Perceptual Coding of Audio Residues
- Waveform Approximating Residual Audio Coding with Perceptual Pre- and Post-Filtering -
Title:
Perceptual Coding of Audio Residues

Master Programme:
Wireless Communication

Project period:
E8, spring semester 2008

Project group:
Group 880

Participants:
Jesper Rindom Jensen
Jesper Kjær Nielsen

Supervisor:
Søren Holdt Jensen
Mads Græsbøll Christensen

Copies: 6
Page numbers: 84

Date of completion: 3rd of June, 2008

The content of this report is freely available, but publication (with reference source) may only be pursued due to agreement with the respective authors.
Preface

This collection of materials is written by project group 08gr880 at the department of Electronic Systems on Aalborg University during the 8th semester in the period spanning from February 4th, 2008 to June 3rd, 2008. The project title was "Perceptual Coding of Audio Residues" and an extended abstract, an article and worksheets have been composed on basis thereof, and have been enclosed in this collection of materials.

The reader should pay attention to the following on perusal of this collection of materials:

- The report is divided into three major parts:
  - The article.
  - The worksheets.
- Figures, tables and equations are numerated consecutively according to the chapter number. Hence, the first figure in chapter one is named figure 1.1, the second figure figure 1.2 and so on.
- The extended abstract, the article and each worksheet have their own bibliography.

Aalborg University June 3rd, 2008

Jesper Rindom Jensen
<jesperrj@es.aau.dk>

Jesper Kjær Nielsen
<jkjaer@es.aau.dk>
Extended Abstract
- Waveform Approximating Residual Audio Coding with Perceptual Pre- and Post-Filtering -
Waveform Approximating Residual Audio Coding with Perceptual Pre- and Post-Filtering

Jesper Kjær Nielsen¹, Jesper Rindom Jensen, Mads Græsbøll Christensen², Søren Holdt Jensen³, and Torben Larsen
Aalborg University
Department of Electronic Systems
Fredrik Bajers Vej 7, DK-9220 Aalborg

Abstract

We investigate waveform approximating residual coding for a sinusoidal parametric audio coder at low bit rates. The residual coding is based on the well-known pre- and post-filtering method with lossless coding [1] which features perceptual weighting for short time segments at a low computational complexity. We compare the incurred perceptual distortion from joint quantization of the residual and the sinusoids for different bit rates. In addition to that we develop a transform coding scheme for the coefficients in the pre- and post-filters which must be send as side information between the encoder and decoder. Our investigations show that the combination of the sinusoidal subcoder and the pre- and post-filtering entails an overall lower perceptual distortion for low as well as high bit rates. Also, the developed transform coding scheme enables efficient coding of the side information at a very low bit rate.

Review Topic: I.4. Narrowband / Wideband Speech and Audio Coding

Extended Summary

Parametric audio coding at low bit rates has in recent years been subject to comprehensive research which among others has led to the inclusion of HILN (harmonics and individual lines plus noise) into the MPEG-4 standard [2]. HILN consists like most other parametric coders of a sinusoidal subcoder and a residual or noise subcoder where the latter codes the remaining audio signal which cannot be extracted by the sinusoidal subcoder. Residual subcoders are typically divided into non-waveform and waveform approximating subcoders. The non-waveform approximating subcoders are typically based on stochastic modelling of the residual and perform well at low bit rates. The audio quality, however, does not in general increase with increasing bit rate which is in contrast to the waveform approximating coders whose main drawback is poor performance at low bit rates.

In our work we consider the use of a waveform approximating residual subcoder at low bit rates and seek to minimize the distortion in the perceptual domain. For this purpose we use the pre- and post-filtering method [1, 3] which features weighted cascade least-mean square (WCLMS) prediction for lossless coding. In the pre- and post-filtering method, a pre-filter adapts its frequency response to the inverse of the masking curve thus mapping the audio signal to a perceptual domain in which irrelevancy reduction can be performed in a straight-forward manner. The inverse filtering is performed by the post-filter whose frequency response equals the masking curve. The adaption of the pre- and post-filter to the masking curve requires side information to be send from the encoder to the decoder. For this reason we also develop an efficient transform coding scheme based on the fixed Karhunen-Loeve Transform (KLT) so that the amount of side information is as small as possible.

The sinusoidal subcoder assumes a simple model for the perceptual weighted audio signal given by

\[ \hat{x}[n] = \sum_{l=1}^{L} A_l \cos(\omega_l n + \phi_l) \]  

¹All correspondence should be directed to J. K. Nielsen, Dept. of Electronic Systems, Aalborg University, Fredrik Bajers Vej 7, DK-9220 Aalborg, Denmark, email: jkjaer@es.aau.dk, phone: +45 41 16 17 87.
²The work of M.G. Christensen was supported by the Parametric Audio Processing project, Danish Research Council for Technology and Production Sciences grant no. 274–06–0521
³The work of S.H. Jensen was partly supported by the Danish Technical Research Council, through the framework project 'Intelligent Sound', www.intelligentsound.org (STVF No. 26-04-0092)
where $A_l$, $\omega_l$ and $\phi_l$ are the amplitude, frequency and phase of the $l$’th sinusoid, respectively. The difference between the signal extracted by this model and the weighted audio signal is termed the weighted residual and given by

$$e[n] = x[n] - \hat{x}[n],$$

and a quantized version of it is the input to the WCLMS lossless encoder. In our implementation we have placed the sinusoidal subcoder between the pre- and post-filter in order to utilize the high time-resolution perceptual weighting in the pre- and post-filters, and it is depicted in figure 1. Also, since the pre-filter performs the perceptual weighting of the audio signal, this setup does not require the perceptual matching pursuit (PMP) [4] for frequency estimation in the sinusoidal subcoder but only the MP algorithm [5] thus reducing the computational complexity. The amplitude, phase and frequency of the sinusoids are quantized according to [6] which does not rely on the masking curve.

In the fixed KLT coding scheme of the masking curves a training database consisting of eight different songs of a total length of approximately 40 minutes was used to find the optimum 2-dimensional fixed KLT transform kernels. The masking curve was parametrized by the linear predictive (LPC) coefficients and represented by the line spectral frequency (LSF) coefficients for every audio signal segment of 4 ms. The training database was also used to design the entropy coded scalar quantizers for the quantization of the transform coefficients.

The evaluation of the audio system consisting of the sinusoidal subcoder and the WCLMS based residual coder was performed with rate-distortion (R-D) measurement in the perceptual domain between the pre- and post-filter. The R-D measurements were carried out for two different music pieces; one with tonal characteristics (a violin signal - Fig. 2) and one with more stochastic and transient behaviour (intro of live recording of Eric Clapton, Layla - Fig. 3). For both songs different setups for the bit allocation were tested: 1) all bits were allocated the sinusoidal subcoder, 2) all bits were allocated the WCLMS subcoder, and 3) the bits were distributed between the two subcoders with a fixed quantity for the sinusoidal subcoder (corresponding to 5, 12 and 25 sinusoids, respectively) and the rest for the WCLMS based residual subcoder. The results show that for low bit-rates, the sinusoidal subcoder in setup 1 is superior to the WCLMS subcoder in setup 2. For higher bit rates, however, the situation is reversed. For the different combinations of the two subcoders (setup 3) the results indicate that a combination will lower the perceptual distortion as compared to setup 1 for increasing bit rate.

For the coding of the masking curves, a test database consisting of approximately 84,000 LSF vectors of dimension $P = 15$ was used. The performance of the fixed KLT was evaluated against the discrete Cosine Transform (DCT) and pulse code modulation (PCM) by use of the log spectral distortion (LSD) measure which is often used for evaluation of speech coders [7]. The LSD measures the average mean square logarithmic distance between the original and reconstructed power spectral density (PSD) and is defined as [8]

$$D_{LS} = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \left(10 \log_{10} \frac{S(\omega)}{\hat{S}(\omega)}\right)^2 d\omega}$$

where $S(\omega)$ and $\hat{S}(\omega)$ are the original and reconstructed PSD, respectively, which in our setup corresponds to the original and reconstructed masking curve. The LSD was computed for different bit rates for each LSF vector in our test database and the sample mean and sample variance for the fixed KLT, the DCT and PCM operating on a block of $M = 10$ consecutive LSF vectors were calculated. Figure 4 shows a plot of the measured values. Clearly, the variances of PCM and the fixed KLT is significantly lower than for the DCT. For the mean value, the fixed KLT is superior to the DCT and PCM, and the DCT performs worse than PCM for average bit rates above 1 bit. It is seen that an average LSD of 0.50 dB for the fixed KLT resulted in an average bit rate of 0.75 bits per transform coefficient. With a frame length of 4 ms and $P = 15$ this amounts to a bit rate of 2812.5 bits/s which is significantly lower than 7 kbit/s to 10 kbit/s obtained in [3] by use of vector quantization.

Our work indicates that the combination of a sinusoidal subcoder with the WCLMS based residual subcoder leads to an interesting interpretation of the optimality with respect to perceptual distortion.
of the two subcoders at different bit rates. Further work should seek to perform an R-D optimal bit allocation between the two subcoders for a given bit rate such that the overall perceptual distortion is minimized.

References


Figure 1: Block diagram of the sinusoidal subcoder integrated in the pre- and post-filtering setup where the WCLMS predictor acts as a waveform approximating residual subcoder.

Figure 2: Measured average rate-distortion of sinusoidal coding, WCLMS coding, and sinusoidal and WCLMS coded residual coding of a violin signal. The sinusoidal and WCLMS coded residual coding is performed for 6, 12 and 25 number of sinusoids.
Figure 3: Measured average rate-distortion of sinusoidal coding, WCLMS coding, and sinusoidal and WCLMS coded residual coding of a mixed signal containing different instruments. The sinusoidal and WCLMS coded residual coding is performed for 6, 12 and 25 number of sinusoids.

Figure 4: Mean (solid) and variance (dashed) of LSD for each LSF vector in the test database for different measured average bit rates.
Article

- Waveform Approximating Residual Audio Coding with Perceptual Pre-and Post-Filtering -
ABSTRACT

We investigate waveform approximating residual coding for a sinusoidal parametric audio coder at low bit rates. The residual coding is based on the well-known pre- and post-filtering method with lossless coding [1] which features perceptual weighting for short time segments at a low computational complexity. We compare the incurred perceptual distortion from joint quantization of the residual and the sinusoids for different bit rates. In addition to that we develop a transform coding scheme for the coefficients in the pre- and post-filters which must be send as side information between the encoder and decoder. Our investigations show that the combination of the sinusoidal subcoder and the pre- and post-filtering entails an overall lower perceptual distortion for low as well as high bit rates. Also, the developed transform coding scheme enables efficient coding of the side information at a very low bit rate.

Index Terms— Perceptual audio coding, residual coding, pre- and post-filtering, sinusoidal parametric coding

1. INTRODUCTION

The reduction of the bit rate for a given fidelity in audio coders has been subject to comprehensive research in the past few decades. This has led to a variety of audio coders of which MPEG-1 layer 3 (MP3) and MPEG-2/4 Advanced Audio Coding (AAC) [2] are the most widespread. These audio coders can typically achieve CD-quality at bit rates of 96 kbit/s and 64 kbit/s for a mono signal [1], respectively, whereas the standard pulse code modulation (PCM) entails a bit rate of 705.1 kbit/s for a mono signal with 16 bit/sample and a sample rate of 44.1 kHz. The large compression factor is achieved by use of perceptual audio coding which comprises irrelevance and redundancy removal. Irrelevance removal is a lossy process in which signal components, that are inaudible to the human ear, are discarded, and redundancy removal is a lossless process that removes statistical dependencies within the signal. In order to determine whether a signal component is audible or not, a psycho-acoustical model is used to derive a masked threshold below which signal components are inaudible. This threshold depends on the time, frequency and amplitude characteristics of the audio signal [3].

Traditional audio coders use subband coding and/or transform coding (see e.g. [4, 5, 6, 7]) in which an audio signal in the encoder is transformed into a perceptual domain where quantization according to a derived masking curve is performed succeeded by lossless coding. In the decoder, the inverse transform is applied and this produces the reconstructed audio signal. For very low bit rates, subband coding and transform coding are, however, not optimal for some audio signals for which reason parametric audio coding has been used as an alternative in the recent years [8, 9]. In parametric audio coding a model of the audio signal is assumed, and the encoding process thus reduces to an estimation of the parameters in the assumed model. A very popular parametric model is the sinusoidal model which recently has been standardized as MPEG-4 HILN (harmonics and individual lines plus noise) [10]. HILN consists like most other parametric coders of a sinusoidal subcoder and a residual or noise subcoder where the latter codes the remaining audio signal which cannot be extracted by the sinusoidal subcoder. Residual subcoders are typically divided into non-waveform and waveform approximating coders. The non-waveform approximating subcoders are typically based on stochastic modelling of the residual and perform well at low bit rates. The audio quality, however, does not in general increase with increasing bit rate which is in contrast to the waveform approximating coders whose main drawback is poor performance at low bit rates.

In this paper, we investigate a computationally efficient waveform approximating residual subcoder for low bit rates. The subcoder is based on the pre- and post-filtering method [1, 11] which features perceptual weighting for very short time segments and de-correlation with the weighted cascade least-mean-square (WCLMS) prediction. In the pre- and post-filtering method, a pre-filter adapts its frequency response to the inverse of the masking curve thus mapping the audio signal to a perceptual domain in which irrelevance reduction can be performed in a straight-forward manner. The inverse
filtering is performed by the post-filter whose frequency response equals the masking curve. The adaption of the pre-and post-filter to the masking curve requires side information to be send from the encoder to the decoder. In this paper, an efficient encoding scheme is also proposed based on transform coding with the fixed Karhunen-Loeve Transform (KLT) whose performance is compared to the Discrete Cosine Transform (DCT) and pulse code modulation (PCM).

The paper is organized as follows. In Section 2, we restate the pre- and post-filtering method, sinusoidal audio coding and 2-dimensional transform coding. Based on this, our implementation of the sinusoidal subcoder and the pre- and post-filter residual subcoder is given in Section 3 along with a description of the implementation of the transform coder of the masking curves. In Section 4 the results are presented while Section 5 concludes this paper.

## 2. FUNDAMENTALS

In this section, we restate the basic theory behind the pre- and post-filtering method, sinusoidal audio coding and transform coding of masking curves.

### 2.1. Pre- and Post-Filtering

The overall pre- and post-filtering system consists of an encoder and a decoder as depicted in Fig. 1. The irrelevance removal is performed by an adaptive psycho-acoustical controlled pre-filter, whose frequency response is the inverse of the masking curve, and a quantizer. The irrelevance removal is performed by a lossless coder based on weighted cascade least-mean-square (WCLMS) prediction. The decoder consists of a lossless decoder, an inverse quantizer and the post-filter whose frequency response is the inverse of the pre-filter and, hence, equals the masking curve. The masking curve obtained from the psycho-acoustic model is parametrized using linear predictive coding (LPC) and the resulting LPC coefficients are used in the pre- and post-filters. The masking curves, and thus the LPC coefficients, are updated every 2 ms to 4 ms. Since a direct switch between old and new filter coefficients leads to audible artifacts [1, 11], interpolation between the coefficients is necessary. The interpolation requirement, however, introduces stability issues in the post-filter since LPC coefficients are not suitable for interpolation. Therefore, the LPC coefficients are converted into either the line spectral frequency (LSF) coefficient representation [12] or the reflection coefficient (PARCOR) representation [13] which are both amenable to interpolation.

The operation of the post-filter requires the frequency response of the pre-filter to be coded and send as side information between the encoder and decoder. In [11] this is done by use of vector quantization of the LSF coefficients which results in bit rates from 7 kbit/s to 10 kbits/s.

### 2.2. Sinusoidal Audio Coding

In this paper, we consider the following sinusoidal model of order \( L \) for a time frame \( n = 0, 1, \ldots, N - 1 \) of an audio signal \( x[n] \)

\[
\hat{x}[n] = \sum_{l=1}^{L} A_l \cos(\omega_l n + \phi_l) \tag{1}
\]

where \( A_l, \omega_l \) and \( \phi_l \) are the amplitude, frequency and phase of the \( l \)th sinusoid, respectively. The difference between the signal extracted by this model and the actual audio signal is termed the residual and is given by

\[
e[n] = x[n] - \hat{x}[n]. \tag{2}
\]

For each time frame, which might overlap the previous, the parameters of Eq. (1) are estimated by use of some suitable estimator as for example the perceptual matching pursuit (PMP) [14] which iteratively seeks to minimize a perceptual norm based on this residual. The sinusoidal model in Eq. (1) is effective for coding stationary tonal signals, but it entails a lot of problems in coding non-stationary and transient signals. For this reason many extensions to the basic sinusoidal model have been proposed which among others comprise adaptive segmentation [15] and amplitude modulation [16]. In this paper, however, we will use the simple model in Eq. (1).

*Fig. 1. Block diagram of the pre- and post-filter audio coding system [1].*
2.3. 2-D Transform Coding

In 2-dimensional transform coding, the matrix $X$ of size $P \times M$ is transformed into the matrix $Y$ of size $P \times M$ with

$$Y = T_P X T_M^H$$

(3)

where $(\cdot)^H$ denotes the complex transpose, and $T_P$ and $T_M$ are separable orthonormal transform kernels of size $P \times P$ and $M \times M$, respectively. The inverse 2-D transform is given by

$$X = T_P^H Y T_M.$$

(4)

The main motivation behind transform coding is that, for a suitable pair of transform kernels, the quantization of the transform coefficients in $Y$ leads to a smaller overall distortion as compared to direct quantization of $X$ for the same bit rate. It can be shown that the optimum transform for a Gaussian source is the Karhunen-Loeve Transform (KLT) whose transform kernels are found from the eigenvalue decomposition in horizontal and vertical direction of $X$ given by

$$R_v = T_P^H \Lambda_v T_P ; \quad R_h = T_M^H \Lambda_h T_M$$

(5)

provided that the autocorrelation function of $X$ is separable in horizontal and vertical direction [17]. The main drawback of the KLT is that it depends on the input statistics. For this reason other suboptimal, but fixed, transforms as the discrete Cosine Transform (DCT) and the discrete Fourier Transform (DFT) have been suggested. Another advantage of these transforms is that they can be implemented in an effective way using the FFT.

For quantization of the transform coefficient entropy coded scalar quantization is considered since it fits well in the sinusoidal coder and pre- and post filtering setup. It can be shown analytically that the uniform quantizer under high resolution approximations is the optimal quantizer for entropy coded scalar quantization with an average rate only 0.255 bits from the Shannon lower bound [18]. Empirical studies have also shown that the uniform quantizer is nearly optimal for low bit rates as well [18]. Under high resolution approximations the optimal step size $\Delta$ of all the uniform quantizers are equal and given by [19]

$$\Delta = 2^{-R} \left[ \prod_{i=1}^{P} \prod_{j=1}^{M} 2^{h(y_{ij})} \right]^{\frac{1}{P^2}}$$

(6)

where $R$ is the desired average bit rate and $h(y_{ij})$ denotes the differential entropy of the $(i, j)$'th transform coefficient. The differential entropies for different probability density functions (pdf) are shown in Table 1.

<table>
<thead>
<tr>
<th>PDF</th>
<th>Differential Entropy, $h(y_{ij})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>$\frac{1}{2} \log_2 2 \pi e \sigma_{y_{ij}}^2$</td>
</tr>
<tr>
<td>Uniform</td>
<td>$\frac{1}{2} \log_2 12 \sigma_{y_{ij}}^2$</td>
</tr>
<tr>
<td>Laplace</td>
<td>$\frac{1}{2} \log_2 2 \pi e \sigma_{y_{ij}}^2$</td>
</tr>
</tbody>
</table>

Table 1. Differential entropies for random variables of different pdfs [20].

3.1. Sinusoidal Coding with Residual Coding based on Pre- and Post-filtering

The coding system considered in this paper is based on the simple sinusoidal subcoder implementing the signal model in Eq. (1) and the pre- and post-filter setup depicted in Fig. 1 acting as waveform approximating residual subcoder. By using a waveform approximating residual subcoder it is ensured that the audio quality converges to the quality of the original signal for increasing bit rate. The pre- and post-filter setup has also been shown to have a good performance at low bit rates [1].

We considered two different solutions for the structural composition of the sinusoidal and residual coding system. The first and most intuitive option was to place the sinusoidal subcoder in front of the pre- and post-filters of the residual subcoder. By doing so the masking curve has to be calculated twice since it is needed in both the PMP algorithm of the sinusoidal subcoder and for determining the filter coefficients for the pre- and post-filters. Furthermore, the masking curves used in the sinusoidal subcoder is updated more slowly than the masking curve used in the pre- and post-filters, because of the different window lengths. In the second option, the sinusoidal subcoder was moved in between the pre- and post-filters. The signal at the input to the sinusoidal subcoder is then perceptually weighted by the pre-filter thus saving the separate perceptual weighting in the sinusoidal subcoder and increasing the time resolution of the masking. While obtaining these improvements compared to the first coder structure, this structure suffers from the sub-optimality introduced by the LPC representation of the frequency response of the pre- and post-filters. The performance of these two different structural compositions was evaluated (see section 4.1) using informal subjective listening tests, which revealed no noticeable difference in the audio quality. Therefore, it was chosen to use the coding structure with the sinusoidal subcoder positioned between the pre- and post-filters, and it is depicted in Fig. 2. The coding of the residual was implemented by applying the WCLMS based subcoder presented in [1] on the residual from...
3.2. Transform Coding of Masking Curves

Efficient coding of the LSF coefficients is widely studied in the field of speech coding, but has not been treated in great detail for coding of the masking curve. The studies in speech coding comprise among others the statistical properties of the LSF coefficients [21, 22] and vector predictive quantization [23, 24] and transform coding [25, 26] of the LSF coefficients. They show that LSF coefficients are highly correlated in the same frame and between frames and that the distributions of the LSF coefficient resemble skewed Gaussian and Laplace distributions. In our studies, the LSF coefficients describe the masking curve, but they seem to have the same statistical properties as the LSF coefficients in speech coding. The normalized histograms for the LSF coefficients are shown in figure 3. These results were found from an analysis of a music training database consisting of eight different songs of a total length of approximately 40 minutes. An LSF column vector of dimension \( P = 15 \) was computed for every music frame of 4 ms which led to a training database of approximately 600,000 LSF vectors.

In our coding scheme, transform coding is used as opposed to vector quantization in [11]. The motivation behind this choice is that transform coding enables the use of simple scalar quantizers. We use and compare the performance of fixed KLT, DCT and PCM where fixed KLT refers to that the transform kernels are found from the training database and fixed during coding. The PCM uses simply the identity matrix as transform kernels. For quantization the entropy coded scalar quantizers are used, and the step size \( \Delta \) is found from Eq. (6) with the Laplace distribution as the model for the transform coefficients. The variances of the transform coefficients were found from the training database. Another music database consisting of approximately 84,000 LSF vectors was used for testing.

4. EXPERIMENTAL RESULTS

The following section describes the results obtained from the measurements of the sinusoidal subcoder combined with the WCLMS based residual subcoder. Also, the evaluation of the transform coding of the LSF coefficients is described.

4.1. Sinusoidal Coding with Residual Coding based on Pre- and Post-filtering

This section describes two different experiments. First, an evaluation of the subcoder performance with respect to the position of the sinusoidal subcoder is carried out. Second, an evaluation of the performance of the sinusoidal subcoder with and without the WCLMS acting as residual subcoder is carried out, and it is compared to the performance of the original pre- and post-filtering setup.

The influence of the sinusoidal subcoder position was evaluated through subjective listening tests. Two different coding system was implemented based on the two structural options mentioned in section 3.1. The experimental setup was as follows. For the sinusoidal coding 32 ms overlapping von Hann windowed time segments were used, and 4 ms time segments for the pre- and post-filtering. The number of sinusoids was set to 30 in the sinusoidal subcoder. Further, the phases of the sinusoids were quantized uniformly using 5 bits while the amplitudes and frequencies were quantized in the logarithmic domain using step-sizes of 0.161 and 0.003, respectively, as proposed in [16]. The pre- and post-filters were implemented using a lattice structure, i.e. PARCOR coefficients. To avoid audible artifacts in the coded signals due to rapidly changing filter coefficients, linear interpolation was applied on the PARCOR coefficients for each input sample. For this experiment, the gain-factor in the pre- and post-filters were held constant since applying a varying gain-factor led to audible artifacts during transient signal periods. With the total bit-rate fixed to around 100 kbit/s in order to compensate for the constant gain-factor, we performed subjective listening tests of the two setups applied on a violin signal and a mixed signal (intro of live recording of Eric Clapton, Layla).
Fig. 3. Normalized histogram for the LSF coefficients. The histograms are computed from the training database consisting of approx. 600,000 LSF vectors, and the normalization is performed for each histogram such that the maximum value is 1.

Fig. 4. Measured average rate-distortion of sinusoidal coding, WCLMS coding, and sinusoidal and WCLMS coded residual coding of a violin signal. The sinusoidal and WCLMS coded residual coding is performed for 6, 12 and 25 number of sinusoids.

The tests showed that no noticeable difference was observed between the coded signals for which reason the sinusoidal subcoder was implemented between the pre- and post-filters in order to reduce the computational complexity.

The evaluation of the sinusoidal subcoder in combination with the WCLMS based coding of the residual was performed by means of rate-distortion measurements. In these measurements the pre- and post-filters were extended by a gain-factor equal to the variance of the input signal as proposed in [11] and, to reduce the effect of rapidly fluctuating variances of consecutive input blocks, the applied gain factor was found through linear interpolation. These measurements were carried out for coding of two types of signals, a violin signal and a mixed signal, for three different types of setups: 1) All bits were allocated the sinusoidal subcoder, 2) all bits were allocated the WCLMS subcoder and 3) the bits were distributed between the two subcoders with fixed quantities for the sinusoidal subcoder (corresponding to 5, 12 and 25 sinusoids, respectively) and the rest for the WCLMS based residual subcoder.

The different setups were first applied to a violin signal and the results are depicted in Fig 4. As it can be seen, allocating all bits to the sinusoidal subcoder or distributing the bits between the two subcoder gives a lower distortion for low bit rates compared to allocating all bits to the WCLMS subcoder. For higher bit rates, however, allocating all bits to the WCLMS subcoder gives the lowest distortion. The different setups were applied on a mixed signal as well containing different musical instruments. The results from these measurements are depicted in Fig. 5. As it can be seen the same characteristics are obtained.

Generally, the performance of the sinusoidal and waveform approximating residual coding system depends on the number of sinusoids and the desired bit rate. For low bit rates, the sinusoidal subcoder should be allocated more bits than the WCLMS based residual subcoder since it results in an overall lower perceptual distortion. As the bit rate increases, the WCLMS based residual subcoder should be allocated more and more bits while the bit rate of the sinusoidal coder should be decreased. This bit allocation should, in contrast to our simple illustrative implementation, be implemented in an R-D optimal way where, for each bit rate, the bit allocation be-
Average MSE per transform coefficient

Average rate per transform coefficient

Fig. 6. Measured average rate-distortion pairs for each transform coefficient of the fixed KLT, the DCT and PCM. The measurements were performed on a test database of approx. 84,000 LSF vectors.

4.2. Transform Coding of Masking Curves

The evaluation of the transform coding scheme of the masking curves is performed by use of two tests: 1) rate-distortion measurements (R-D) for the fixed KLT, the DCT and PCM and 2) log spectral density measurements for the fixed KLT, the DCT and PCM.

Figure 6 depicts the measured R-D points for the fixed KLT, the DCT and PCM operating on a block of $M = 10$ consecutive LSF column vectors of size $P = 15$. The distortion is the mean squared error (MSE) between the unquantized and quantized transform coefficient, and the average rate is estimated as

$$\hat{R} = \frac{1}{PM} \sum_{i=1}^{P} \sum_{j=1}^{M} \left[ \frac{A_{ij}}{N} \sum_{k=1}^{c_{kij}} \log_2 \frac{c_{kij}}{N} \right]$$

(7)

where $A_{ij}$ is the cardinality of the alphabet $A_{ij}$ representing the $(i, j)$th transform coefficient in the $N$ blocks. The quantity $c_{kij}$ denotes the cardinality of the $k$th symbol in the alphabet $A_{ij}$. Figure 6 shows that the fixed KLT is superior to PCM and the DCT while the DCT performs worse than PCM for average bit rates above 1 bit.

Since the MSE distortion measure does not in general correspond to subjective measures, the system performance was also tested using the log spectral distortion (LSD) measure which is often used for evaluation of speech coders [25]. The LSD measures the average mean square logarithmic distance between the original and reconstructed power spectral density (PSD) and is defined as [27]

$$D_{LS} = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \left[ 10 \log_{10} \frac{S(\omega)}{\hat{S}(\omega)} \right]^2 d\omega}$$

(8)

where $S(\omega)$ and $\hat{S}(\omega)$ are the original and reconstructed PSD, respectively, which in our setup corresponds to the original and reconstructed masking curve. The LSD was computed for each LSF vector in our test database and the sample mean and sample variance for fixed KLT, the DCT and PCM for different bit rates were calculated. Figure 7 shows a plot of the measured values. Clearly, the variances of PCM and the fixed KLT is significantly lower than for the DCT while the mean values resembles the observed pattern of the objective distortion measure in figure 6. That is, the fixed KLT is superior to the DCT and PCM, and the DCT performs worse than PCM for average bit rates above 1 bit. Figure 8 shows the worst fit in terms of LSD of an original and reconstructed masking curve with an LSD of 1.27 dB for the fixed KLT with an average rate of 1 bit.

It is interesting to note that an average LSD of 0.50 dB for the fixed KLT resulted in an average bit rate of 0.75 bits. With a frame length of 4 ms and $P = 15$ this amounts to a bit rate of 2812.5 bits/s which is significantly lower than 7 kbit/s to 10 kbit/s obtained in [11] by use of vector quantization. In speech coding, an LSD of 1 dB is typically considered as a limit of perceptual significance [25]. If this is also true for
obtained at a bit rate of 2.8 kbit/s.

Applying entropy coded scalar quantization of the transform post-filters could be coded in an efficient way. It was shown that a gain-factor should be investigated such that transientsig-

Further work should aim to implement the rate-distortion optimal bit allocation between the sinusoidal subcoder and the WCLMS based subcoder. Also, the issue with the gain-factor should be investigated such that transient signals sounds less distorted, and listening tests should be performed.

5. CONCLUSION

In this paper we focused on two topics. First, we investigated the combination of a simple sinusoidal subcoder and a computationally efficient waveform approximating residual subcoder for low bit rates based on pre- and post-filtering. The results showed that it was possible to use the sinusoidal subcoder with the chosen residual subcoder to reduce the perceptual distortion at low bit rates, whereas, for higher bit rates the perceptual distortion was lowest using only a WCLMS based subcoder, which is a part of the chosen residual subcoder system. In order to keep the distortion as low as possible for all bit rates, the two coding structures should be allocated bits jointly in and rate-distortion optimal way. Second, it was investigated how the LPC coefficients for the pre- and post-filters could be coded in an efficient way. It was shown that by applying transform coding using the fixed KLT and by applying entropy coded scalar quantization of the transform coefficients, the MSE as well as the LSD could be improved compared to using the DCT and PCM. For a block length of 4 ms and filter order of 15 an average LSD of 0.5 dB could be obtained at a bit rate of 2.8 kbit/s.

Further work should aim to implement the rate-distortion optimal bit allocation between the sinusoidal subcoder and the WCLMS based residual subcoder. Also, the issue with the gain-factor should be investigated such that transient signals sounds less distorted, and listening tests should be performed.

6. REFERENCES


[16] M.G. Christensen and S. van de Par, “Efficient parametric codd-


---

**Fig. 8.** Worst fit in terms of LSD for the fixed KLT with an average rate of 1 bit. The LSD for the shown plot is 1.27 dB.


Worksheets

- Perceptual Coding of Audio Residues -
# Table of Contents

1. **Psychoacoustics for Audio Coding**
   - 1.1 The Physics of the Human Ear ........................................ 1
   - 1.2 Hearing Range and Threshold ........................................... 2
   - 1.3 Masking ........................................................................... 3
   - Bibliography ........................................................................... 6

2. **General Perceptual Audio Coding**
   - 2.1 Overview of a Generic Perceptual Audio Coder ....................... 9
   - 2.2 Audio Coding Types ........................................................ 10
   - Bibliography ........................................................................... 14

3. **Sinusoidal Coding with Pre- and Post-Filtering based Residual Coding**
   - 3.1 Overview of the Coding System ........................................... 15
   - 3.2 Sinusoidal Coding ............................................................ 16
   - 3.3 Residual Coding ............................................................... 17
   - 3.4 Implementation and Evaluations of the Coding System .......... 26
   - Bibliography ........................................................................... 27

4. **Transform Coding of Line Spectral Frequency Coefficients**
   - 4.1 Line Spectral Frequency Coefficients .................................. 31
   - 4.2 Transform Coding ............................................................. 36
   - 4.3 Entropy Coded Quantization of Transform Coefficients ........... 38
   - 4.4 Application: Transform Coding of Line Spectral Frequency Coefficients ................................................................. 42
   - 4.5 Conclusion ....................................................................... 51
   - Bibliography ........................................................................... 55
Psychoacoustics for Audio Coding

Lossy audio coding algorithms achieve a high compression ratio from a good understanding of the human auditory system. This enables the algorithms to discard nonaudible information from the source signal and place quantization noise within the signal without introducing audible distortion. Psychoacoustics is the science that deals with the human auditory system, and an overview of the main concepts is given in this worksheet.

1.1 The Physics of the Human Ear

The human ear is shown in figure 1.1 and it consists of three major parts: The outer, middle and inner ear. These parts transform pressure variations in the air (the outer ear) into mechanical motion (the middle ear) which create pressure variations in a fluid that surrounds sensory cells (the inner ear) [Zwicker and Fastl, 1999, pp. 23-28]. The sensory cells produce electric signals which enter the brain through the auditory nerve.

Two parts of the human ear are of special interest from an audio coding perspective: The ear canal in the outer ear and the cochlea in the inner ear. The ear canal has a length of approximately 20 mm which corresponds to a resonant mode in air at around 4 kHz [Bosi and Goldberg, 2003, p. 169]. For this reason the human ear is most sensitive to frequencies around 4 kHz which is discussed in connection with the hearing range and threshold in the next section. The cochlea in the inner ear is vital to the understanding of masking. Masking is a phenomenon that refers to the fact that some sounds are inaudible in the presence of other sounds and is described in a later section in this worksheet. The cochlea is a coiled tube with a length of approximately 32 mm and consists of three channels that run from the base to the apex of the cochlea [Zwicker and Fastl, 1999, p. 25]. Two of the channels are only separated by a very thin membrane for which reason they can be regarded as one unit from a hydro-mechanical point of view. The last channel is separated from the other channels by the basilar membrane which extends from the base of the cochlea and almost until the apex where a small window enables direct contact between the fluids in the channels [Bosi and Goldberg, 2003, p. 170].
The basilar membrane oscillates in response to the pressure variations in the fluid in the cochlea. The location of the maximum of the oscillations depends on the frequency of the pressure variations. It has been shown that the basilar membrane works as a spectrum analyzer that maps a frequency to a location on the basilar membrane [Bosi and Goldberg, 2003, p. 172]. This mapping is illustrated in figure 1.2. Notice that the maximum is located near the apex for low frequencies and near the base for high frequencies. The oscillation of the basilar membrane is detected by hair cells attached to the membrane, and this creates electrical impulses to the brain.

1.2 Hearing Range and Threshold

The human auditory system is capable of sensing sound pressures at different levels and at different frequencies. At best, the human ear can detect sound pressures in the interval from around 20 $\mu$Pa to around 100 Pa. This wide range is often described in terms of
1.3. Masking

Figure 1.2: Frequency to location mapping along the basilar membrane [Plack, 2003].

sound pressure level defined by [Bosi and Goldberg, 2003, p. 150]

\[
\text{SPL} = 20 \log_{10} \frac{p}{p_0} \quad \text{[dB]}
\] (1.1)

where \( p_0 = 20 \mu \text{Pa} \). The human hearing range is in terms of sound pressure level in the interval from 0 dB to around 135 dB.

The sensitivity of the ear with respect to sound pressure level depends on the frequency and changes with the sound pressure level. The audible frequency range is generally considered to be in the interval from 20 Hz to 20 kHz as shown in figure 1.3 with the interval from 3 kHz to 5 kHz being the most sensitive range as discussed in the previous section [Bosi and Goldberg, 2003, p. 151].

The curves in figure 1.3 describes the subjective response to different sound pressure levels with the bottom line marked with '0' corresponding to the threshold of hearing. The curves define the phon scale which by definition coincides with the sound pressure level scale at 1 kHz [Watkinson, 2002, p. 32]. Therefore, they illustrate the subjective sound pressure level at different frequencies perceived as being of equal amplitude to the sound pressure level at 1 kHz.

1.3 Masking

Masking is the most important concept in lossy audio coding, and it arises from the function of the basilar membrane described in section 1.1. Masking utilizes the fact that some sound are inaudible in the presence of other sounds. The masking phenomenon is typically divided into two different types: Simultaneous masking and temporal masking. Simultaneous masking occurs in the frequency domain while temporal masking occurs in the time domain. The next two subsections will give an introduction to both of them.
1 Psychoacoustics for Audio Coding

1.3.1 Simultaneous Masking

Recall from section 1.1 that the cochlea performs a frequency to location mapping along the basilar membrane. This mapping is a result of the location specific oscillations of the basilar membrane in response to pressure variations in the surrounding fluid inside the cochlea. The envelopes of these oscillations for three different sinusoidal stimuli are shown in figure 1.4. Notice that the base of the cochlea is to the right and the apex is to the left.

Figure 1.4: Envelope of basilar membrane oscillations in response to sinusoidal stimuli at 400 Hz, 1600 Hz and 6400 Hz [Painter et al. 2007, p. 116].

The figure shows that the basilar membrane oscillates in a certain interval around the
1.3. Masking

stimulating frequency. A small interval around the stimulating frequency is referred to as the critical band, and it is very important in the design of audio coders [Painter et al., 2007, p. 116]. The critical band is asymmetric since it decreases much faster for frequencies lower than the stimulating frequency as compared to frequencies above the stimulating frequency. Additionally, the critical band is level and frequency dependent. The frequency dependency is often described by the analytical expression [Zwicker and Fastl, 1999, p. 164]

\[
\Delta f_c = \left[25 + 75 \cdot (1 + 1.4 \cdot 10^{-6} \cdot f^2)^{0.64}\right] \text{ [Hz]}
\]  

(1.2)

which is plotted in figure 1.5. From this it is seen that the critical bandwidth is approximately 100 Hz for frequencies lower than 500 Hz and approximately 0.2 times the center frequency for higher frequencies.

![Figure 1.5: Critical bandwidth as a function of stimulating frequency.](image)

Audio coders typically employ a filter bank interpretation of the critical bands where the frequency range is divided into a discrete set of bandpass filters. The Bark scale is a nonlinear mapping from frequency in Hertz into Barks and is often used to form these filters. The mapping is given by [Zwicker and Fastl, 1999, p. 164]

\[
z_b = \left[13 \cdot \arctan(0.00076 \cdot f) + 3.5 \cdot \arctan\left(\left(\frac{f}{7500}\right)^2\right)\right] \text{ [Bark]}
\]

(1.3)

and is plotted in figure 1.6. A critical bandwidth is thus one Bark.

Different kinds of simultaneous masking with different properties can occur. Traditionally three types are considered [Painter et al., 2007, pp. 121-125]: Noise-masking-tone (NMT), tone-masking-noise (TMN) and noise-masking-noise (NMN) where the last two
Figure 1.6: Critical band rate measured in Bark as a function of stimulating frequency.

types are important since they allow quantization noise to be masked. Figure 1.7(a) and figure 1.7(b) show examples of NMT and TMN, respectively. In NMT narrowband noise with bandwidth of one Bark masks a tone, in TMN a tone masks narrowband noise, and in NMN narrowband noise masks narrowband noise. The difference in sound pressure level (SPL) between the masker sound and the masked sound depends on the type of masking, the SPL, and the frequency of the tone (if any) with respect to the center frequency of the critical band in which the masking occurs. In NMT the tone will be masked if its SPL is more than approximately 4 dB lower than the SPL of the narrowband noise. This difference is considerably increased in the case of TMN and NMN where the difference in SPL must be greater than approximately 21 dB and 26 dB, respectively [Painter et al., 2007, pp. 123-124]. These differences are referred to as the minimum signal-to-mask ratio (SMR).

1.3.2 Temporal Masking

Besides simultaneous masking, which occurs in the frequency domain, another masking phenomenon exists in the time domain, and it is referred to as temporal (or nonsimultaneous) masking. As shown in figure 1.8, the sensitivity of the human ear decreases before and after the presence of a masker. This is referred to as pre- and postmasking with a duration of the significant part of approximately 1 ms - 2 ms and 50 ms - 300 ms, respectively [Painter et al., 2007, p. 127]. The physical reason behind temporal masking is the integration time of the ear [Bosi and Goldberg, 2003, p. 158]. It takes some time to build up the perception of a sound which explains the somewhat counter-intuitive phenomenon of premasking.
BIBLIOGRAPHY

Figure 1.7: Examples of the simultaneous masking phenomenon for noise and tones [Painter et al., 2007, p. 123].

Figure 1.8: Temporal masking [Painter et al., 2007, p. 127].

Bibliography


Bibliography


General Perceptual Audio Coding

This worksheet serves as an introduction to perceptual audio coding. First, an overview of general perceptual audio coding is given through a short description of the blocks featured in a generic perceptual audio coder. The rest of the worksheet gives a more in-depth description of different coding types.

2.1 Overview of a Generic Perceptual Audio Coder

Audio coding algorithms are used to obtain short digital representations of audio signals. Since the amount of information that has to be transmitted through the Internet increases and it is desired to store as much audio on a single storage media as possible, the necessity of such algorithms increases. The goal in audio coding is therefore, to present an audio signal with a minimum number of bits without decreasing the quality of the audio signal, i.e. the coding should be transparent. That is, a listener should not be able to distinguish between the original signal and the coded version of the signal.

To make it possible to obtain a high compression ratio while still having transparent coding, perceptual audio coding could be used, and here the idea is to exploit both perceptual irrelevancies and statistical redundancies. Throughout the years, these two subjects have been dealt with by basing the audio coding on the principles of psychoacoustics (see worksheet 1) and by removing statistical redundancies, respectively [Painter, Spanias, and Atti, 2007] [Bosi and Goldberg, 2003].

In the last few years several perceptual audio coding schemes and compression standards have been proposed and many of these can be represented by the generic block diagram depicted in Figure 2.1.

Following, a brief description of the block diagram of the generic perceptual audio coder is given. The first step is to divide the audio signal into segments, typically with a length of 2 ms to 50 ms [Painter et al., 2007, p. 4]. Then, the input signal is fed to the
2 General Perceptual Audio Coding

Figure 2.1: Block diagram of a generic perceptual audio coder.

The control of the perceptual distortion is achieved through a psychoacoustic analysis block. This block finds the masking threshold which is a measure of the maximum amount of distortion that can be allowed in each point in the time-frequency domain without introducing audible artifacts. By using the information given by these thresholds the quantization block is able to exploit perceptual irrelevancies. Furthermore, the quantization block can remove some of the statistical redundancies by using techniques such as differential pulse code modulation (DPCM) or adaptive differential pulse code modulation (ADPCM) [Watkinson, 2002, p. 5].

To remove the remaining statistical redundancies, the quantized parametric set is fed to the entropy coding block before it is send through a channel. Generally, the psychoacoustic features are very signal dependent for which reason most perceptual audio coding algorithms has to be of variable rate. The rest of the worksheet will give a more in-depth description of different coding types which can be used in the time-frequency analysis block of the generic perceptual audio coder.

2.2 Audio Coding Types

When it comes to the actual encoding of the audio signal several different methods exist. In general, these can be split into two categories: Source coders and numeric coders. In source coders the redundancy in the signal is removed by estimating a model of the source generation mechanism. Source coders can be both lossy and lossless but typically they will be lossy. Numeric coders, on the other hand, uses abstract numerical methods to
remove the redundancy and typically these are lossless coders. This section, will focus on
the three source coding types; transform coding, subband coding and sinusoidal coding.

2.2.1 Transform Coding

Transform coders are characterized by their use of an unitary transform, which could be
discrete Fourier transform (DFT), discrete cosine transform (DCT), etc. Many transform
coders use the DCT, since it converts the input signal into a form where redundancy can
be easily detected and removed [Watkinson, 2002, p. 86]. The DCT works on blocks of
windowed samples. Following, an example is given where the DCT is used on a small
block of eight samples. In practice several hundred samples might be used.

\[
\begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\end{array}
\]

Figure 2.2: Wave table for an eight-point DCT.

In Figure 2.2 a wave table for an eight-point DCT can be seen. The idea behind using
the DCT, is then, that the input signal can be represented by a linear combination of
the waves shown in the wave table, where the DCT coefficients determines the amount
of each wave to be added. The first coefficient will represent the DC component of the
signal whereas the the remaining coefficients will represent increasing frequencies. The
coded signal, given by the coefficients, could then be decoded by using an inverse DCT.

It should be remarked that the DCT itself gives no compression of the input signal,
since the number of coefficients at the output always equals the number of input sam-
pies. However, by exploiting that some coefficients are equal to zero or of small value a
coding gain can be achieved, by transmitting these low-valued coefficients with shorter
wordlength and the zero-valued coefficients need not to be transmitted at all.

2.2.2 Subband Coding

Subband coding resembles transform coding in many ways. Like in transform coding,
subband coding also exploits the signal redundancy and psychoacoustics irrelevancy in
the frequency domain. One thing that differ between the two coding types is, that trans-
form coding typically performs a high-resolution frequency analysis whereas subband
coders rely on a coarse division of the frequency spectrum.

Subband coding exploits the fact that real sounds do not have uniform spectral energy.
In pulse code modulation (PCM) the wordlength is based on the desired dynamic range
and is typically constant for all frequencies. Therefore, when a PCM scheme is used
on a signal with an uneven frequency spectrum only the loudest frequency components
would use the whole dynamic range whereas the other components would be coded with a
huge headroom. Instead, subband coding proposes a method where the whole frequency
range of interest is divided into smaller frequency bands. The wordlength for each band
is then adjusted, such that bands with only a little energy will have a short word length
assigned and vice versa. Thus, each band will have a variable wordlength and the sum
of the wordlengths of all bands will generally be smaller than that of PCM. Thereby, it
can be concluded that subband coding will entail a coding gain.

![Diagram](image)

**Figure 2.3:** Critical condition for a subband coding noise masking scheme.

The noise that can be masked in each band and thereby the compression that can be
obtained depends on the width of the subbands. The narrower the subbands are the more
noise can masked as depicted in Figure 2.3. But decreasing the width of the subbands
will, however, raise the complexity and the coding delay. Splitting the frequency range
into subbands is rather complex and requires a lot of computation. One typical way of
doing it is to use quadrature mirror filtering (QMF) which will not be described here.

### 2.2.3 Sinusoidal Coding

Sinusoidal coding, which is a parametric coding technique, has been used successfully
since the 1980s for speech coding and later its use was extended to audio coding as
well. One of the advantages of sinusoidal coding is that it can achieve high quality at
relatively small bit rates which is desirable in applications such as Internet streaming

---
The simplest form of sinusoidal coding assumes that the signal $s[n]$ to be coded, can be represented entirely by a sum of sinusoids

$$s[n] \approx \sum_{k=1}^{K} A_k \cos(\omega_k[n]n + \phi_k[n]) ,$$

where $A_k$ denotes the amplitude, $\omega_k[n]$ denotes the instantaneous frequency, and $\phi_k$ denotes the instantaneous phase of the $k$-th sinusoid. The goal of sinusoidal coding is to estimate the parameters (amplitudes, frequencies and phases) of the assumed model. It turns out that it is relatively easy to estimate the amplitude and the phase of the individual sinusoids whereas the estimation of the frequencies is rather complex. Typically, subspace-based methods or nonlinear least-squares (NLS) methods are used for this purpose. The decomposition into a deterministic signal part and a stochastic signal part is done sequentially. First, the sinusoids are extracted and their parameters are estimated and the remaining signal, also known as the residual, is coded using a residual coder. This procedure is depicted in Figure 2.4.
The coding of residuals can be done in several different ways. The problem with coding the residuals is that they are not well modeled by sinusoids. Even though this is the case, residuals have been modeled using sinusoids in speech coding. It turns out that this is an acceptable approach for narrowband signals, whereas, it performs bad when applied to wideband signals [Christensen, 2005, p. 24].

The different types of residual coders can be divided into two categories, namely waveform approximating and non-waveform approximating residual coders. The first group generally perform better at higher bit-rates since the reconstructed signal here may converge to the original signal, whereas, the other group, sometimes denoted noise coders, performs best at low bit-rates [Christensen, 2005, p. 24].

Bibliography


Sinusoidal Coding with Pre- and Post-Filtering based Residual Coding

The following worksheet serves as an introduction to the chosen coding system setup, which is constituted by a sinusoidal coder and a residual coder. First, an overview of the chosen coding system is given followed by a more detailed description of the sinusoidal coder and the residual coder, respectively. The worksheet concludes with simulations and subjective evaluations of the coding system.

### 3.1 Overview of the Coding System

The structure of the coding system is based on the block diagram shown in fig. 3.1 in worksheet 2. To give a more detailed view of the actual implementation of the coding structure the block diagram was extended as depicted in fig. 3.1.

![Detailed block diagram of the sinusoidal and residual coding system.](image)

Since, the main topic of this worksheet is the coding of the residual, the actual structure of the sinusoidal coder is not outlined on the same level of detail as the residual coder.
In the block diagram, seen in fig. 3.1 the residual coder is already outlined and it is constituted by the lossless encoder placed after the sinusoidal coder and the perceptual weighting of the input signal is performed by the pre- and post-filters. From the block diagram it can be seen that the sinusoidal coder is placed in between the pre- and post-filters. This has both advantages and disadvantages. As described later in this worksheet, the masking curve is needed to compute the filter coefficients of the pre- and post filters. This means that the masking curve has to be found twice if the pre-filter is placed after the sinusoidal coder but by placing it before this is avoided since the input to the sinusoidal coder is now already weighted.

Another advantage by doing so is that the input blocks to the pre- and post-filters are only 4 ms whereas the input segments to the sinusoidal coder are 30 ms long which means that the masking curve is updated more frequently. The cost of this is, though, that the spectral resolution of the masking curve is decreased and the frequency response of the pre- and post-filters will only be an approximation of the real masking curve since the filters are based on linear prediction.

In the following sections the coding system described by the block diagram in fig. 3.1 is described further, and the performance is evaluated.

### 3.2 Sinusoidal Coding

For the purpose of estimating the dominating sinusoids of the audio signal to be coded it was chosen to use a simple sinusoidal coder. Therefore, the signal model was assumed to be a sum of constant amplitude sinusoids and it can thereby be described by

\[
\hat{x}[n] = \sum_{l=1}^{L} A_l \cos(\omega_l n + \phi_l)
\]  

(3.1)

where \(A_l, \omega_l, \text{ and } \phi_l\) are the amplitude, frequency and phase of the \(l\)th sinusoid. The described model fits well on tonal and stationary segments whereas it is a poor approximation for transient and non-stationary signals, however, since the focus is on the coding of the residual it is chosen to use this simple sinusoidal coder. The parameters of the sinusoidal model is then estimated using the perceptual matching pursuit algorithm [Christensen and Jensen 2006] or just the matching pursuit algorithm if the input signal to the sinusoidal coder is already perceptually weighted as in fig. 3.1. The signal that is left after the sinusoidal coder can be described by

\[
e[n] = x[n] - \hat{x}[n]
\]  

(3.2)

and it is typically termed the residual.
3.3 Residual Coding

In this worksheet a pre- and post-filtering method combined with a lossless coder, for coding of the residual, is used. One of the critical pitfalls of traditional transform coding is that it needs to run in two modes - one for stationary signal parts and one for non-stationary signal parts. Another issue is that the traditional coding scheme typical introduces a long delay, because of the large number of subbands needed.

To improve this a pre- and post-filtering method is used, where the idea is to separate the irrelevancy and redundancy reduction [Schuller, Yu, Huang, and Edler, 2002]. The irrelevancies is reduced by the pre- and post-filters, which should have a frequency response that approximates the masking curve and the inverse of it, respectively. Secondly, the redundancies are reduced by using predictive coding which is shown to have the same asymptotic coding gain as transform coding [Nitadori, 1970]. The following section describes the reduction of the irrelevancies and the redundancies.

3.3.1 Irrelevancy Reduction using Pre- and Post-Filtering

As mentioned previously it is desired to reduce the irrelevance and the redundancy of the audio signals. The main goal, when trying to reduce the amount of irrelevance, is to shape the quantization noise such that it fits the perceptual masking threshold as good as possible.

The idea in the pre- and post-filtering setup is therefore, to filter the output signal of the decoder in a way such that the quantization noise gets shaped according to the masked threshold. This is fulfilled by the post-filter. But, if only the post-filter was applied the shape of the audio signal would be distorted, for which reason a pre-filter is needed. Therefore, to preserve the original shape of the audio signal, the frequency response of the pre-filter should approximate the inverse of the masked threshold.

Since both the pre- and post-filter are related to the masked threshold, they are controlled by a psychoacoustic model. This is done by adapting some of the theory from linear prediction.

Linear Prediction

Linear prediction deals with the problem of predicting a future sample of a process on basis of a given set of past samples of the same process. In other words the objective is to find a predicted value \( \hat{u}(n|U_{n-1}) \) of a sample \( u(n) \) where \( U_{n-1} \) spans the past samples \( u(n-1), u(n-2), \ldots, u(n-M) \). Using a linear transversal filter which is optimized in the mean squared error sense, the problem of predicting the future sample can be solved using [Haykin, 2001].
\[ \hat{u}(n|U_{n-1}) = \sum_{k=1}^{M} w_{f,k} u(n-k) \]  

(3.3)

where \( w_{f,k} \) is the \( k \)’th tap weight of the filter and \( M \) is the filter order. The prediction error is then defined as being the difference between the input sample and the predicted value of it

\[ f_M(n) = u(n) - \hat{u}(n|U_{n-1}). \]  

(3.4)

The resulting prediction error filter from equation 3.4 is depicted in fig. 3.2. Ideally no correlation will be present in the prediction error signal for which reason it can be transmitted more efficiently than the actual input signal itself. The decoder then has to apply the inverse filter, i.e. a filter which has a frequency response that approximates the input signal spectrum, to the prediction error such that the input signal is obtained again at the output. The inverse prediction filter is also depicted in fig. 3.2.

Figure 3.2: Block diagram of the a) prediction error filter and b) the inverse prediction error filter.

To find the coefficients of the optimum FIR predictor a set of equations is needed. For this purpose, the Wiener-Hopf equations may be adapted such that

\[ R w_f = r \]  

(3.5)

where \( R \) is the correlation matrix of the filter tap inputs and \( r \) is the cross-correlation vector between the tap inputs and the desired signal. Furthermore, the prediction-error power is defined as

\[ P_M = r(0) - r^T w_f \]  

(3.6)

where \( r(0) \) is the autocorrelation function of the input process for lag 0. Equation 3.5 and 3.6 are then combined into the matrix equation

\[
\begin{bmatrix} r(0) & r^T \end{bmatrix} \begin{bmatrix} 1 & \ & \ & \ & \end{bmatrix} = \begin{bmatrix} P_M \end{bmatrix}
\]

(3.7)
3.3. Residual Coding

The prediction error filter coefficients can then be found by inserting \( a_M = [1 \ - w_f]^T \) and solving for \( a_M \).

Pre- and Post-Filtering using Linear Prediction Theory

It is then desired to use the same filter structure as for the linear predictor, but instead of approximating the signal spectra, the masked thresholds obtained from the psychoacoustic model should be approximated. The psychoacoustic model describes the masked threshold as a power spectral density (PSD) function. To be able to use the same procedure as in linear prediction, the PSD function is converted to the corresponding autocorrelation function trough an inverse discrete Fourier transform (DFT)

\[
s_{mm}(n) = \mathcal{F}^{-1}\{M(\omega)\}.
\]  

(3.8)

It is then possible to find the coefficients of the pre- and post-filters using equation 3.7.

3.3.2 Interpolation of Filter Coefficients

As such, the filter coefficients of the pre- and post-filters are updated for each input frame. Therefore, the filter coefficients are constant over each frame which means that abrupt changes in the coefficients from one frame to another most likely will occur. Such transients in the coefficients can introduce distortion in the reproduced signal. To circumvent this issue the coefficients can be interpolated across frames [Kleijn and K.K.Paliwal, 1995].

One problem with interpolating the filter coefficients is that instability can occur in IIR filters. Therefore, the interpolation is rarely applied directly on the LP coefficients but rather in a transform domain where stability is guaranteed. Some well known transform parameters are log area ratio (LAR), line spectral pairs (LSP), reflection coefficients (PARCOR) and autocorrelation parameters. Interpolating between LSP’s performs better than interpolating between the other parameters but it has been shown that interpolating between reflection coefficients is sufficient for audio signals [Kleijn and K.K.Paliwal, 1995, Edler and Schuller, 2000].

Since reflection coefficient can be inserted directly into a lattice filter structure it is chosen to use such coefficients instead of LSP’s which have to be converted back to LP coefficients before they can be implemented into a direct form filter structure. Furthermore, it has been shown that simple linear interpolation for every input sample to the pre- and post-filter between such transform parameters eliminates the transition problems [Schuller et al., 2002].

Linear Interpolation

The general equation for linear interpolation is defined as
\[ y(i) = (1 - r_i)y_{\text{past}} + r_iy_{\text{future}} \]  
\[(3.9)\]

where \( r_i \) is the normalized distance from the past sample \( y_{\text{past}} \) to the sample \( y(i) \) to be interpolated. That is, \( r_i \) can be found as

\[ r_i = \frac{x(i) - x_{\text{past}}}{x_{\text{future}} - x_{\text{past}}} . \]  
\[(3.10)\]

When the linear interpolation is applied in the pre- and post-filters it is assumed that the true coefficients is located in the middle of each frame and the coefficients at the samples between those coefficients is then interpolated as shown in fig. 3.3.

![Figure 3.3: Values of the linear interpolation weights across neighbouring frames.](image)

### 3.3.3 Filtering Using a Lattice Structure

As mentioned in the previous section it is chosen to use reflection coefficients for the pre- and post-filters. To see how the lattice coefficients are found a short review of the FIR filter structure is given. The general equation of a FIR filter as shown in fig. 3.4 is

\[ y(n) = x(n) + \sum_{k=1}^{M} \alpha_{M,k} x(n - k) . \]  
\[(3.11)\]

![Figure 3.4: Block diagram of a \( M \)'th order FIR filter.](image)
It can be shown that such a filter is closely related with the topic of linear prediction, since the prediction error can be written as

\[ \hat{x}(n) = - \sum_{k=1}^{M} \alpha_{M,k} x(n-k) \]  

(3.12)

where \( \alpha_{M,k} = w_{f,k} \) and if eq. 3.12 is inserted in eq. 3.11 the general equation of linear prediction from eq. 3.4 is obtained.

\[
\sum_{z} z^{-1} x(n) e_0(n) = \sum_{z} z^{-1} \tilde{e}_0(n) \kappa_1 \\
\sum_{z} z^{-1} e_m(n) = \sum_{z} z^{-1} \tilde{e}_m(n) \kappa_m \\
\vdots \\
\sum_{z} z^{-1} e_M(n) = \sum_{z} z^{-1} \tilde{e}_M(n) \kappa_M \\
y(n) = \sum e_1(n) = y(n)
\]

Figure 3.5: Block diagram of a \( M \)th order lattice FIR filter.

The general set of equations for an \( M \)th-order lattice filter, as depicted in fig. 3.5, is then [Proakis and Manolakis 1996]

\[ e_0(n) = \tilde{e}_0(n) \]  

(3.13)

\[ e_m(n) = e_{m-1}(n) + \kappa_m \tilde{e}_{m-1}(n-1) \quad m = 1, 2, \ldots, M \]  

(3.14)

\[ \tilde{e}_m(n) = \kappa_m e_{m-1}(n) + \tilde{e}_{m-1}(n-1) \quad m = 1, 2, \ldots, M . \]  

(3.15)

To compare this filter structure with the direct form FIR filter structure two examples are given. First, an example for a first order FIR filter is given. The inputs to the first stage of the corresponding lattice filter are set to the same value as the input to the FIR filter such that

\[ e_0(n) = \tilde{e}_0(n) = x(n) \]  

(3.16)

and the output of the FIR filter is equated with one of the outputs of the lattice filter which leads to

\[ y(n) = e_1(n) \Leftrightarrow x(n) + \alpha_{1,1} x(n-1) = x(n) + \kappa_1 x(n-1) \Rightarrow \]

\[ \alpha_{1,1} = \kappa_1 . \]  

(3.17)

(3.18)

That is, for a first order lattice filter the reflection coefficient \( \kappa_1 \) is of the same value as the first coefficient \( \alpha_{1,1} \) of the FIR filter, if the outputs of the two filters are equated. Next, a second order FIR filter is considered. The output from the first stage from the corresponding lattice filter is


3 Sinusoidal Coding with Pre- and Post-Filtering based Residual Coding

\[ e_1(n) = x(n) + \kappa_1 x(n - 1) \] (3.19)
\[ \tilde{e}_1(n) = \kappa_1 x(n) + x(n - 1) \] (3.20)

and the output of interest from the second stage is then

\[ e_2(n) = e_1(n) + \kappa_2 \tilde{e}_1(n - 1) = x(n) + \kappa_1 x(n - 1) + \kappa_2 (\kappa_1 x(n - 1) + x(n - 2)) \] (3.21)
\[ = x(n) + \kappa_1 (1 + \kappa_2) x(n - 1) + \kappa_2 x(n - 2) . \] (3.22)

If \( e_2(n) \) is equated with the output \( y(n) \) from the corresponding second order FIR filter the following reflection coefficient values are obtained

\[ \kappa_2 = \alpha_{2,2}, \quad \kappa_1 = \frac{\alpha_{2,1}}{1 + \alpha_{2,2}} . \] (3.23)

As seen, the reflection coefficient of the last stage in the lattice filters equates the last coefficient of the corresponding FIR filter, which in general holds [Proakis and Manolakis, 1996].

As induced by the two examples the remaining coefficients are found by stepping down from the coefficient of the last stage. The general equation for calculating those remaining reflection coefficients is [Proakis and Manolakis, 1996]

\[ A_{m-1}(z) = \frac{A_m(z) - \kappa_m B_m(z)}{1 - \kappa_m^2} \quad m = M, M - 1, \ldots, 1 \] (3.24)

where

\[ A_m(z) = \frac{E_m(z)}{X(z)} = \frac{Z[e_m(n)]}{X(z)} \] (3.25)
\[ B_m(z) = \frac{\tilde{E}_m(z)}{X(z)} = \frac{Z[\tilde{e}_m(n)]}{X(z)} . \] (3.26)

The procedure is then to find \( \kappa_m = \alpha_{m,m} \) from eq. (3.24) for \( m = M, M - 1, \ldots, 1 \).

The lattice FIR filter structure is used in the pre-filter while the inverse of it is needed in the post-filter. Therefore, an all-pole IIR lattice filter, as depicted in fig. 3.6 should be used, which can be described by the following set of equations

\[ e_M(n) = x(n) \] (3.27)
\[ e_{m-1}(n) = e_m(n) - \kappa_m \tilde{e}_{m-1}(n - 1) \quad m = M, M - 1, \ldots, 1 \] (3.28)
\[ \tilde{e}_m(n) = \kappa_m e_{m-1}(n) + \tilde{e}_{m-1}(n - 1) \quad m = M, M - 1, \ldots, 1 \] (3.29)
\[ y(n) = e_0(n) = \tilde{e}_0(n) . \] (3.30)
3.3. Residual Coding

3.3.4 Coding and Quantization of Sinusoidal Coding Residual

The remaining signal, i.e. the residual, that is left after the estimated sinusoids are subtracted from the pre-filtered signal, will typically contain a considerable amount of correlation. Several methods exist for removing this remaining correlation, but here it is chosen to use weighted cascaded least-mean-squares (WCLMS) predictors followed by an entropy coder as it is a part of the original perceptual pre- and post-filtering setup [Schuller et al., 2002]. The WCLMS algorithm is based on the normalized least squares (NLMS) algorithm, for which reason a introduction to this algorithm is given first, before the WCLMS algorithm is described.

Normalized Least-Mean-Squares

The normalized least-mean-squares (NLMS) algorithm is an extension of the LMS algorithm, so a short introduction to the LMS algorithm is given. In fig. 3.7 a block diagram of a generic LMS filter is presented. As it can be seen, the LMS algorithm consists of two steps. The first step is a filter, which generates an output of a linear filter from an input signal and it computes an error signal which is the difference between the filter output and the desired response. The second step of the algorithm is an update of the filter coefficients which is done such that the gradient of the LMS cost-function converges to zero [Haykin, 2001].

Without going further into the theory behind the LMS algorithm, it is summarized by the following mathematical expressions

\[
e(n) = d(n) - \hat{w}^T(n)u(n) \tag{3.31}
\]

\[
\hat{w}(n+1) = \hat{w}(n) + \mu u(n)e(n) \tag{3.32}
\]

where \(u(n) = [u(n), u(n-1), \ldots, u(n-M+1)]^T\), \(M\) is the filter length and \(\mu\) is the step-size. One problem with the LMS algorithm is that the adjustment of the filter coefficients is directly proportional to \(u(n)\) and thereby, suffers from gradient noise amplification when \(u(n)\) becomes large [Haykin, 2001]. To overcome this problem, the adjustment factor can be normalized by the Euclidean norm of the input vector which leads to the NLMS algorithm.
Adaptive weight-control mechanism

Transversal filter $w(n)$

$e(n) = d(n) - \hat{w}^T(n)u(n)$ (3.33)

$\hat{w}(n+1) = \hat{w}(n) + \frac{\mu}{|u(n)|^2}u(n)e(n)$ (3.34)

Weighted Cascaded Normalized LMS

The weighted cascaded NLMS (WCLMS) algorithm builds on a cascade coupling of three NLMS filters. It has been found that cascading such adaptive filters has some nice properties with respect to rate of convergence, prediction accuracy and numerical stability [Prandoni and Vetterli, 1998]. Therefore, such a filter structure is used in the WCLMS algorithm but applied in slightly different manner than in [Prandoni and Vetterli, 1998]. Instead of using only the output from the last predictor as the final prediction error, the prediction from each of the predictors in the cascade coupling are weighted and added as depicted in fig. 3.8. Following, the idea is then to use this refined prediction error which is the output from the WCLMS algorithm to calculate the final prediction error. The prediction error is then coded and transmitted to the decoder where the inverse prediction is applied. This is depicted in fig. 3.9.

From fig. 3.8 the predictors are given by

$P_1(x(n-1)) = w_{L_1}^T x(n-1)$ (3.35)

$P_2(x(n-1)) = P_1(x(n-1)) + \hat{e}_1(n)$ (3.36)

$P_3(x(n-1)) = P_2(x(n-1)) + \hat{e}_2(n)$ (3.37)

where $w_{L_1}$ is the first NLMS filter of order $L_1$ and, $\hat{e}_1(n)$ and $\hat{e}_2(n)$ are the predicted values of
3.3. Residual Coding

As mentioned the three predictors then have to be weighted before they are summed

\[
\nu_i(n-1) \geq 0 \land \sum_{i=1}^{3} \nu_i(n-1) = 1
\]

where \(\nu_i(n-1)\) is the weighting factor related to the \(i\)’th predictor which measures how well the \(i\)’th predictor has predicted the signal in the past. These weighting factors should then be updated every time a new prediction is made and in this case the update is based on the predictive minimum description length (PMDL) principle [Schuller et al., 2002].

The idea is here to find the joint probability of \(x(1), x(2), \ldots, x(n)\) in a predictive way. Assuming that the prediction error is Laplacian distributed and introducing a forgetting
because of non-stationarity, the expression for the PMDL weights is then
\[ \nu_i(n - 1) \propto e^{-\left(1 - \mu\right) \sum_{j=1}^{n-1} |e_i(n - j)| \cdot \mu^{j-1}} \]
(3.42)
where \( b \) is the scale parameter of the Laplacian distribution.

### 3.4 Implementation and Evaluations of the Coding System

Based on the theory described in the previous sections the coding system depicted in fig. 3.10 is proposed. To evaluate the proposed coding system two different tests were performed. First, it was investigated how the overall performance would be when the sinusoidal coder was placed before the pre-filter and between the pre- and post-filters, respectively. The influence of the sinusoidal coder position was evaluated through subjective listening tests. The experimental setup for these tests was as follows. For the sinusoidal coding 32 ms overlapping von Hann windowed time segments were used, and 4 ms time windows were used for the pre- and post-filtering. Further, the number of sinusoids were set to 30 in the sinusoidal coder. The quantizers for the amplitude, frequency and phases of the sinusoids was chosen according to [Christensen and van de Par, 2006] such that logarithmic quantizers with step-sizes of 0.161 and 0.003 were used for the amplitude and the frequency, respectively, and a 5 bit uniform quantizer was used for the phase. The pre- and post-filter were implemented as described in the previous sections but the gain-factor of the filters, determined by the variance of the input signal, was not applied, since it entailed audible artifacts during transient signal periods. To compensate for this a relatively high total bit-rate was needed and it was therefore fixed to 100 kbit/s.

![Figure 3.10: Block diagram of the sinusoidal subcoder integrated in the pre- and post-filtering setup where the WCLMS predictor acts as a waveform approximating residual subcoder.](image)

Two different coding system were build based on the described setup, but with two different placements of the sinusoidal coder. One with the sinusoidal coder in front of the pre-filter and one with the sinusoidal coder being positioned between the filters. The
two coding systems were applied on a tonal violin signal and a more stochastic mixed instrumental signal. No noticeable differences were observed between the signals coded with the two different coding system, which justifies the use of the coding structure presented in section 3.1.

The performance gain achieved by combining the sinusoidal coder with the pre- and post-filtering based residual coder, was measured through rate-distortion measurements. In these measurements the gain factor mentioned previously was applied to the pre- and post filters through linear interpolation. Besides this, the three different setups were chosen:

- Setup 1: All bits were allocated the sinusoidal coder.
- Setup 2: All bits were allocated a WCLMS coder.
- Setup 3: The bits were distributed between the two coders for fixed numbers of sinusoids (6, 12 and 25) in the sinusoidal coder.

The three setups were applied on a violin signal and a mixed instrumental signal as for the subjective listening tests, and the results are depicted in fig. 3.11 and 3.12 respectively. Generally, the results from the two measurements show that for low bit-rates the lowest average distortion is obtained by allocating all bits to the sinusoidal coder or by distributing the bits between the WCLMS coder and the sinusoidal coder. For higher bit-rates it would be more efficient to allocate the bits only to the WCLMS coder. In the depicted measurements it is only the number of sinusoids which is varied, but further work could as well investigate how the distortion would be if the step-size of the quantizers in the sinusoidal coder was varied. Furthermore, it can be deduced that the overall average distortion can be minimized by, for a given bit rate, allocating the bits as in either setup 1, 2 or 3.

Bibliography


Figure 3.11: Measured average rate-distortion of sinusoidal coding, WCLMS coding, and sinusoidal and WCLMS coded residual coding of a violin signal. The sinusoidal and WCLMS coded residual coding is performed for 6, 12 and 25 number of sinusoids.
Figure 3.12: Measured average rate-distortion of sinusoidal coding, WCLMS coding, and sinusoidal and WCLMS coded residual coding of a mixed signal containing different instruments. The sinusoidal and WCLMS coded residual coding is performed for 6, 12 and 25 number of sinusoids.
Bibliography


Transform Coding of Line Spectral Frequency Coefficients

The LSF coefficients are an representation of the linear predictive coefficients and efficient coding of them is an important topic in the field of speech coding which is often based on linear predictive coding (LPC). In this worksheet, however, the LSF coefficients originate from LPC analysis of masking curves for the audio coder described in worksheet 3. The audio coder uses the pre- and post-filter setup from [Schuller, Yu, Huang, and Edler, 2002] where the perceptual weighting of an audio signal is, in the encoder, performed by the pre-filter whose frequency response is the inverse of the masking curve. The post-filter in the decoder has the same frequency response as the masking curve and performs thus the inverse weighting on the quantized signal.

The masking curve does not change much under nearly stationary parts of the input audio signal for which reason the filter or LPC coefficients of the pre- and post-filter is highly correlated. Since the LPC coefficient must be sent from the encoder to the decoder along with the encoded audio signal as side information, an efficient compression scheme should be used which exploits this correlation and reduces the bit rate. The compression scheme can be both lossless or lossy, but in this worksheet only lossy compression will be considered since it entails larger compression.

4.1 Line Spectral Frequency Coefficients

LPC coefficients are not suited for direct quantization since

- there is no easy stability check
- there is no simple relationship between the coefficients and the frequency response
- they are not amenable to interpolation
the dynamic range of the coefficients is relatively large

Therefore other representations of the LPC coefficients have been proposed. This comprises among others PARCOR coefficients \[\text{Rabiner and Schafer, 1978}\] and line spectral frequencies (LSF) \[\text{Itakura, 1975}\] where the latter is generally accepted as one of the best representations for quantization and compression of LPC coefficients \[\text{Farvardin and Laroia, 1989}\].

The \(M\)th order prediction error filter is an FIR filter given by

\[
A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M}
\] (4.1)

where \(\{a_1, a_2, \cdots, a_M\}\) are the LPC coefficients. If the polynomials \(P(z)\) and \(Q(z)\) are defined as

\[
P(z) = A(z) + z^{-(M+1)} A(z^{-1})
\] (4.2)

and

\[
Q(z) = A(z) - z^{-(M+1)} A(z^{-1}) ,
\] (4.3)

then \(P(z)\) and \(Q(z)\) are symmetric and anti-symmetric, respectively, and the following three important properties apply \[\text{Soong and Juang, 1984}\]:

1. All the zeros of the polynomials \(P(z)\) and \(Q(z)\) are on the unit circle.
2. The zeros of the polynomials \(P(z)\) and \(Q(z)\) are interlaced.
3. Minimum phase property can be preserved after quantization in a simple manner.

The first property means that the zeros of \(P(z)\) and \(Q(z)\), where both polynomials are of order \(M + 1\), can be expressed as

\[
z_k = e^{j\omega_k} , \quad k = 0, 1, \cdots, 2M + 1.
\] (4.4)

From the definition of \(P(z)\) and \(Q(z)\) in equation 4.2 and 4.3 it follows that the zeros appear in complex conjugated pairs. Also, since

\[
Q(e^{j0}) = Q(1) = A(1) - 1A(1) = 0
\] (4.5)

and

\[
P(e^{j\pi}) = P(-1) = A(-1) + (-1)^{-(M+1)}A(-1) = 0 , \quad \text{for } M \text{ even}
\] (4.6)

\[
Q(e^{j\pi}) = Q(-1) = A(-1) - (-1)^{-(M+1)}A(-1) = 0 , \quad \text{for } M \text{ uneven}
\] (4.7)

then \(\omega_0 = 0\) and \(\omega_{M+1} = \pi\) are fixed zeros. Thus, all the information is captured in the \(M\) coefficients \(\omega_k\) for \(k = 1, 2, \cdots, M\) in the interval \([0, \pi]\), and they are called the LSF
coefficients. This definition reveals that the LSF coefficients have a close relationship
with the frequency response of the prediction error filter in equation 4.1, and it enables
smooth spectral interpolation between LSF coefficients and the use of psychovisual or
acoustical weighted quantization [Kabal and Ramachandran, 1986].

The second and the third property are both related to the stability of the inverse filter
of the prediction error filter $A(z)$. If $A(z)$ is minimum phase, then its inverse will be
stable. The minimum phase property is guaranteed if the LSF coefficients are ordered
in ascending order [Farvardin and Laroia, 1989].

### 4.1.1 Statistics of LSF-coefficients

The ordering property of the LSF coefficients indicates that there is intraframe correla-
tion between the LSF coefficients (i.e. the correlation between LSF coefficients in the
same frame). Also, since the masking curve is almost constant during stationary parts of
the input signal, there is some interframe correlation between neighboring LSF vectors
(i.e. the correlation between LSF coefficients in adjacent frames). Table 4.1 and 4.2
show the intraframe and interframe correlation coefficients, respectively, computed from
some training data. The training data consists of approximately 600,000 LSF vectors
generated from a 15th-order LPC analysis of masking curves. The masking curves are
computed from 4 ms blocks obtained from eight different songs of a total length of approx-
imately 35 minutes. The songs are sampled with 44.100 kHz which gives a block
length of 176 samples. Thus, the training data is an $M \times N$ matrix $X_t$, where $M$ is the
LPC order and $N$ is the number of LSF vectors, with the LSF vectors as column vectors.

Table 4.1 shows the intraframe sample correlation coefficient matrix with the $(i,j)$th
element computed from

$$
\hat{\rho}_{ij}^{\text{intra}} = \frac{1}{N-1} \sum_{n=1}^{N} \frac{(X_{in} - \hat{\mu}_i)(X_{jn} - \hat{\mu}_j)}{\hat{\sigma}_i \hat{\sigma}_j}
$$

(4.8)

where $X_{in}$ denotes the $(i,n)$th element of $X$, and $\hat{\mu}_i$ and $\hat{\sigma}_i^2$ denote the sample mean
and sample variance for the $i$th row of $X$, respectively. Table 4.2 shows the interframe
sample correlation coefficient vector for all rows of $X$, i.e. each row is split into blocks of
$L = 15$ consecutive samples where the $(i,j)$th sample correlation coefficient is computed
from

$$
\hat{\rho}_{ij}^{\text{inter}} = \frac{1}{\lfloor N/L \rfloor - 1} \sum_{n=1,L+1,\ldots,\lfloor N/L \rfloor L+1} \frac{(X_{in} - \hat{\mu}_i)(X_{i(n+j)} - \hat{\mu}_i)}{\hat{\sigma}_i^2}
$$

(4.9)

where $\lfloor \cdot \rfloor$ denotes the floor function.

The tables indicate a strong intraframe correlation and a strong interframe correlation
between neighboring LSF coefficients. which is also concluded in [Farvardin and Laroia,
### Table 4.1: Intraframe sample correlation coefficient matrix of training LSF vectors.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.31</td>
<td>0.36</td>
<td>0.36</td>
<td>0.38</td>
<td>0.34</td>
<td>0.36</td>
<td>0.33</td>
<td>0.36</td>
<td>0.34</td>
<td>0.31</td>
<td>0.30</td>
<td>0.32</td>
<td>0.27</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.34</td>
<td>0.55</td>
<td>0.55</td>
<td>0.40</td>
<td>0.39</td>
<td>0.31</td>
<td>0.33</td>
<td>0.25</td>
<td>0.27</td>
<td>0.20</td>
<td>0.21</td>
<td>0.04</td>
<td>0.36</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.32</td>
<td>0.30</td>
<td>0.18</td>
<td>0.74</td>
<td>0.68</td>
<td>0.64</td>
<td>0.70</td>
<td>0.67</td>
<td>0.65</td>
<td>0.35</td>
<td>0.35</td>
<td>0.28</td>
<td>0.36</td>
<td>0.32</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.39</td>
<td>0.74</td>
<td>0.74</td>
<td>0.73</td>
<td>0.73</td>
<td>0.72</td>
<td>0.72</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.16</td>
<td>0.40</td>
<td>0.64</td>
<td>0.78</td>
<td>1.00</td>
<td>0.85</td>
<td>0.74</td>
<td>0.56</td>
<td>0.42</td>
<td>0.34</td>
<td>0.21</td>
<td>0.05</td>
<td>0.34</td>
<td>0.56</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.33</td>
<td>0.30</td>
<td>0.62</td>
<td>0.69</td>
<td>0.67</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.14</td>
<td>0.34</td>
<td>0.59</td>
<td>0.74</td>
<td>0.85</td>
<td>1.00</td>
<td>0.86</td>
<td>0.75</td>
<td>0.64</td>
<td>0.49</td>
<td>0.34</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.09</td>
<td>0.13</td>
<td>0.56</td>
<td>0.63</td>
<td>0.73</td>
<td>0.86</td>
<td>1.00</td>
<td>0.88</td>
<td>0.77</td>
<td>0.61</td>
<td>0.50</td>
<td>0.13</td>
<td>0.00</td>
<td>0.77</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.09</td>
<td>0.26</td>
<td>0.43</td>
<td>0.48</td>
<td>0.55</td>
<td>0.69</td>
<td>0.75</td>
<td>0.88</td>
<td>1.00</td>
<td>0.89</td>
<td>0.75</td>
<td>0.26</td>
<td>0.26</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.09</td>
<td>0.27</td>
<td>0.38</td>
<td>0.44</td>
<td>0.51</td>
<td>0.61</td>
<td>0.71</td>
<td>0.87</td>
<td>1.00</td>
<td>0.89</td>
<td>0.77</td>
<td>0.80</td>
<td>0.74</td>
<td>0.83</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.05</td>
<td>0.20</td>
<td>0.36</td>
<td>0.36</td>
<td>0.36</td>
<td>0.49</td>
<td>0.61</td>
<td>0.75</td>
<td>0.89</td>
<td>1.00</td>
<td>0.91</td>
<td>0.09</td>
<td>0.48</td>
<td>0.78</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.02</td>
<td>0.21</td>
<td>0.28</td>
<td>0.32</td>
<td>0.32</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-0.06</td>
<td>-0.04</td>
<td>0.06</td>
<td>0.02</td>
<td>-0.08</td>
<td>-0.02</td>
<td>-0.21</td>
<td>-0.26</td>
<td>-0.40</td>
<td>-0.59</td>
<td>-0.77</td>
<td>-1.00</td>
<td>-0.25</td>
<td>-0.23</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-0.18</td>
<td>-0.55</td>
<td>-0.49</td>
<td>0.47</td>
<td>0.54</td>
<td>0.62</td>
<td>0.80</td>
<td>0.82</td>
<td>0.54</td>
<td>0.48</td>
<td>0.29</td>
<td>0.25</td>
<td>1.00</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>-0.15</td>
<td>-0.41</td>
<td>-0.52</td>
<td>-0.56</td>
<td>-0.62</td>
<td>-0.69</td>
<td>-0.80</td>
<td>-0.85</td>
<td>-0.76</td>
<td>-0.69</td>
<td>-0.23</td>
<td>-0.80</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4.2: Interframe sample correlation coefficient matrix of the elements in the training LSF vectors.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.34</td>
<td>0.55</td>
<td>0.55</td>
<td>0.40</td>
<td>0.39</td>
<td>0.31</td>
<td>0.33</td>
<td>0.25</td>
<td>0.27</td>
<td>0.20</td>
<td>0.21</td>
<td>0.04</td>
<td>0.36</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.39</td>
<td>0.74</td>
<td>0.74</td>
<td>0.73</td>
<td>0.73</td>
<td>0.72</td>
<td>0.72</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.16</td>
<td>0.40</td>
<td>0.64</td>
<td>0.78</td>
<td>1.00</td>
<td>0.85</td>
<td>0.74</td>
<td>0.56</td>
<td>0.42</td>
<td>0.34</td>
<td>0.21</td>
<td>0.05</td>
<td>0.34</td>
<td>0.56</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.33</td>
<td>0.30</td>
<td>0.62</td>
<td>0.69</td>
<td>0.67</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.14</td>
<td>0.34</td>
<td>0.59</td>
<td>0.74</td>
<td>0.85</td>
<td>1.00</td>
<td>0.86</td>
<td>0.75</td>
<td>0.64</td>
<td>0.49</td>
<td>0.34</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.09</td>
<td>0.13</td>
<td>0.56</td>
<td>0.63</td>
<td>0.73</td>
<td>0.86</td>
<td>1.00</td>
<td>0.88</td>
<td>0.77</td>
<td>0.61</td>
<td>0.50</td>
<td>0.13</td>
<td>0.00</td>
<td>0.77</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.09</td>
<td>0.26</td>
<td>0.43</td>
<td>0.48</td>
<td>0.55</td>
<td>0.69</td>
<td>0.75</td>
<td>0.88</td>
<td>1.00</td>
<td>0.89</td>
<td>0.75</td>
<td>0.26</td>
<td>0.26</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.09</td>
<td>0.27</td>
<td>0.38</td>
<td>0.44</td>
<td>0.51</td>
<td>0.61</td>
<td>0.71</td>
<td>0.87</td>
<td>1.00</td>
<td>0.89</td>
<td>0.77</td>
<td>0.80</td>
<td>0.74</td>
<td>0.83</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.05</td>
<td>0.20</td>
<td>0.36</td>
<td>0.36</td>
<td>0.36</td>
<td>0.49</td>
<td>0.61</td>
<td>0.75</td>
<td>0.89</td>
<td>1.00</td>
<td>0.91</td>
<td>0.09</td>
<td>0.48</td>
<td>0.78</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.02</td>
<td>0.21</td>
<td>0.28</td>
<td>0.32</td>
<td>0.32</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>-0.06</td>
<td>-0.04</td>
<td>0.06</td>
<td>0.02</td>
<td>-0.08</td>
<td>-0.02</td>
<td>-0.21</td>
<td>-0.26</td>
<td>-0.40</td>
<td>-0.59</td>
<td>-0.77</td>
<td>-1.00</td>
<td>-0.25</td>
<td>-0.23</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-0.18</td>
<td>-0.55</td>
<td>-0.49</td>
<td>0.47</td>
<td>0.54</td>
<td>0.62</td>
<td>0.80</td>
<td>0.82</td>
<td>0.54</td>
<td>0.48</td>
<td>0.29</td>
<td>0.25</td>
<td>1.00</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-0.15</td>
<td>-0.41</td>
<td>-0.52</td>
<td>-0.56</td>
<td>-0.62</td>
<td>-0.69</td>
<td>-0.80</td>
<td>-0.85</td>
<td>-0.76</td>
<td>-0.69</td>
<td>-0.23</td>
<td>-0.80</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>
4.1. Line Spectral Frequency Coefficients

The distribution of the individual LSF coefficient are important in order to design an efficient quantizer. In [Farvardin and Laroia, 1989] it is argued, that it is reasonable to assume the LSF coefficients to be Gaussian distributed, while in [Erkelens and Broersen, 1995] and [Lois and Vu, 1997] the empirical distributions of the LSF coefficients are found based on a speech training set. The results show that the distribution resembles skewed Gaussian or skewed Laplace distributions. Figure 4.1 shows the empirical histograms for the LSF coefficients from the music training data, described previously, along with the sample mean and variance of the corresponding LSF coefficient. The plotted distributions resembles the results in [Erkelens and Broersen, 1995] and [Lois and Vu, 1997].

![Normalized histograms of LSF coefficients](image)

**Figure 4.1:** Histograms of LSF coefficients. The histograms in the top plot are normalized such that the maximum value for each histogram is one.
4.2 Transform Coding

As discussed in the previous section, there is a strong correlation between LSF coefficients in the same frame (intraframe correlation) and in adjacent frames (interframe correlation). This means that there is a significant amount of redundancy present in the LSF coefficients for which reason direct quantization could be ineffective. The redundancy is removed by removing the correlation between the LSF coefficients, and there exist several tools that can accomplish this decorrelation. This comprises among others predictive coding, sub-band coding and transform coding. The coding gain with respect to direct pulse code modulation (PCM) coding is asymptotically the same for all of these coding schemes [Gray and Gersho, 1992, p. 244, p. 251].

Sub-band coding and transform coding are similar in the way that they first analyse the signal to obtain an intermediate representation of the signal which is amenable to quantization, and then synthesize the quantized intermediate signal to obtain a quantized version of the original signal. The analysis/synthesis is for sub-band coding performed by use of a filter-bank, while the analysis/synthesis for transform coding is performed by some orthogonal transform. Both of the coding schemes suffer from a delay which in some application can be too high (see [Schuller et al., 2002]), but the implementation and computational complexity are typically much lower as compared to vector predictive coding. Another important property of the sub-band and transform coding schemes is that the intermediate representation often relates better to human perception than the original signal for which reason perceptual quantization can be used. Predictive coding does not suffer from any coding delay, but in the case of vector predictive coding in a non-stationary environment the computational complexity and amount of side information to be send are immense. Thus, simplification of the predictor (see for example [Shoham, 1987] and [Yong, Davidson, and Gersho, 1988]) is often introduced in order to obtain a feasible vector predictive coding scheme. In this worksheet, transform coding will be used to quantize the LSF coefficients because of its simplicity.

Figure 4.2: Block diagram of transform coder.

Figure 4.2 shows the three basic steps of a transform coder. In the analysis step, the input vector $x$ is transformed into the transform vector $y$ with the transform kernel

$$T$$

and then quantized by $Q_1$, $Q_2$, $Q_M$. In the synthesis step, the quantized transform vector $\hat{y}$ is transformed back into the quantized version of the original signal $\hat{x}$. The transform kernel in the synthesis step is denoted by $T^{-1}$. The quantizers $Q_1$, $Q_2$, $Q_M$ are used to quantize the transform coefficients.
or matrix $T$. Each of the transform coefficients in the transform vector $y$ is then, in the second step, quantized with a quantizer $Q_i$ which produces the quantized transform vector $\hat{y}$. In the last step, the reconstructed version $\hat{x}$ of the input vector $x$ is formed by transforming the quantized transform vector $\hat{y}$ with the inverse of the transform kernel or matrix $T$. The three steps can be summarized in the following equations

$$y =Tx$$  
$$\hat{y}_i = Q(y_i) \quad \text{for } i = 1, \cdots, M$$  
$$\hat{x} = T^{-1}\hat{y}.$$  

As discussed previously, the transform coding scheme enables the quantization to be performed on the transform coefficients which, for a suitable transform kernel $T$, are less correlated than the coefficient in the original input signal. In this way, the available bit-pool is not wasted on encoding redundant information but, in the case of completely uncorrelated samples, only used to encode the information.

From this discussion it is evident that the transform kernel should be chosen so that it produces as decorrelated transform coefficients as possible. Another important feature is that the inverse transform kernel should not amplify quantization errors. The latter requirement is fulfilled if the transform kernel is restricted to be orthonormal, i.e.

$$T^{-1} = TT^T,$$  

which follows from the observation that [Gray and Gersho, 1992, p. 239]

$$D_{TC} = E\{\|y - \hat{y}\|^2\} = E\{\|Tx - T\hat{x}\|^2\}$$  
$$= E\{(x - \hat{x})^TT^TT(x - \hat{x})\} = E\{\|x - \hat{x}\|^2\}. \quad (4.15)$$

That is, the overall distortion $D_{TC}$ introduced by quantizing $y$ is the same in signal and transform domain and unaffected by the transform kernel if the transform kernel is an orthogonal matrix.

To find the transform kernel that produces completely uncorrelated transform coefficient, consider the auto-correlation matrix $R_{yy}$ of the transform vector $y$. A simple rewriting yields

$$R_{yy} = E\{yy^T\} = E\{Tx x^TT^T\} = TR_{xx}T^T.$$  

If the auto-correlation matrix $R_{xx}$ of the input vector $x$ is replaced by its eigenvalue decomposition, so that

$$R_{yy} = TUAU^TT^T,$$  

it is clear that $R_{yy} = \Lambda$ if $T = U^T$. That is, the transform coefficients are completely uncorrelated, i.e. $R_{yy}$ is a diagonal matrix, if the transform kernel has the eigenvectors of the auto-correlation matrix $R_{xx}$ as row vectors. The transform satisfying this condition is known as the discrete Karhunen-Loeve Transform (KLT) or Hotelling Transform
The main drawback of the KLT is that it depends on the statistics of the input signal. That is, the optimal transform kernel changes with the statistics of the input signal and is only constant in the case of a stationary input signal. For this reason, several signal independent transform kernels like the discrete Hadamard Transform (DHT), discrete Cosine Transform (DCT) and discrete Fourier Transform (DFT) have been suggested, and they have the additional advantage that fast implementations of them are known [Jayant and Noll, 1984, pp. 546-563]. The main drawback of these transforms is that they do not produce completely uncorrelated transform coefficients and are thus sub-optimal in this respect.

4.3 Entropy Coded Quantization of Transform Coefficients

The quantization step of figure 4.2 typically consists of restricting the bit-rate to be below some threshold while keeping the introduced distortion as small as possible. In mathematical terms, this can be formulated as

\[
\min D(x, \hat{x}_i) \quad \text{s.t. } R(i) \leq \bar{R}
\]

where \(D(\cdot)\) is some distortion measure associated by the representation of \(x\) with \(\hat{x}_i\) in \(R(i)\) bits, and \(\bar{R}\) is the maximum allowed rate or rate constraint. A commonly used distortion measure is the mean square error \(D(x, \hat{x}_i) = E\{\|x - \hat{x}_i\|^2\}\) which will be used in the rest of this worksheet.

4.3.1 Rate-Distortion Theory

Consider a continuous amplitude, discrete time memoryless, i.e. independent, signal \(x\). Shannon showed that the solution to equation 4.18 is bounded by [Cover and Thomas, 2006, p. 310]

\[
R(D) \geq h(x) - h(x - \hat{x})
\]

where \(h(\cdot)\) denotes the differential entropy and \(D = E\{(x - \hat{x})^2\}\). Thus, the minimum required rate is in loose terms the difference between the information in the input signal and the information in the quantization error. For an input signal \(x\) with a Gaussian pdf, it can be shown that the bound equals [Cover and Thomas, 2006, p. 310]

\[
R(D)_G \geq \frac{1}{2} \log_2 2\pi e \sigma_x^2 - \frac{1}{2} \log_2 2\pi e D = \frac{1}{2} \log_2 \frac{\sigma_x^2}{D} .
\]

For input signals with a non-Gaussian pdf, it is not possible to obtain an explicit expression for the bound, but it can be shown that the bound is confined in the interval [Jayant and Noll, 1984, p. 639]

\[
R(D)_L = h(x) - \frac{1}{2} \log_2 2\pi e D \leq R(D) \leq R(D)_G .
\]
4.3. Entropy Coded Quantization of Transform Coefficients

Figure 4.3 shows the Shannon lower bound \( R(D)_L \) for three different input pdf’s whose differential entropy is listed in Table 4.3.

<table>
<thead>
<tr>
<th>pdf, ( p_X(x) )</th>
<th>differential entropy, ( h(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>( \frac{1}{2} \log_2 2\pi e \sigma_x^2 )</td>
</tr>
<tr>
<td>Uniform</td>
<td>( \frac{1}{2} \log_2 12\sigma_x^2 )</td>
</tr>
<tr>
<td>Laplace</td>
<td>( \frac{1}{2} \log_2 2e^2 \sigma_x^2 )</td>
</tr>
</tbody>
</table>

*Table 4.3: Differential entropy for different input pdf’s [Jayant and Noll, 1984, p. 154]*

![Figure 4.3: Rate-Distortion bounds for memoryless input signals with different pdf's](image)

4.3.2 Optimal Quantizer for Entropy Coded Quantization

An entropy coded quantizer consists of two elements: A quantizer and an entropy coder. The entropy coder operates on the output signal from the quantizer and ideally removes the redundancy of the signal so that the resulting bit rate of the entropy coded quantizer equals the entropy of the output signal from the quantizer. The distortion is introduced in the quantizer for which reason the design of the quantizer is very important. It can be shown that, for high resolution quantizers, the best quantizer is a uniform quantizer.
4 Transform Coding of Line Spectral Frequency Coefficients

Empirical studies have also shown that the uniform quantizers are nearly optimal for low bit rates as well [Gray and Gersho, 1992, p. 300]. The main drawback of entropy coded quantization is that it produces variable bit rate with a mean value equal to the entropy. If constant bit rate is required, pdf-optimized quantizers with a certain resolution determined by bit allocation schemes should be used.

The differential entropy of the quantization error for the combination of a uniform quantizer and an entropy coder is

\[ h(x - \hat{x}) = \mathbb{E}\{-\log_2 p_q(x)\} = -\int_{-\infty}^{\infty} p_q(x) \log_2 p_q(x) dx \] (4.22)

where \( p_q(x) \) is the pdf of the quantization error. If high resolution conditions are assumed, i.e. the input pdf is constant in the quantization intervals and that no overload regions exist [Gray and Gersho, 1992, p. 161], the differential entropy of the quantization error can be rewritten as

\[ h(x - \hat{x}) = -\int_{-\Delta/2}^{\Delta/2} \frac{1}{\Delta} \log_2 \frac{1}{\Delta} dx = \log_2 \Delta \] (4.23)

where \( \Delta \) is the step size of the uniform quantizer. Inserting this into equation 4.19 yields

\[ R(D) \geq h(x) - \log_2 \Delta = h(x) - \frac{1}{2} \log_2 2\pi e D + \frac{1}{2} \log_2 \frac{\pi e}{6} \approx R(D)_L + 0.255 . \] (4.24)

Thus, the optimal entropy coded quantizer is approximately 0.255 bits from the Shannon lower bound \( R(D)_L \). The quantity \( \frac{1}{2} \log_2 \frac{\pi e}{6} \approx 0.255 \) can be explained by the memoryless quantization and can be reduced if a quantizer with memory is used [Jayant and Noll, 1984, p. 155]. In many situations, however, it is not worth using quantizers with memory since it increases the complexity [Gray and Gersho, 1992, p. 300].

To summarize the discussion, the design rule for the step-size can now be formulated as: If, for an ideal entropy coder, a given input pdf and for a desired rate \( \bar{R} \), the step size \( \Delta \) of the uniform quantizer is chosen to

\[ \Delta = 2^{h(x) - \bar{R}} , \] (4.25)

then the entropy coded quantizer will minimize equation 4.18 for a memoryless high resolution quantizer.

### 4.3.3 Optimal Bit Allocation

Until now, the discussed rate-distortion theory has only covered quantization of a single input signal with constant variance \( \sigma_x^2 \). In this subsection, the quantization of \( N \) independent and non-identically distributed input signals is treated. The generalization of
4.3. Entropy Coded Quantization of Transform Coefficients

equation 4.18 to the multiple input signals scenario can be formulated as

\[
\min \sum_{j=1}^{M} D(x_j, \tilde{x}_i) \quad (4.26)
\]

s.t. \( \sum_{j=1}^{M} R_j(i) \leq R \)

where \( R = M\bar{R} \) is the total available bitpool which is distributed among the \( R_j(i)s \) to minimize the overall distortion \( D \). Since the input signals are assumed independent, the overall solution of equation 4.26 is the sum of the single input solutions to equation 4.18 for each \( j \) \cite{Cover:2006}. Thus, following the same arguments as in the previous subsection, the solution

\[
R(D) = \sum_{j=1}^{M} R(D_j) = \sum_{j=1}^{M} (h(x_j) - \log_2 \Delta_j) \quad (4.27)
\]

minimizes equation 4.26 for the memoryless high resolution entropy coded quantizer. From this, the distortion-rate function given by

\[
D(R) = \sum_{j=1}^{M} \sigma^2_{q_j} = \sum_{j=1}^{M} \frac{\Delta_j^2}{12} = \frac{1}{12} \sum_{j=1}^{M} 2^{2h(x_j)} 2^{-2R_j} \quad (4.28)
\]

can be obtained. This function can be minimized by the use of Lagrange multipliers. Thus, the minimum of the function

\[
J(R) = D(R) + \lambda \sum_{j=1}^{M} R_j \quad (4.29)
\]

minimizes the constrained optimization problem in equation 4.26 \cite{Shoham:1988, Gray:1992}. The solution is given by \cite{Gray:1992} p. 229

\[
R_j = \bar{R} + h(x_j) - \frac{1}{M} \sum_{i=1}^{M} h(x_i) \ , \quad \text{for } j = 1, \ldots, M \ . \quad (4.30)
\]

Using this equation, the step size \( \Delta_j \) for the \( j \)th quantizer can be found from equation 4.28 as

\[
\Delta_j = 2^{h(x_j)-R_j} = 2^{-\bar{R}+1/M \sum_{i=1}^{M} h(x_i)} = 2^{-\bar{R} \prod_{i=1}^{M} 2^{h(x_i)}}^{1/M} = \Delta . \quad (4.31)
\]

Thus, all the quantizers have the same step size which is computed from the average bit rate \( \bar{R} \) and all the differential entropies \( h(x_i) \). If the \( h(x_i) \)'s are all equal, then equation 4.31 reduces to equation 4.25.
4.4 Application: Transform Coding of Line Spectral Frequency Coefficients

In this section, transform coding with entropy coded quantization of the LSF coefficients is described. This encompasses both the design and the evaluation.

The input signal of the transform coder is a matrix $X$ which is a block of $L$ consecutive LSF vectors with $M$ elements. The matrix $X$ is transformed into the matrix $Y$ with a 2D-transform which is a straightforward extension of the 1D-transform. Each of the transform coefficients is hereafter quantized before an inverse 2D-transform is applied in order to obtain the reconstructed matrix $\hat{X}$.

4.4.1 2D-DCT and 2D-KLT

As mentioned in section 4.2, there exist several orthogonal transform kernels among which KLT is optimal. However, since the transform kernel of the KLT is dependent on the input data, it is not the most popular one. The most popular is the DCT which is widely used for image and video coding, and a modified version of it (the MDCT) is also very popular in audio coding. For these reasons, two different transform coders are designed in this worksheet based on the KLT and DCT, respectively, and their performance is compared to pulse code modulation (PCM).

Since the input signal is a matrix, the transform coder must use a 2D-transform in order to reduce both intraframe and interframe correlation. For an $M \times L$ input matrix $X$, equation 4.10 through 4.12 can be written in the 2D case as

$$Y = T_M X T_L^T$$

$$\hat{Y}_{ij} = Q(Y_{ij}) \text{, for } i = 1, \cdots, M \text{, } j = 1, \cdots, L$$

$$\hat{X} = T_{M}^{T} \hat{Y} T_{L}.$$  

where $T_{M}$ and $T_{L}$ are $M \times M$ and $L \times L$ transform kernels, respectively. For the KLT the transform kernels must be found from eigenanalysis of the input signal while the transform kernels for the DCT are static and can be found from

$$T_K = \sqrt{\frac{2}{K}} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} & \cdots & 1/\sqrt{2} \\ \cos \frac{\pi}{2K} & \cos \frac{3\pi}{2K} & \cos \frac{5\pi}{2K} & \cdots & \cos \frac{(2K-1)\pi}{2K} \\ \cos \frac{2\pi}{2K} & \cos \frac{6\pi}{2K} & \cos \frac{10\pi}{2K} & \cdots & \cos \frac{(2K-2)\pi}{2K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \cos \frac{(K-1)\pi}{2K} & \cos \frac{3(K-1)\pi}{2K} & \cos \frac{5(K-1)\pi}{2K} & \cdots & \cos \frac{(2K-1)(K-1)\pi}{2K} \end{bmatrix}. \quad (4.35)$$

for $K = L$ and $K = M$, respectively.
Figure 4.4: Histograms of the low frequency transform coefficients for the DCT. The histograms in the top plot are normalized such that the maximum value for each histogram is one.
Figure 4.5: Histograms of the low frequency transform coefficients for the KLT. The histograms in the top plot are normalized such that the maximum value for each histogram is one.
4.4. Application: Transform Coding of Line Spectral Frequency Coefficients

4.4.2 Statistics of Transform Coefficients

The statistics of the transform coefficients are important for the quantization process. Figure 4.4 and 4.5 show the histograms of the transform coefficients in the upper $3 \times 3$ sub-matrix of $Y$, whose size is $15 \times 10$, for the DCT and KLT, respectively. A direct comparison with the histogram of the LSF coefficients in figure 4.1 reveals that the histogram of the transform coefficients are more symmetric and that the resemblance to the Laplace distribution is stronger.

4.4.3 Bit Allocation and Design of Quantizer

The quantizer design is based on section 4.3 and the observed histograms of the transform coefficients. Recall that the quantizer is uniform and specified by the step size which is calculated from equation 4.31 to satisfy an average rate constraint $\bar{R}$ for the transform coefficients. Besides the rate constraint, the step size also depends of the differential entropy of the transform coefficients. The differential entropies listed in table 4.3 depend on the variance of the input signal which are estimated from the training data. Since the transform coefficients for the DCT and KLT resemble the Laplace distribution more than the Gaussian and Uniform distribution, the step size is found from

$$\Delta = 2^{-\bar{R}} \left[ \prod_{i=1}^{M} 2^{h(x_i)} \right]^{1/M} = 2^{-\bar{R}} \left[ 2e^{2} \prod_{i=1}^{M} \sigma_i^2 \right]^{1/M} \quad (4.36)$$

where the variances $\sigma_i^2$ of the transform coefficients are estimated from the training data.

4.4.4 Verification of LSF Compression System

Before the overall system performance is evaluated on a real signal, the implementation is verified in a controlled environment. A couple of verification tests have been carried out and this subsection presents one of them. In the presented verification test, the input signal is a one dimensional random sequence $x[n]$, i.e. $N = 1$, generated by a first order auto-regressive (AR) source driven by white Gaussian noise with variance $\sigma_u^2$. The AR(1) random sequence satisfies a couple of properties which makes it ideal for the verification: It is stationary and it is possible to calculate the average distortion analytically under high resolution assumptions.

The analytical expression for the distortion is found from equation 4.28 and equation
4.31 to

\[
D = \frac{\Delta^2}{12} = \frac{2^{-2\bar{R}}}{12} \left[ \prod_{i=1}^{M} 2^{h(x_i)} \right]^{2/M} = \frac{2^{-2\bar{R}}}{12} \left[ \prod_{i=1}^{M} \sqrt{2\pi e\sigma^2_i} \right]^{2/M}
\]

\[
= \frac{2^{-2\bar{R}}}{12} 2\pi e \left[ \prod_{i=1}^{M} \sigma^2_i \right]^{1/M} = \frac{2^{-2\bar{R}}}{12} 2\pi e \left[ \prod_{i=1}^{M} \bar{t}_i R_{xx} \bar{t}_i \right]^{1/M}
\]  

(4.37)

where \( R_{xx} \) is the auto-correlation matrix of the AR(1) sequence and \( \bar{t}_i \) is the \( i \)th row vector of the transform kernel \( T \). The analytical expression for the auto-correlation matrix is found by forming the toeplitz matrix of the auto-correlation vector which is computed from a reformulation of the normal equations [Hayes, 1996, p. 148]. This reformulation is given by

\[
r_{xx} = H^{-1} v
\]  

(4.38)

where

\[
r_{xx} = \begin{bmatrix} r_{xx}[0] & r_{xx}[0] & \cdots & r_{xx}[M-1] \end{bmatrix}^T,
\]

(4.39)

\[
v = \begin{bmatrix} \sigma_u^2 & 0 & \cdots & 0 \\ 0 & -a & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -a & 1 \end{bmatrix}^T,
\]

(4.40)

\[
H = \begin{bmatrix}
1 & -a & 0 & \cdots & 0 \\
-a & 1 & 0 & \cdots & 0 \\
0 & -a & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & -a & 1
\end{bmatrix}
\]

(4.41)

\[
(4.42)
\]

where \( r_{xx}[k] = r_{xx}[-k] = E\{x[n]x[n-k]\} \), \( \sigma_u^2 \) is the variance of the driving white Gaussian noise and \( a \) is the coefficient of the AR(1) sequence.

Figure 4.6 shows the rate-distortion plot for the coding of the AR(1) sequence. The plot was created using the design rules listed in the previous sections, i.e. (1) find the KLT transform kernel, mean and variances of transform coefficients from a training sequence and (2) compute the optimal step size of the uniform quantizer based on the desired target bit rate and the obtained variances of the transform coefficients. Another realization from the AR(1) source was used for testing. The driving white Gaussian noise had a variance of \( \sigma_u^2 = 2 \), and the AR(1) coefficient was \( a = 0.92 \). Using equation (4.37), the distortion for a desired bit-rate was calculated and the results are shown in table 4.5. Table 4.5 shows the measured distortion and entropy for PCM, DCT and KLT, and they are approximately equal to the values found from the analytical expression. Notice, that the distortions associated with DCT and KLT are almost equal which was expected since the DCT transform kernel is almost optimum for an AR(1) sequence with \( a \to 1 \).
4.4. Application: Transform Coding of Line Spectral Frequency Coefficients

Jayant and Noll [1984, p. 558]. The measured entropies are also close to the desired rate \( R \) except for the lowest bit rates. This is explained by the fact that the high resolution assumption does not hold for low bit rates. Figure 4.6 shows that the KLT and DCT performs equally well and significantly better than PCM.

<table>
<thead>
<tr>
<th>( \bar{R} )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{PCM} )</td>
<td>4.63</td>
<td>1.16</td>
<td>0.29</td>
<td>72 \times 10^{-3}</td>
<td>18 \times 10^{-3}</td>
<td>4.5 \times 10^{-3}</td>
<td>1.1 \times 10^{-3}</td>
<td>0.28 \times 10^{-3}</td>
<td>0.28 \times 10^{-3}</td>
<td>0.28 \times 10^{-3}</td>
</tr>
<tr>
<td>( D_{KLT} )</td>
<td>0.86</td>
<td>0.21</td>
<td>54 \times 10^{-3}</td>
<td>13 \times 10^{-3}</td>
<td>3.4 \times 10^{-3}</td>
<td>0.84 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
</tr>
<tr>
<td>( D_{DCT} )</td>
<td>0.86</td>
<td>0.22</td>
<td>54 \times 10^{-3}</td>
<td>14 \times 10^{-3}</td>
<td>3.4 \times 10^{-3}</td>
<td>0.84 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
</tr>
</tbody>
</table>

Table 4.4: Analytical values for the distortion associated with uniform entropy coded quantization with \( \bar{R} \) bits per transform coefficients for PCM, KLT and DCT under high resolution approximations.

<table>
<thead>
<tr>
<th>( R )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{PCM} )</td>
<td>4.7</td>
<td>1.2</td>
<td>0.29</td>
<td>73 \times 10^{-3}</td>
<td>18 \times 10^{-3}</td>
<td>4.6 \times 10^{-3}</td>
<td>1.1 \times 10^{-3}</td>
<td>0.29 \times 10^{-3}</td>
<td>0.29 \times 10^{-3}</td>
<td>0.29 \times 10^{-3}</td>
</tr>
<tr>
<td>( D_{KLT} )</td>
<td>0.71</td>
<td>0.21</td>
<td>53 \times 10^{-3}</td>
<td>13 \times 10^{-3}</td>
<td>3.3 \times 10^{-3}</td>
<td>0.83 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
</tr>
<tr>
<td>( D_{DCT} )</td>
<td>0.72</td>
<td>0.21</td>
<td>54 \times 10^{-3}</td>
<td>13 \times 10^{-3}</td>
<td>3.3 \times 10^{-3}</td>
<td>0.84 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
<td>0.21 \times 10^{-3}</td>
</tr>
</tbody>
</table>

Table 4.5: Measured values for the distortion and entropy associated with uniform entropy coded quantization with \( R \) bits per transform coefficients for PCM, KLT and DCT under high resolution approximations. Figure 4.6 is a plot of the rate-distortion points in this table.

4.4.5 Evaluation of Performance

In the last section of this worksheet, the transform coding of the LSF coefficients is evaluated. The evaluation is based on two different tests:

1. The performance of the entropy coded quantizer
2. The overall system performance with respect to distortion of the masking curves

For the evaluations a test set of 84,000 new LSF vectors has been used.

Evaluation of Entropy Coded Quantizer

Figure 4.7 shows the rate-distortion performance for entropy coded quantization with KLT, DCT and PCM in the case of \( P = 15 \) and \( L = 10 \). The three curves show the average distortion versus the average rate per transform coefficient. The step size is found from equation 4.31 with assumed Laplace distributed inputs. In the case of PCM, the transform matrices are simply identity matrices, and the transform coefficients are therefore equal to the original LSF coefficients. It is clear from the figure that the KLT performs better than the DCT and PCM as expected. However, the DCT performs worse than PCM at high bit-rates which is somewhat surprising.

In figure 4.8 rate-distortions plots for the DCT are shown for different block length \( L \) with assumed Laplace distributed inputs. The plot confirms the expectation that
4 Transform Coding of Line Spectral Frequency Coefficients

Figure 4.6: Rate-distortion plot for PCM, DCT and KLT of AR(1) source with coefficient $a = 0.92$ and driven by white Gaussian noise with variance $\sigma_u^2 = 2$. 
4.4. Application: Transform Coding of Line Spectral Frequency Coefficients

Figure 4.7: Rate-distortion plot for the KLT, DCT and PCM in case of assumed Laplace distributed input.
Figure 4.8: Rate-distortion plot for the DCT of different block length $L$ in case of assumed Laplace distributed input.
longer block leads to better overall coding performance, but the plot also shows that the performance is not improved much for high values of \( L \) as compared to medium values of \( L \). Since \( L \) is proportional to the introduced delay, it seems reasonable to choose a value of 1 to 10 depending on the application.

**Evaluation of overall system performance**

In speech coding systems, the overall system performance of an LPC coding system is often evaluated by use of the log spectral distortion (LSD) \cite{Farvardin and Laroia, 1989}. The LSD measures the average mean square logarithmic distance between the original and reconstructed power spectral density (PSD) and is defined as \cite{Gray and Markel, 1976}

\[
D_{\text{LS}} = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \left[ 10 \log_{10} \frac{S(\omega)}{\hat{S}(\omega)} \right]^2 d\omega}
\]  

(4.43)

where \( S(\omega) \) and \( \hat{S}(\omega) \) are the original and reconstructed PSD, respectively.

Figure 4.9, 4.10 and 4.11 show the LSD for the DCT, KLT and PCM, respectively, for an average bit-rate of 1 bits per transform coefficient. The average LSD was 0.68 dB for the DCT while it was 0.40 dB for the KLT and 0.69 dB for PCM. This once again illustrates that the KLT is superior to the DCT and PCM with respect to coding performance, but the price is that the KLT transform kernels are dependent on the input statistics. In this test, however, the two KLT transform kernels were constant and calculated from the training data. The DCT and PCM performs, at this bit-rate of 1 bits per transform coefficient, equally well which was expected from the rate-distortion plot in figure 4.7.

**4.5 Conclusion**

This worksheet has described transform coding of LSF coefficients. Two typical transforms, the KLT and DCT, have been used and their performance in this application have been compared to PCM. The comparison has shown that the KLT yields a lower distortion for the same rate as compared to DCT and PCM. DCT performs better than PCM for bit rates lower than 1 bit per transform coefficient whereas PCM performs better than the DCT for higher bit rates.

The design of the quantizer was based on statistics of the transform coefficients. The statistics were found from training data and the performance was evaluated by use of test data. Both the training and test data were obtained from real songs. This statistical approach to rate-distortion optimal quantizer design involved a couple of assumptions on the statistics of the transform coefficient. For example the Laplace distribution was assumed as a statistical model for the transform coefficient and their variances were found from the training data. It could be interesting to take the operational approach
Figure 4.9: Measured log-spectral distortion for the 84,000 LSF test vectors using the DCT (top), frequency response for worst fit in terms of LSD (middle) and frequency response for best fit in terms of LSD (bottom). The average bit-rate was 1 bits and the average LSD was 0.68 dB.
Figure 4.10: Measured log-spectral distortion for the 84,000 LSF test vectors using the KLT (top), frequency response for worst fit in terms of LSD (middle) and frequency response for best fit in terms of LSD (bottom). The average bit-rate was 1 bits and the average LSD was 0.40 dB.
Figure 4.11: Measured log-spectral distortion for the 84,000 LSF test vectors using the PCM (top), frequency response for worst fit in terms of LSD (middle) and frequency response for best fit in terms of LSD (bottom). The average bit-rate was 1 bits and the average LSD was 0.69 dB.
to rate-distortion theory and compare the performance to the statistical approach, but due to time limitation this has to be left as further work.

**Bibliography**


Bibliography


