanical vibration of objects, where the vibrating motion travels through physical media, causing acoustic waves. In most cases, the physical medium corresponds to air while the sound waves represent the variations of atmospheric pressure. The magnitude of sound is represented as a time-varying pressure, expressed in units of Pascal (Pa). Relevant values of sound pressure vary between 10^{-5} Pa, which is close to the limits of human hearing at the most sensitive frequencies, to 10^2 Pa, which corresponds to the threshold of pain [56]. Given the extent of this range, we can describe sound pressures in logarithmic units and define the sound pressure level (SPL) in units of decibel (dB) as

$$L = 20 \log_{10} \left(\frac{p}{p_0} \right) \text{ dB}, \tag{4.1}$$

in which p is the pressure produced by a sound source and p_0 is the reference pressure of 20 μPa that corresponds to the minimum audible threshold of human hearing for a 1 kHz tone. Sounds are also described in terms of sound intensity, which represents the sound energy transmitted per second through a unit area of a sound field and the SPL in terms of sound intensity can be expressed as

$$L = 10 \log_{10} \left(\frac{I}{I_0} \right) \text{ dB}, \tag{4.2}$$

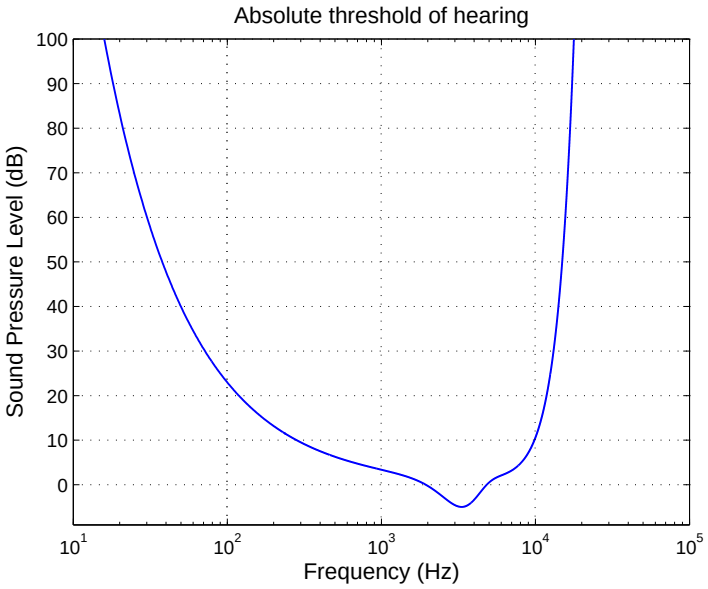

Figure 4.2: Absolute threshold of hearing.

where the reference intensity I_0 has value of 10^{-12} W/m^2. One way of evaluating the level of a sound is from its frequency structure. For discrete spectra (*e.g.*, periodic signals) the overall sound level is calculated by summing the levels of individual spectral components, which are directly related to the squared magnitude of the signal's Fourier series coefficients.

The most important characteristic of loudness perception, especially for the design of modern audio codecs, is the dependence of loudness on frequency. This kind of dependence is described with the frequency dependent function $T_q(f)$, called absolute threshold of hearing

$$T_q(f) = 3.64\Big(\frac{f}{1000}\Big)^{-0.8} - 6.5\, e^{-0.6\big(f/1000-3.3\big)^2} + 10^{-3}\Big(\frac{f}{1000}\Big)^4 \text{ dB}, \tag{4.3}$$

where f is expressed in Hz. The absolute threshold of hearing, which is easily measured through hearing experiments, characterizes the amount of energy needed in a pure tone such that it can be detected by a listener in a noiseless environment. A mean threshold is obtained by averaging the individual thresholds of numerous listeners. From Figure 4.2 we can see that the human ear is less sensitive to low frequencies and thus a much higher SPL level is required to produce the same perceived loudness as that of a sound at high frequencies. Besides, the absolute threshold of hearing is extremely important for audio

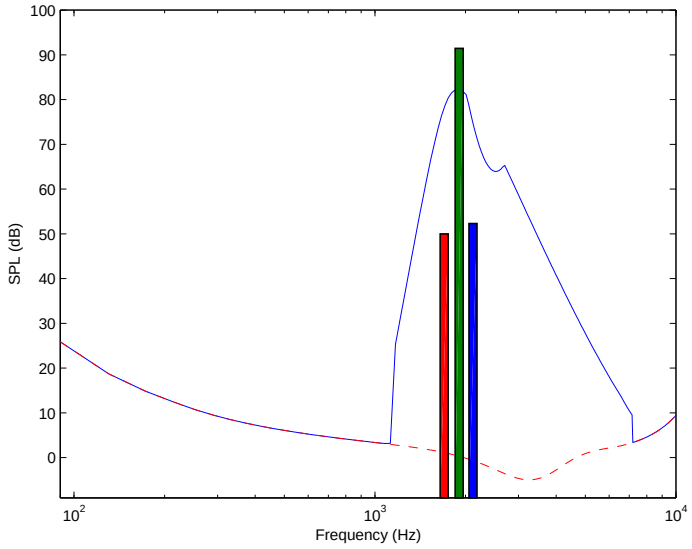

Figure 4.3: Example of frequency masking.

coding applications since a signal's frequency components that fall below this level are not perceived by the human ear and therefore they do not need to be transmitted. In addition, if the quantization noise frequency components that are transmitted falls below this level, then it is not detectable by the human auditory system.

Auditory masking is another characteristic of the human hearing, by which the perception of a sound is suppressed by another one. Masking is characterized by an increase in the audibility threshold of a sound in the presence of a louder sound. The amount of masking corresponds to the quantity by which the threshold is augmented above the absolute threshold of hearing. The masking sound is commonly referred to as the masker, while the sound being masked is referred to as the maskee.

Masking effects can be divided into two categories. The first category is called frequency masking and it is observed when a loud masking sound occurs at the same time as the maskee sound and it is no longer possible to hear the normally audible maskee. The second type of masking is called temporal masking and it is created when the masker and maskee sounds occur at different times. Temporal masking in turn can be subdivided into backward masking in which the maskee is generated before the masker, and into forward masking in which the maskee comes after the masker.

As an example of frequency masking is shown in Figure 4.3, where we can see a loud

signal (green bar) masking two other signals (red and blue bar) at nearby frequencies. The blue solid line is the *masking curve* which represents the audibility threshold for signals in the presence of the masking signal (in our case, the green bar). Thus, other signals that are below this curve will not be heard when the masker is present. The masked signals in Figure 4.3 fall below the masking curve, so they are not heard even though they are above the absolute threshold of hearing, denoted by the red dashed line. The masking curve can be exploited for coding purposes in the same way as with the absolute threshold of hearing. Signals that are below the masking curve are inaudible and, thus they do not need to be coded.

Masking phenomena can extend in time outside the period when the masker is present, *i.e.*, a masking effect can be created when the masker and maskee have occurred at different times. This type of masking is depicted in Figure 4.4. Backward masking or pre-masking takes place before the onset of the masker, while forward masking or post-masking takes place after the masker is removed. From a auditory system point of view, temporal masking can be attributed to the integration time that auditory system requires in order to build the perception of sound and to the fact that louder sounds require longer integration intervals than subdued ones.

Masking effects must be taken into account in the perceptual audio coding. The assumption in such coders is that masking effects derived from simple maskers can be extended to a complex signal. Masking thresholds are computed by identifying masking signals in the frequency domain, then by developing frequency and temporal masking curves based on the characteristics of each identified masker and finally, by combining the individual masking curves with each other and with the absolute threshold of hearing to create the global threshold representing audibility for the signal. The global threshold is used to identify imperceptible signal components and to choose the amount of bits needed for quantizing the audible signal components.

Another important notion in psychoacoustics is the critical bandwidth. It is associated with the frequency selectivity of the hearing system, which can be affected by a frequency to place conversion that occurs in the cochlea along the basilar membrane. Mechanical movements are transformed to travelling wave in the cochlea. For sinusoidal stimuli, the

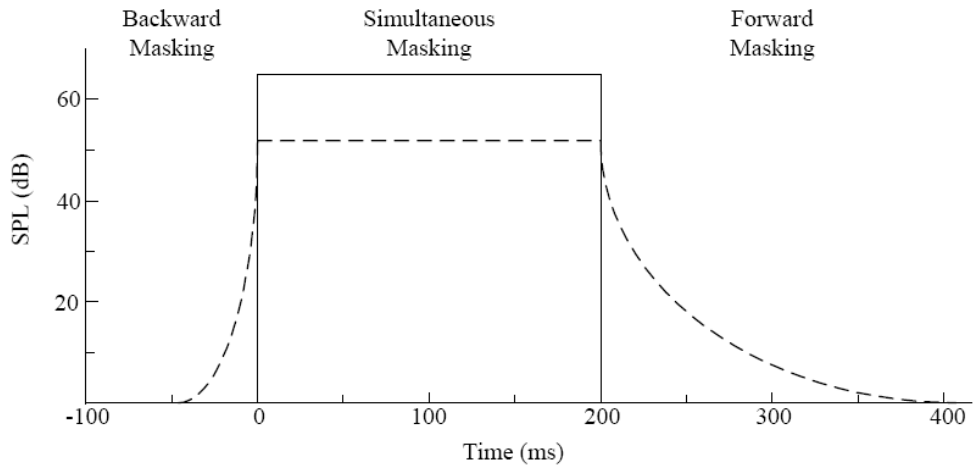

Figure 4.4: Example of temporal masking.

travelling wave on the basilar membrane propagates from the oval window until a region which has the same resonant frequency. At this point the magnitude increases to a peak and this location is called characteristic frequency. From a signal processing perspective, in this transformation the cochlea can be viewed as a bank of overlapped bandpass filters. The magnitude responses are nonuniform functions of frequency. Besides, the filter passbands have nonlinear bandwidths which increase by frequency. The critical bandwidth is a function of frequency that measures the width of the cochlear filter passbands.

There have been many interpretations of critical bands [56]. One interpretation is the perceived level of a noise. For a narrowband noise the perceived loudness remains constant for a constant SPL even when the noise bandwidth is increased up to the critical band width. If we increase the bandwidth beyond the critical bandwidth, the loudness begins to increase. Another notion is the audibility threshold of a narrowband noise masked with two tones. The detection threshold for the noise remains constant as long as the frequency separation between the tones is within one critical bandwidth. Beyond this bandwidth the threshold rapidly decreases.

A perceptual scale has been introduced based on the importance of critical band concept. The critical band rate is obtained by adding one critical band to the next in such a way that the upper limit of the lower band corresponds to the lower limit of the next higher critical band. Accordingly, there is a one-to-one mapping between frequency and the number of critical bands. The critical band rate is expressed in units of Bark, where an

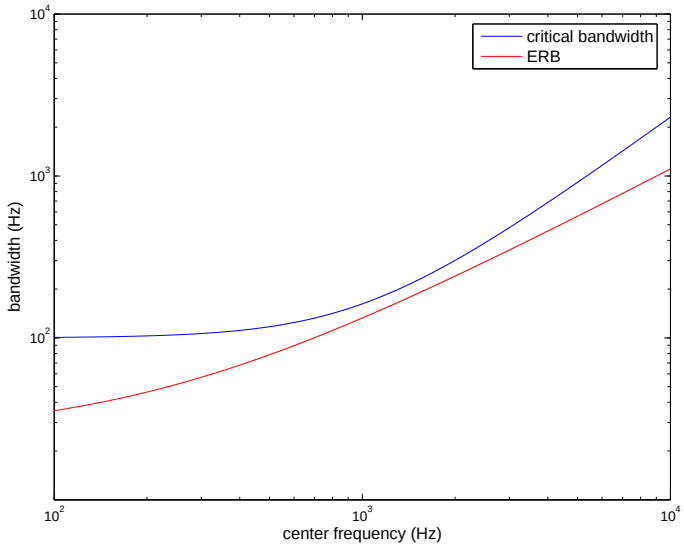

Figure 4.5: ERB versus critical bandwidth as a function of center frequency.

increment of one Bark corresponds to one critical band. In [56] is introduced an analytical expression that characterizes the dependence of critical band rate on frequency

$$z(f) = 13 \arctan(760f) + 3.5 \arctan(f/7500)^2, \tag{4.4}$$

where $z(f)$ is expressed in Bark units and the frequency f is expressed in Hz. For an average listener, critical bandwidth, as a function of its central frequency, is conveniently approximated by

$$BW(f) = 25 + 75(1 + 1.4(f/1000)^2)^{0.69}, \tag{4.5}$$

where $BW(f)$ is expressed in Hz.

Another measure for the perceptual frequency of the ear is proposed in [31], called Equivalent Rectangular Bandwidth (ERB). The ERB of a filter corresponds to the bandwidth of the rectangular filter which has the same peak transmission and passes the same power given a white noise input. The equation that relates the ERB to the center frequency of an auditory filter is given by

$$ERB(f) = 24.7(4.37(f/1000) + 1). \tag{4.6}$$

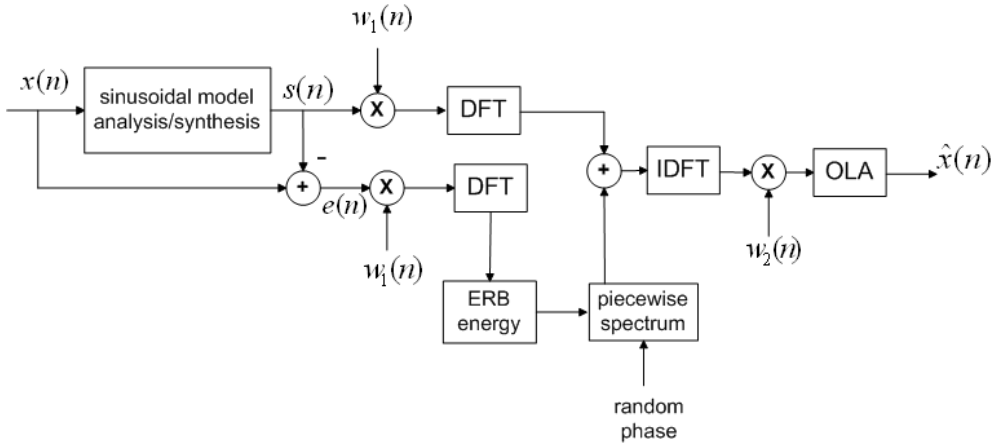

Figure 4.6: Signal reconstruction based on a sinusoids plus noise model, where the noise part is modeled using a filterbank based on the human auditory system.

As shown in Figure 4.5, the ERB function specified differs from the critical bandwidth expression. The ERB scale implies that auditory filter bandwidths decrease below 500 Hz, whereas the critical bandwidth remains essentially flat. The apparent increased frequency selectivity of the auditory system below 500 Hz has implications for optimal filter bank design, as well as for perceptual bit allocation strategies [35].

4.2 Noise modeling using a filterbank model of the auditory system

In this section, it is described a noise modeling method, in which the noise signal's spectrum is divided into critical bands and the spectral envelope is estimated by retaining the energy in each band. Then, the piecewise constant envelope is added to the sinusoidal part, in the frequency domain, in terms of rectangular coordinates, where the envelope's phase spectrum is chosen to be a uniformly distributed random phase. The approximated signal is finally computed by taking the Inverse Fourier Transform of the aforementioned spectral sum.

The method is shown in Figure 4.6. In the analysis procedure of the noise component $e(n)$, a sliding window, $w_1^i(n) = w_1(n - iH)$, of length N is used to obtain the frames,

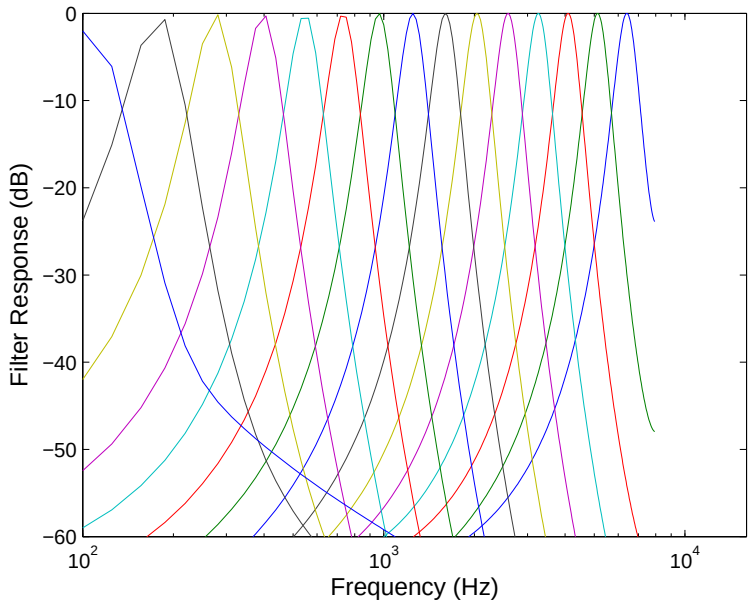

Figure 4.7: Filterbank according to ERB model.

at times spaced by the analysis hop size H. Discrete Fourier transform is applied in each frame to obtain its spectrum. Fast Fourier transform (FFT) algorithm, of size M, is applied for computing the discrete spectrum, where $M \geq N$. The values of M, N and H need not correspond to those used in the sinusoidal analysis process. Next, the spectrum is divided into bands according to the ERB model given in Equation 4.6. An example of an ERB filterbank with 15 bands is shown in Figure 4.7. In other words, the model of noise perception is based on the division of the spectrum into a set of equivalent rectangular bands (ERBs) [31]. In perceiving a broadband noise, the auditory system is primarily sensitive to the total short-time energy in each of these bands, and not to the specific distribution of energy within the bands.

The energy in each band is computed from the FFT magnitude

$$R_q^i = \frac{1}{M} \sum_{k \in b_q} |E_i(k)|^2, \tag{4.7}$$

where b_q denotes the bins that fall in the q^{th} ERB and $E_i(k)$ is the FFT of the i^{th} windowed frame of $e(n)$. It should be noted that the negative frequency components are included in the energy computation, because of the spectrum's conjugate symmetry. In this FFT-based analysis, the energies R_q^i serve as the parameters for the i^{th} frame of the noise

component $e(n)$. The sum of the band energies across the spectrum yields the signal energy of Parseval's theorem

$$\sum_{q=1}^{Q} R_q^i = \frac{1}{M}\sum_{q=1}^{Q}\sum_{k\in b_q}|E_i(k)|^2 = \frac{1}{M}\sum_{k=1}^{M-1}|E_i(k)|^2 = \sum_{n=0}^{N-1}(w_1(n-iH)e(n))^2. \tag{4.8}$$

From Equation 4.7 we can see that the FFT phase is not used in the ERB energy calculation, since the auditory system is primarily sensitive to the magnitude of the short-time spectrum.

The modeled noise component $e(n)$ is synthesized using inverse FFT. In specific, the ERB energies are converted into a piecewise constant spectrum wherein the magnitude of each constant piece is determined by the corresponding ERB analysis parameter. An example of this is given in Figure 4.8, which shows the magnitude (blue solid line) of an analysis frame extracted from the noise component of a classical music audio signal and the corresponding piecewise constant spectral estimate (red solid line) based on fifteen ERBs.

The ERB energies preservation in the analysis and the synthesis process of the noise component, is verified by the following relations

$$R_q^i = \frac{1}{M}\sum_{k\in b_q}|E_i(k)|^2 = \frac{1}{S}\sum_{k'\in s_q}|\hat{E}_i(k')|^2, \tag{4.9}$$

where $\hat{E}_i(k)$ is the piecewise constant spectral estimate derived at the synthesis stage, S is the size of the synthesis IFFT and s_q are the bins in the q^{th} synthesis band. At the synthesis stage the magnitude spectrum is constant in each band, thus for any bin $k' \in s_q$, Equation 4.9 can be rewritten as

$$R_q^i = \frac{c_q}{S}|\hat{E}_i(k')|^2 \Rightarrow |\hat{E}_i(k')| = \sqrt{\frac{S}{c_q}R_q^i}, \tag{4.10}$$

where c_q is the number of bins in the q^{th} ERB at the synthesis stage. As we can see in Figure 4.6, after the piecewise constant magnitude spectrum estimation, a uniform random phase is applied on a bin-by-bin basis. Next, the spectrum of the noise component $e(n)$ and the spectrum of the harmonic part $s(n)$ are summed (in rectangular coordinates) and

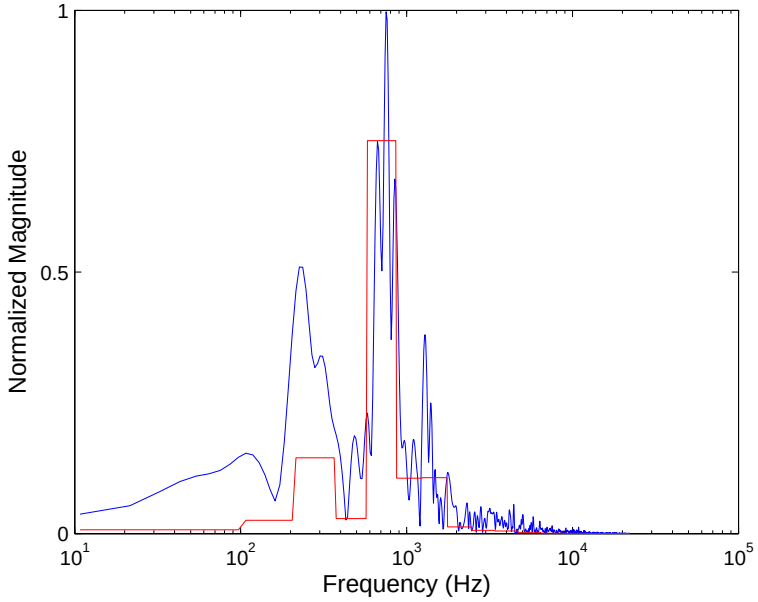

Figure 4.8: Piecewise constant ERB estimate (red solid line) of the noise component's magnitude spectrum (blue solid line) for a frame of a classical music audio signal.

transformed into a time-domain signal by the IFFT and OLA.

Finally, to avoid perceptual loss of the original noise component we have to take into consideration the various parameters that may affect the loudness of the noise component. These parameters are the multiple windowing steps, the possible analysis and synthesis frame sizes and sampling rates. Perceptual losslessness can be achieved through an appropriate normalization of the noise component. In [14] it is shown that at the analysis stage, the noise component's frame have to be multiplied by the scale factor

$$SC_1 = \frac{1}{\sqrt{\sum_{n=0}^{N-1} w_1(n)^2}}$$

before the ERB-based estimation of the piecewise constant magnitude spectrum. Similarly, in the synthesis, the reconstructed signal should be multiplied by the factor

$$SC_2 = \frac{1}{\sqrt{\frac{2}{L^2}\sum_{n=0}^{L-1} v(n)(v(n) + v_1(n) + v_2(n))}},$$

where L is the length (in samples) of the synthesis frame in the OLA method and $v(n)$ is

the window used in the OLA at the synthesis stage with

$$v_1(n) = \begin{cases} v(n + \frac{L}{2}) , & 0 \leq n < \frac{L}{2} \\ 0 , & \frac{L}{2} \leq n < L \end{cases} \tag{4.11}$$

$$v_2(n) = \begin{cases} 0 , & 0 \leq n < \frac{L}{2} \\ v(n - \frac{L}{2}) , & \frac{L}{2} \leq n < L. \end{cases} \tag{4.12}$$

Hence, $v_1(n)$ and $v_2(n)$ correspond to the second half of the synthesis window from the previous frame and the first half of the synthesis window from the next frame where a 50% overlapping is assumed in the derivation.

4.3 Noise modeling based on perceptual linear predictive analysis

The current section describes a noise modeling method based on the estimation of the envelope of the sinusoidal model's noise part using a perceptually motivated Linear Predictive (LP) analysis. This method, called Perceptual Linear Predictive Coding (PLPC), minimizes a perceptual modelling error and represents only the frequency components of the sinusoidal noise part that are of perceptual importance, while automatically discarding components masked by the sinusoidal part.

In specific, it has been shown (see [29]) that the LP coefficients can be found by minimization of the quantity $E = \int_0^1 S(f)/\hat{S}(f)df$, where $S(f)$ is the power spectral density of the original signal and $\hat{S}(f)$ is the power spectral density of the signal estimated with LP model. From the previous relation we can conclude that the approximation of $S(f)$ by $\hat{S}(f)$ is more accurate at points of the spectrum where more energy is concentrated, *i.e.*, at the spectral peaks [29]. This property of the LPC leads to modeling problems when the sinusoidal noise part does not only contain perceptually important noise components, but also perceptually unimportant sinusoidal components, which dominate the noise part's power

spectral density because they contain more energy than the perceptually more important noise-like components of the spectrum. Thus, modeling of the power spectral density $S(f)$ by $\hat{S}(f)$ will cause an inaccurate modeling of perceptual important noise-like components of the spectrum.

This problem can be dealt with the minimization of a perceptual distortion measure

$$\|\varepsilon(n)\|_\pi^2 = \int_0^1 R(f)|\mathcal{E}(f)|^2 df, \tag{4.13}$$

where $\varepsilon(n)$ is the LP modeling error, $R(f)$ is the Fourier transform of the reciprocal of the signal's global masking threshold and $\mathcal{E}(f)$ is the Fourier transform of the error $\varepsilon(n)$. The global masking threshold of the sinusoidal noise part is computed using the algorithm mentioned in [35] where it is assumed that the masking curves are mainly dominated by the tonal components of the original audio signal. With this assumption, $R(f)$ can be determined from the sinusoidal part. The relation between the LP modeling error $\varepsilon(n)$ and the sinusoidal noise part $e(n)$ is given by (see in section 3.4)

$$\varepsilon(n) = e(n) - \sum_{k=1}^{p} \alpha_k e(n-k). \tag{4.14}$$

Since the reciprocal threshold $R(f)$ is positive and real for all frequencies f, Equation 4.13 defines a (perceptual) norm which is referred to as $\|.\|_\pi$. Thus, the problem is to find LP coefficients α_k which minimize the perceptual norm 4.13

$$\begin{aligned}
\min_{\alpha_k} \|\varepsilon(n)\|_\pi^2 &= \min_{\alpha_k} \int_0^1 R(f)|\mathcal{E}(f)|^2 df \\
&= \min_{\alpha_k} \int_0^1 |H(f)\mathcal{E}(f)|^2 df \\
&= \min_{\alpha_k} \|h(n) * \varepsilon(n)\|_2^2 \\
\Rightarrow \min_{\alpha_k} \|\varepsilon(n)\|_\pi^2 &= \min_{\alpha_k} \|\varepsilon_\pi(n)\|_2^2,
\end{aligned} \tag{4.15}$$

where $H(f) = \sqrt{R(f)}$ and the convolution $h(n) * \varepsilon(n)$ defines a transformation of the LP modeling error $\varepsilon(n)$ to a modeling error $\varepsilon_\pi(n)$ in the perceptual domain. Thus, minimization of the LP modeling error's perceptual norm can be achieved if LP analysis is applied

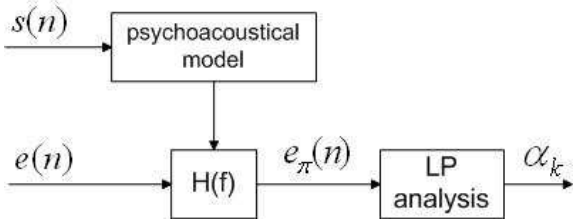

Figure 4.9: PLPC analysis process. [18]

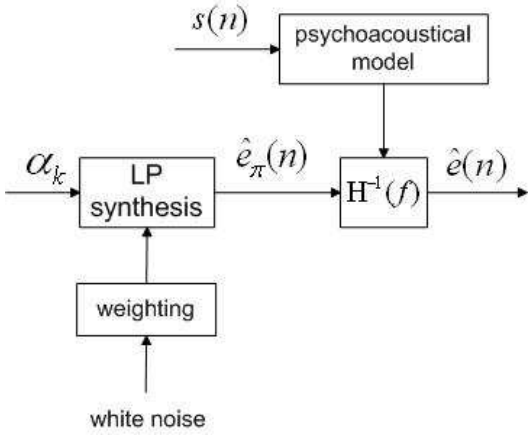

Figure 4.10: PLPC synthesis process. [18]

to the signal transformed to the perceptual domain.

In Figure 4.9 the PLPC analysis process is depicted. The noise part $e(n)$ is filtered by the filter $H(f) = \sqrt{R(f)}$, which is determined by the psychoacoustical model. The input to psychoacoustical model is the sinusoidal part $s(n)$ which is used to compute the masking threshold $R(f)$. The filtered noise part denoted by $e_\pi(n)$ is analyzed using Linear Prediction where the LP coefficients α_k are computed.

The PLPC synthesis process is shown in Figure 4.10. The signal $e_\pi(n)$ is reconstructed by filtering a colored noise signal from the LP synthesis filter which is formed using the coefficients α_k. Then, $\hat{e}_\pi(n)$ is filtered by $H^{-1}(f)$ and the output of the filter is the reconstructed noise part $\hat{e}(n)$. Colored noise is used in the LP filtering because the perceptual transformation $H(f)$ depends on the reciprocal of the absolute threshold of hearing, shown in Figure 4.11, which is not flat across the frequency range. This causes the perceptual modeling error $\varepsilon_\pi(n)$ to not be whitened over the whole frequency range. Thus, after the

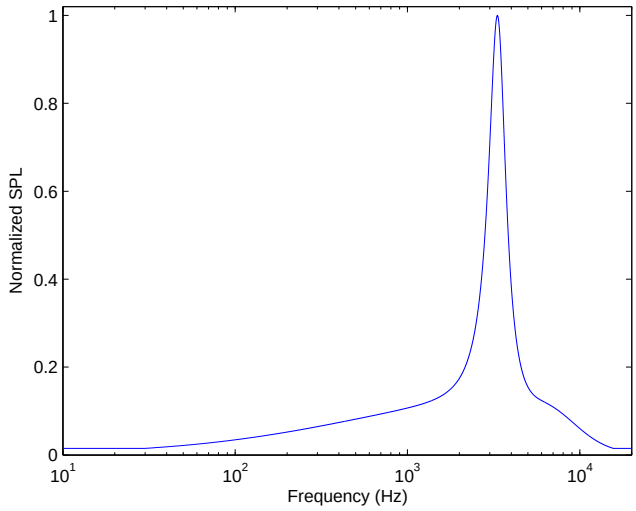

Figure 4.11: Reciprocal of the absolute threshold of hearing.

filtering of $e(n)$ with $H(f)$, the signal $e_\pi(n)$ will dominate its frequency component at low and high frequencies. This type of spectral shaping cannot be modeled well with a low-order LP model, so the PLPC modeling error become non-white in nature. Thus, in order to preserve the non-white property of the PLPC modeling error at the synthesis stage, the "weighting" block depicted in Figure 4.10 is used to shape the white excitation signal by the absolute threshold of hearing at low and high frequencies.

Chapter 5

Density estimation using Gaussian mixture models

5.1 Description of the Gaussian mixture model

Density estimation of random processes can been done using two different approaches. One is the parametric data estimation approach called maximum likelihood parameter estimation method [39]. The probability density function (PDF) $f_{\mathbf{x}}(\mathbf{x})$ of the random vector $\mathbf{x}$ is assumed to belong to a family of parametric densities $f_{\mathbf{x}}(\mathbf{x}|\,\boldsymbol{\theta})$, and the goal is to find the parameters $\boldsymbol{\theta}$ such that it best explains the random vector. Maximum likelihood is a computationally efficient approach, since the parameter can be estimated from a fairly small number of observations which linearly grows with the number of parameters [42]. However, the drawback of this method is the enforcement of an a priori structure on the observed data which may cause poor estimation of parameters.

The second approach of density estimation is the nonparametric method [39]. This estimation approach does not make any prior assumption about the unknown PDF. Therefore, the estimation is consistent which roughly means that if the number of samples is large enough, each estimated value is very close to true value irrespective of the unknown PDF. However, nonparametric techniques need a large number of observations.

Mixture models constitute a density estimation method which lies between the two aforementioned approaches by exploiting the advantages of the parametric and nonpara-

metric methods. Specifically, mixture models try to model the unknown PDF as a mixture of parametric PDF's

$$f_{\mathbf{x}}(\mathbf{x}|\,\boldsymbol{\theta}) = \sum_{i=1}^{M} p_i f_{\mathbf{x}}^{i}(\mathbf{x}|\,\boldsymbol{\theta}_i), \tag{5.1}$$

where $\Theta = [M, p_1, \ldots, p_M, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M]$ is the space of possible parameters. The parameters p_i are the weight constants which have to sum up to one. Parameterized by $\boldsymbol{\theta}_i$, $f_{\mathbf{x}}^{i}(\mathbf{x}|\,\boldsymbol{\theta}_i)$ is an individual parametric density. Each of the parametric PDF's $f_{\mathbf{x}}^{i}(\mathbf{x}|\,\boldsymbol{\theta}_i)$ are also called clusters. Equation 5.1 can considered as a functional decomposition of the unknown PDF in terms of parametric PDF's which make the basis functions for this decomposition.

Mixture models belong to the general category of unsupervised classifiers, *i.e.*, we assume that the training samples used to design a classifier are not labeled by their category membership, in contrast with the procedures that use labeled samples and called supervised. In other words, in the unsupervised case we are given a collection of samples without being told their category (class). In many signal processing applications, the Gaussian mixture model (GMM) is used for estimating the unknown PDF of a random vector $\mathbf{x}$. Each individual parametric density $f_{\mathbf{x}}^{i}(\mathbf{x}|\,\boldsymbol{\theta}_i)$ in Equation 5.1 is a D-variate Gaussian function of the form

$$f_{\mathbf{x}}^{i}(\mathbf{x}|\,\boldsymbol{\theta}_i) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_i|^{1/2}} \exp\Big\{ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\Big\}, \tag{5.2}$$

with mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. The complete Gaussian mixture density is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation

$$\boldsymbol{\theta}_i = \{p_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}\,,\ i = 1, \ldots, M. \tag{5.3}$$

GMM can have several different forms depending on the choice of covariance matrices. The model can have one covariance matrix per Gaussian component as indicated in Equation 5.3. The covariance matrix can also be full or diagonal.

There are several techniques available for estimating the parameters $\{\boldsymbol{\theta}_i\}$ of a GMM.

The goal is to estimate the parameters of the GMM, which in some sense best matches the distribution of the training samples. One method for doing this is the Maximum Likelihood (ML) estimation, which was mentioned in the first paragraph of the current section. The aim of ML estimation is to find the model parameters which maximize the likelihood of the GMM, given the training samples. Specifically, given a set of observation data in a matrix $\mathbf{X}$ and a set of observation parameters $\boldsymbol{\theta}$, the ML parameter estimation aims at maximizing the likelihood $\mathcal{L}(\boldsymbol{\theta}|\,\mathbf{X})$ of the observation data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}|\,\mathbf{X}). \tag{5.4}$$

Assuming that we have independent, identically distributed data, likelihood function can be rewritten as follows

$$\mathcal{L}(\boldsymbol{\theta}|\,\mathbf{X}) = p(\boldsymbol{X}|\,\boldsymbol{\theta}) = p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n|\,\boldsymbol{\theta}) = \prod_{i=1}^{n} p(\boldsymbol{x}_i|\,\boldsymbol{\theta}). \tag{5.5}$$

The maximum value of Equation 5.5, assuming an analytical solution, can be found by taking the derivative and set it equal to zero, $\frac{\partial}{\partial \boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}|\,\mathbf{X}) = 0$. This maximization problem is considered to be a nonlinear optimization problem, where direct maximization is not easy. However, there are other ways to find the parameter $\boldsymbol{\theta}$ which maximize the likelihood $\mathcal{L}(\boldsymbol{\theta}|\,\mathbf{X})$, but the most popular method is the Expectation Maximization (EM) algorithm [39]. The basic idea of the EM algorithm is (beginning with an initial parameter set $\boldsymbol{\theta}$) to estimate a new parameter set $\hat{\boldsymbol{\theta}}$, such that $p(\boldsymbol{X}|\,\hat{\boldsymbol{\theta}}) \geq p(\boldsymbol{X}|\,\boldsymbol{\theta})$. Then, the new set $\hat{\boldsymbol{\theta}}$ becomes the initial set and a new iteration begins. This iterative process continues until a given threshold is reached.

In more detail, assuming that the observed data is given in the matrix $\mathbf{X}$, we want to maximize the probability $p(\mathbf{X}|\,\hat{\boldsymbol{\theta}})$. We, also, assume that $\mathbf{Y}$ denotes the hidden data, *i.e.*, the data that cannot be observed directly and gives information about the state of the underlying model. In the EM algorithm it is assumed that we have an estimate of the parameter set $\boldsymbol{\theta}$ and want to estimate the probability that each hidden parameter $\mathbf{y}$ occurs. Instead of maximizing the likelihood function in a direct manner, an auxiliary function based on the expected value of the complete data $(\mathbf{X}, \mathbf{Y})$, *i.e.*, the combination

of the hidden and observed data, is built

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i) = E\{\log p(\mathbf{X}, \mathbf{y}|\, \boldsymbol{\theta})|\, \mathbf{X}, \boldsymbol{\theta}^i\} = \sum_{\mathbf{y} \in \mathbf{Y}} p(\boldsymbol{y}|\, \boldsymbol{X}, \boldsymbol{\theta}^i)\, \log p(\boldsymbol{X}, \boldsymbol{y}|\, \boldsymbol{\theta}), \tag{5.6}$$

where $\boldsymbol{\theta}^i$ denotes the estimation of the parameter set $\boldsymbol{\theta}$ at the i^{th} iteration. It should be noted that the maximization process of the auxiliary function $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$ always increases the likelihood $p(\mathbf{X}|\, \boldsymbol{\theta}^{i+1})$ of the observed data $\mathbf{X}$, and a maximum of the auxiliary function corresponds to a maximum of the likelihood.

Theorem 5.1.1 ***Entropy inequality.*** *Let f and g be probability densities with respect to a measure μ. Suppose $f > 0$ and $g > 0$ almost everywhere relative to μ. If E_f denotes expectation with respect to the probability measure $f\, d\mu$, then $E_f(\ln f) \geq E_f(\ln g)$, with equality only if $f = g$ almost everywhere relative to μ.*

From the definition of the auxiliary function $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i) = E\{\log p(\mathbf{X}, \mathbf{y}|\, \boldsymbol{\theta})|\, \mathbf{X}, \boldsymbol{\theta}^i\}$ and using the Theorem 5.1.1, the ascent property $\log p(\mathbf{X}|\, \boldsymbol{\theta}^{i+1}) > \log p(\mathbf{X}|\, \boldsymbol{\theta}^i)$ of the EM algorithm can be proved [26]. The four steps of the EM algorithm can be summarized as followed:

1. **Initialization**: choose an initial value for the model parameter set $\boldsymbol{\theta}$ (this initialization is usually done by a clustering procedure such as $k-$means algorithm [39])

2. **Expectation step** (E-step): compute the function

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i) = E\{\log p(\mathbf{X}, \mathbf{y}|\, \boldsymbol{\theta})|\, \mathbf{X}, \boldsymbol{\theta}^i\} = \sum_{\mathbf{y} \in \mathbf{Y}} p(\boldsymbol{y}|\, \boldsymbol{X}, \boldsymbol{\theta}^i)\, \log p(\boldsymbol{X}, \boldsymbol{y}|\, \boldsymbol{\theta})$$

3. **Maximization step** (M-step): let $\boldsymbol{\theta}^{i+1}$be that value of $\boldsymbol{\theta}$ that maximizes $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$,

$$\boldsymbol{\theta}^{i+1} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$$

4. **Iteration**: set $\boldsymbol{\theta} = \boldsymbol{\theta}^{i+1}$and repeat the steps 2 and 3 until the convergence is reached (*e.g.*, the algorithm converges when $\|\boldsymbol{\theta}^{i+1} - \boldsymbol{\theta}^i\| < \epsilon$ for some ϵ some appropriate distance measure $\|.\|$)

5.2 EM for Gaussian mixtures parameter estimation

The EM algorithm can be applied to the estimation of the parameter set of a Gaussian mixture model. The PDF of each observation data vector at a specific time t and the parameter set $\boldsymbol{\theta}$ are expressed as

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}_t) &= \sum_{k=1}^{M} p(\omega_k) g_k(\mathbf{x}_t) && (5.7) \\ \boldsymbol{\theta} &= \{p_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\} \ , \ k = 1, \ldots, M && (5.8) \\ \sum_{k=1}^{M} p_k &= 1, && (5.9) \end{aligned}$$

where

$$g_k(\mathbf{x}_t) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp\Big\{ -\frac{1}{2} (\mathbf{x}_t - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_k) \Big\}$$

is the $D-$dimensional Gaussian distribution. The parameter $p(\omega_k)$ is the k^{th} mixture weight (prior probability of the Gaussian component), $\boldsymbol{\mu}_k$ is the k^{th} mixture mean vector and $\boldsymbol{\Sigma}_k$ is the k^{th} mixture covariance matrix. The observation data matrix can be expressed in time as $\mathbf{X} = \mathbf{x}_1, \ldots, \mathbf{x}_N$ and the set $C = \{c_1, \ldots, c_N\}$ denotes the component class indices, where $c_t \in [1, M]$. The likelihood of $\mathbf{X}$ can be expressed as

$$p(\mathbf{X}|\, \boldsymbol{\theta}) = \sum_{C} p(\mathbf{X}, C|\, \boldsymbol{\theta}) = \sum_{C} \prod_{t=1}^{N} p(\omega_{c_t}) g_{c_t}(\mathbf{x}_t). \tag{5.10}$$

From Equations 5.6 and 5.10, we can write the function $\mathcal{Q}$ in the following form

$$\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \sum_{t=1}^{N} \sum_{j=1}^{M} \log \big(\hat{p}(\omega_{c_t}) \hat{g}_{c_t}(\mathbf{x}_t) \big) \alpha_t(j), \tag{5.11}$$

where

$$\alpha_t(j) = p(\mathbf{X}|\, \boldsymbol{\theta}) p(c_t = j|\, \mathbf{x}_t, \boldsymbol{\theta}) \ \text{ with } \ p(c_t = j|\, \mathbf{x}_t, \boldsymbol{\theta}) = \frac{p(\omega_j) g_j(\mathbf{x}_t)}{\sum_{l=1}^{M} p(\omega_l) g_l(\mathbf{x}_t)}. \tag{5.12}$$

Thus, the new parameter set $\hat{\boldsymbol{\theta}}$ can be computed by solving the equation

$$\frac{\partial\, \mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})}{\partial\, \hat{\boldsymbol{\theta}}} = 0. \tag{5.13}$$

In order to estimate the parameters of the Gaussian mixture model, we need to solve Equation 5.13 for the specific case of GMM. In specific, in each iteration i the posterior probabilities $p(\omega_m|\, \mathbf{x}_t, \boldsymbol{\theta})$ are estimated using the relation (E-step of the EM algorithm)

$$p^{(i)}(\omega_m|\, \mathbf{x}_t, \boldsymbol{\theta}) = \frac{p^{(i)}(\omega_m) g_m^{(i)}(\mathbf{x}_t)}{\sum_{k=1}^{M} p^{(i)}(\omega_k) g_k^{(i)}(\mathbf{x}_t)}, \tag{5.14}$$

where $p(\omega_m|\, \mathbf{x}_t, \boldsymbol{\theta})$ is the posterior probability that given the observation vector $\mathbf{x}_t$ and the parameter $\boldsymbol{\theta}$ the data is generated by the m^{th} cluster. After the computation of the posterior probabilities 5.14, the parameters of the Gaussian mixture are estimated by the relations

$$p^{(i+1)}(\omega_m) = \frac{1}{N} \sum_{t=1}^{N} p^{(i)}(\omega_m|\, \mathbf{x}_t, \boldsymbol{\theta}) \tag{5.15}$$

$$\boldsymbol{\mu}_m^{(i+1)} = \frac{\sum_{t=1}^{N} p^{(i)}(\omega_m|\, \mathbf{x}_t, \boldsymbol{\theta}) \mathbf{x}_t}{\sum_{t=1}^{N} p^{(i)}(\omega_m|\, \mathbf{x}_t, \boldsymbol{\theta})} \tag{5.16}$$

$$\boldsymbol{\Sigma}_m^{(i+1)} = \frac{\sum_{t=1}^{N} p^{(i)}(\omega_m|\, \mathbf{x}_t, \boldsymbol{\theta})(\mathbf{x}_t - \boldsymbol{\mu}_m^{(i+1)})(\mathbf{x}_t - \boldsymbol{\mu}_m^{(i+1)})^T}{\sum_{t=1}^{N} p^{(i)}(\omega_m|\, \mathbf{x}_t, \boldsymbol{\theta})} \tag{5.17}$$

For more detailed information about the derivation of the above equations, see [2].

A common problem that arises in the application of EM algorithm in Gaussian mixture models, is the implementation of the algorithm when the number of parameters in $\boldsymbol{\theta}$ increases. If more parameters are used, the obtained freedom may cause the problem of overfitting, since the random properties of the training data may have been reflected in the model. However, a large number of parameters will increase the computational complexity. Therefore, the number of parameters could be sufficiently decreased if diagonal covariance matrices are used instead of full matrices.

Chapter 6

Source Coding

6.1 Entropy

Let us assume that the source is modeled by a discrete-time random process $\{X_i\}_{-\infty}^{\infty}$. The alphabet over which the random variables X_i are defined can be either discrete or continuous depending on the nature of the information source. Let set $\mathcal{A} = \{a_1, a_2, \ldots, a_N\}$ denote the set in which the random variable X takes its values and let the probability mass function for the discrete random variable X be denoted by $p_X(x) = P(X = x)$ for all $x \in \mathcal{A}$. The information content of the information source is known as the *entropy* [8] of the source and is defined as follows

$$H(X) = -\sum_{x \in \mathcal{A}} p_X(x)\ \log_2(p_X(x)). \tag{6.1}$$

For instance, if we consider a binary source with probabilities p and $1 - p$, respectively, the entropy of the source is $H(X) = -p \log_2 p - (1 - p) \log_2(1 - p)$, which is shown in Figure 6.1.

The concept of the joint and conditional entropy can be introduced when dealing with two or more random variables. The *joint entropy* of two random variables (X, Y) with alphabet $\mathcal{A}$ and $\mathcal{B}$, respectively, is defined by

$$H(X, Y) = -\sum_{x \in \mathcal{A}, y \in \mathcal{B}} p_{X,Y}(x, y)\ \log_2(p_{X,Y}(x, y)). \tag{6.2}$$

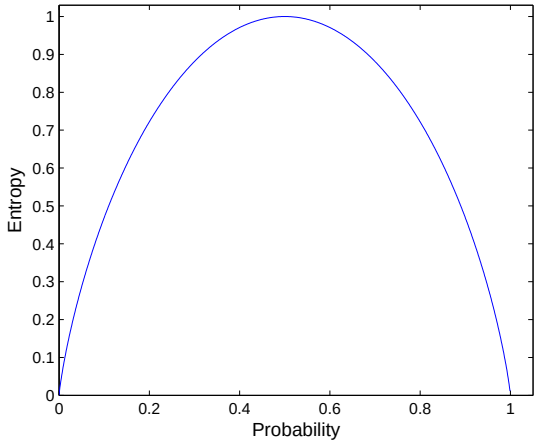

Figure 6.1: The binary entropy function.

The *conditional entropy* of the random variable X given the random variable Y is defined by

$$H(X|Y) = E\{log_2(p_{X|Y}(x,y))\} = -\sum_{x\in\mathcal{A},y\in\mathcal{B}} p_{X,Y}(x,y)\ \log_2(p_{X,Y}(x,y)). \tag{6.3}$$

Intuitively, the conditional entropy expresses the amount of uncertainty in X when one knows that $Y = y$. From the definition of the conditional probability we have that $p(x,y) = p(y)p(x|y)$. Using this relation we can show that

$$H(X,Y) = H(Y) + H(X|Y). \tag{6.4}$$

This relation indicates that the information content of the pair (X,Y) is equal to the information content of Y plus the information content of X after Y is known.

The *mutual information* $I(X;Y)$ between two random variables X and Y is a quantity that plays an important role in information theory and particularly to rate-distortion theory. The mutual information is defined as

$$I(X;Y) = H(X) - H(X|Y), \tag{6.5}$$

and expresses the amount of information provided by the random variable Y about random variable X. Using the definitions of entropy and mutual information, we can express the

mutual information in terms of the probability mass functions of X and Y,

$$I(X;Y) = \sum_{x\in\mathcal{A}}\sum_{y\in\mathcal{B}} p_{X,Y}(x,y)\log_2\Big(\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}\Big). \tag{6.6}$$

So far, the entropy and mutual information were defined for the case of discrete sources. However, entropy is not defined for a random variable with continuous alphabets. The exact description of a random variable that can take any value within a continuous alphabet is impossible with a finite number of bits. Hence, we define a new quantity for the continuous case, called *differential entropy*, that resembles entropy. For a random variable X with probability density function $f_X(x)$, the differential entropy is defined as follows

$$h(X) = -\int_{-\infty}^{\infty} f_X(x)\log_2(f_X(x))\,dx. \tag{6.7}$$

The definitions of joint differential entropy and conditional differential entropy are straightforward from Equations (6.4, 6.3):

$$h(X,Y) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(x,y)\log_2(f_{X,Y}(x,y))\,dx\,dy \tag{6.8}$$

$$h(X|Y) = h(X,Y) - h(Y). \tag{6.9}$$

The following equation

$$I(X;Y) = h(Y) - h(Y|X) = h(X) - h(X|Y) \tag{6.10}$$

defines the mutual information between two continuous random variables X and Y.

6.2 Rate-distortion theory

In general, source coding can be divided into two categories according with the distortion introduced to the coded source compared to the initial one. Lossless coding is the coding of a source without any distortion introduced, while lossy coding is referred to the distorted coding of a source. The rate-distortion theory is an information theoretic approach used for determining bounds on optimal rate-distortion performance when coding a particular

source.

Suppose that the probability density function of a source and a distortion measure are given. The rate-distortion function defines the lowest achievable average rate for a given average distortion of the source. Similarly, the distortion-rate function specifies the lowest achievable average distortion at a given average rate. The source coding theorem [36] gives a formal statement about the minimum rate required to code a source at a given distortion

Theorem 6.2.1 *A source with entropy H can be encoded with arbitrarily small error probability at any rate R (bits/source output) as long as $R > H$. Conversely, if $R < H$, the error probability will be bounded away from zero, independent of the complexity of the encoder and the decoder employed.*

Let us examine more thoroughly the concept of rate and distortion via the context of *quantization* [13], which is a very critical step in digital signal compression. Quantization can be defined as the mapping of continuous amplitude values into codes that can be represented with a finite number of bits. More formally, let $\mathbb{R}^n$ denote the n-dimensional Euclidean space. A quantizer $\mathcal{Q}$ of dimension n and size N maps a point (vector) $\mathbf{x} \in \mathbb{R}^n$ to a point $\mathbf{y}_i \in \mathbb{R}^n$, from a finite set of N reproduction points, $\mathcal{C} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$. The set $\mathcal{C}$ is called the *codebook* of size N, and the reproduction points $\mathbf{y}_i$ are called *codevectors*. The quantized point is denoted by $\hat{\mathbf{x}} = \mathcal{Q}(\mathbf{x}) = \mathbf{y}_i$. The quantizer is associated with a partition of the Euclidean space into non-overlapping quantization *cells* $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$, *i.e.*, $\bigcup_{i=1}^{N} S_i = \mathbb{R}^n$ and $S_i \cap S_j = \emptyset$ of $i \neq j$.

The quantization process can be decomposed into two steps; *encoding* and *decoding*. The encoder $\mathcal{E}$ maps a point $\mathbf{x} \in \mathbb{R}^n$ to an integer quantization index i from the index set $\mathcal{I} = \{1, 2, \ldots, N\}$ such that $\mathcal{E}(\mathbf{x}) = i$ if and only if $\mathbf{x} \in S_i$. On the other hand, the decoder $\mathcal{D}$ selects the corresponding reproduction point $\mathbf{y}_i$ from the codebook $\mathcal{C}$, *i.e.*, $\mathcal{D}(i) = \mathbf{y}_i$. Hence, the overall encoding-decoding process can be formalized by the relation

$$\hat{\mathbf{x}} = \mathcal{Q}(\mathbf{x}) = \mathcal{D}(\mathcal{E}(\mathbf{x})) = \mathbf{y}_i. \tag{6.11}$$

If the dimension $k = 1$ we have the case of scalar quantization, while if $k \geq 2$ the quantization process is called vector quantization.

Distortion is used as a way of measuring the closeness of the reproducted (output) n-dimensional vector $\hat{\mathbf{x}}$ to the original (input) vector $\mathbf{x}$, and is denoted by $d(\mathbf{x}, \hat{\mathbf{x}})$. It is assumed that we are dealing with a per-letter distortion measure, hence, the distortion between $\mathbf{x}$ and $\hat{\mathbf{x}}$ is the average of the distortion between their components

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{M} \sum_{m=1}^{M} d(x_m, \hat{x}_m), \tag{6.12}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_M)^T$ and $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_M)^T$. $d(\mathbf{X}, \hat{\mathbf{X}})$ is a random variable, since the source output $\mathbf{X}$ forms a random vector, with the lower-case letter $\mathbf{x}$ denoting a particular realization of the random vector. Thus, the distortion for the source is defined as the expected value of this random variable,

$$D = E\{d(\mathbf{X}, \hat{\mathbf{X}})\} = \int_{\mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x}) d(\mathbf{x}, \hat{\mathbf{x}})\, d\mathbf{x}, \tag{6.13}$$

where $f_{\mathbf{X}}(\mathbf{x})$ denoted the probability density function of the input vector $\mathbf{x}$.

A widely used distortion measure is the *squared error*,

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{M} \|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \frac{1}{M} (\mathbf{x} - \hat{\mathbf{x}})^T (\mathbf{x} - \hat{\mathbf{x}}) = \frac{1}{M} \sum_{m=1}^{M} (x_m - \hat{x}_m)^2. \tag{6.14}$$

Another interesting distortion measure is the *weighted squared error*

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{M} (\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{W} (\mathbf{x} - \hat{\mathbf{x}}), \tag{6.15}$$

where $\mathbf{W}$ is a symmetric positive-definite weighting matrix. If $\mathbf{W}$ is a diagonal matrix with diagonal elements $w_{m,m} > 0$, we have that

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{M} \sum_{m=1}^{M} w_{m,m} (x_m - \hat{x}_m)^2, \tag{6.16}$$

where the weights $w_{m,m}$ give a different emphasis to the squared errors of the individual vector elements. The distortion measure 6.15 is used in coding applications such as perceptual coding of audio signals, where the matrix $\mathbf{W}$ is chosen to account for perceptual

effects of quantization process. The weighting matrix $\mathbf{W}$ can also depend on the input vector $\mathbf{x}$. Thus, the distortion in Equation 6.15 can be written as

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{M}(\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{W}(\mathbf{x})(\mathbf{x} - \hat{\mathbf{x}}). \tag{6.17}$$

The second concept related with quantization is rate. The rate is the cost in terms of the number of bits needed to describe the quantization indices to the decoder during the quantization process. If all the quantization indices are represented with an equal number of bits then we define the quantizer as *fixed rate*. In this case, the rate is given by

$$R_f = \frac{1}{M}\log_2(N). \tag{6.18}$$

However, if the quantization individual indices are represented with an unequal number of bits then we define the quantizer as *variable rate*, and the rate is given by

$$R_v = \frac{1}{M}\sum_{i=0}^{M} p_I(i) r(i), \tag{6.19}$$

where $p_I(i) = \int_{S_i} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}$ are the probabilities of the individual indices and $r(i)$ are the rates used for the individual indices. The lowest possible average rate for the variable rate case is given by the entropy of the quantization indices $H(I)$ [16],

$$R_{v,min} = \frac{1}{M}H(I) = -\frac{1}{M}p_I(i)\log_2(p_I(i)). \tag{6.20}$$

If the rate is expressed as a function of distortion, $R(D)$, called *rate-distortion function*, then the rate R is a decreasing function of distortion D [8]. Thus, if we need high-fidelity reproduction (low D) we require a high R. The following theorem gives the general form of the rate-distortion function.

Theorem 6.2.2 *The minimum number of bits per source output required to reproduce a source (without memory) with distortion less than or equal to D is called rate-distortion*

function, denoted by $R(D)$ and given by

$$R(D) = \min_{p(\hat{\mathbf{x}}|\mathbf{x}: E\{d(\mathbf{X},\hat{\mathbf{X}})\} \leq D)} I(\mathbf{X}; \hat{\mathbf{X}})$$

In practical applications, quantizers are designed to minimize a distortion under a given rate constraint. A quantizer is referred to be *resolution constrained* if the rate constraint is formulated as a given fixed rate and a fixed number of quantization cells, while if the rate constraint is formulated as a given average constraint a quantizer is referred to as an *entropy constrained* quantizer resulting in variable rate. Entropy constrained quantizers achieve lower average rates compared to resolution constrained quantizers.

6.3 High-rate theory

High-rate theory [16] can be considered as a general quantization theory, which enables the coding of symbol sequences that need not be long, it allows to find a relation between rate and distortion for fixed-rate coders and it also leads to the design of practical quantizers. In high-rate theory we assume a large number of quantization cells or, equivalently, small quantization errors. In other words, the high-rate assumption means that the probability density function of the input can be accurately approximated as being constant within quantization cells, and that the distortion resulting from unbounded cells is negligible. In detail, if the quantization rate is high, then the quantization cell S_i is small enough to assume that the probability density function $f_x(x)$ of the data to be quantized is constant in S_i: $f_x(x) \approx p_x(\hat{x}_i)$ when $x \in S_i$, where $\hat{x}_i$ is the i^{th} codevector.

High-rate quantizers specify the asymptotically optimal *quantization point density function* $g_{\mathbf{X}}(\mathbf{x})$, which describes the density of quantization points in the Euclidean space $\mathbb{R}^n$, without specifying the quantization points with an exact manner. The optimal quantization point density can be found analytically, which is very important in the case where quantizers have to adapt fast to changing bitrate requirements. In addition, quantizers designed using high-rate theory have shown good performance even at low rates, despite the assumption of high rate [40].

In detail, in the asymptotic case of a very large number of quantization cells, the

continuous quantization point density at point $\mathbf{x}$ is approximately the inverse of the volume of the quantization cell containing $\mathbf{x}$, *i.e.*,

$$g_{\mathbf{X}}(\mathbf{x}) \approx \frac{1}{volume(S_i)}\,, \text{ if } \mathbf{x} \in S_i. \tag{6.21}$$

In the case of a scalar quantizer (of dimension 1) the quantization point density function is defined as the inverse of the quantizer step size Δ_i, where i denotes the i^{th} cell.

Consider the case of the input weighted squared error measure in Equation 6.17. Then, the average distortion can be expressed as a function of the continuous quantization point density as follows

$$g_{\mathbf{X}}(\mathbf{x}) \approx \frac{1}{volume(S_i)}\,, \text{ if } \mathbf{x} \in S_i. \tag{6.22}$$

The optimal quantization point density that minimizes a distortion measure under a given bitrate constraint can be computed using either standard integration inequalities [15], for finding the quantization point density that meets the minimum distortion bound, or calculus of variations [24] (such as Lagrange multipliers method and Euler-Lagrange equation) for solving the constrained minimization problem.

For the resolution constrained quantizer, the optimal quantization point density and distortion measure are [11]

$$g_{\mathbf{X}}(\mathbf{x}) = 2^{nR_f} \frac{(f_{\mathbf{X}}(\mathbf{x})[\det(\mathbf{W}(\mathbf{x}))]^{\frac{1}{n}})^{\frac{n}{n+2}}}{\int_{\mathbb{R}^n}(f_{\mathbf{X}}(\mathbf{x})[\det(\mathbf{W}(\mathbf{x}))]^{\frac{1}{n}})^{\frac{n}{n+2}}\,d\mathbf{x}} \tag{6.23}$$

$$D = C(n,2)2^{-2R_f}\Big(\int_{\mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x})[\det(\mathbf{W}(\mathbf{x}))]^{\frac{1}{n}})^{\frac{n}{n+2}}\,d\mathbf{x}\Big)^{\frac{n+2}{n}}, \tag{6.24}$$

where $C(n,2)$ is the coefficient of quantization at dimension n [12]. The entropy constrained quantizer's optimal quantization point density and distortion measure are [28]

$$g_{\mathbf{X}}(\mathbf{x}) = 2^{nR_v - h(\mathbf{X})} \frac{(\det(\mathbf{W}(\mathbf{x})))^{\frac{1}{2}}}{\int_{\mathbb{R}^n}(\det(\mathbf{W}(\mathbf{x})))^{\frac{1}{2}}\,d\mathbf{x}} \tag{6.25}$$

$$D = C(n,2)2^{-2R_v + \frac{2}{n}h(\mathbf{X})}2^{\frac{1}{n}\int_{\mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x})\log_2(\det(\mathbf{W}(\mathbf{x})))\,d\mathbf{x}}, \tag{6.26}$$

where $h(\mathbf{X}) = -\int_{\mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x})\log_2(f_{\mathbf{X}}(\mathbf{x}))\,d\mathbf{x}$ is the differential entropy of the random vector

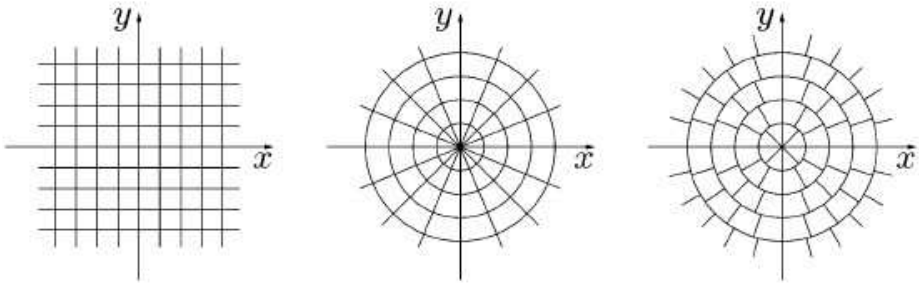

Figure 6.2: Partitions of the input space for (a) rectangular quantization, (b) strictly polar quantization, and (c) unrestricted polar quantization. [52]

X.

6.3.1 Quantization of a bivariate variable

In many coding applications, the random variable which is to be quantized is bivariate. The variable can be quantized either using a two-dimensional (vector) quantizer (VQ) or using a combination of two scalar (one dimensional) quantizers in order to achieve lower complexity. The scalar quantizers can be applied to a rectangular representation or a polar representation of the bivariate variable. The rectangular representation in terms of the real and imaginary components X and Y, and the polar representation in terms of the amplitude and phase components A and Φ, are related as $X = \text{real}\{Ae^{j\Phi}\}$, $Y = \text{imag}\{Ae^{j\Phi}\}$, and $A = (X^2 + Y^2)^{1/2}$, $\Phi = \arctan(Y, X)$.

In the polar representation, the amplitude A and phase Φ can be quantized in an independent way, called *strictly polar quantization* (SPQ), or, alternatively, the phase quantization process can be made dependent on the amplitude, called *unrestricted polar quantization* (UPQ) [54]. In Figure 6.2 are shown examples of partitions of the input space in the case of rectangular quantization, SPQ and UPQ. In [54] a numerical method is used to show that for a bivariate Gaussian input variable and the mean-squared error distortion measure, UPQ is significantly superior to SPQ and close to rectangular quantization in performance, if the number of quantization cells is fixed and entropy coding is applied to quantization indices.

Chapter 7

Modeling of spot microphone signals

In this chapter we propose a method for modeling microphone signals for immersive audio applications based on the sinusoids plus noise model. The method takes advantage of the interchannel similarities in order to achieve the final objective of a multichannel recording's low bitrate transmission (see the next chapter for the coding approach). In particular, each microphone signal is modeled using the sinusoidal parameters (harmonic part) and the short-time spectral envelope of the noise (modeling noise part). For resynthesis of each microphone signal, the harmonic part is added to the noise part which is recreated by using the corresponding noise envelope with the noise residual obtained from just one of the signals (the so-called "reference signal").

In the next sections we give an overview of the multiple microphone recordings' acquisition for multichannel rendering and we then describe the proposed modeling method. Finally, we give modeling results obtained with subjective listening tests.

7.1 Microphone signals of a multichannel recording

In this study, we mainly focus on multichannel recordings obtained from live concert hall performances. A number of microphones is used to capture several characteristics of the recording venue, resulting in an equal number of microphone signals, called stem recordings. Our main goal is to design a system that is able to recreate at the receiving end all of these target microphone signals from a smaller set (or even only one, which could be a downmix sum signal) of reference microphone signals. The result would be a significant

reduction in transmission requirements, while enabling interactivity at the receiver. For achieving high quality resynthesis, we propose the use of some additional information for each microphone with the constraint that this additional information requires minimal datarates for transmission. By examining the acoustical characteristics of the various stem recordings, the distinction of microphones is made into reverberant and spot microphone signals.

Spot microphones are microphones that are placed close to the sound source. The recordings of these microphones heavily depend on the instruments that are near the microphone and not so much on the hall acoustics; these recordings recreate the sense that the sound source is not a point source but rather distributed such as in an orchestra. Hence, resynthesizing the signals captured by these microphones involves enhancing certain instruments and diminishing others, which in most cases overlap in the time and frequency domains.

Reverberant microphones are the microphones placed far from the sound source, that mainly capture the reverberation information of the venue. Here, we focus on the recordings made by spot microphone signals since modeling their spectral properties is more challenging compared to reverberant microphone signals. Modeling of the latter signals has been considered in an earlier work [33], where linear time-invariant filters were proposed for transforming a reference signal into a given reverberant signal.

7.2 Noise transplantation based on sinusoids plus noise model

Before proceeding to the analysis of the proposed noise transplantation method, let us sum up the main points of the sinusoids plus noise model (SNM) which comprises the core of the proposed approach. In section 2.2 we saw that an accurate representation of audio signals is achieved with the SNM model. As we have mentioned, the sinusoidal model captures the harmonics of the original audio signal well if the number of harmonics used is carefully chosen. However, especially for music signals, the harmonic part of the signal is not sufficient for high-quality synthesis; its structured nature and the lack of

"randomness" in the signal is audible even if a high number of sinusoidal functions is used. The noise part $e(n)$, which contains the spectral information which is considered of random nature, is necessary for high-quality audio synthesis. It mostly contains higher-frequency information, and adds the acoustically needed "randomness" to the sinusoidal part. The expression used for modeling audio signals with the SNM model is the following

$$s(n) = \sum_{l=1}^{L} \alpha_l(n)\, \cos(\theta_l(n)) + e(n). \tag{7.1}$$

Practically, after the sinusoidal parameters are estimated, the noise part $e(n)$ is computed by subtracting the sinusoidal part from the original signal.

In our work, the noise part $e(n)$ is modeled as the result of filtering a residual noise component, *i.e.*, the AR modeling error of the sinusoidal noise part, with an autoregressive (AR) filter that models the noise spectral envelope. Linear Predictive (LP) analysis is applied to estimate the spectral envelope of the sinusoidal noise part. In other words, we assume the following relation for the signal $e(n)$

$$e(n) = \sum_{i=1}^{p} b(i)\, e(n-i) + r_e(n), \tag{7.2}$$

where $r_e(n)$ is the residual of the noise and p is the AR filter order. The $p+1^{th}$-dimensional vector $\mathbf{b} = (1, -b_1, -b_2, ..., -b_p)^T$ represents the spectral envelope of the noise part $e(n)$. In accordance with section 3.4, Equation 7.2 can be written as follows in the frequency domain

$$S_e(e^{j\omega}) = \left|\frac{1}{P_b(e^{j\omega})}\right|^2 S_{r_e}(e^{j\omega}), \tag{7.3}$$

where $S_e(e^{j\omega})$ and $S_{r_e}(e^{j\omega})$ is the power spectral density of $e(n)$ and $r_e(n)$, respectively, while

$$P_b(e^{j\omega}) = 1 - \sum_{i=1}^{p} b(i)\, e^{-j\omega i} \tag{7.4}$$

is the frequency response of the LP filter $\mathbf{b}$. For the sake of clarity, we have to denote that since in this chapter there are two noise quantities introduced, *i.e.*, the sinusoidal model noise $e(n)$ and its whitened version $r_e(n)$, we will refer to $e(n)$ as the (sinusoidal) *noise* signal and to $r_e(n)$ as the *residual* (noise) of $e(n)$.

Generally, in coding applications, the noise signal $e(n)$ will require a much higher degree in terms of datarates compared to the sinusoidal part, exactly due to its quasi-random nature. Thus, we are interested here to propose a method that is based on the sinusoidal part of the audio signal, but can result in high-quality audio synthesis at the decoder. In order to achieve this objective, we propose a scheme that is similar to the Spatial Audio Coding philosophy. In other words, we propose that given a collection of spot microphone signals that correspond to the same multichannel recording and thus have similar content, we encode as a full audio channel only one of the spot signals, called reference signal. The remaining spot microphone signals are modeled with the SNM model, retaining their sinusoidal parts and the noise spectral envelope (filter **b** in Equation 7.2).

In the resynthesis process, we model the reference spot signal with the SNM in order to obtain its noise signal $e(n)$, and from it we obtain the LP residual $r_e(n)$ using LP analysis. Finally, we reconstruct each spot microphone signal using its sinusoidal part and its noise LP filter; its sinusoidal part is added to the noise part that we obtain by filtering with the signals LP noise shaping filter the LP residual of the sinusoidal noise from the reference signal.

Let us analyze the aforementioned idea for the trivial case of a 2-channel (stereo) recording. We start by considering two spot microphone signals of a music performance, in which the two microphones are placed close to two distinct groups of instruments of the orchestra. The first microphone signal is denoted by $s_L(n)$. For simplicity we refer to the signal $s_L(n)$ as the left channel, which should not be confused with the channels of the multichannel mix. The second microphone signal is denoted by $s_R(n)$, called right channel.

Each of these microphone signals mainly captures the sound from the closest group of instruments, but also captures the sound from all the other instruments of the orchestra (this is especially true for live concert hall performances). Thus, the two recordings are similar in content, and this is apparent in most multichannel recordings in such settings. Alternatively, one of the channels (the reference signal) could be a sum signal of all the spot recordings. Here, we focus on a particular frame of the 2 signals (left and right channel), which corresponds to exactly the same music part (*i.e.*, some time-alignment

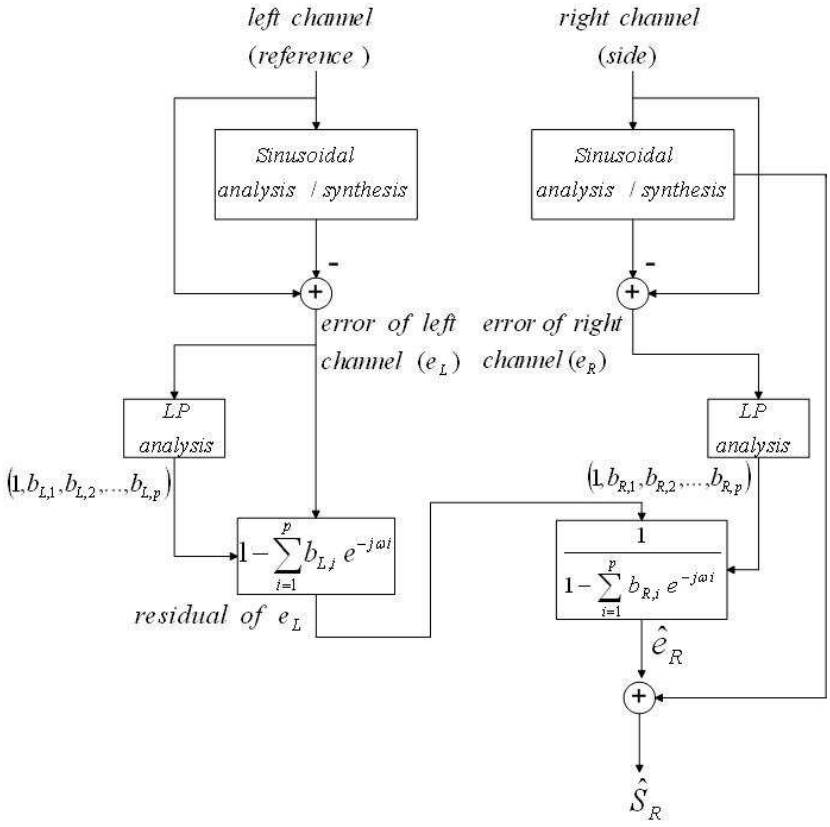

Figure 7.1: Noise transplantation. The LP residual of the reference signal's noise part is filtered by the side signal's noise envelope and added to its sinusoidal part.

procedure might be necessary to align the two microphone signals). The two audio frames are modeled with the SNM model as follows

$$s_L(n) = \sum_{l=1}^{L} \alpha_{L,l}(n)\, \cos(\theta_{L,l}(n)) + e_L(n), \tag{7.5}$$

$$s_R(n) = \sum_{l=1}^{L} \alpha_{R,l}(n)\, \cos(\theta_{R,l}(n)) + e_R(n), \tag{7.6}$$

where $s_L(n)$, $s_R(n)$ represents the left and right channel respectively. The goal is to resynthesize the right channel by using the estimated sinusoidal parameters $\{\alpha_{R,l}\,,\, \theta_{R,l}\}_{l=1}^{L}$ to create the sinusoidal part, while the noise part $e_R(n)$ is estimated using the AR modeling error of the $e_L(n)$.

The sinusoidal modeling error signals $e_L(n)$ and $e_R(n)$ will contain similar frequency content, because of the facts that: (1) the main spectral information that characterizes each signal is captured by the sinusoidal model and (2) $s_L(n)$ and $s_R(n)$ contain similar music

content because they constitute a pair of spot microphone signals. Thus, the assumption is that, as the harmonics capture most of the important information for each spot microphone signal and the LP coefficients capture most of the channel-specific noise characteristics, the residual noise part that remains will be similar for all the spot microphone signals. This assumption is in fact verified in Section 7.3. In the stereophonic recording, by taking the reference residual (whitened sinusoidal noise) and filtering it with the noise envelope of the side channel, we can obtain a noise signal with very similar spectral properties to the initial noise component of the side channel (right channel). This procedure is depicted in the diagram of Figure 4.1.

In specific, we want to resynthesize the right channel using the AR modeling error of the left channel to estimate the noise part $e_R(n)$. Firstly, the left channel is segmented into overlapping frames with short time duration. The sinusoidal parameters are estimated in each analysis frame and they are used in Overlap-Add (OLA) method to form the sinusoidal part, as we can see in Figure 7.1. The sinusoidal part is then subtracted from $s_L(n)$ to obtain the approximation error $e_L(n)$ of the sinusoidal modeling.

Next, each frame of the segmented $e_L(n)$ is filtered through the all-zero AR filter to obtain its AR modeling error ("residual of $e_L(n)$"). The AR modeling error of the left channel is then filtered through the all-pole AR filter (represented by the LP coefficients of the $e_R(n)$) to form the estimation $\hat{e}_R(n)$ of the right channel's noise part. Intuitively, the spectrum of the modeling error of the right channel is modulated by the spectral envelope of the left channel's residual $e_L(n)$. Finally, each frame of the estimated $e_R(n)$ is summed with the corresponding frame (in time) of the sinusoidal part to create the final estimated frame $\hat{s}_R(n)$ of the right channel, where OLA is used again to synthesize the final signal.

The noise transplantation method can be formalized as follows: consider a multichannel recording with M spot microphone signals. We introduce the general relation for the resynthesis of one of the *side* spot microphone signals s_k (as opposed to the *reference* spot signal $s_{(ref)}$),

$$\hat{s}_k(n) = \sum_{l=1}^{L} \alpha_{k,l}(n)\, cos(\theta_{k,l}(n)) + \hat{e}_k(n)\ , k = 1, ..., M, \tag{7.7}$$

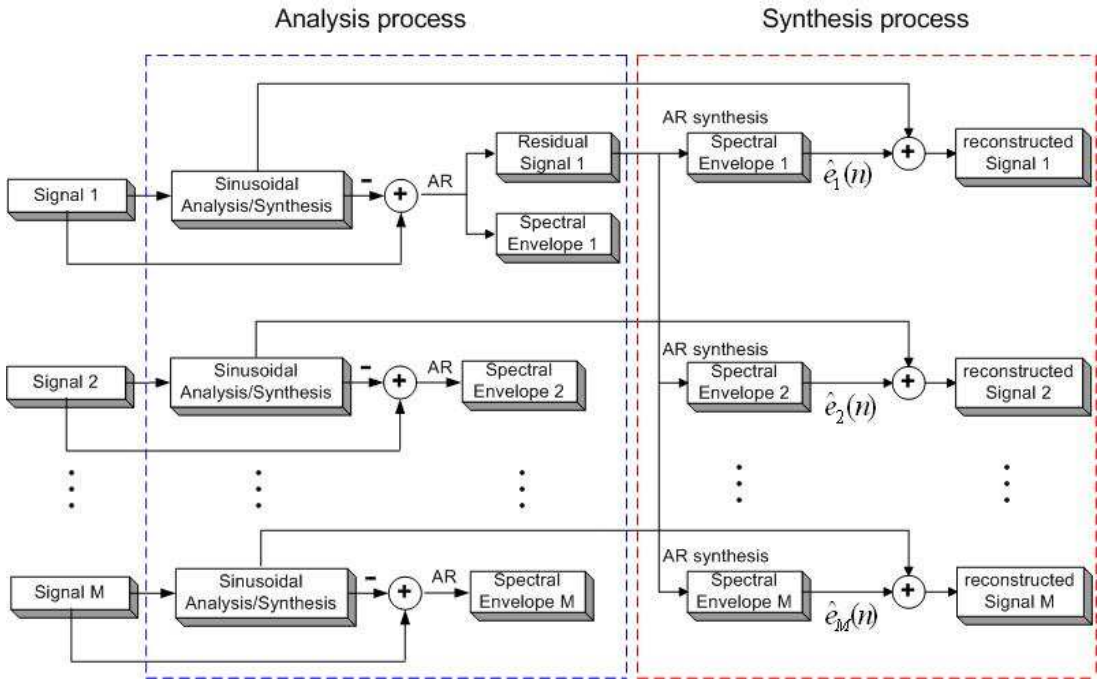

Figure 7.2: Noise transplantation approach for the general case of M spot microphone signals.

where $\hat{e}_k(n)$ is represented in the frequency domain as follows

$$S_{\hat{e}_k}(e^{j\omega}) = \left| \frac{1}{1 - \sum_{i=1}^{p} b_k(i)\, e^{-j\omega i}} \right|^2 S_{r_{e_{(ref)}}}(e^{j\omega}). \tag{7.8}$$

In Equations (7.7, 7.8), $\alpha_{k,l}(n)$ and $\theta_{k,l}(n)$ are the estimated sinusoidal parameters of the spot microphone signal k, $\{b_k\}$ is the signal's LP noise shaping filter, while $\hat{e}_k(n)$ is the estimated noise part of the k^{th} spot microphone using the noise transplantation procedure described. The residual of the noise part of the reference spot signal can be found as

$$S_{r_{e_{(ref)}}}(e^{j\omega}) = \left| 1 - \sum_{i=1}^{p} b_{(ref)}(i)\, e^{-j\omega i} \right|^2 S_{e_{(ref)}}(e^{j\omega}). \tag{7.9}$$

Thus, $S_{r_{e_{(ref)}}}(e^{j\omega})$ is the power spectral density of the reference spot signal's noise residual, and $e_{(ref)}$ is the sinusoidal noise part obtained from the reference spot microphone. The general case of resynthesis M spot microphone signals is shown in Figure 7.2, in which the blue bounding box corresponds to the analysis procedure, while the red bounding box is related with reconstruction of the M spot signals using the proposed noise transplantation method.

From the previous discussion, we note that the proposed method presents scalability with regard to the quality. It can be controlled through the number of sinusoids, the order of the LP filter and the percentage of frame overlapping. By decreasing (increasing) these parameters we can achieve minimum (maximum) datarate requirements. It is also important to clarify that for the reference signal, the SNM model is applied only for obtaining the noise residual. This signal is assumed to be encoded and transmitted as a monophonic audio signal, *e.g.*, MP3, to the receiver. Also, it is possible that more than one reference signals might be necessary for the method to perform well in practice, depending on the nature of the multiple microphone signals of a particular multichannel recording or when backwards compatibility with stereo decoders is required.

7.3 Performance evaluation of modeling

In this section, we are interested in illustrating that the use of the proposed method results in a modeled signal that is subjectively very close to the original recording. The objective is to estimate the noise part of a side microphone signal from the reference signal along with low-dimensional sinusoidal model parameters. In order to show the high-quality modeling results, we conduct two type of subjective (listening) tests; Degradation Category Rating (DCR) test and ABX test. The results are mentioned in this section given the fact that the importance of the noise part in sinusoidal modeling of music can mostly be quantified subjectively. In the DCR test, the quality of the resynthesized signal is evaluated using a 5-grade scale [25]. On the other hand, in the ABX test, each listener is presented with the audio files A and B in random order, and is asked to associate the audio file X with A or B depending on which audio signal is acoustically closer to the X.

For the performance evaluation, we used spot microphone signals obtained from a concert hall performance in US[1]. The main objective is to prove, using the aforementioned subjective listening tests, that the proposed modeling approach result in high-quality recordings. For the results of this section, we use two of these spot microphone signals of the multichannel recording, where one of the microphones captures mainly the female voices of

[1]Provided by Prof. Kyriakakis of the University of Southern California

the orchestra's chorus and is used here as the side signal (see Figure 7.1), while the other one mainly captures the male voices and is used here as the reference signal. Apart from that signals, some additional sound signals are used, in Section 7.3.1, where the crosstalk issue is examined thoroughly.

The DCR-based listening tests that conducted for evaluating the quality of the resynthesized signals, use a 5-grade scale from 1 (very annoying perceived quality) to 5 (not perceived difference in quality). We chose three parts of the performance of about 10 sec. duration each (referred to as Signals 1-3 here), which were listened by sixteen volunteers individually (authors are not included) using high-quality headphones. The tests conducted in this section are based on the fact that our goal is to resynthesize each microphone signal independently of the others, with the use of only one reference signal, which can be the sum of at least two spot signals depending on the application field, and the model parameters (sinusoids and LP filter) that characterize the side microphone signal.

The implementation of the proposed method is based on a 20 msec. analysis and synthesis frame for the sinusoidal model with 50% overlapping (with overlap-add method). The Linear Prediction order for the autoregressive noise shaping filters is 25 and the sampling frequency of the recordings is 44.1 kHz. In Figure 7.3 are depicted the average DCR tests for each of the three 2-channel testing signals. Each of the three figure corresponds to a different choice of sinusoidal parameters per frame. In particular, the upper plot corresponds to 80 sinusoids, the middle plot to 40 sinusoids, and the lower plot to 10 sinusoids per frame. In each plot of the Figure 7.3, the solid line corresponds to the sinusoidal model resynthesis ("sin"), while the dotted line corresponds to our proposed method ("sin plus LPC noise"). The dashed line corresponds to adding the noise part, which is obtained with the Critical Band Energy (CBE) approach mentioned in section 4.2, of the side signal to the sinusoidal part of the side signal ("sin plus CBE noise"). Finally, the dashed-dotted line corresponds to adding to the sinusoidal part of the side signal the noise of the reference signal which is modeled using the perceptual Linear Predictive Coding (PLPC) noise shaping model of section 4.3 ("sin plus PLPC noise"). A graphical representation of the 95% confidence interval are shown also in the Figure 7.3, where the x's mark the mean value and the vertical solid lines indicate the limits of the confidence interval.

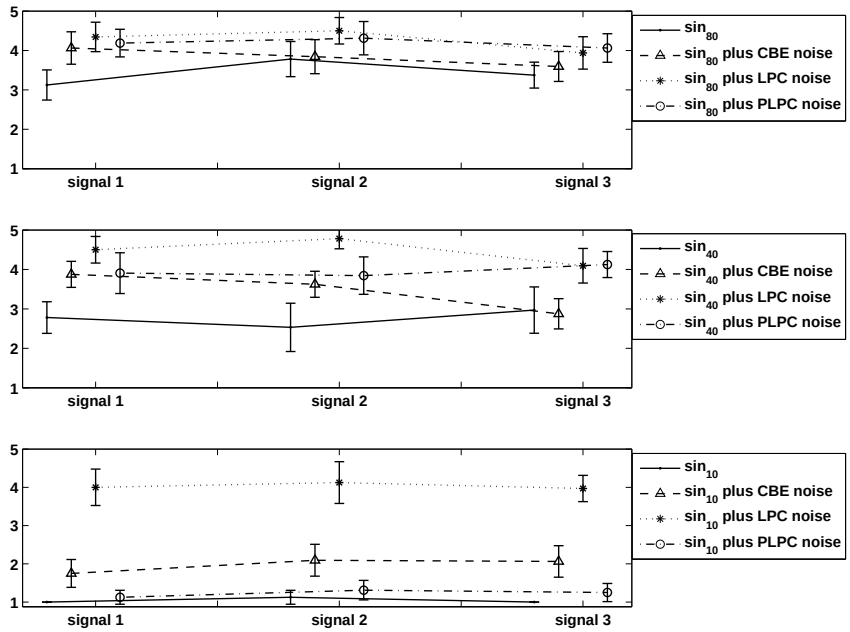

Figure 7.3: Results from the quality rating DCR listening tests corresponding to sinusoidal modeling with (a) 80 sinusoids per frame (upper), (b) 40 sinusoids per frame (middle), and (c) 10 sinusoids per frame (lower).

These results show clearly that the three noise-based methods are superior in comparison to the model based on sinusoidal parameters only. Thus, it is apparent that the noise has to be treated to achieve high-quality resynthesis of the spot microphone signals. Both CBE and PLPC approaches give slightly worse results compared to our method, in the case of high enough number of sinusoids (40 and 80 sinusoids in total). However, in the 10 sinusoids case, our Linear Prediction-based method still achieves a grade around 4.0, while the two other noise-based methods achieve a grade below 2.0. This can be attributed to the fact that PLPC and CBE methods treat the envelope of the noise part only, while our method provides a residual signal for the noise as well (based on similarities among the various spot microphone signals). Finally, it is important to note that, since the subjective audio quality remains high even for a very small number of sinusoids, *i.e.*, the case of 10 sinusoids in total, we can achieve the final objective of increased coding performance (low datarate), since it translates into decreasing the bitrate needed for encoding the sinusoidal part.

7.3.1 Downmix subjective tests

In this subsection, we examine if it is possible to resynthesize the various spot microphone signals from a downmix sum signal. This is important in cases when spot signals do not contain very similar audio content, which is often the case in studio recordings. In particular, we are interested to test the amount of crosstalk that is introduced, and whether there are implications regarding the quality of the resynthesized signals. It is expected that it will be more difficult to resynthesize good quality spot signals from the sum signal compared to the reference signal that was used in the previous section since the sum signal will contain frequency components which were not at all present in some spot signals. Also, crosstalk will be more audible in separate track recordings.

We used seven audio signals for the test. Each sound file used was a sum of two original recordings, and more specifically the following signals were created: (1) bass[2] plus soprano singer[2], (2) guitar[3] plus rock singer[3], (3) harpsichord[2] plus violin[2], (4) female[4] plus male speech[4], (5) trumpet[2] plus violin, (6) violin plus guitar, and (7) violin plus harpsichord. These seven signals correspond 1-1 to Signals 1-7 in the DCR results depicted in Fig. 7.4. The instrument that is referred first in the above list is the instrument (side signal) that we wanted to resynthesize from the sum signal (reference signal). 13 volunteers (the authors are not included) participated at this DCR listening test. The modeling parameters used for the experiments of this subsection are: a 20 msec analysis/synthesis frame is used for the sinusoidal model with 50% overlapping and the LP order for the AR noise shaping filters is 20, while the sampling rate for the recordings used is 44.1 kHz, except for the speech signals[4] which have 22 kHz rate.

Figure 7.4 shows the DCR results for the seven test signals. We can notice that in the 40 sinusoids case, the Signals 1-5 achieve a grade above 4.0, while the Signals 6-7 achieve a grade around 3.0 because the percussive sounds cannot be adequately modeled by the SNM, and significant information remains in the residual. Thus, it is a difficult task to diminish the percussive signal when resynthesizing another spot signal, but the opposite is not as hard. Besides, in the 10 sinusoids case, the grade of the Signals 1-2 and Signal

[2]http://sound.media.mit.edu/mpeg4/audio/sqam/
[3]courtesy of rock band "Orange Moon"
[4]http://www.cslu.ogi.edu/corpora/voices/

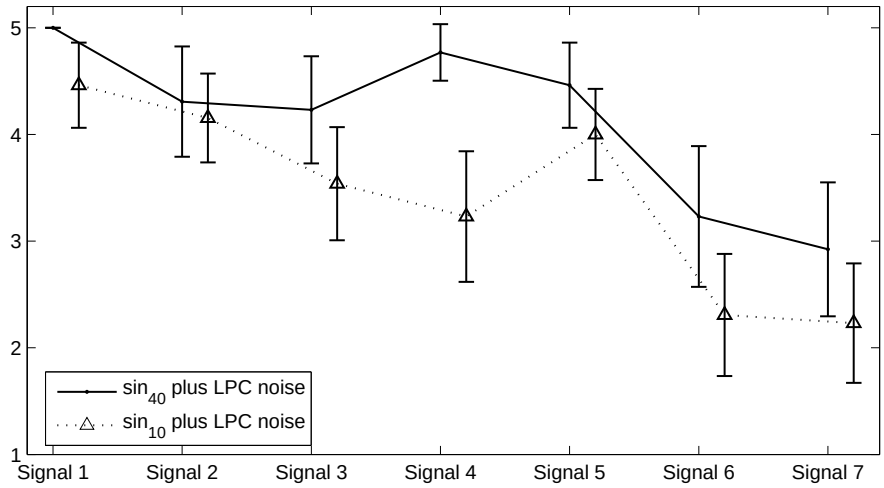

Figure 7.4: Results from the quality rating DCR listening tests for the downmix case, corresponding to sinusoidal modeling with (a) 40 sinusoids per frame (solid line), and (b) 10 sinusoids per frame (dotted line).

5 is above 4.0. The Signals 1-7 were also used to conduct an ABX test for the case of 40 and 10 sinusoids, respectively. In the ABX test, each listener was presented with the original two instrument recordings that were used to obtain the sum signal as signals A and B (in random order), as well as the resynthesized signal (Signal X), and was asked to associate X with A or B depending on which instrument prevails in the recording. The total ABX score in both cases was 100%, which means that *no crosstalk is introduced using the proposed method.*

As a general conclusion, resynthesis from a downmix sum signal is more challenging than from a signal which originally contains common information with all spot signals. Thus, the downmix case will be further examined in our future research.

Chapter 8

Coding of spot microphone signals

In this chapter, we present the second part of our proposed scheme which concerns the coding procedure of the modeling parameters. The coding process can be divided into two tasks; the coding of the sinusoidal parameters and the coding of the noise spectral envelopes for each side signal (for each short-time frame). Section 8.1 analyzes the coding framework of the sinusoidal parameters, while Section 8.2 describes the coding of the noise spectral envelopes. Figure 8.1 depicts the proposed coding scheme. In particular, the reference signal (Signal 1) is fully encoded (*e.g.* using an MP3 encoder at 64 kbps), while the remaining $M-1$ signals are reconstructed using the quantized sinusoidal and LP parameters, and the LP residual obtained from the reference channel.

8.1 Coding of the sinusoidal parameters

As we mentioned in Section 2.2, an audio signal can be decomposed into two parts; a sinusoidal part and a noise part. Thus, the sinusoids plus noise model (SNM) evaluated over an audio signal's segment of length N samples is

$$s(n) = \sum_{l=1}^{L} \alpha_l \cos(\omega_l n + \phi_l) + e(n)\,, \;\; n = 0, \ldots, N-1, \tag{8.1}$$

where L is number of sinusoids, N is the length (in samples) of the frame to be analyzed and $\{\alpha_l\,,\,\omega_l\,,\,\phi_l\}_{l=1}^{L}$ are the constant amplitudes, frequencies and phases respectively. We adopt the coding scheme of [53], developed for jointly optimal quantization of sinusoidal

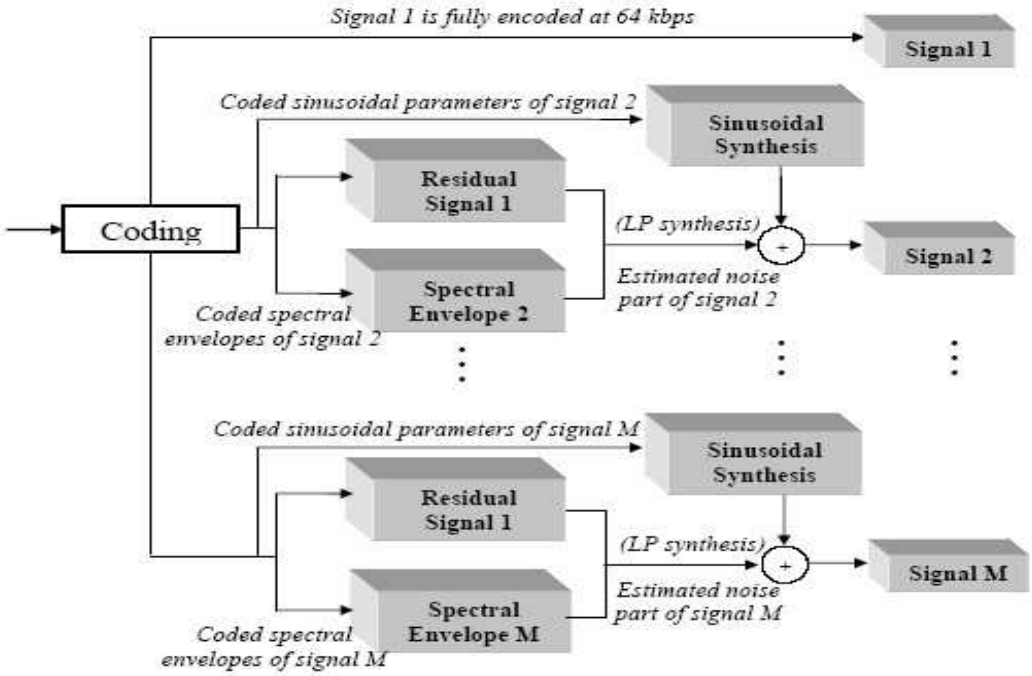

Figure 8.1: Diagram of the coding procedure.

frequencies, amplitudes and phases.

The sinusoidal parameters are quantized directly in polar form (see section 6.3.1), assuming that the frequency quantization is made dependent on the amplitude and phase quantization is made, also, dependent on amplitude and frequency. This is considered to be an unresticted polar quantization (UPQ) scheme, and represents a combination of three scalar quantizers. To derive the quantizers we make a high-rate quantization assumption, *i.e.*, the probability density function (pdf) of the data to be quantized does not change significantly within the quantization cell.

8.1.1 Formulation of quantization problem

In order to derive the quantizers, the goal is to minimize, in a segment-by-segment basis, the average weighted mean squared distortion (WMSE) for L sinusoids

$$D = \frac{1}{L} \sum_{l=1}^{L} w_l \, D_l \tag{8.2}$$

under the entropy constraint

$$\begin{aligned} H &= \frac{1}{L}\sum_{l=1}^{L} H(I_{\alpha l}, I_{\omega l}, I_{\phi l}) \\ &= \frac{1}{L}\sum_{l=1}^{L} (H(I_{\alpha l}) + H(I_{\omega l}|I_{\alpha l}) + H(I_{\phi l}|I_{\alpha l})). \end{aligned} \tag{8.3}$$

The mean squared error (MSE) D_l introduced by the quantization of the l^{th} sinusoid, is assigned with a perceptual weight w_l, which is defined as $w_l = 1/m_{th\,l},\ l = 1, \ldots L$, where $m_{th\,l}$ is the masking threshold at frequencies of the corresponding sinusoids.

In particular, the perceptual (sinusoidal) weights $w_l,\ l = 1, \ldots, L$ (where L is the number of sinusoids per analysis frame), are defined to be the reciprocal of the masking threshold $m_{\text{th l}}$ at frequencies of the corresponding sinusoids. This means that sinusoids with higher masking threshold values are assigned lower weights. The masking threshold $m_{\text{th l}}$ is computed using the frequency masking model of [51]. According to this model, the threshold at a certain frequency l is calculated as follows

$$m_{\text{th l}} = m_{\text{th quiet}} + \sum_{k=1}^{L} m_{\text{th masker k}}, \tag{8.4}$$

where $m_{\text{th quiet}}$ is the absolute threshold of hearing corresponding to the l^{th} frequency and quantities $m_{\text{th masker k}}$ in the summation $\sum_{k=1}^{L} m_{\text{th masker k}}$ correspond to the masking curves due to individual sinusoidal maskers. The absolute threshold of hearing expressed in dB sound pressure level (SPL) is given by

$$m_{\text{th quiet}}^{\text{dB SPL}} = \min\{60,\ 3.64 f_l^{-0.8} - 6.5 e^{-0.6(f_l - 3.3)^2} + 0.001 f_l^4\}, \tag{8.5}$$

where f_l is the frequency (in kHz) of the l^{th} sinusoid.

In Figure 8.2 the individual masking curve $m_{\text{th masker k}}$, expressed in dB SPL, of a sinusoidal masker k is shown, where P_k denotes the power of the k^{th} masker. The masking curve is described by the lower (s_l) and the upper (s_u) slope of masking and by the common offset (s_b) of the slope. The lower and upper slope denote the masking contribution to the frequencies below and above the (sinusoidal) masker frequency. The parameters used

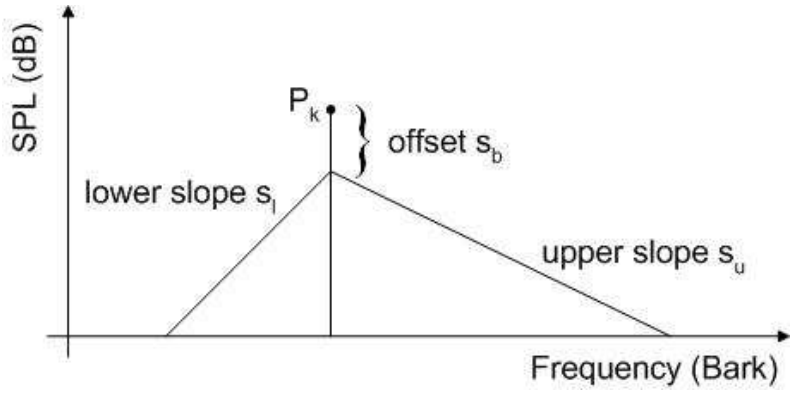

Figure 8.2: Masking curve due to an individual sinusoid.

in the computation of the perceptual weights are $s_l = -22$ dB/Bark, $s_u = (0.25P_k - 18)$ dB/Bark and $s_b = 10$ dB.

The given total entropy per sinusoid, *i.e.*, the triad amplitude plus frequency plus phase, is denoted by H. The entropy $H(I_{\alpha l}, I_{\omega l}, I_{\phi l})$ in Equation 8.3 expresses the joint entropy of the amplitude, frequency and phase quantization indices of the l^{th} sinusoid, while the entropies $H(I_{\alpha l})$, $H(I_{\omega l}|I_{\alpha l})$ and $H(I_{\phi l}|I_{\alpha l})$ express the entropies of the individual quantization indices. In UPQ, the frequency quantization dependency on amplitude means that $H(I_{\omega l}|I_{\alpha l}) < H(I_{\omega l})$, where $H(I_{\omega l})$ is the entropy of frequency quantization indices, and the phase quantization dependency on amplitude means, also, that $H(I_{\phi l}|I_{\alpha l}) < H(I_{\phi l})$, where $H(I_{\phi l})$ is the entropy of phase quantization indices.

The frequency quantization point density depends by definition on both frequency and amplitude, and is denoted by $g_\Omega(\omega, \alpha)$. Similarly, the phase quantization point density, denoted by $g_\Phi(\phi, \alpha, w_l)$, depends by definition on phase, amplitude and the individual weight w_l associated with the l^{th} sinusoid. By making the assumptions that the reproduction points of the amplitude, frequency and phase quantizers are in the middle of the corresponding quantization intervals and using the high-rate assumption that the probability density functions of the sinusoidal parameters are constant within the quantization interval [24, 53], the entropies of amplitude, frequency and phase quantization indices can

be written as

$$H(I_{\alpha l}) \approx h(A) + \int f_A(\alpha) \log_2(g_A(\alpha))\, d\alpha \tag{8.6}$$

$$H(I_{\omega l}|I_{\alpha l}) \approx h(\Omega) + \int\int f_{A,\Omega}(\alpha,\omega) \log_2(g_\Omega(\omega,\alpha))\, d\alpha\, d\omega \tag{8.7}$$

$$H(I_{\phi l}|I_{\alpha l}) \approx h(\Phi) + \frac{1}{L}\sum_{l=1}^{L}\int\int f_{A,\Phi}(\alpha,\phi) \log_2(g_\Phi(\phi,\alpha,w_l))\, d\alpha\, d\phi, \tag{8.8}$$

where $h(A)$, $h(\Omega)$ and $h(\Phi)$ are the differential entropies of the amplitude, frequency and phase variables, respectively, $f_A(\alpha)$ denotes the marginal pdf of the amplitude variable, $f_{A,\Omega}(\alpha,\omega)$ is the joint pdf of the amplitude and frequency variables, and $f_{A,\Phi}(\alpha,\phi)$ is the joint pdf of the amplitude and phase variables. Thus, the entropy constraint in Equation 8.3 can be written as follows

$$\underbrace{H - h(A) - h(\Omega) - h(\Phi)}_{\tilde{H}} \approx \int f_A(\alpha) \log_2(g_A(\alpha))\, d\alpha + \iint f_{A,\Omega}(\alpha,\omega) \log_2(g_\Omega(\omega,\alpha))\, d\alpha\, d\omega + \\ + \frac{1}{L}\sum_{l=0}^{L-1}\iint f_{A,\Phi}(\alpha,\phi) \log_2(g_\Phi(\phi,\alpha,w_l))\, d\alpha\, d\phi. \tag{8.9}$$

8.1.2 Derivation of the entropy constrained quantizers

The MSE D_l over a segment of length N, can be expressed as

$$D_l = E\left\{\frac{1}{N}\sum_{n=-(N-1)/2}^{(N-1)/2}(\alpha_l\, \cos(\omega_l\, n + \phi_l) - \hat{\alpha}_l\, \cos(\hat{\omega}_l\, n + \hat{\phi}_l))\right\}, \tag{8.10}$$

where $\{\alpha_l,\, \omega_l,\, \phi_l\}$ and $\{\hat{\alpha}_l,\, \hat{\omega}_l,\, \hat{\phi}_l\}$ are the unquantized and quantized sinusoidal parameters respectively, and $E\{\cdot\}$ denotes the expectation value. After some algebraic manipulation [53, 52, 24], D_l can be written in the form

$$D_l = E\left\{\frac{(\alpha_l - \hat{\alpha}_l)^2}{2} + \alpha_l\hat{\alpha}_l\left(\frac{(\omega_l - \hat{\omega}_l)^2\, N^2}{24} + \frac{(\phi_l - \hat{\phi}_l)^2}{2}\right)\right\}, \tag{8.11}$$

From Equations 8.2, 8.11 we conclude that it is preferable to quantize with higher accuracy sinusoids associated with higher perceptual weights w_l. Thus, it is obvious that the amplitude, frequency and phase quantizers should depend on the weights w_l. However, only the phase quantizers should depend on the weights w_l because in a practical coding scheme, the w_l do not have to be transmitted on the decoder side since they can be reconstructed using the quantized sinusoidal amplitudes and frequencies.

By making, again, high-rate assumptions and assuming that the reproduction points of the amplitude, frequency and phase quantizers are in the middle of the corresponding quantization intervals, Equation 8.11 can be written as

$$D_l \approx \iiint f_{A,\Omega,\Phi}(\alpha,\omega,\phi)\left(\frac{g_A^{-2}(\alpha)}{24}+\alpha^2\frac{g_\Omega^{-2}(\omega,\alpha)N^2}{288}+\alpha^2\frac{g_\Phi^{-2}(\phi,\alpha,w_l)}{24}\right)d\alpha\,d\omega\,d\phi. \tag{8.12}$$

Thus, the optimization problem is to minimize the WMSE in Equation 8.2 under the constraint expressed in Equation 8.9. This constrained minimization problem can be solved using the method of Lagrange multipliers, where the criterion to optimize is

$$\begin{aligned}\rho = &\iiint f_{A,\Omega,\Phi}(\alpha,\omega,\phi)\left(\frac{g_A^{-2}(\alpha)}{24}+\alpha^2\frac{g_\Omega^{-2}(\omega,\alpha)N^2}{288}+\alpha^2\frac{g_\Phi^{-2}(\phi,\alpha,w_l)}{24}\right)d\alpha\,d\omega\,d\phi+\\ &+\lambda\left(\int f_A(\alpha)\log_2(g_A(\alpha))\,d\alpha+\iint f_{A,\Omega}(\alpha,\omega)\log_2(g_\Omega(\omega,\alpha))\,d\alpha\,d\omega+\right.\\ &\left.+\frac{1}{L}\sum_{l=0}^{L-1}\iint f_{A,\Phi}(\alpha,\phi)\log_2(g_\Phi(\phi,\alpha,w_l))\,d\alpha\,d\phi\right),\end{aligned} \tag{8.13}$$

where λ is the Lagrange multiplier which corresponds to the constraint in Equation 8.9.

We evaluate the Euler-Lagrange equations with respect to the quantization point densities $g_A(\alpha)$, $g_\Omega(\omega,\alpha)$ and $g_\Phi(\phi,\alpha,w_l)$ in order to obtain the optimum quantization point

densities

$$g_A(\alpha) = g_A = \frac{w_\alpha^{\frac{1}{6}} 2^{\frac{1}{3}\tilde{H}-\frac{2}{3}b(A)}}{w_g^{\frac{1}{6}} (\frac{N^2}{12})^{\frac{1}{6}}} \tag{8.14}$$

$$g_\Omega(\omega,\alpha) = g_\Omega(\alpha) = \frac{\alpha w_\alpha^{\frac{1}{6}} \left(\frac{N^2}{12}\right)^{\frac{1}{3}} 2^{\frac{1}{3}\tilde{H}-\frac{2}{3}b(A)}}{w_g^{\frac{1}{6}}} \tag{8.15}$$

$$g_\Phi(\phi,\alpha,w_l) = g_\Phi(\alpha,w_l) = \frac{\alpha w_l^{\frac{1}{2}} 2^{\frac{1}{3}\tilde{H}-\frac{2}{3}b(A)}}{w_\alpha^{\frac{1}{3}} w_g^{\frac{1}{6}} (\frac{N^2}{12})^{\frac{1}{6}}}, \tag{8.16}$$

where $w_\alpha = (1/L)\sum_{l=1}^{L} w_l$ and $w_g = (\prod_{l=1}^{L} w_l)^{1/L}$ is the arithmetic and geometric mean of the perceptual weights of the L sinusoids, respectively, $\tilde{H} = H - h(A) - h(\Omega) - h(\Phi)$ and $b(A) = \int f_A(\alpha)\log_2(\alpha)\,d\alpha$. If the Equations 8.14 − 8.16 are substituted into the Equation 8.12 and the Equation 8.2, we derive the optimal average mean squared distortion

$$D = \frac{w_\alpha^{\frac{2}{3}} w_g^{\frac{1}{3}} \left(\frac{N^2}{12}\right)^{\frac{1}{3}}}{8} 2^{-\frac{2}{3}\tilde{H}+\frac{4}{3}b(A)}. \tag{8.17}$$

At this point, we have to remind that the quantization point density is defined as the inverse of the quantizer step size Δ, that is, as the quantization point density is taking high values, the step size Δ becomes smaller, so we have high quantization accuracy. Thus, we observe from Equation 8.15 that the accuracy of the uniform frequency quantizer increases with amplitude α and segment length N. From Equation 8.16, we can, also, observe that the accuracy of the uniform phase quantizer increases with amplitude α and perceptual weight w_l.

8.2 Coding of the spectral envelopes

In Section 7.2, it is mentioned that the noise part of a spot microphone signal can be synthesized by passing the autoregressive (AR) modeling error of the reference spot microphone signal through the all-pole Linear Predictive (LP) filter of the spot microphone we want to synthesize. In the current section, the algorithm which quantizes the spectral envelopes of the sinusoidal error of the spot signals is described. We follow the quantization

scheme mentioned in [48].

In specific, the LP coefficients of each spot signal frame's AR modeling error are transformed to LSF's (Line Spectral Frequencies). Next, each vector that contains the LSF coefficients of each spot signal is modeled with the use of a Gaussian Mixture Model (see section 5.1),

$$g(\boldsymbol{x}) = \sum_{i=1}^{C} p_i N(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{8.18}$$

where $N(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the normal multivariate distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, p_i is the prior probability that the observation $\boldsymbol{x}$ has been generated by cluster i and C is the number of clusters.

The covariance matrix of each cluster can be diagonalized using eigenvalue decomposition as

$$\boldsymbol{\Sigma}_i = \mathbf{Q}_i \boldsymbol{\Lambda}_i \mathbf{Q}_i^T, \tag{8.19}$$

where $i = 1, \ldots, C$ and $\boldsymbol{\Lambda}_i = \text{diag}(\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,p})$. The matrix $\boldsymbol{\Lambda}_i$ is diagonal and contains the corresponding eigenvalues of $\boldsymbol{\Sigma}_i$, while $\mathbf{Q}_i$ is the matrix containing the corresponding set of orthogonal eigenvectors of $\boldsymbol{\Sigma}_i$, for the i^{th} Gaussian class of the model. Then, the Karhunen-Loève transform (KLT) substitutes each LSF vector for time segment k, $\boldsymbol{z}_k$, with another decorrelated vector $\boldsymbol{w}_k$, where $\boldsymbol{w}_k = \mathbf{Q}_i^T(\boldsymbol{z}_k - \boldsymbol{\mu}_i)$. Afterwards, the components of the vector $\boldsymbol{w}_k$ are passed through a nonuniform quantizer, *i.e.*, through a compressor, a uniform quantizer and an expander.

In the decoder side of the quantization procedure, the correlated version of the quantized vector is reconstructed by left multiplying the reconstructed $\boldsymbol{w}_k$ with the matrix $\mathbf{Q}_i$. Finally, the cluster mean $\boldsymbol{\mu}_i$ is added to obtain the quantized value of $\boldsymbol{z}_k$ by the i^{th} cluster, $\hat{\boldsymbol{z}}_k$. Figure 8.3 depicts the overall process for quantizing each LSF vector.

In order to choose the GMM cluster that best models a particular LSF vector, we evaluate the vector's relative distortion value and we choose the distortion with the minimum value. This procedure os shown in Figure 8.4. Here, the Log Spectral Density (LSD) is used as a measure of distance. LSD is computed for each GMM class, and the vector is

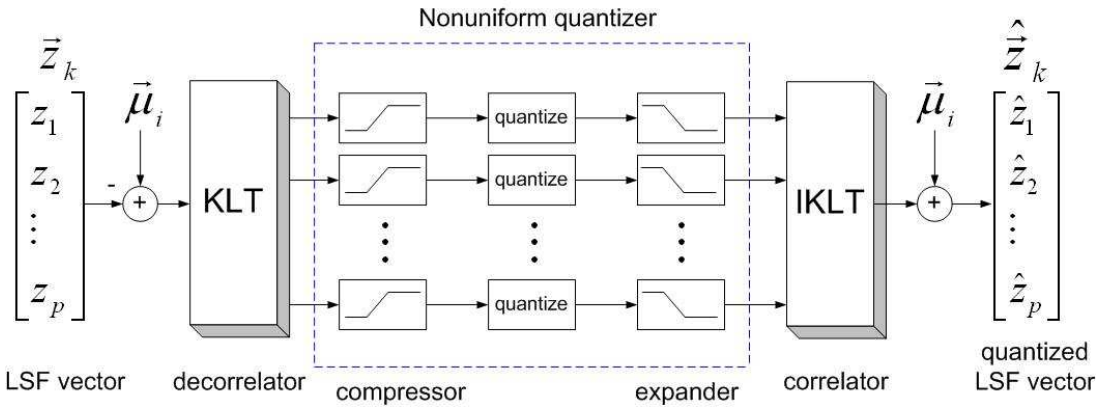

Figure 8.3: LSF quantization scheme.

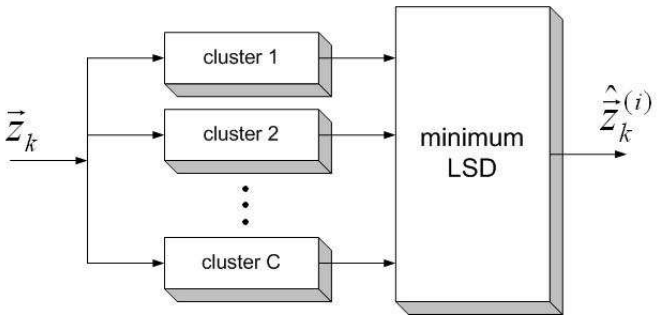

Figure 8.4: LSF's classification scheme using the minimum LSD value.

classified to the cluster associated with the minimal LSD, which is defined as

$$LSD(i) = \left(\frac{1}{F_s} \int_0^{F_s} \left[10 \log_{10} \left(\frac{S(f)}{\hat{S}^{(i)}(f)} \right) \right]^2 df \right)^{\frac{1}{2}}, \tag{8.20}$$

where F_s is the sampling rate, $S(f)$, $\hat{S}^{(i)}(f)$ are respectively the LP power spectra corresponding to the original vector $\boldsymbol{z}_k$ and the quantized vector $\hat{\boldsymbol{z}}_k^{(i)}$, for each cluster $i = 1, \ldots, C$.

A bit allocation scheme for the uniform quantizer (see Figure 8.3) is needed in order to allocate the total available bits (denoted by b_{tot} and specified by the user) for quantizing the source, among the various clusters of the GMM. Let b_i be the bits for quantizing cluster i, and q_i the quantity

$$q_i = \left[\prod_{j=1}^{p} \lambda_{i,j} \right]^{\frac{1}{p}}, \quad i = 1, \ldots, C, \tag{8.21}$$

where p is the dimensionality of the LSF vector and $\lambda_{i,j}$ is the j^{th} eigenvalue of cluster i. The bit allocation scheme can be either *fixed rate* or *variable rate*. In the fixed rate bit allocation scheme, the length of the codewords is fixed and can be easily found to satisfy the constraint

$$2^{b_{tot}} = \sum_{i=1}^{C} 2^{b_i}. \tag{8.22}$$

The optimal bit allocation which minimizes the total average mean square distortion under the constraint 8.22, is given by

$$b_i = b_{tot} - \log_2 \left[\sum_{j=1}^{C} (p_j q_j)^{\frac{p}{p+2}} \right] + \frac{p}{p+2} \log_2(p_i q_i),\ i = 1, \ldots, C. \tag{8.23}$$

In the variable rate bit allocation scheme, some of the total bits (denoted b_c) are used for the cluster identification. Thus, the variable rate constraint becomes

$$b_q = b_{tot} - b_c, \tag{8.24}$$

where $b_c = \log_2 C$. In a variable rate quantizer, the *average* rate of the quantizer is fixed, which translates into the constraint

$$b_q = \sum_{i=1}^{C} p_i b_i. \tag{8.25}$$

The optimal bit allocation which minimizes the total average mean square distortion under the constraint 8.25, is given by

$$b_i = b_q + \frac{p}{2} \left[\log_2 q_i - \sum_{j=1}^{C} p_j \log_2 q_j \right], i = 1, \ldots, C. \tag{8.26}$$

After the evaluation of the cluster allocated bits, the bit allocation among the cluster dimensions is given by

$$b_{i,j} = \tfrac{b_i}{p} + \tfrac{1}{2} \log_2 \left[\tfrac{\lambda_{i,j}}{q_i} \right], \quad i = 1, \ldots, C \quad j = 1, \ldots, p, \tag{8.27}$$

where $b_{i,j}$ is the allocated bits to the j^{th} component of the i^{th} cluster. For more accurate bit allocation, we rounded $b_{i,j}$ in the nearest integer number. In this study, we focus on the variable rate bit allocation scheme.

8.3 Performance evaluation of coding

In this section, we are interested to examine the coding performance of the proposed system, with respect to the resulting audio quality. For this purpose we performed subjective (listening) tests by employing DCR and ABX tests. For our listening tests, we used three signals, referred to as Signals 1-3. These signals are parts of a multichannel recording of a concert hall performance. These three signals are the same as the signals used for evaluating the quality of the modeling in Section 7.3 in which we used the recordings from two different microphones, one of which captured mainly the female voices of the orchestra chorus, while the second one captured mainly the male voices. The former was used in our experiments as the side channel, and the latter as the reference signal. Our main objective is to test whether the side signal can be accurately reproduced when using the residual from the reference signal.

In Section 7.3 we showed that the proposed noise transplantation approach results in very good quality (around 4.0 grade in DCR tests) for the three signals. Thus, in this section our objective is to examine the lower limit in bitrates which can be achieved by our proposed system without loss of audio quality below the grade achieved by modeling alone (*i.e.* 4.0 grade for the three signals tested here).

Regarding the parameters used for deriving the waveforms used in the tests, the sampling rate for the audio data was 44.1 kHz and the LP order for the AR noise shaping filters was 10. The analysis/synthesis frame for the implementation of the sinusoidal model is 30 msec with 50% overlapping between successive frames. The coding efficiency for the sinusoidal parameters was tested for a given (target) entropy of 28 and 20 bits per sinusoid (amplitudes, frequencies and phases in total). The given entropy of 28 and 20 bits per sinusoid (amplitudes, frequencies and phases in total) gives a bitrate of 18.67 and 13.3 kbps, respectively.

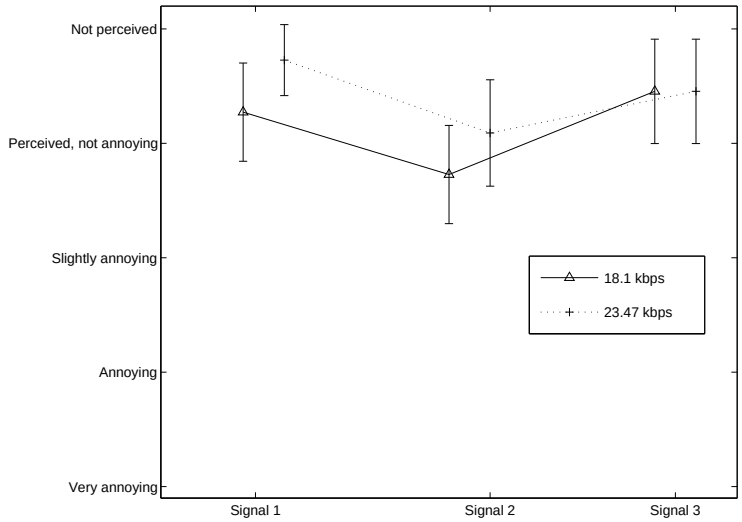

Figure 8.5: Results from the quality rating DCR listening tests, corresponding to coding with (a) 23.47 kbps (dotted), (b) 18.1 kbps (solid). Each frame is modeled with 10 sinusoids and 10 LP parameters.

Regarding the coding of the LP parameters (noise spectral envelope), 28 bits were used per LSF vector. With 23 msec frame and 75 % overlapping, this corresponds to 4.8 kbps for the noise envelopes. Thus, the resulting bitrates that were tested are 23.47 kbps and 18.1 kbps (adding the bitrate of the sinusoidal parameters and the noise envelopes). A training audio dataset of about 100,000 LSF vectors (approximately 9.5 min of audio) was used to estimate the parameters of a 16-class GMM. The training database consisted of recordings of the classical music performance (corresponding to the recording from which Signals 1-3 originated, but a different part of the recording than the one used for testing).

Eleven volunteers participated in the DCR tests, using high-quality headphones. The results of the DCR tests are depicted in Figure 8.5, where the 95% confidence interval are shown (the vertical lines indicate the confidence limits). The solid line shows the results for the case of coding with a bitrate of 18.1 kbps, while the dotted line shows the results for the 23.47 kbps case. The results of the figure verify that the quality of the coded audio signals is good and the proposed algorithm offers an encouraging performance, and that this quality can be maintained at as low as 18 kbps per side signal. We note that the reference signal was PCM coded with 16 bits per sample, however similar results were obtained for the side signals when the reference signal was MP3 coded at 64 kbps (monophonic case).

In order to have an objective quality measure for the quantization evaluation, we define

audio signal	segmental SQNR (23.47 kbps)	segmental SQNR (18.1 kbps)
Signal 1	6.5407	6.4473
Signal 2	5.5531	5.2958
Signal 3	5.8855	5.7393

Table 8.1: Segmental SNR for the 23.47 kbps and 18.1 kbps bitrate.

the following signal-to-quantization noise ratio (SQNR) for the original signal frame x and its quantized counterpart $\tilde{x}$

$$SQNR = 10 \log_{10} \left(\frac{\|x\|_2^2}{\|x - \tilde{x}\|_2^2} \right) \text{ (dB)}. \tag{8.28}$$

We use the segmental SQNR defined as the average value taken across the signal frames in question. The segmental SQNR of the Signals 1-3, for the cases of 18.1 kbps and 23.47 kbps is shown in Table 8.1. If we notice Table 8.1 we can affirm the subjective results shown in Figure 8.5.

We, also, conducted an ABX test using the same Signals 1-3. In this case, the objective was to test whether our method introduces crosstalk to the side signals from the reference, and whether this affects the coding procedure. This is an important issue in our approach, since all side signals are synthesized using the residual of the reference recording. Each listener was asked to decide whether X is more similar to A or B, where X is the coded signal while A and B are the corresponding reference and side signals in random order. The total ABX score in the case of 18.1 kbps was 95% and for the 23.47 kbps case was 98.3%, which verifies that the crosstalk issue does not affect the coding process.

Chapter 9

Conclusion and Future work

In the present work, we described a sinusoids plus noise model that is specifically tailored for multichannel audio, with the objective of low bitrate coding by taking advantage of the similarities among the various spot microphone signals. Spot signals were treated here since preserving their content and quality is important when interactivity between the listener and the acoustic environment is needed, as in truly immersive environments.

The proposed approach offers the possibility of employing the flexible sinusoidal model into low bitrate multichannel audio coding, following a similar Spatial Audio Coding philosophy. At the same time, by focusing on the spot signals before those are mixed into the final multichannel mix, our method allows for many applications that are not feasible if the spot signals are not available to the decoder.

The proposed method is divided into the modeling and coding stage. The modeling scheme is implemented via an enhanced adaptation of the sinusoids plus noise model. Sinusoids cannot be used per se for high-quality audio modeling because they do not represent all the audible information of a recording, thus the noise part has to be treated to avoid an artificial sounding resynthesis of the audio signal. Subjective listening tests in Chapter 7 demonstrate that high-quality resynthesis of spot microphones can be achieved using the *noise transplantation procedure*, in which the noise part for each spot microphone (side) signal (before the mixing stage) can be obtained by using its noise envelope to transform the noise part of just one of the signals (the so-called "reference signal"). In Chapter 8 the coding procedure of the modeling parameters is presented. The coding

process can be divided into the coding of the sinusoidal parameters and the coding of the noise spectral envelopes for each spot (side) signal. It is shown that the proposed coding method allows for good-quality audio coding for as low as about 18 kbps per spot signal.

In the future, we intend to improve the system by better modeling of the transient signals and by using multiresolution sinusoidal analysis. We intend to further reduce the bitrates that are associated with the proposed method. Finally, a validity of the coding process will be examined via more subjective listening tests towards using more downmix reference signals.

Bibliography

[1] F. Baumgarte and C. Faller. Binaural cue coding - Part I: Psychoacoustic fundamentals and design principles. *IEEE Trans. Speech and Audio Processing*, 11(6):509–519, November 2003.

[2] J. Bilmes. A gentle tutorial of the EM algorithm and its application for Gaussian mixture and hidden Markov models. Technical report, Berkeley, 1998.

[3] K. Brandenburg and F. Bosi. ISO/IEC MPEG-2 advanced audio coding: Overview and applications. *Proc.* 103^{rd} *Convention of the Audio Engineering Society (AES)*, preprint No. 4641, (New York, NY), September 1997.

[4] K. Brandenburg and G. Stoll. ISO-MPEG-1 audio: A generic standard for coding of high-quality digital audio. *Journal of Audio Engineering Society*, pages 780–792, October 1994.

[5] J. Breebaart, S. van de Par, A. Kohlrausch, and E. Schuijers. Parametric coding of stereo audio. *EURASIP Journal on Applied Signal Processing*, pages 1305–1322, 2005:9.

[6] J. Breebaart et al. MPEG Spatial Audio Coding / MPEG Surround: Overview and current status. In *Proc. AES* 119^{th} *Convention, Paper 6599*, October 2005.

[7] M. G. Christensen, A. Jakobsson, S. V. Andersen, and S. H. Jensen. Linear AM decomposition for sinusoidal audio coding. *Proceedings IEEE International Conference Acoustics, Speech and Signal Processing*, 3:165–168, March 2005.

[8] T. M. Cover and J. A. Thomas. Elements of information theory. *New York, Wiley*, 1991.

[9] M. Davis. The AC-3 multichannel coder,. *in Proc.* 95^{th} *Convention of the Audio Engineering Society (AES), preprint No. 3774*, October 1993.

[10] C. Faller and F. Baumgarte. Binaural cue coding - Part II: Schemes and applications. *IEEE Trans. Speech and Audio Processing*, 11(6):520–531, November 2003.

[11] W. R. Gardner and B. D. Rao. Theoretical analysis of the high-rate vector quantization of lpc parameters. *IEEE Trans. Speech and Audio Processing*, 3:5:367–381, September 1995.

[12] A. Gersho. Asymptotically optimal block quantization. *IEEE Trans. Information Theory*, 25:4:373–380, July 1979.

[13] A. Gersho and R. M. Gray. *Vector quantization and signal compression.* Kluwer academic publishers, 1997.

[14] M. Goodwin. Residual modeling in music analysis-synthesis. *Proceedings IEEE International Conference Acoustics, Speech and Signal Processing*, 2:1005–1008, May 1996.

[15] R. M. Gray. Source coding theory. *Kluwer academic publishers*, 1990.

[16] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Trans. Information Theory*, 44:6:968–979, October 1998.

[17] S. Haykin. *Adaptive filter theory.* Prentice Hall, 1996.

[18] R. C. Hendriks, R. Heusdens, and J. Jensen. Perceptual linear predictive noise modelling for sinusoid-plus-noise audio coding. *Proceedings IEEE International Conference Acoustics, Speech and Signal Processing*, 4:189–192, May 2004.

[19] J. Herre, K. Brandenburg, and D. Lederer. Intensity stereo coding. In *Proc.* 96^{th} *Convention of the Audio Engineering Society (AES), preprint No. 3799*, 1994.

[20] J. Jensen, R. Heusdens, and S. H. Jensen. A perceptual subspace approach for modeling of speech and audio signals with damped sinusoids. *IEEE Trans. Speech and Audio Processing*, 12(2):121–132, March 2004.

[21] J. D. Johnston and A. J. Ferreira. Sum-difference stereo transform coding. In *Proceedings IEEE International Conference Acoustics, Speech and Signal Processing*, pages 569–572, 1992.

[22] K. Karadimou, A. Mouchtaris, and P. Tsakalides. Multichannel audio modeling and coding using a multiband source/filter model. *in Proc.* 39^{nd} *Annual Asilomar Conference on Signals, Systems and Computers*, October 30 - November 2 2005.

[23] B. W. Kleijn and K. K. Paliwal. Speech coding and synthesis. Elsevier, 1995.

[24] W. B. Kleijn. A basis for source coding. *KTH (Royal institute of technology), Stockholm, Sweden*, 2004.

[25] W. B. Kleijn and K. K. Paliwal, editors. *Speech Coding and Synthesis.* Elsevier Science, 1995.

[26] K. Lange. *Applied probabilty.* New York, Springer-Verlag, 2003.

[27] J. Lee. Optimized quadtree for Karhunen-Loeve transform in multispectral image coding. *IEEE Trans. Image Processing*, 8(4):453–461, April 1999.

[28] J. Li, N. Chaddha, and R. M. Gray. Asymptotic performance of vector quantizers with a perceptual distortion measure. *IEEE Trans. Information Theory*, 45:4:1082–1091, May 1999.

[29] J. Makhoul. Spectral linear prediction: properties and applications. *IEEE Trans. Acoustics, Speech and Signal Processing*, 23(3):283–296, June 1975.

[30] R. J. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. Acoustics, Speech and Signal Processing*, 34(4):744–754, August 1986.

[31] B. C. J. Moore and B. R. Glasberg. Suggested formulae for calculating auditory-filter bandwidths and excitation patterns. *Journal of the Acoustical Society of America*, 74(3):750–753, September 1983.

[32] A. Mouchtaris. Time-frequency and adaptive signal processing methods for immersive audio virtual acquisition and rendering. *Ph.D Thesis*, May 2003.

[33] A. Mouchtaris, S. S. Narayanan, and C. Kyriakakis. Virtual microphones for multichannel audio resynthesis. *EURASIP Journal on Applied Signal Processing*, 2003:10:968–979, September 2003.

[34] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-time signal processing.* Prentice Hall, Second ed., 1999.

[35] T. Painter and A. Spanias. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–513, April 2000.

[36] J. Proakis and M. Salehi. *Communication systems engineering.* Prentice Hall, 2002.

[37] T. F. Quatieri and R. J. McAulay. Shape invariant time-scale and pitch modification of speech. *IEEE Trans. Acoustics, Speech and Signal Processing*, 40(3):497–510, March 1992.

[38] A. Rezayee and S. Gazor. An adaptive KLT approach for speech enhancement. *IEEE Trans. Speech and Audio Processing*, 9(2):87–95, February 2001.

[39] P. E. H. Richard, O. Duda, and D. G. Stork. *Pattern Classification.* Wiley Interscience, Second ed., 2000.

[40] J. Samuelsson and P. Hedelin. Recursive coding of spectrum parameters. *IEEE Trans. Speech and Audio Processing*, 9:5:492–503, July 2001.

[41] A. Sawchuk, E. Chew, R. Zimmermann, C. Papadopoulos, and C. Kyriakakis. From remote media immersion to distributed immersive performance. *in Proc. ACM SIGMM Workshop on Experiential Telepresence (ETP)*, November 2003.

[42] D. W. Scott. *Multivariate density estimation: Theory Practice and Visualization.* New York, Wiley, 1992.

[43] X. Serra. A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition. *PhD thesis, Stanford University, Stanford, CA*, 1989.

[44] X. Serra and J. O. Smith. Spectral modeling sythesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24, Winter 1990.

[45] F. K. Soong and B. H. Juang. Line spectrum pair (lsp) and speech data compression. *Proceedings IEEE International Conference Acoustics, Speech and Signal Processing*, pages 37–40, March 1984.

[46] Y. Stylianou. Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification. PhD. diss. Ecole Nationale Supèrieure des Tèlècommunications, January 1996.

[47] Y. Stylianou. Applying the harmonic plus noise model in concatinative speech synthesis. *IEEE Trans. Speech and Audio Processing*, 9(1):21–29, January 2001.

[48] A. D. Subramaniam and B. D. Rao. Pdf optimized parametric vector quantization of speech line spectral frequencies. *IEEE Trans. Speech and Audio Processing*, 11:365–380, March 2003.

[49] C. Tzagkarakis, A. Mouchtaris, and P. Tsakalides. Modeling spot microphone signals using the sinusoidal plus noise approach. *in IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October 2007.

[50] C. Tzagkarakis, A. Mouchtaris, and P. Tsakalides. Sinusoidal modeling of spot microphone signals based on noise transplantation for multichannel audio coding. *in Proc. European Signal Processing Conference*, September 2007.

[51] R. Vafin, S. V. Andersen, and W. B. Kleijn. Exploiting time and frequency masking in consistent sinusoidal analysis-synthesis. *Proceedings IEEE International Conference Acoustics, Speech and Signal Processing*, 2:901–904, Junuary 2000.

[52] R. Vafin and W. B. Kleijn. Entropy-constrained polar quantization and its application to audio coding. *IEEE Trans. Speech and Audio Processing*, 13(2):220–232, March 2005.

[53] R. Vafin, D. Prakash, and W. B. Kleijn. On frequency quantization in sinusoidal audio coding. *IEEE Signal Processing Letters*, 12(3):210–213, March 2005.

[54] S. G. Wilson. Magnitude/phase quantization of independent Gaussian variates. *IEEE Trans. Communications*, 28:11:1924–1929, November 1980.

[55] D. Yang, H. Ai, C. Kyriakakis, and C-C. J. Kuo. High-fidelity multichannel audio coding with Karhunen-Loève transform. *IEEE Trans. Speech and Audio Processing*, 11:365–380, July 2003.

[56] E. Zwicker and H. Fastl. *Psychoacoustics: facts and models*. Springer-Verlag, Second ed., 1999.