Low-bit-rate Informed Source Separation Using Decoder NTF and Efficient Parameter Coding

Von der Fakultät für Elektrotechnik und Informationstechnik der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von
Diplom-Ingenieur
Christian Rohlfing
aus Krefeld

Berichter:
Univ.-Prof. Dr.-Ing. Jens-Rainer Ohm
Univ.-Prof. Dr.-Ing. Peter Jax

Tag der mündlichen Prüfung: 26.04.2018

Vorwort

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Nachrichtentechnik der RWTH Aachen. An dieser Stelle möchte ich mich bei all denen bedanken, die mir die Fertigstellung dieses Buches erleichtert und mich bei den damit verbundenen Höhen und Tiefen unterstützt haben:

An erster Stelle danke ich herzlich Prof. Dr.-Ing. Jens-Rainer Ohm für die Betreuung meiner Arbeit. Als mein Doktorvater stand er mir stets mit guten Ratschlägen zur Seite und unterstützte so meinen wissenschaftlichen Werdegang. Prof. Dr.-Ing. Peter Jax danke ich für die Übernahme des Zweitgutachtens und das rege Interesse an meiner Arbeit.

Mein besonderer Dank gilt außerdem Dr.-Ing. Mathias Wien, bei dem ich in verschiedenen Forschungs- und Lebenslagen immer ein offenes Ohr finden konnte. Ich danke weiterhin meinem langjährigen Audiobürokollegen Julian Becker für die tolle Zusammenarbeit und seine Expertise, von der ich nicht nur in Sachen Audioquellentrennung profitieren durfte. Max Bläser danke ich besonders für die erstklassige Reiseleitung bei meinen Ausflügen in Teile der Video- und Entropiecodierung. Des Weiteren möchte ich mich bei den Probelesern meiner Arbeit und den Probehörern meiner Abschlussvorträge bedanken. Danke Kjell, Cordula, Jan, Jens und Max! Mein Dank gilt auch den Studierenden, die während ihrer Abschlussarbeiten oder Hilfskrafttätigkeiten zum Gelingen dieser Arbeit beigetragen haben.

I want to especially thank Antoine Liutkus for all the awesome discussions during the last years and the gentle introduction to the French way of signal processing. Merci pour tout! I also owe special thanks to Adrian Murtaza from Fraunhofer IIS for helping me with the reference software for Spatial Audio Object Coding and to Jérémy E.Cohen and Antoine Deleforge for taking me on journeys through the worlds of tensor decomposition and phase unmixing respectively.

Ich danke dem Kollegium für die tolle Atmosphäre am Institut, insbesondere bei Kaffeepausen, Mensagängen und Aktivitäten außerhalb des Instituts, und für all die guten Gespräche, Diskussionen und Blödeleien: Manon Bratschke, Gabriele Kaschel, Gudrun Klein, Ingrid Reißel und Myrjam Schiermeyer; Helmut Flasche und Ali Doha; Clemens Jansen, Felix Breuer, Nils Hochschwender und André Kleinen; Hossein Bakhshi-Golestani, Julian Becker, Max Bläser, Christopher Bulla, Olena Chubach, Christian Feldmann, Volker Gnann, Konstantin Hanke, Iris Heisterklaus, Cordula Heithausen, Peter Hosten, Fabian Jäger, Abin Jose, Maria Meyer, Thibaut Meyer, Ningqing Qian, Johannes Sauer, Jens Schneider, Martin Spiertz, Aleksandar Stojanovic, Uday Thakur, Mathias Wien und Bin Zhang. Besonders bedanken möchte ich mich außerdem bei den Mitgliedern der IENT-Band für all die legendären Auftritte und die nicht weniger legendären Proben; stellvertretend erwähnen möchte ich hier deren Bandleader Mathias und die sich niemals so nennende Frontfrau Cordula.

Zu guter Letzt danke ich meiner Familie, insbesondere meinen Eltern, und Kjell für ihre liebevolle Unterstützung und Geduld.

Aachen, im Mai 2018

Contents

1 Introduction **1**
1.1 Main Contributions . . . 2
1.2 Outline . . . 3

2 Fundamentals **5**
2.1 Time-Frequency Transform . . . 5
2.1.1 Short-Time Fourier Transform . . . 6
2.1.2 Modified Discrete Cosine Transform . . . 8
2.1.3 Logarithmic Frequency Mapping . . . 10
2.2 Nonnegative Factorization . . . 11
2.2.1 Cost-function and Update Rules . . . 12
2.2.2 Application to Audio Source Separation . . . 15
2.3 Quantization . . . 18
2.3.1 Uniform Quantization . . . 19
2.3.2 Non-uniform Quantization . . . 20
2.3.3 Lloyd-Max Algorithm . . . 21
2.3.4 Rate-distortion Optimized Quantization . . . 22
2.4 Entropy Coding . . . 22
2.4.1 Context-based Adaptive Binary Arithmetic Coding . . . 23
2.4.2 Run-length Coding . . . 27
2.5 Time-Frequency Masking for Source Separation . . . 28
2.5.1 Oracle Masks . . . 28
2.5.2 Masks obtained by Nonnegative Tensor Factorization . . . 28
2.6 Phase Re-estimation . . . 29
2.6.1 Griffin-Lim Algorithm . . . 29
2.6.2 Consistent Wiener Filtering . . . 30
2.7 Audio Object Coding – State of the Art . . . 30
2.7.1 Informed Source Separation . . . 31
2.7.2 Spatial Audio (Object) Coding . . . 32
2.8 Evaluation Environment . . . 33
2.8.1 Quality Assessment . . . 34
2.8.2 Rate-quality Optimization . . . 36
2.8.3 Test Sets . . . 38

3 Reference Algorithms **39**
3.1 Reference Algorithm for Informed Source Separation . . . 39
3.1.1 Encoder . . . 40
3.1.2 Decoder . . . 41
3.1.3 Residual Transmission: Coding-based Informed Source Separation . . . 41

3.2 Preliminary Experiments . . . 43
3.2.1 Time-frequency Transform . . . 44
3.2.2 Factorization . . . 45
3.2.3 Quantization . . . 46
3.3 Reference Algorithm for Blind Source Separation . . . 47
3.4 Summary . . . 49

4 Efficient Parameter Encoding **51**
4.1 Preliminary Evaluation . . . 52
4.2 Context Design . . . 53
4.2.1 Context Modeling Based on Bin Values . . . 55
4.2.2 Context Modeling Based on Integer Values . . . 59
4.3 Experimental Results . . . 60
4.4 Summary . . . 64

5 Parameter Re-estimation at Decoder **65**
5.1 Nonnegative Factorization of the Mixture . . . 65
5.1.1 Decoder Factorization Model . . . 67
5.1.2 Decoder Configurations . . . 68
5.1.3 Experimental Results . . . 70
5.2 Decoder NTF Constraints . . . 82
5.2.1 Approximation of Quantization Characteristic . . . 83
5.2.2 Constraint Formulation . . . 84
5.2.3 Preliminary Evaluation . . . 85
5.2.4 Experimental Results . . . 87
5.3 Summary . . . 89

6 Residual Parameter Coding **91**
6.1 Phase Residual . . . 91
6.2 Quantization of Phase Residual . . . 92
6.3 Generalized Complex Residual . . . 95
6.4 Experimental Results . . . 97
6.4.1 Phase Residual . . . 97
6.4.2 Complex Residual . . . 99
6.5 Summary . . . 102

7 Complete Algorithm **103**
7.1 Overview . . . 103
7.1.1 Low Bit Rate Scenario . . . 104
7.1.2 High Bit Rate Extension . . . 105
7.2 Comparison to SAOC . . . 106
7.3 Influence of Lossy-Encoded Mixture . . . 110
7.4 Summary . . . 113

8 Conclusions and Outlook **115**
8.1 Conclusions . . . 115
8.2 Outlook . . . 116

A Test Sets **119**
A.1 Test Set $\mathscr{A}$ 119
A.2 Test Set $\mathscr{B}$ 119

B Quantized Matching Derivations **123**

C CABAC **125**
C.1 CABAC Context Identifiers 125
C.2 Bin-value based Context Model Interpretation 125
C.3 Results with GZIP Jointly Encoding Parameters as Baseline 127

D Lower Bound **129**

Bibliography **133**

Abbreviations

AAC Advanced Audio Coding

BAC Binary Arithmetic Coding

BD-BR Bjøntegaard Delta Bit Rate

BSS Blind Source Separation

CABAC Context-based Adaptive Binary Arithmetic Coding

CISS Coding-based Informed Source Separation

CWF Consistent Wiener Filtering

DCT Discrete Cosine Transform

DFT Discrete Fourier Transform

EG Exponential Golomb

GL Griffin-Lim Algorithm

IDFT Inverse Discrete Fourier Transform

IMDCT Inverse Modified Discrete Cosine Transform

ISS Informed Source Separation

ISTFT Inverse Short Time Fourier Transform

IS Itakura Saito

KL Kullback Leibler

KLT Karhunen–Loève Transform

LM Lloyd-Max Algorithm

MDCT Modified Discrete Cosine Transform

NMF Nonnegative Matrix Factorization

NTF Nonnegative Tensor Factorization

RDOQ Rate-distortion Optimized Quantization

RLC Run-length Coding

SAC Spatial Audio Coding

SAOC Spatial Audio Object Coding

SAR Signal-to-Artifacts Ratio

SBSS Semi-blind Source Separation

SDR Signal-to-Distortion Ratio

SIR Signal-to-Interferences Ratio

SNR Signal-to-Noise Ratio

STFT Short Time Fourier Transform

SVD Singular Value Decomposition

TF Time-frequency

TU Truncated unary

Notation

Mathematical Definitions

$\mathbb{C}$, $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{N}$	set of complex, real, nonnegative real and natural numbers
$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M,1} & a_{M,2} & \cdots & a_{M,N} \end{pmatrix}$	$M \times N$ matrix $\mathbf{A}$. Matrices and tensors are denoted with a bold, upright symbol without being indexed. Single scalar elements are denoted with lowercase, non-bold, italic symbols and are indexed with subscripts.
$\mathbf{a}_{\bullet,n} = \left(a_{1,n}, \cdots, a_{M,n}\right)^\top$	nth column vector of matrix $\mathbf{A}$.
$\mathbf{a}_{m,\bullet} = \left(a_{m,1}, \cdots, a_{m,N}\right)$	mth row vector of matrix $\mathbf{A}$
$\mathbf{A}_{\bullet,\bullet,o}$	oth slice of a $M \times N \times O$ tensor $\mathbf{A}$. The result is a matrix of size $M \times N$. The operator $\bullet$ can be used on any dimension of a tensor to index all elements of the tensor in this dimension.
$\mathbf{A}^\top$	matrix transpose
$\mathbf{A} \cdot \mathbf{B}$, $\frac{\mathbf{A}}{\mathbf{B}}$	element-wise multiplication and division of matrices $\mathbf{A}$ and $\mathbf{B}$ of same size
$\nabla_{\mathbf{A}} f\left(\mathbf{A}\right)$	gradient of function $f\left(\mathbf{A}\right)$ with respect to matrix $\mathbf{A}$
$\jmath = \sqrt{-1}$	imaginary unit
$\underline{a} = \mathrm{Re}\left\{\underline{a}\right\} + \jmath\, \mathrm{Im}\left\{\underline{a}\right\}$	complex numbers are denoted with an underlined symbol and consist of real and imaginary part
$\underline{a}^*$	complex conjugate
$\lvert a \rvert = \sqrt{a a^*}$	modulus of complex number

Variables and Signals in the Time-frequency Domain

J	number of sources
F_s	sampling frequency
N_w, N_h	window length and hop size of STFT/STFT
N_ny	number of spectral coefficients per STFT frame
T	number of STFT/MDCT frames
F	number of mel filters

$\underline{s}_{f,t,j}$, $\underline{\mathbf{S}} \in \mathbb{C}^{N_{\text{ny}} \times T \times J}$	source in time-frequency domain
$\underline{\tilde{s}}_{f,t,j}$, $\underline{\tilde{\mathbf{S}}} \in \mathbb{C}^{N_{\text{ny}} \times T \times J}$	estimated source in time-frequency domain
$\underline{x}_{f,t}$, $\underline{\mathbf{X}} \in \mathbb{C}^{N_{\text{ny}} \times T}$	mixture in time-frequency domain
$v_{f,t,j}$, $\mathbf{V} \in \mathbb{R}_+^{F \times T \times J}$	general notation of (mel-filtered) amplitude spectrogram
$v_{\mathbf{s},f,t,j}$, $\mathbf{V}_\mathbf{s} \in \mathbb{R}_+^{F \times T \times J}$	(mel-filtered) source amplitude spectrograms
$v_{\mathbf{x},f,t,j}$, $\mathbf{V}_\mathbf{x} \in \mathbb{R}_+^{F \times T}$	(mel-filtered) mixture amplitude spectrogram

Nonnegative Tensor Factorization Parameters

K	number of NTF components
β	parameter of β-divergence
N_{it}	number of iterations
$\hat{v}_{f,t,j}$, $\hat{\mathbf{V}}(\Theta) \in \mathbb{R}_+^{F \times T \times J}$	NTF approximation $\hat{\mathbf{V}}$
$\Theta = \{\mathbf{W}, \mathbf{H}, \mathbf{Q}\}$	NTF parameters: Frequency basis $\mathbf{W} \in \mathbb{R}_+^{F \times T}$, temporal activations $\mathbf{H} \in \mathbb{R}_+^{T \times K}$ and grouping $\mathbf{Q} \in \mathbb{R}_+^{J \times K}$ with elements w_{fk}, h_{tk} and q_{jk}
$\Theta_\mathbf{s} = \{\mathbf{W}_\mathbf{s}, \mathbf{H}_\mathbf{s}, \mathbf{Q}_\mathbf{s}\}$	source NTF parameters, calculated at encoder with input $\mathbf{V}_\mathbf{s}$
$\Theta_\mathbf{x} = \{\mathbf{W}_\mathbf{x}, \mathbf{H}_\mathbf{x}, \mathbf{Q}_\mathbf{x}\}$	mixture NTF parameters, calculated at decoder with input $\mathbf{V}_\mathbf{x}$

Quantization

N_{q}	number of reconstruction values
x_m, $\mathbf{x} \in \mathbb{R}^M$	general input of quantizer of length M
c_g, $\mathbf{c} \in \mathbb{R}^{N_{\text{q}}}$	reconstruction values
g_m, $\mathbf{g} \in \mathbb{N}^M$	quantization indices mapping a reconstruction value c_g to an element of x_m with $1 \leq g_m \leq N_{\text{q}}$
$b_n \in \{0, 1\}$	bin at position n of bin-string $\mathbf{b}$
$\bar{x}_m$, $\bar{\mathbf{x}} \in \mathbb{R}^M$	quantized version of $\mathbf{x}$ with $\bar{x}_m = c_{g_m}$
A	A-law companding factor

Quantized Nonnegative Tensor Factorization Parameters

$\bar{\Theta}_\mathbf{s} = \left\{\bar{\mathbf{W}}_\mathbf{s}, \bar{\mathbf{H}}_\mathbf{s}, \bar{\mathbf{Q}}_\mathbf{s}\right\}$	quantized source NTF parameters
$\mathbf{c}_{\mathbf{W}_\mathbf{s}}$, $\mathbf{c}_{\mathbf{H}_\mathbf{s}}$	reconstruction values corresponding to $\mathbf{W}_\mathbf{s}$ and $\mathbf{H}_\mathbf{s}$
$\mathbf{G}_{\mathbf{W}_\mathbf{s}}$, $\mathbf{G}_{\mathbf{H}_\mathbf{s}}$	quantization indices corresponding to $\mathbf{W}_\mathbf{s}$ and $\mathbf{H}_\mathbf{s}$, of same size as $\mathbf{W}_\mathbf{s}$ and $\mathbf{H}_\mathbf{s}$
$\tilde{\Theta}_\mathbf{x} = \left\{\tilde{\mathbf{W}}_\mathbf{x}, \tilde{\mathbf{H}}_\mathbf{x}, \mathbf{Q}_\mathbf{x}\right\}$	soft-quantized mixture NTF parameters
$g_{f,k}$, $\mathbf{G} \in \mathbb{N}^{F \times T}$	abbreviation for quantization indices $\mathbf{G}_{\mathbf{W}_\mathbf{s}}$ with $1 \leq g_{f,k} \leq N_{\text{q}}$
$b_n^{f,k} \in \{0, 1\}$, $\mathbf{b}^{f,k}$	bin at position n of bin-string $\mathbf{b}^{f,k}$ corresponding to binarization of $g_{f,k}$

1 Introduction

Source separation deals with the problem of extracting sources out of a mixture. In the case of audio source separation, the sources are usually recordings of musical instruments, playing alongside each other. The mixture is often produced by a professional sound engineer during the mixing process in the studio. Applications such as re-mixing or spatialization need isolated recordings of the single sources of a music mixture. The field of audio source separation can be roughly categorized by the amount of *prior information* about the sources. In the case of Blind Source Separation (BSS), no information about the particular mixture at hand is available. These algorithms often use generic models of the sources to conduct the separation. Supervised source separation needs either some sort of interaction with the user or pre-trained models, e.g. obtained by deep learning. Some methods take even the score of the musical piece played by the instruments or videos capturing the motion of the instrument's player as input. These algorithms lead to improved separation quality compared to BSS by exploiting the prior information about the sources. However, the separation quality yielded in these cases is often not sufficient for large audience applications.

The special case of Informed Source Separation (ISS) uses source separation methods for efficient *coding* of the sources, in this field also referred to as objects. The basic idea is to use a source separation algorithm to extract the sources given the mixture; the same general objective as for source separation. The main point is that the source separation step is supported by a compact set of *side information* which is extracted at the encoding side. Here, the original recordings of each source must be at hand. The resulting side information is then transmitted to the decoder which applies source separation given the mixture. This scheme is denoted as Informed Source Separation (ISS) and unifies due to its nature the research fields of source separation and audio coding. The audio coding community itself has brought forward several audio coding standards dealing with encoding multi-channel audio signals or, more recently, multiple audio objects.

This thesis deals with ISS algorithms compressing the audio objects with nonnegative factorization methods such as Nonnegative Tensor Factorization (NTF). These methods are widely used in the source separation community as they allow for an efficient description of single sound events present in audio recordings. This property can be exploited for compression as well: Several ISS methods exist using NTF for encoding the objects in the Time-frequency (TF) domain. The resulting parameters are quantized, encoded to a bit stream and sent to the decoder. Here, the source separation step usually consists of TF masking of the mixture, also referred to as Wiener filtering. The corresponding TF masks are calculated in the decoder given the NTF parameters in the side information.

In this thesis, several limitations of existing NTF-based ISS methods are addressed:

- In most ISS methods using NTF as compression, the resulting parameters are quantized and subsequently coded with entropy coding methods such as Huffman Coding or Arithmetic Coding. These methods are usually not adapted to the task of encoding the

quantized NTF parameters which are typically strongly structured. This structure is exploited only to some extent by these methods.

- In the decoder of most of the NTF-based ISS methods, the source separation step solely consists of TF masking. This requires to always send the full set of NTF parameters which prevents operating at very low bit rates.

- For higher bit rates, methods exist combining both ISS and source coding, denoted as Coding-based Informed Source Separation (CISS). Here, additional residuals accounting for possible deviations of the source estimates from the original sources are transmitted. However, CISS requires computational complexity available at the decoder to apply the residual in the TF domain.

These limitations are addressed in this thesis. The main contributions and the outline of this thesis are summarized in Sections 1.1 and 1.2 respectively. Throughout this work, two ideal assumptions on the mixture are made:

1. It can be shown that the quality of the estimated audio objects decreases with the quality of the encoding of the mixture as shown in Section 7.3. Hence, it is typically assumed that the mixture is transmitted with high quality to the decoder when dealing with audio object coding. In practice, two scenarios are probable. The mixture is either compressed losslessly (pulse-code modulation (PCM) as used in the .wav format or audio-CD) or encoded lossy with an existing audio coding scheme, e.g. Advanced Audio Coding (AAC). In case of lossless compression, the side information is embedded with high-rate data hiding techniques as proposed in e.g. [PGB14] and in the case of AAC compressing the mixture, the side information is simply stored in the AAC metadata. In this thesis, the second scenario is considered in an extra chapter of this thesis. Here, the impact of the lossy mixture encoding on the ISS performance is evaluated. Note that backward compatibility is ensured because a standard AAC decoder is still able to play back the mixture.

2. The mixture is assumed to be mono. A multi-channel (e.g. stereo) mixture requires the estimation of a mixing matrix, describing the contributions of each source to each channel of the mixture. Estimation algorithms, as proposed in e.g. [DVG10; LBR13], are not in the focus of this thesis.

1.1 Main Contributions

The contributions of this thesis can be roughly divided into three parts as they tackle advanced parameter encoding, an extension of the decoder allowing for low bit rates and an efficient method of quantizing a residual between original sources in the encoder and estimated sources in the decoder. In the following, the aspects of these contributions are summarized.

Most of the existing ISS methods use standard coding schemes such as Huffman Coding or Arithmetic Coding. The first contribution aims at encoding the quantized NTF parameters with a more advanced coding method. The NTF parameters are strongly structured when applied on audio data. This structure can be exploited by an efficient coding algorithm:

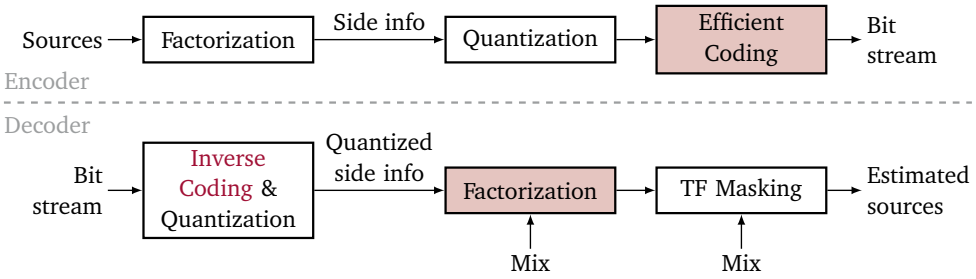


Figure 1.1 Contributions to Informed Source Separation. Third contribution not depicted.

Context-based Adaptive Binary Arithmetic Coding (CABAC), used in the video coding community, adapts to local statistics in the data and approaches conditional entropy depending on the design of so-called *context models*. In this thesis, CABAC is adapted to the field of factorization-based ISS by proposing suitable context models based on the typical structure of the NTF parameters. These context models are thoroughly evaluated and CABAC is compared to other existing coding schemes.

The second contribution is focused on enabling lower bit rates. In many NTF-based ISS methods, the source separation step in the decoder solely consists of TF masking. In this thesis, the decoder is extended to use a complete NTF-based *blind* source separation algorithm by adding a factorization block to the decoder. The encoder can decide to omit transmission of certain NTF parameters which are then estimated by the proposed NTF in the decoder with the mixture as observation and the transmitted parameters as initialization. This extension also permits the decoder to run blindly without any transmitted side information. Furthermore, constraints are proposed which steer the decoder factorization process during its updates.

In contrast to the second contribution, the third contribution aims at higher bit rates. The main limitation of the Wiener filter in the decoder is that all estimated sources are combined with the mixture's phase. In addition to this, the magnitude estimation in the decoder may cause errors as well. To correct possible errors of the source separation step in the decoder, the encoder can compute residuals in the TF domain. In this thesis, it is proposed to quantize these residuals under a rate-quality constraint with Rate-distortion Optimized Quantization (RDOQ), increasing separation performance for higher bit rates while efficiently constraining the rate necessary to transmit the residuals.

1.2 Outline

This thesis is structured as follows. In Chapter 2, fundamentals of source separation, quantization and entropy coding are summarized. A summary of state-of-the-art audio object coding schemes is given as well and the evaluation environment is outlined. In Chapter 3, a reference ISS algorithm is introduced and evaluated which is then used throughout the remainder of this thesis as a baseline on which to improve and to compare against. Additionally, a reference BSS algorithm is briefly summarized as well.

The main contributions, as summarized in Section 1.1, are subject of the following chapters: Chapter 4 comprises the first contribution of this thesis: CABAC as an efficient entropy coding method is introduced to ISS for coding the quantized NTF parameters. Different con-

text models are proposed and evaluated which are then used for comparing CABAC against other entropy coding schemes. In Chapter 5, the second contribution of this thesis is introduced. The ISS reference decoder is extended to contain a complete BSS algorithm by adding a second factorization block to the decoder. This block is initialized with the quantized NTF parameters originally used for Wiener filtering. Additional constraints to the decoder factorization are proposed to prevent the factorization process from deviating too much from its initialization. In Chapter 6, the third contribution is presented. Aiming at higher bit rates, a residual in TF domain between original and estimated sources can be calculated, quantized, and transmitted to the decoder to correct possible errors in the source estimation process of the decoder.

In Chapter 7, the three contributions are summarized and jointly evaluated and compared to other audio object coding methods. Finally, conclusions and an outlook on possible future work are given in Chapter 8.

2 Fundamentals

This thesis deals with coding of digital audio signals. In the following, all signals are assumed to be represented digitally, meaning both sampled with sampling frequency F_s and quantized. The test signals used for evaluation throughout this thesis are available in Audio CD quality, sampled with $F_s = 44100\,\text{Hz}$ and quantized with at least 16 bit per sample yielding high amplitude resolution. Therefore, the quantization process of the analog-to-digital conversion is neglected from now on. Moreover, M samples of a discrete, sampled signal are stacked to a real-valued vector $\mathbf{x}$ with

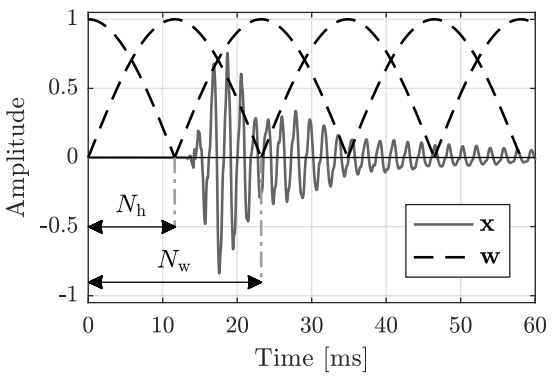
$$\mathbf{x} = \left(x_1, x_{2,}, \ldots, x_m, \ldots, x_M\right)^\top. \tag{2.1}$$

In the following sections of this chapter, basic methods and concepts are summarized. This thesis deals with ISS schemes utilizing factorization methods for compressing the sources in the TF domain. Two exemplary transforms, mapping time-domain signals to their TF representations, are discussed in Section 2.1. The factorization process used throughout this thesis is detailed in Section 2.2 and yields compact side information which has to be sent to the decoder. Quantization and entropy coding are summarized in Sections 2.3 and 2.4 respectively. The source separation step in the decoder utilizes TF masking as described in Section 2.5 whereas phase re-estimation algorithms and a generalized TF masking process are discussed in Section 2.6. State-of-the-art algorithms for audio object coding, including ISS, are summarized in Section 2.7. To assess the performance of ISS algorithms, the algorithms are tested on different test mixtures. The quality of the estimated sources as well as the bit rate, which is necessary to transmit the side information, have to be evaluated. The evaluation procedure is summarized in Section 2.8.

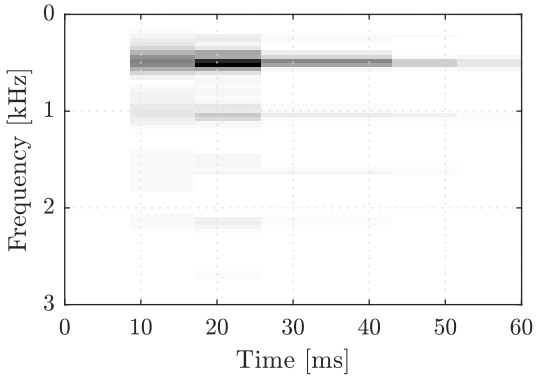
2.1 Time-Frequency Transform

Typical properties of audio signals such as pitch or timbre are exposed more clearly in the frequency domain. The Fourier transform is commonly used to transform a time-domain signal to the frequency domain. However, audio signals usually vary over time. To analyze the signal's frequency content for given time instances, the Short Time Fourier Transform (STFT) [AR77] is often used to transform time-domain signals to the Time-frequency (TF) domain.

In the following, the STFT and some of its properties are briefly explained in Section 2.1.1. Section 2.1.2 discusses another transform, the Modified Discrete Cosine Transform (MDCT) [PB86; PJB87], which has usually better coding properties than the STFT. In Section 2.1.3, a logarithmic mapping of the frequency axis of the TF-transformed signals is briefly discussed.



(a) Time-domain signal $\mathbf{x}$ and shifted versions of window function $\mathbf{w}$ of size $N_w = 23.22$ ms with shift $N_h = 11.61$ ms.



(b) Magnitude spectrogram $|\underline{\mathbf{Y}}| = |\text{STFT}\{\mathbf{x}\}|$.

Figure 2.1 Illustration of a spectrogram of a harmonic note [Spi12].

2.1.1 Short-Time Fourier Transform

To be able to understand the Short Time Fourier Transform (STFT), it is important to first explain the Discrete Fourier Transform (DFT) which the STFT uses as its main building block. The DFT of time-domain vector $\underline{\mathbf{x}}$ as well as the corresponding inverse transform, the Inverse Discrete Fourier Transform (IDFT), is given as

$$\underline{y}_f = \sum_{m=1}^{M} \underline{x}_m \exp\left(-\jmath 2\pi \frac{(m-1)(f-1)}{N}\right), \quad \underline{x}_m = \frac{1}{M} \sum_{f=1}^{M} \underline{y}_f \exp\left(\jmath 2\pi \frac{(m-1)(f-1)}{N}\right) \tag{2.2}$$

with frequency bin $1 \leq f \leq M$. In this thesis, only audio signals are transformed. Therefore, $\mathbf{x}$ is assumed to be real-valued from now on. In this case, the corresponding spectrum $\underline{\mathbf{y}}$ is Hermitian with

$$\underline{y}_{f+1} = \underline{y}^*_{M-f-1} \tag{2.3}$$

for $1 \leq f \leq M$. Due to this symmetry, it is sufficient to store and process only the first $N_{ny} = \frac{M}{2} + 1$ coefficients. Prior to the IDFT, the missing coefficients are simply calculated with Equation (2.3).

The Short Time Fourier Transform (STFT) is based on the windowed DFT with window $\mathbf{w}$ of length N_w [AR77]. The window is applied prior to the DFT to prevent the spectral leakage effect [Mal08]. Instead of transforming the whole time-domain signal, segments of length N_w are extracted from $\mathbf{x}$, multiplied with window $\mathbf{w}$ and transformed with the DFT independently. The segments are shifted by N_h samples where N_h is often called hop size. This procedure is summarized in the following equation as

$$\underline{y}_{f,t} = [\text{STFT}\{\mathbf{x}\}]_{f,t} = \sum_{m=1}^{N_w} x_{m+(t-1)N_h} w_m \exp\left(-\jmath 2\pi \frac{(m-1)(f-1)}{N_w}\right), \tag{2.4}$$

with segment index $1 \leq t \leq T$ and

$$T = \left\lfloor \frac{M - N_w}{N_h} \right\rfloor + 1 \tag{2.5}$$

denoting the number of segments. Note that for $w_m = 1$ for all m and $T = 1$, Equation (2.4) falls back to the DFT as shown in Equation (2.2). In this thesis, the DFT size is fixed to the window length N_w. The output of the STFT is the complex valued matrix $\underline{\mathbf{Y}}$ which is of size $N_\mathrm{w} \times T$. Since $\mathbf{x}$ is still assumed to be real-valued and Equation (2.3) still holds for each spectrum $\underline{\mathbf{y}}_{\bullet,t}$, only $N_\mathrm{ny} = \frac{N_\mathrm{w}}{2} + 1$ frequency bins are stored, thus decreasing the size of $\underline{\mathbf{Y}}$ to $N_\mathrm{ny} \times T$. Figure 2.1 shows an exemplary time-domain representation $\mathbf{x}$ of a harmonic note in Figure 2.1a and the corresponding magnitude spectrogram $\left|\underline{\mathbf{Y}}\right| = |\mathrm{STFT}\{\mathbf{x}\}|$ in Figure 2.1b. Additionally, shifted versions of window function $\mathbf{w}$ are shown in Figure 2.1a to illustrate the STFT procedure as given in Equation (2.4): $\mathbf{w}$ is shifted by multiples of N_h, namely by $(t-1)N_\mathrm{h}$. Input $\mathbf{x}$ is then consecutively multiplied by these shifted versions and transformed to the frequency domain by the DFT. The resulting spectrum is stored in a column $\underline{\mathbf{y}}_{\bullet t}$ of $\underline{\mathbf{Y}}$ as shown in Figure 2.1b.

The inverse transform, Inverse Short Time Fourier Transform (ISTFT), is conducted in two steps: First, the spectrum $\underline{\mathbf{y}}_{\bullet,t}$ of frame t is transformed back to time-domain by the IDFT. The resulting samples are then synthesized by overlap-add. These steps are summarized as

$$x_{m+(t-1)N_\mathrm{h}} \leftarrow x_{m+(t-1)N_\mathrm{h}} + w_m \frac{1}{N_\mathrm{w}} \sum_{f=1}^{N_\mathrm{w}} \underline{y}_{f,t} \exp\left(\jmath 2\pi \frac{(m-1)(f-1)}{N_\mathrm{w}}\right) \tag{2.6}$$

and abbreviated with $\mathbf{x} = \mathrm{ISTFT}\left\{\underline{\mathbf{Y}}\right\}$.

In the following, the overlap is fixed to 50% which means that $N_\mathrm{h} = \frac{N_\mathrm{w}}{2}$. On the one hand, larger overlaps (e.g. 75%) are reported to increase separation quality in a blind source separation scenario [Bec16]. On the other hand, in the informed case dealt with in this thesis, the number of parameters to transmit is significantly increased. This can be explained by the fact that decreasing N_h results in an increasing number of segments T, cf. (2.5).

In this thesis, the same window is used for analysis (2.4) and synthesis (2.6). To obtain perfect reconstruction, $\mathbf{x} = \mathrm{ISTFT}\{\mathrm{STFT}\{\mathbf{x}\}\}$, the window $\mathbf{w}$ has to fulfill

$$\sum_{t=-\infty}^{\infty} w^2_{m-tN_\mathrm{h}} = 1 \text{ , for all } m.$$

Many different window functions $\mathbf{w}$ are applicable, some choices are e.g. detailed in [Mal08]. Throughout this thesis, the square root of the Hann-window is used for both the STFT and the MDCT

$$w_m = \sqrt{\frac{1}{2} - \frac{1}{2}\cos\left(\frac{2\pi m}{N_\mathrm{w}}\right)}\,,\ 1 \le m \le N_\mathrm{w}\,. \tag{2.7}$$

One particular contribution of this thesis, further discussed in Section 2.6, deals with the re-estimation of distorted phase spectrograms. With Euler's formula, spectrogram $\underline{\mathbf{Y}}$ can be decomposed into magnitude $\left|\underline{\mathbf{Y}}\right|$ and phase $\angle\underline{\mathbf{Y}}$ as

$$\underline{\mathbf{Y}} = \left|\underline{\mathbf{Y}}\right| \exp\left(\jmath \angle \underline{\mathbf{Y}}\right).$$

Some of the considered algorithms for phase re-estimation as further presented in Section 2.6, make use of the consistency property of the STFT as depicted in Figure 2.2. Computing the STFT of any real-valued input signal $\mathbf{x} \in \mathbb{R}^N$ always results in a consistent spectrogram. However, this does not hold true the other way round. Successive columns of a

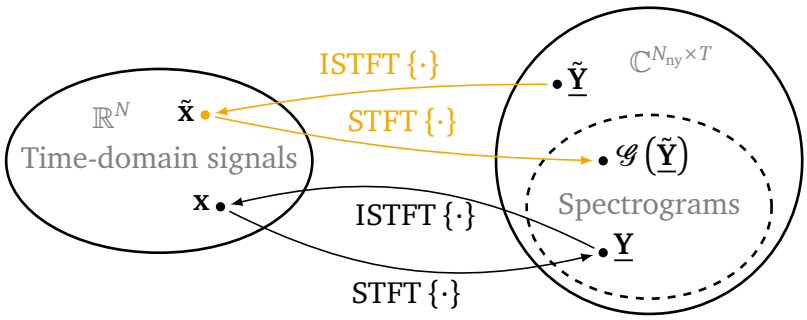


Figure 2.2 STFT consistency: $\underline{\mathbf{Y}}$ is a consistent spectrogram whereas $\tilde{\underline{\mathbf{Y}}}$ is an inconsistent spectrogram which can be mapped to the set of consistent STFT spectrograms with function $\mathscr{G}\left(\tilde{\underline{\mathbf{Y}}}\right)$ as marked in orange. [SD11; GKR15].

consistent STFT spectrogram $\underline{\mathbf{Y}}$ are calculated from overlapping frames (c.f. Equation (2.4)) and share signal information to some extent [GKR15]. Due to this fact, it becomes clear that not any complex matrix $\tilde{\underline{\mathbf{Y}}} \in \mathbb{C}^{N_{\text{ny}} \times T}$ is a proper STFT spectrogram. This is already the case if a formerly consistent spectrogram is modified (e.g. by Wiener filtering, cf. Section 2.5). However, any complex $N_{\text{ny}} \times T$ matrix $\tilde{\underline{\mathbf{Y}}}$ can be mapped to the set of consistent STFT spectrograms as shown in Figure 2.2 with the following operation

$$\mathscr{G}\left(\tilde{\underline{\mathbf{Y}}}\right) = \text{STFT}\left\{\text{ISTFT}\left\{\tilde{\underline{\mathbf{Y}}}\right\}\right\} \tag{2.8}$$

which allows for a mathematical definition of consistent spectrograms: A spectrogram $\underline{\mathbf{Y}}$ is consistent if $\underline{\mathbf{Y}} = \mathscr{G}\left(\underline{\mathbf{Y}}\right)$. Equation (2.8) is used by some phase re-estimation algorithms as summarized further in Section 2.6.

2.1.2 Modified Discrete Cosine Transform

The Modified Discrete Cosine Transform (MDCT) [PB86; PJB87] is widely used in the audio coding community, e.g. for AAC [MPE99] or OPUS [VVT12]. The MDCT is critically sampled, meaning that the number of elements of the input signal is equal to the number of transform coefficients.

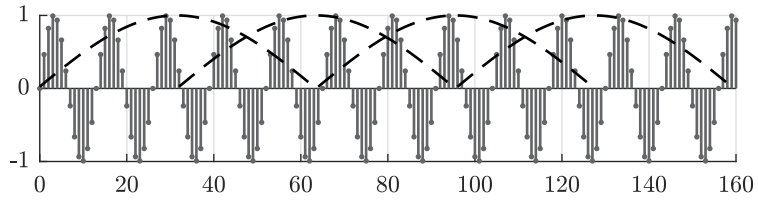
The Modified Discrete Cosine Transform (MDCT) is based on the Discrete Cosine Transform (DCT)-IV [BG03]. Similar to the STFT, overlapping segments of length N_{w} are windowed and transformed by the DCT-IV. As discussed in Section 2.1.1, the hop size is fixed here to $N_{\text{h}} = \frac{N_{\text{w}}}{2}$ as well. The resulting coefficients $\mathbf{Z}$ are real-valued

$$z_{f,t} = \left[\text{MDCT}\left\{\mathbf{x}\right\}\right]_{f,t} = \sqrt{\frac{4}{N_{\text{w}}}} \sum_{m=1}^{N_{\text{w}}} x_{m+(t-1)N_{\text{h}}} w_m \cos\left[\frac{2\pi}{N_{\text{w}}}\left(m-1+N_{\text{off}}\right)\left(f-1+\frac{1}{2}\right)\right], \tag{2.9}$$

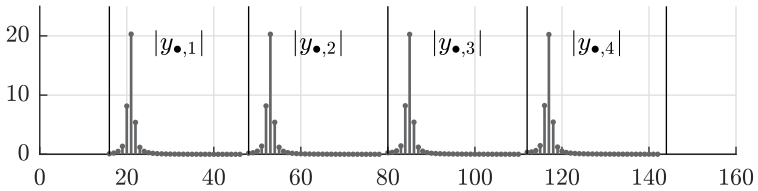
with bins $1 \leq f \leq \frac{N_{\text{w}}}{2}$ and the fixed offset $N_{\text{off}} = \frac{1}{2}\left(\frac{N_{\text{w}}}{2}+1\right)$. The result is real-valued $\frac{N_{\text{w}}}{2} \times T$ matrix $\mathbf{Z}$. As window function, the square-root Hann-window in Equation (2.7) is used.

The inverse operation, the Inverse Modified Discrete Cosine Transform (IMDCT), is abbreviated with $\mathbf{x} = \text{IMDCT}\left\{\mathbf{Z}\right\}$ and computed with overlap-add as well as the ISTFT shown in Equation (2.6)

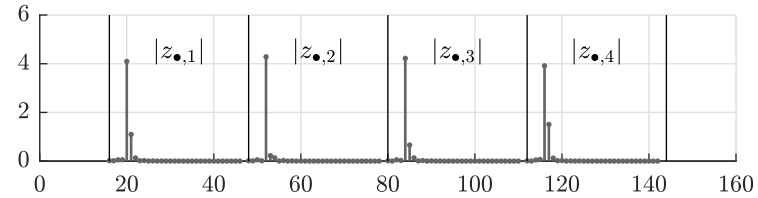
$$x_{m+(t-1)N_{\text{h}}} \leftarrow x_{m+(t-1)N_{\text{h}}} + w_m \sqrt{\frac{4}{N_{\text{w}}}} \sum_{f=1}^{N_{\text{w}}/2} z_{f,t} \cos\left[\frac{2\pi}{N_{\text{w}}}\left(m-1+N_{\text{off}}\right)\left(f-1+\frac{1}{2}\right)\right]. \tag{2.10}$$



(a) Time-domain input **x** and windows **w**.



(b) STFT magnitude spectra $\left|\underline{\mathbf{y}}\right|_{\bullet,t}$.



(c) MDCT magnitude spectra $\left|\mathbf{z}_{\bullet,t}\right|$.

Figure 2.3 Comparison of MDCT to STFT [Nia06].

In the following, the MDCT is briefly compared to the STFT as summarized in the previous Section 2.1.1. Due to the nature of the used basis functions, the STFT and the MDCT are closely related. The MDCT can even be computed by means of a modified STFT [BG03]. In this thesis however, two other properties are of special interest:

- The discussed TF transforms are used in the field of audio coding in this thesis. It is hence of interest whether the TF transform is critically sampled assuming perfect reconstruction. This is only fulfilled if the transform yields the same number of TF coefficients as the number of input samples.

- The magnitude of the TF coefficients are subject to a subsequent factorization process as further discussed in Section 2.2. The magnitude coefficients should be suitable for factorization, meaning that the factorization should be able to yield a compact description of the magnitude and not be restrained too much.

When using overlapping windows, the STFT matrices have more elements than the corresponding signals in time-domain: Given M real-valued input values, the complex STFT output $\underline{\mathbf{Y}}$ has $MN_\mathrm{w}/N_\mathrm{h}$ real-valued elements[1] which are needed for perfect reconstruction

[1] Due to symmetry (2.3), each of the T spectra stored in the columns $\underline{\mathbf{y}}_{\bullet,t}$ of $\underline{\mathbf{Y}}$ have $N_\mathrm{ny} = \frac{N_\mathrm{w}}{2} + 1$ coefficients. Two of them, at $f = 1$ and $f = N_\mathrm{ny}$, are real-valued, the other $\frac{N_\mathrm{w}}{2} - 1$ values for $1 < f < N_\mathrm{ny}$ are complex-valued. In total, each of $T \approx M/N_\mathrm{h}$ columns comprises N_w real-valued elements which approximately adds up to $MN_\mathrm{w}/N_\mathrm{h}$ real-valued coefficients of $\underline{\mathbf{Y}}$.

to completely describe the input. Since it is assumed that the hop size is fixed to $N_\mathrm{h} = \frac{N_\mathrm{w}}{2}$, the total amount of real-valued elements is $2M$. In contrast to this, the real-valued MDCT output $\mathbf{Z}$ has the same number of coefficients as the number of input samples[2]. The MDCT is critically sampled which is an advantage over the STFT [BSH08].

Figure 2.3 shows both MDCT and STFT coefficients for a sine signal given in Figure 2.3a. Comparing the magnitude STFT coefficients $\left|\underline{\mathbf{Y}}\right|$ in Figure 2.3b to the magnitude coefficients $|\mathbf{Z}|$ obtained by the MDCT, it becomes clear that the MDCT coefficients are not invariant with respect to phase-shifts of input $\mathbf{x}$. This is a disadvantage compared to the STFT magnitudes with respect to the subsequent factorization algorithm (cf. Section 2.2) which takes magnitude coefficients as input. In the case of the STFT output as shown in Figure 2.3b, only one basis spectrum (e.g. at $t = 1$) would be sufficient to approximate the whole magnitude spectrogram (itself and the other spectra at $2 \leq t \leq 4$). For the MDCT coefficients, several spectra are needed for describing all TF points (at least the spectra at $t = 1$ and $t = 2$ since the second highest peaks deviate from each other). Thus, compared to the STFT, the expected factorization would be inferior.

In summary, the STFT needs more coefficients than the MDCT but its magnitude is invariant to phase shifts which could lead to better separation results. STFT and MDCT are compared experimentally in Section 3.2.1.

2.1.3 Logarithmic Frequency Mapping

As already proposed in the literature e.g. [FCC08; Spi12], a logarithmic frequency mapping of the spectral domain of the TF coefficients can be applied. In [Spi12], a more detailed discussion and comparison to other logarithmic transforms such as the constant Q transform [Bro91] is given. Here, the motivation and procedure of the used logarithmic frequency mapping are only briefly summarized.

The mel scale [SVN37]

$$f_\mathrm{Mel} = 2595 \log_{10}\left(1 + \frac{f_\mathrm{Hertz}}{700}\right) \tag{2.11}$$

is a frequency scale which maps linear frequencies f_Hertz to logarithmic frequencies f_Mel to approximate the human frequency perception which behaves rather logarithmically than linearly. The coefficients in Equation (2.11) are chosen such that $1\,\mathrm{kHz} = 1\,\mathrm{kMel}$. In e.g. [FCC08; Spi12] it is then proposed to map the STFT spectrograms, which are in the linear frequency domain, to the mel domain by applying a mel filter bank, consisting of overlapping triangular filters. The center frequency and width of each filter is spaced according to (2.11), resulting in narrower filters for lower linear frequencies and wider filters for higher frequencies. The resulting F triangular filters with $F < N_\mathrm{ny}$ are stored in the columns of mel filter bank $\mathbf{H}_\mathrm{mel}$ of size $N_\mathrm{ny} \times F$ as shown in Figure 2.4. Mel filtering of the magnitude STFT spectrogram $\mathbf{Y} = \left|\underline{\mathbf{Y}}\right|$ of size $N_\mathrm{ny} \times T$ is then obtained by

$$\mathbf{Y}_\mathrm{mel} = \mathbf{H}_\mathrm{mel}^\top \mathbf{Y}, \tag{2.12}$$

resulting in the mel spectrogram $\mathbf{Y}_\mathrm{mel}$ of size $F \times T$.

[2] Each of the $T \approx 2M/N_\mathrm{w}$ DCT spectra has $N_\mathrm{w}/2$ real-valued coefficients resulting in approximately M real-valued coefficients in total for $\mathbf{Z}$.

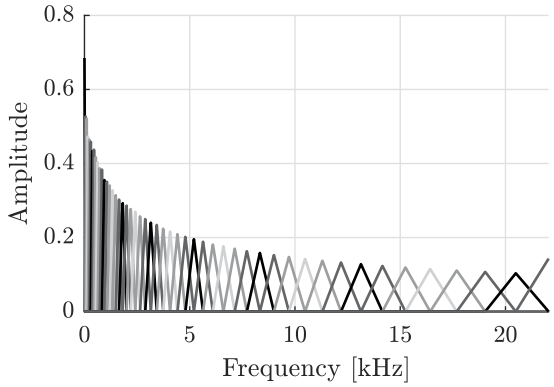


Figure 2.4 Mel filter bank $\mathbf{H}_{\text{mel}}$ with $F = 50$ filters.

Figure 2.4 shows $F = 50$ triangular filters which are linearly spaced on the mel scale. As discussed in [Spi12], an inverse operation of (2.12) yielding perfect reconstruction is not possible. However, it can be approximated by

$$\mathbf{Y}' = \mathbf{H}_{\text{mel}}\mathbf{Y}_{\text{mel}}.$$

To minimize the reconstruction error, it is proposed in [Spi12] to normalize $\mathbf{H}_{\text{mel}}$ such that $\sum_{f'}\left[\mathbf{H}_{\text{mel}}\mathbf{H}_{\text{mel}}^{\top}\right]_{f,f'} = \text{const}$ for all f.

Applying the mel filter bank has some other advantages than approximating the human frequency perception as pointed out in [Spi12]: It speeds up the subsequent factorization process since the number of filters is chosen to be smaller than the number of spectral coefficients N_{ny} of the STFT. This fact is especially interesting for ISS, since the factorization of less spectral coefficients results in less transmitted parameters and thus in a smaller bit rate (cf. Section 2.2). In addition to that, it was shown and evaluated in [Spi12] that mel filtering has another advantage, namely suppressing vibrato effects: These effects can have negative impact on the factorization, since multiple components may be needed to represent a note with vibrato. Vibrato notes usually span over multiple frequency bins which may be grouped by mel filtering to one mel bin. Although the precision of the frequency analysis is lower after mel filtering, the separation quality is increased. The suppression of vibrato effects has stronger influence on the quality than the limited resolution at higher mel bins.

2.2 Nonnegative Factorization

Nonnegative Tensor Factorization (NTF) approximates a D-dimensional, nonnegative tensor by a product of D nonnegative matrices [Cic+09]. The special case of $D = 2$ is also called Nonnegative Matrix Factorization (NMF) since the input is a matrix. The NMF grew popular after Lee and Seung introduced it to a broader audience by publishing a short overview in Nature [LS99] and shortly after a summary of easy-to-use algorithms for solving NMF in [LS01]. The generalization of the NMF to tensors, NTF, is also referred to as nonnegative PARAFAC (Parallel factor decomposition) and discussed in detail e.g. in [Cic+09].

In this thesis, only tensors $\mathbf{V}$ with $D = 3$ dimensions are considered as input for the NTF process. Matrices ($D = 2$) are a special case thereof as shown below. NTF approximates an

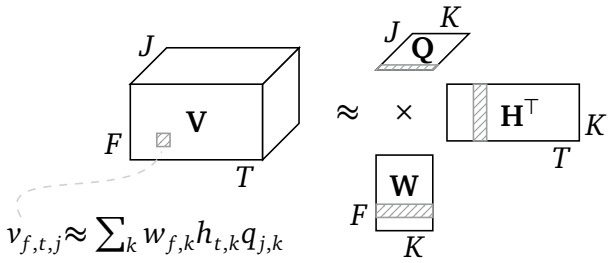


Figure 2.5 NTF of $F \times T \times J$ tensor $\mathbf{V}$ with K components. The elements of the NTF parameters needed to represent input value $v_{f,t,j}$ are marked gray.

$F \times T \times J$ nonnegative tensor $\mathbf{V}$ by a product of three nonnegative matrices $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{Q}$

$$v_{f,t,j} \approx \hat{v}_{f,t,j} = \sum_k w_{f,k} h_{t,k} q_{j,k}, \tag{2.13}$$

where $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{Q}$ are of size $F \times K$, $T \times K$ and $J \times K$ as depicted in Figure 2.5. The matrices $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{Q}$ are gathered under the general notation $\Theta = \{\mathbf{W}, \mathbf{H}, \mathbf{Q}\}$. The approximation $\hat{\mathbf{V}}$ is of same size as $\mathbf{V}$. For $J = 1$, Equation (2.13) becomes the well-known NMF model $\mathbf{V} \approx \mathbf{W}\mathbf{H}^\top$ (with $q_{1,k} = 1$ for all k). Equation (2.13) can also be expressed in a vectorized notation as

$$\mathbf{V}_{\bullet,\bullet,j} \approx \hat{\mathbf{V}}_{\bullet,\bullet,j}(\Theta) = \mathbf{W}\,\mathrm{diag}\left(\mathbf{q}_{j,\bullet}\right)\mathbf{H}^\top = \sum_k q_{j,k} \underbrace{\mathbf{w}_{\bullet,k}\mathbf{h}_{\bullet,k}^\top}_{=\mathbf{C}_{\bullet,\bullet,k}}, \tag{2.14}$$

which shows another aspect of the NTF model: Each jth slice $\mathbf{V}_{\bullet,\bullet,j}$ of $\mathbf{V}$ can be approximated by a sum of K weighted rank-1 matrices $\mathbf{C}_{\bullet,\bullet,k}$ of size $F \times T$ with weights $q_{j,k}$

$$\mathbf{C}_{\bullet,\bullet,k} = \mathbf{w}_{\bullet,k}\mathbf{h}_{\bullet,k}^\top, \tag{2.15}$$

which correspond to the related NTF components. The total number of components K is a user-defined parameter. It has both influence on the factorization quality, as discussed further in Section 2.2.1, and the parameter bit rate: NTF is able to significantly reduce the number of elements from $F \times T \times J$ of tensor $\mathbf{V}$ to $(F + T + J)K$ elements of the NTF parameters in Θ.

In the following, the NTF cost function and the resulting multiplicative update rules used throughout this thesis will be summarized in Section 2.2.1. With this at hand, the application of NTF to source separation is summarized in Section 2.2.2.

2.2.1 Cost-function and Update Rules

In this section, the derivation of update rules for estimating the NTF parameters iteratively is summarized. Starting from a cost function between input and estimation, it is possible to derive update rules which guarantee that the parameters stay nonnegative during the estimation process. To obtain parameters $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{Q}$ which approximate the tensor $\mathbf{V}$ as given in (2.13), a cost function between $\mathbf{V}$ and $\hat{\mathbf{V}}(\Theta)$ has to be chosen to measure the factorization quality appropriately. Lee and Seung proposed using the Kullback Leibler (KL) divergence and the Euclidean distance in [LS01] for this task. NMF minimizing the Itakura

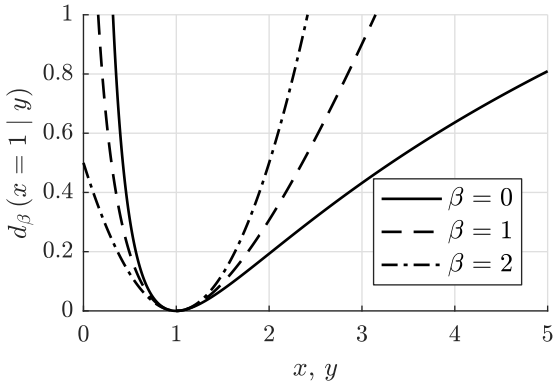


Figure 2.6 β-divergence for $x = 1$ fixed.

Saito (IS) distance was thoroughly investigated in [FBD09]. In this thesis, the β-divergence is chosen as the NTF cost function

$$d_\beta(x \mid y) = \begin{cases} \frac{x}{y} - \log\frac{x}{y} - 1 & \text{if } \beta = 0 \\ x\log\frac{x}{y} - x + y & \text{if } \beta = 1 \\ \frac{1}{\beta(\beta-1)}\left(x^\beta + (\beta-1)\,y^\beta - \beta x y^{\beta-1}\right) & \text{if } \beta \in \mathbb{R}\setminus\{0,1\} \end{cases} \tag{2.16}$$

which includes e.g. IS distance ($\beta = 0$), KL divergence ($\beta = 1$) and Euclidean distance ($\beta = 2$) as shown in Figure 2.6. The derivative of $d_\beta(x \mid y)$ with respect to y is given as

$$\frac{\partial\, d_\beta(x \mid y)}{\partial\, y} = y^{\beta-1} - x y^{\beta-2} \tag{2.17}$$

and is used for the deriving update rules in the following. More detail on the β-divergence used in NTF can be found in e.g. [Cic+09; FI11]. To derive rules for updating the parameters $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{Q}$, it is necessary to formulate a cost function between all elements of input $\mathbf{V}$ and its estimate $\hat{\mathbf{V}}$ as

$$\min\ d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}(\Theta)\right) = \min \sum_{f,t,j} d_\beta\left(v_{f,t,j} \mid \hat{v}_{f,t,j}(\Theta)\right) \tag{2.18}$$

with $d_\beta(x \mid y)$ given in Equation (2.16) and $\hat{\mathbf{V}}(\Theta)$ in (2.14).

In the following, update rules are derived for calculating $\Theta = \{\mathbf{W}, \mathbf{H}, \mathbf{Q}\}$ iteratively. Starting from an initial guess for Θ, the basic idea is to fix two of the three parameters of Θ and to update the unfixed one. This thesis uses multiplicative update rules to assure the nonnegativity of the updated parameters. These update rules can be derived using positive and negative gradient parts [LS01; FI11; Cic+09] which are defined for the gradient of (2.18) with respect to $\mathbf{W}$ as

$$\nabla_{\mathbf{W}} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right) = \underbrace{\nabla^+_{\mathbf{W}} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right)}_{\geq \mathbf{0}} - \underbrace{\nabla^-_{\mathbf{W}} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right)}_{\geq \mathbf{0}} \tag{2.19}$$

where $\nabla_{\mathbf{W}} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right)$ is real-valued and both $\nabla^+_{\mathbf{W}} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right)$ and $\nabla^-_{\mathbf{W}} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right)$ are nonnegative. Θ is omitted in the argument of $\hat{\mathbf{V}}$ for conciseness. The gradient parts with respect to the

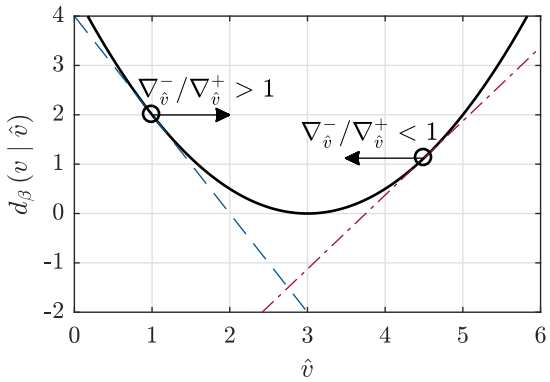


Figure 2.7 Multiplicative update $\hat{v} \leftarrow \hat{v} \cdot \nabla_{\hat{v}}^{-}/\nabla_{\hat{v}}^{+}$ approaches the minimum of $d_\beta(v \mid \hat{v})$ with abbreviations $\nabla_{\hat{v}}^{+} = \nabla_{\hat{v}}^{+} d_\beta(v \mid \hat{v})$ and $\nabla_{\hat{v}}^{-} = \nabla_{\hat{v}}^{-} d_\beta(v \mid \hat{v})$ [Liu12].

other two parameters $\mathbf{H}$ and $\mathbf{Q}$ are defined in the same manner. The multiplicative update rules can then be written as [LS01]

$$\mathbf{W} \leftarrow \mathbf{W} \cdot \frac{\nabla_{\mathbf{W}}^{-} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right)}{\nabla_{\mathbf{W}}^{+} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right)}, \quad \mathbf{H} \leftarrow \mathbf{H} \cdot \frac{\nabla_{\mathbf{H}}^{-} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right)}{\nabla_{\mathbf{H}}^{+} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right)}, \quad \mathbf{q}_{j\bullet} \leftarrow \mathbf{q}_{j\bullet} \cdot \frac{\operatorname{diag}\left(\nabla_{\mathbf{q}_{j\bullet}}^{-} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right)\right)}{\operatorname{diag}\left(\nabla_{\mathbf{q}_{j\bullet}}^{+} d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}\right)\right)}. \tag{2.20}$$

Since the gradient parts defined in (2.19) are nonnegative, the update rules in (2.20) assure that the NTF parameters stay nonnegative after updating them. Using (2.17), (2.20) and the derivative of $\hat{\mathbf{V}}(\Theta)$ with respect to one parameter finally yields the multiplicative update rules [Liu12]

$$\begin{aligned}
\mathbf{W} &\leftarrow \mathbf{W} \cdot \frac{\sum_j \left[\left(\mathbf{V}_{\bullet\bullet j} \cdot \hat{\mathbf{V}}_{\bullet\bullet j}^{\beta-2}\right)\mathbf{H}\right]\operatorname{diag}\left(\mathbf{q}_{j\bullet}\right)}{\sum_j \left[\hat{\mathbf{V}}_{\bullet\bullet j}^{\beta-1}\mathbf{H}\right]\operatorname{diag}\left(\mathbf{q}_{j\bullet}\right)}, \\
\mathbf{H} &\leftarrow \mathbf{H} \cdot \frac{\sum_j \left[\left(\mathbf{V}_{\bullet\bullet j} \cdot \hat{\mathbf{V}}_{\bullet\bullet j}^{\beta-2}\right)^{\top}\mathbf{W}\right]\operatorname{diag}\left(\mathbf{q}_{j\bullet}\right)}{\sum_j \left[\left(\hat{\mathbf{V}}_{\bullet\bullet j}^{\beta-1}\right)^{\top}\mathbf{W}\right]\operatorname{diag}\left(\mathbf{q}_{j\bullet}\right)} \text{ and} \\
\mathbf{q}_{j\bullet} &\leftarrow \mathbf{q}_{j\bullet} \cdot \frac{\operatorname{diag}\left(\left\{\left(\mathbf{V}_{\bullet\bullet j} \cdot \hat{\mathbf{V}}_{\bullet\bullet j}^{\beta-2}\right)\mathbf{H}\right\}^{\top}\mathbf{W}\right)}{\operatorname{diag}\left(\left(\hat{\mathbf{V}}_{\bullet\bullet j}^{\beta-1}\mathbf{H}\right)^{\top}\mathbf{W}\right)}.
\end{aligned} \tag{2.21}$$

These update rules are proven to lead to a local minimum of (2.18) for $\beta \in [1, 2]$. For $\beta = 0$, empirical studies report convergence, too [FI11; Cic+09]. The update rules converge to a local minimum as previously mentioned. The separation quality is dependent on the choice of initial values of Θ. Several methods for initialization are further discussed in e.g. [BMR15]. The initialization procedure used in this thesis is described in Section 2.2.2.

Figure 2.7 illustrates the convergence of multiplicative update rules. It shows the cost function $d_\beta(v \mid \hat{v})$ given in (2.18) for scalar values v and $\hat{v}$. The gradient of d_β with respect to $\hat{v}$ can be expressed as $\nabla_{\hat{v}} d_\beta(v \mid \hat{v}) = \nabla_{\hat{v}}^{+} d_\beta(v \mid \hat{v}) - \nabla_{\hat{v}}^{-} d_\beta(v \mid \hat{v}) = \nabla_{\hat{v}}^{+} - \nabla_{\hat{v}}^{-}$ with both $\nabla_{\hat{v}}^{+}$ and $\nabla_{\hat{v}}^{-}$ nonnegative as already shown in (2.19). In this example, the gradient is given

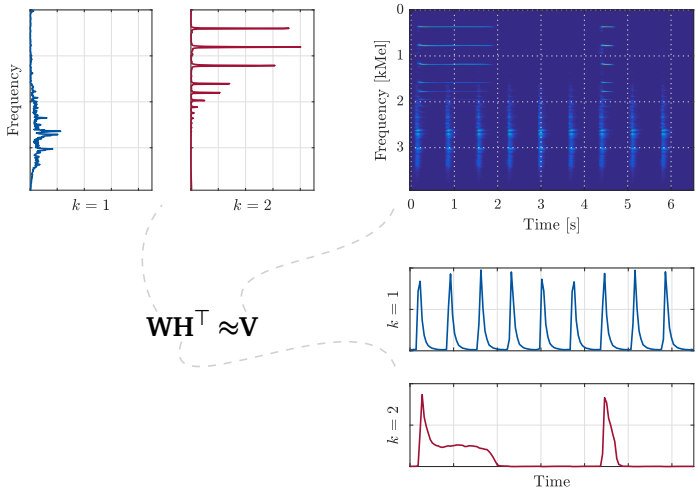


Figure 2.8 Detailed example of an NTF factorization of a single $F \times T$ spectrogram $\mathbf{V}$ of a tambourine and a trumpet with $K = 2$ components and $J = 1$. Each column of $\mathbf{W}$ and $\mathbf{H}$ is plotted and color-coded to distinguish the two different components [Hen11; Liu12].

with (2.17) as $\nabla_{\hat{v}} d_\beta(v \mid \hat{v}) = \hat{v}^{\beta-1} - v\hat{v}^{\beta-2}$ and the gradient terms are thus $\nabla^+_{\hat{v}} d_\beta(v \mid \hat{v}) = \hat{v}^{\beta-1}$ and $\nabla^-_{\hat{v}} d_\beta(v \mid \hat{v}) = v\hat{v}^{\beta-2}$. As shown in Figure 2.7, the multiplicative update $\hat{v} \leftarrow \hat{v} \cdot \nabla^-_{\hat{v}} / \nabla^+_{\hat{v}}$ approaches the minimum of $d_\beta(v \mid \hat{v})$ for a given value of $\hat{v}$ [Liu12].

2.2.2 Application to Audio Source Separation

One of the first reports on applying NTF in the field of audio signal processing was [SB03], using NTF for music transcription. A vast variety of algorithms using NTF for audio source separation exist, e.g. [Vir07; OF10; Spi12; OVB12] for blind source separation. The algorithms of e.g. [Liu+11; Oze+13; Nik15] use NTF for ISS. Several variants of NTF exist which adapt it to audio processing, such as additional constraints as discussed in Section 2.2.2.2, versions of NTF using matrices instead of vectors for components [Sma04; FCC05; BR14], or taking complex-valued inputs [Kam+09].

This thesis deals with ISS which means that the original sources are completely known at an encoding stage. In the following it is assumed that NTF is used for compression of the spectrograms of the original sources. Another use case of NTF is to factorize the (multi-channel) mixture spectrogram for separating the mixture in the context of BSS, as done in e.g. [Spi12]. However, the input of the NTF process are audio spectrograms which leads to considering both cases, BSS and ISS, simultaneously to explain the structure of the resulting NTF parameters. After the TF transform, the resulting $F \times T$ nonnegative magnitude spectrograms of all J sources can be stacked together yielding an $F \times T \times J$ nonnegative tensor $\mathbf{V}$ (the same holds true for a multi-channel mixture of different sources with J channels in the case of BSS). Factorizing this nonnegative tensor yields the NTF parameters Θ which can be interpreted as follows:

- $\mathbf{W}$ holds K frequency basis functions which can be understood as spectral templates, one for each component or sound event k.

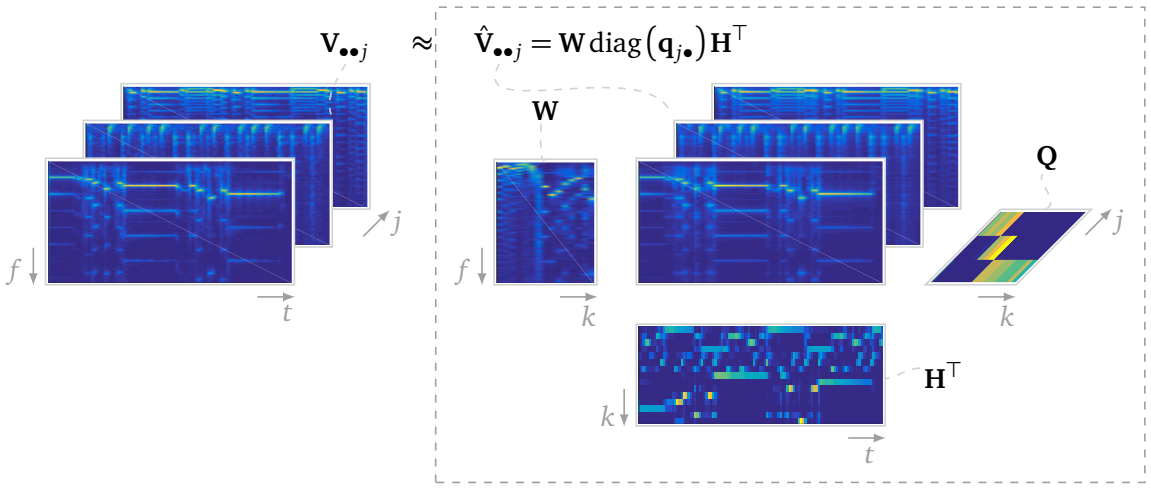


Figure 2.9 NTF model for $J = 3$ sources (guitar, drums and keys) and $K = 15$ components.

- $\mathbf{H}$ consists of the corresponding temporal activations, giving information about the gain of each component k at each time bin t.
- The mapping of each component k to each source (or channel) j is stored in $\mathbf{Q}$.

Each component $\mathbf{C}_{\bullet,\bullet,k}$ as given in Equation (2.15) describes a particular sound event, such as e.g. a harmonic or percussive note, onset or noise, which are present in $\mathbf{V}$. These events are then weighted with $q_{j,k}$ indicating the activity of each sound event in each source.

A single ($J = 1$) audio spectrogram $\mathbf{V}$ of an exemplary mixture of a trumpet and a tambourine signal is shown in Figure 2.8. NTF was conducted here with $K = 2$ components. The basis spectra of each component are stored in the columns $\mathbf{w}_{\bullet,k}$ of $\mathbf{W}$ whereas $\mathbf{h}_{\bullet,k}$, the columns of $\mathbf{H}$, hold the corresponding temporal envelopes or activations. In this example, each note is separated in a component: Component $k = 1$ holds the percussive tambourine note with a rather flat spectrum stored in frequency basis $\mathbf{w}_{\bullet,1}$ and the corresponding short temporal envelopes in $\mathbf{h}_{\bullet,1}$. $\mathbf{w}_{\bullet,2}$ holds the harmonic spectrum of the trumpet note. The harmonics in $\mathbf{w}_{\bullet,2}$ are not equally spaced for higher frequencies because spectrogram $\mathbf{V}$ was subject to mel filtering as discussed in Section 2.1.3. The temporal activation $\mathbf{h}_{\bullet,2}$ is quite continuous and shows that the trumpet note is played twice.

Figure 2.9 shows NTF parameters for factorizing $J = 3$ sources simultaneously with $K = 15$ components as another example. The three source magnitude spectrograms $\mathbf{V}_{\bullet\bullet j}$ of exemplary guitar ($j = 1$), drums ($j = 2$) and keyboard ($j = 3$) signals are factorized such that the first four components describe the keyboard, the next three the drums and the remaining eight components the guitar. This grouping is given by $\mathbf{Q}$. $\mathbf{W}$ holds frequency basis functions and $\mathbf{H}$ temporal activations for all three sources jointly.

In the following sections, different adaptations of the NTF to the field of audio processing are briefly summarized. Suitable initialization methods of Θ are discussed in Section 2.2.2.1 and additional constraints to the NTF cost function are discussed in Section 2.2.2.2.

2.2.2.1 Initialization

In the study of [BMR15], several initialization methods for the NTF parameters $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{Q}$ in the field of source separation are discussed and compared. Initialization with random val-

ues was compared to initial values calculated by different algorithms based on the Singular Value Decomposition (SVD) with $\mathbf{V}$ as input. In this thesis, the SVD-based method yielding the highest separation quality in the experimental comparison conducted in [BMR15] is chosen and briefly discussed in the following. Refer to [BMR15] for a summary of other state-of-the-art initializations. The selected method calculates the SVD with the complex-valued mixture as input. The initialization methods were evaluated in [BMR15] for monaural source separation. The complex SVD is therefore calculated on the complex $N_{\text{ny}} \times T$ mixture spectrogram $\underline{\mathbf{X}}$ as

$$\underline{\mathbf{X}} = \underline{\mathbf{U}}\mathbf{\Sigma}\underline{\mathbf{V}}^{*},$$

with $N_{\text{ny}} \times N_{\text{ny}}$ and $T \times T$ complex, unitary matrices $\underline{\mathbf{U}}$ and $\underline{\mathbf{V}}$ and diagonal matrix $\mathbf{\Sigma}$ of size $N_{\text{ny}} \times T$ storing singular values $\sigma_{f,f}$ on its diagonal. In a second step, the K largest values $\sigma_{k,k}$ and the corresponding column vectors $\underline{\mathbf{u}}_{\bullet,k}$ and $\underline{\mathbf{v}}_{\bullet,k}$ with $1 \leq k \leq K$ are selected. The initial values for $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{Q}$ are then calculated as

$$\mathbf{w}_{\bullet,k} \leftarrow \mathbf{H}_{\text{mel}}^{\top} \left|\underline{\mathbf{u}}\right|_{\bullet,k} \sqrt{\sigma_{k,k}}, \quad \mathbf{h}_{\bullet,k} \leftarrow \left|\underline{\mathbf{v}}\right|_{\bullet,k} \sqrt{\sigma_{k,k}}, \quad q_{j,k} \leftarrow 1.$$

2.2.2.2 Constraints

A detailed overview and evaluation of constraints for NTF in the context of audio source separation is given e.g. in [Bec16]. These constraints exploit typical structures of the NTF parameters $\mathbf{W}$ and $\mathbf{H}$. In this thesis however, only two basic and widely used NTF constraints are considered which were originally proposed in [Vir07] and are only constraining the temporal activation matrix $\mathbf{H}$. The exemplary $\mathbf{H}$ depicted in Figure 2.8 also shows the properties exploited by the constraints of [Vir07]:

- Percussive notes ($k = 1$ in Figure 2.8) usually have short temporal activations since they usually only consist of attack and decay with only little or no sustain.
- Harmonic notes ($k = 2$ in Figure 2.8) have longer, continuous temporal activations. Compared to percussive notes, harmonic notes are usually played for a longer period of time.

These two facts lead to the two constraints proposed by [Vir07]:

Temporal continuity favors continuously played notes, such as harmonic notes, and is calculated as the squared difference of two neighboring elements of one component of $\mathbf{H}$

$$d_{\text{tc}}(\mathbf{H}) = \sum_{t} \frac{1}{\frac{1}{T}\sum_{t'} h_{t'k}^{2}} \sum_{t=2}^{T} \left(h_{tk} - h_{t-1,k}\right)^{2}.$$

The second constraint, sparseness, is widely used and given as

$$d_{\text{s}}(\mathbf{H}) = \sum_{k} \sum_{t} \left| \frac{h_{tk}}{\sqrt{\frac{1}{T}\sum_{t'} h_{t'k}^{2}}} \right|.$$

Sparseness is useful for components holding percussive notes which are typically played for a short time. This means that the corresponding temporal activations are sparse. These

constraint functions are added to the cost function $d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}(\Theta)\right)$ between input $\mathbf{V}$ and reconstruction $\hat{\mathbf{V}}$ in Equation (2.18), resulting in a total cost function

$$\min\ d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}(\Theta)\right) + \gamma_{\mathrm{tc}}\left[d_{\mathrm{tc}}(\mathbf{H})\right] + \gamma_{\mathrm{s}}\left[d_{\mathrm{s}}(\mathbf{H})\right]$$

with weighting factors $\gamma_{\mathrm{tc}}, \gamma_{\mathrm{s}} \geq 0$. Multiplicative update rules can be derived as already shown for $d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}(\Theta)\right)$ in Section 2.2.1 by splitting up the gradients $\nabla_{\mathbf{H}} d_{\mathrm{tc}}(\mathbf{H})$ and $\nabla_{\mathbf{H}} d_{\mathrm{s}}(\mathbf{H})$ into positive and negative parts. The update rules are given in [Vir07]. It should be noted that the underlying assumptions of these constraints conflict each other [Bec16] since a component is either of harmonic or percussive nature. The temporal continuity and sparseness constraints are therefore activated exclusively.

In [BR15; BRR15] it was proposed to adapt the constraint weights γ to each NTF component. A novel constraint cost function was therefore proposed in [BR15] as

$$\min\ d_\beta\left(\mathbf{V} \mid \hat{\mathbf{V}}(\Theta)\right) + \gamma_{\mathrm{tc}} \log\left[d_{\mathrm{tc}}(\mathbf{H})\right] + \gamma_{\mathrm{s}} \log\left[d_{\mathrm{s}}(\mathbf{H})\right],$$

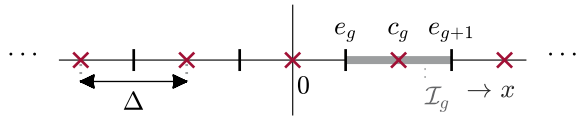
yielding a weighted derivative of constraint $d(\mathbf{H})$, here either d_{tc} or d_{s}: The derivative $\nabla_{\mathbf{H}} \log\left[d(\mathbf{H})\right] = \frac{\nabla_{\mathbf{H}} d(\mathbf{H})}{d(\mathbf{H})}$ weights $\nabla_{\mathbf{H}} d(\mathbf{H})$, the derivative of the constraint, with the current value $d(\mathbf{H})$ of the constraint.

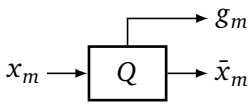
2.3 Quantization

Quantization maps continuous values to a finite set of values, also called *reconstruction values*. The amplitudes of the reconstruction values are either predefined or estimated during the quantization process. The quantization output is then the mapping of each continuous value to a certain reconstruction value which can be expressed by an integer number, also referred to as *quantization index*. Prior to transmission, the quantization indices are subject to coding which maps each integer value to a sequence of binary numbers to compose a *bit stream*. Coding is explained in Section 2.4.

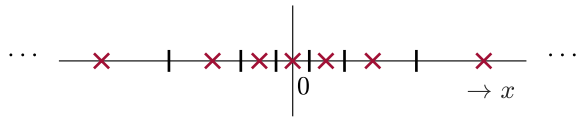
The signals to be quantized and encoded are represented digitally, e.g. as a vector $\mathbf{x}$ with M elements x_m. These values are not necessarily assumed in the time domain. Instead, they are only assumed to be in a vectorized form. Here, only scalar quantization, applied on each element independently, is used. The more efficient but also more complex vector quantization is out of the scope of this thesis. Note that the signals at hand were already subject to quantization during the analog-to-digital conversion (ADC) briefly discussed in the beginning of this chapter. To be precise, the methods discussed in this section should be referred to as re-quantization. Since the ADC is conducted with very high precision, the input signals are assumed to be continuous. A detailed overview over quantization in general is given in e.g. [GN98]. One contribution of this thesis, as discussed in Section 5.2, deals with re-estimation of quantized parameters. Several algorithms exist for this task, e.g. [BGL16; ZBC10], which use compressive-sensing-related methods at the decoder to refine the quantized signals at hand.

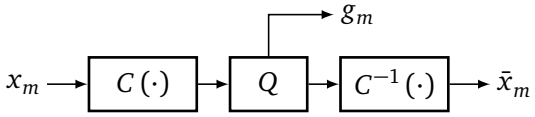
The task of scalar quantization is to map a real-valued scalar $x_m \in \mathbb{R}$ to one of N_{q} predefined reconstruction values c_g with $1 \leq g \leq N_{\mathrm{q}}$. This is a lossy and thus non-invertible operation. Here, scalar quantization will be used on each element x_m of a vector $\mathbf{x}$ independently (cf. Equation (2.1)). Formally, the set of real values $\mathbb{R}$ (or nonnegative real values $\mathbb{R}_+$)





(a) Uniform quantization.





(b) Non-uniform quantization.

Figure 2.10 Scalar quantization.

is partitioned into N_q intervals $\mathscr{I}_g = \left[e_g, e_{g+1}\right)$ defined by $N_\mathrm{q}+1$ boundaries e_g with outermost left and right boundaries as $e_1 = -\infty$ and $e_{N_\mathrm{q}+1} = \infty$. If input x_m lies inside one particular interval $\mathscr{I}_g$, it is mapped to the corresponding reconstruction value c_g

$$q\left(x_m\right) = \bar{x}_m = c_g, \quad x_m \in \mathscr{I}_g. \tag{2.22}$$

The function $q\left(x\right)$ is also called quantization characteristic. Practically, the integer-valued quantization index $1 \leq g_m \leq N_\mathrm{q}$ is stored in vector $\mathbf{g}$ of same size as $\mathbf{x}$ such that the reconstructed value at position m is set to $\bar{x}_m = c_{g_m}$. Figure 2.10a shows reconstruction values c_g, edges e_g and one exemplary interval $\mathscr{I}_g$ for uniform quantization.

Uniform quantization is summarized in Section 2.3.1. Non-uniform quantization using companding and expanding functions prior and subsequent to uniform quantization is detailed in Section 2.3.2. The Lloyd-Max Algorithm (LM), which iteratively finds non-uniform reconstruction values by minimizing a distortion measure between quantization input and output, is briefly discussed in Section 2.3.3. In Section 2.3.4, Rate-distortion Optimized Quantization (RDOQ) is explained which jointly minimizes distortion and rate spent on transmitting the quantization indices.

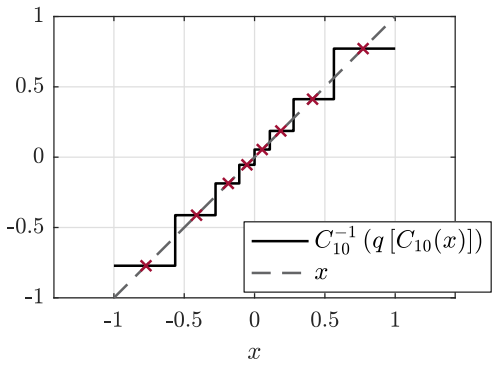
2.3.1 Uniform Quantization

In the case of uniform scalar quantization, the reconstruction values c_g have uniform distance Δ. The signals to be quantized in this thesis are usually sparse. Due to this fact, this section deals with so-called midtread quantizers which can represent the value 0. For nonnegative signals, the distance Δ is determined as $\Delta = \frac{x_\mathrm{max}}{N_\mathrm{q}-1}$ and for real-valued signals as $\Delta = \frac{2x_\mathrm{max}}{N_\mathrm{q}-1}$ with x_max the maximum input value in both cases. It is assumed that the sign of x_m is coded independently. Therefore the quantization indices are calculated given the absolute value of input x_m as

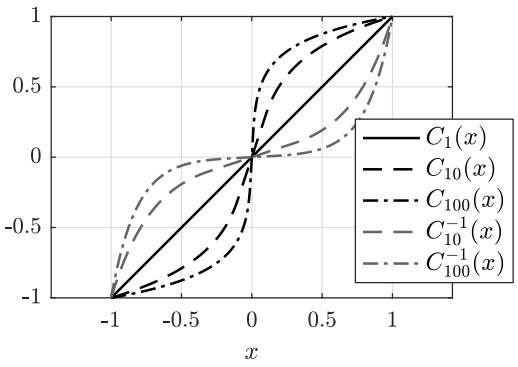
$$g_m = \left\lfloor \frac{|x_m|}{\Delta} + \frac{1}{2} \right\rfloor,$$

and the corresponding reconstruction values are linearly spaced with

$$c_g = \Delta\left(g-1\right).$$



(a) Non-uniform quantization. Input of quantizer is companded, the resulting symbols are expanded.



(b) A-law companding and expanding functions $C_A(x)$ and $C_A^{-1}(x)$.

Figure 2.11 Non-uniform quantization with A-law companding.

2.3.2 Non-uniform Quantization

The procedure of non-uniform quantization is shown in Figure 2.10b and can be achieved by mapping input values x_m by a so-called companding function $C(\cdot)$ prior to quantization. Assuming an inverse to $C(\cdot)$, $C^{-1}(\cdot)$, exists, the quantization process can be summarized as

$$\bar{x}_m = C^{-1}(q[C(x_i)]),$$

were $q(\cdot)$ denotes uniform scalar quantization. Here, the A-law *companding* function [VM06] is used

$$C_A(x) = \frac{\operatorname{sign}(x)}{1+\log A}\begin{cases} A|x|/x_{\max}, & |x|/x_{\max} < \frac{1}{A} \\ 1+\log(A|x|/x_{\max}), & \frac{1}{A} \le |x|/x_{\max} \le 1 \\ 1+\log A, & |x|/x_{\max} > 1 \end{cases} \tag{2.23}$$

with its inverse, the *expanding* function, defined as

$$C_A^{-1}(y) = \frac{\operatorname{sign}(y)\, x_{\max}}{A}\begin{cases} |y|(1+\log A), & |y| < \frac{1}{1+\log A} \\ \exp(|y|(1+\log A)-1), & \frac{1}{1+\log A} \le |y| < 1. \end{cases} \tag{2.24}$$

Note that for $A = 1$, the linear interval spans over the whole value range. In this case, the quantizer is operated as a uniform quantizer. In the following chapters, the corresponding derivatives of C_A and C_A^{-1} are needed. The derivative of C_A can be calculated as

$$\frac{\partial C_A(x)}{\partial x} = \frac{1}{1+\log A}\begin{cases} A/x_{\max}, & |x|/x_{\max} < \frac{1}{A} \\ \frac{1}{|x|}, & \frac{1}{A} \le |x|/x_{\max} \le 1 \end{cases} \tag{2.25}$$

and the derivative for the expanding function C_A^{-1} as

$$\frac{\partial C_A^{-1}(y)}{\partial y} = \frac{x_{\max}(1+\log A)}{A}\begin{cases} 1, & |y| < \frac{1}{1+\log A} \\ \exp(|y|(1+\log A)-1). & \frac{1}{1+\log A} \le |y| < 1. \end{cases} \tag{2.26}$$

An exemplary non-uniform quantization characteristic with $A = 10$ as well as companding and expanding functions for $A \in \{10, 100\}$ are shown in Figure 2.11.

2.3.3 Lloyd-Max Algorithm

The Lloyd-Max Algorithm (LM) [Llo82; Max60] finds optimum reconstruction values c_g by minimizing the distortion between input x_m and output $\bar{x}_m$ of the quantizer. It is assumed here that a training set of M values, x_m, stored in vector $\mathbf{x}$, exists and that the reconstruction values c_g and the corresponding indices g_m are optimized for this training set in the following. This is done by minimizing the distortion between $\mathbf{x}$ and quantization output $\bar{\mathbf{x}}$

$$d(\mathbf{x}, \bar{\mathbf{x}}) = \sum_{m=1}^{M} (x_m - \bar{x}_m)^2,$$

using an Euclidean distortion measure. The distortion can also be expressed depending on the N_q reconstruction values c_g as

$$d(\mathbf{x}, \bar{\mathbf{x}}) = d(\mathbf{x}, \mathbf{c}) = \sum_{g=1}^{N_\mathrm{q}} \sum_{x_m \in \mathscr{I}_g} \left(x_m - c_g\right)^2, \tag{2.27}$$

where the quantization interval is defined by the edges e_g as $\mathscr{I}_g = \left[e_g, e_{g+1}\right)$. Now, the task is to find both c_g and e_g which minimize (2.27). This can be done in an iterative manner, updating quantization boundaries and reconstruction values alternating as summarized in the following steps:

1. Choose N_q initial reconstruction values c_g with $c_1 < c_2 \cdots < c_{N_\mathrm{q}}$, e.g. by uniform quantization (cf. Section 2.3.1).

2. Fix c_g and determine the *quantization boundaries* e_g lying in the middle of the two corresponding neighboring reconstruction values

   $$e_g = \frac{c_{g-1} + c_g}{2} \quad \text{for all } g > 1,$$

   and update the corresponding quantization interval to $\mathscr{I}_g = \left[e_g, e_{g+1}\right)$.

3. Fix e_g and determine *reconstruction values* c_g as the expected value of all input values x_m lying in the quantization interval given by the quantization boundaries e_g

   $$c_g = \mathrm{E}\left[x_m \mid x_m \in \mathscr{I}_g\right] \quad \text{for all } g.$$

   Here, the expected value can be estimated by the mean of all input values x_m lying in $\mathscr{I}_g$ as $c_g = \sum_{x_m \in \mathscr{I}_g} x_m / \left|\mathscr{I}_g\right|$ with $\left|\mathscr{I}_g\right|$ denoting the number of elements of $\mathscr{I}_g$.

4. Repeat steps 2. and 3. until a convergence criterion is met. In this thesis, it is simply checked whether the reconstruction values c_g did not change anymore from one iteration to the next one.

Finally, set the quantization output to $\bar{x}_m = c_{g_m}$ with $x_m \in \left[e_{g_m}, e_{g_m+1}\right)$. The derivation of these update rules is summarized in [VM06]. In contrast to the quantizers mentioned in the sections before, not only the group indices $\mathbf{g}$ but also the reconstruction values $\mathbf{c}$ have to be transmitted to the decoder to reconstruct $\bar{\mathbf{x}}$.

2.3.4 Rate-distortion Optimized Quantization

The LM, as described in previous Section 2.3.3, estimates reconstruction values $\mathbf{c}$ such that the distortion between input values $\mathbf{x}$ and $\mathbf{c}$ is minimized with the distortion given in Equation (2.27). However, this optimization procedure does not take the bit rate into account which is necessary to transmit the quantization indices $\mathbf{g}$. Assume quantization indices which were found by the LM, thus minimizing solely the distortion. However, for some input values, it could be interesting to choose another, neighboring quantization interval as it could reduce the bit rate, accepting a small increase of distortion. In Rate-distortion Optimized Quantization (RDOQ), as summarized e.g. in [SW98], a joint criterion is minimized, taking both distortion and bit rate into account simultaneously. The following criterion sets the distortion $d(\mathbf{x}, \tilde{\mathbf{x}})$ in (2.27) and the rate $r(\tilde{\mathbf{x}})$ spent to encode the quantization indices $\mathbf{g}$ and reconstruction values $\mathbf{c}$ into relation by a factor λ

$$\min_{\tilde{\mathbf{x}}} d(\mathbf{x}, \tilde{\mathbf{x}}) + \lambda r(\tilde{\mathbf{x}}). \tag{2.28}$$

The Lagrangian multiplier λ in Equation (2.28) weights the influence of rate over distortion. For $\lambda = 0$, RDOQ falls back to the LM. Equation (2.28) can be minimized in an iterative manner. The distortion can be easily calculated for a particular quantizer setting as already exploited in the LM. For evaluating the bit rate $r(\tilde{\mathbf{x}})$, the following assumption is made: Since coding methods, as further discussed in Section 2.4, all approach entropy, the rate is simply measured by the entropy obtained by the current quantizer setting.

2.4 Entropy Coding

Entropy coding is used in this thesis as a subsequent step of quantization to encode the quantization indices to a bit stream. In the following, the quantization indices are also referred to as *symbols*. Entropy coding is a lossless mapping of these symbols or sequences thereof to variable-length bin-strings or code words. All coding methods as described here are prefix-free meaning that no code word contains prefixes of other code words so that the symbol can be decoded uniquely. The methods considered in this thesis can be roughly divided into two groups: The first group approaches the (first-order) *entropy* of the input symbol sequence whereas the second group approaches the *conditional entropy* which is smaller than or equal to the first-order entropy. For more detail on entropy coding refer to e.g. [Say05; CT06].

In the following sections, two algorithms are described which are theoretically able to approach conditional entropy, namely Context-based Adaptive Binary Arithmetic Coding (CABAC) in Section 2.4.1 and Run-length Coding (RLC) in Section 2.4.2. First, coding schemes approaching the first-order entropy are briefly summarized below.

Systematic Codes

Systematic codes are constructed by a certain predetermined rule. Fixed-length coding converts an alphabet of N_q symbols simply to a binary representation with $\lceil \log_2(N_\mathrm{q} - 1) \rceil$ bits. In this work, these symbols are assumed to be integers g with $1 \leq g \leq N_\mathrm{q}$. Two systematic codes yielding *variable-length* codes, Truncated unary (TU) and Exponential Golomb (EG) codes, are detailed further in Section 2.4.1.1.

Huffman Coding

Huffman coding assigns a binary code word to each input symbol. Given the distribution of symbols, probable symbols are mapped to shorter codes whereas less probable symbols are mapped to longer codes. Huffman coding is able to find an optimum code book given the symbol's probabilities and assigning a code word to each symbol. This optimality holds true only for independent symbols which are identically distributed and assuming symbol-by-symbol encoding [CT06]. The symbol probabilities have to be transmitted for decoding. Adaptive methods for estimating the probabilities at run-time exist but are less efficient than the adaptive arithmetic coding schemes as discussed below.

Arithmetic Coding

In contrast to Huffman codes or systematic variable-length codes, Arithmetic Coding does not assign codes to each symbol but encodes whole *sequences* of symbols instead. This enables arithmetic coding to yield a fractional number of bits per symbol instead of integer number of bits for Huffman coding [CT06]. However, in contrast to Huffman coding, instantaneous decoding is not possible when using arithmetic coding. To yield fractional number of bits per symbol, arithmetic coding calculates the probability interval of each sequence of symbols which is given by the cumulative joint probability of all symbols in the sequence. If symbols are added to the sequence, the interval is *subdivided* further. Each interval can thus be identified with a number in the interval $[0, 1)$ which requires high arithmetic precision. The arithmetic precision influences how close entropy can be approached. For these calculations, the probabilities of each symbol have to be known beforehand or estimated at run-time. The latter procedure is called adaptive arithmetic coding[3].

GZIP

GZIP (GNU zip) [Deu+96] is based on the DEFLATE algorithm which uses LZ77 in combination with Huffman coding [CT06]. LZ77 approaches entropy [Say05] and builds a dictionary consisting of highly probable symbol vectors of a predefined length. Each dictionary element is addressed with a fixed-length code. DEFLATE encodes the pointers to the dictionary elements subsequently with Huffman coding.

2.4.1 Context-based Adaptive Binary Arithmetic Coding

Context-based Adaptive Binary Arithmetic Coding (CABAC) is used in state-of-the-art video coding schemes such as H.264/AVC [MSW03] and its successor High Efficiency Video Coding (HEVC) [Sul+12] for entropy coding. At its core, CABAC uses arithmetic coding which is able to assign fractional numbers of bits to a symbol as discussed above. However, the main difference to arithmetic coding is *context modeling* which enables adapting to local conditional statistics within the data and thus approaching *conditional entropy*. CABAC is implemented multiplication-free, enabling high coding throughput which is important in the field of video

[3] In this thesis, a MATLAB wrapper `http://www.diegm.uniud.it/~bernardi/Software/Matlab/index.html` of an adaptive arithmetic coding engine written in C provided by `http://www.fredwheeler.org/ac/` is used.

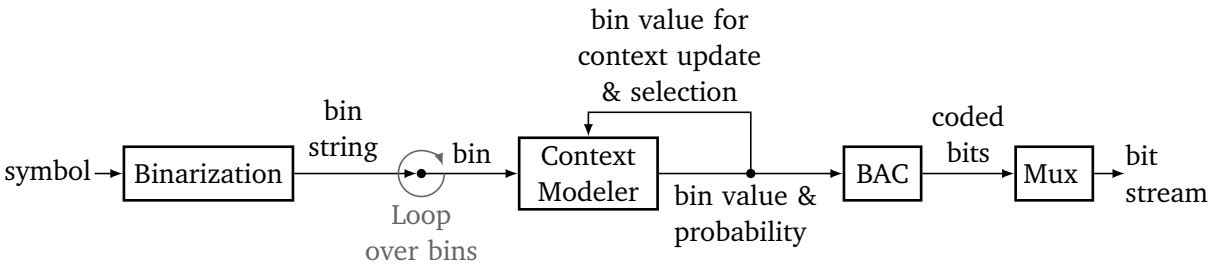


Figure 2.12 Block diagram of CABAC excluding the bypass engine [MSW03].

coding. These constraints also lead to the fact that the core steps in the arithmetic coding block, interval subdivision and probability update, are performed only for binary sources. Therefore, non-binary input symbols have to be *binarized* in a first step, resulting in bin-strings as further discussed in Section 2.4.1.1. These bin-strings are then encoded by the Binary Arithmetic Coding (BAC), given a *context model* which models a particular state of information available at the decoder. This procedure enables CABAC to adapt to local *conditional* statistics and is steered in the *context modeler* which chooses the appropriate context model for each bin to be coded. To each context model belongs a conditional probability, modeling the probability of the current bin to-be-coded given the state of information. When a context model is chosen by the context modeler, the current bin value is also used to update the context model, more precisely the corresponding conditional probability value, at run-time. Context modeling is further discussed in Section 2.4.1.2 and the probability estimation procedure in Section 2.4.1.3. In this thesis, CABAC is adapted to the field of ISS by choosing appropriate binarization schemes and proposing novel context models. The core steps of the BAC are not modified and therefore not described in detail, refer to e.g. [Wie14] for a more in-depth summary. The CABAC engine was extracted from the HEVC test model (HM)[4] provided by the Joint Collaborative Team on Video Coding (JCT-VC) for yielding the experimental results in this thesis.

Figure 2.12 depicts the main building blocks of CABAC. The core steps are performed for a binary source, therefore the first step is binarization, yielding a bin-string. Each bin is then fed into the context modeler, choosing the appropriate context model depending on e.g. previously decoded bins. Given the chosen context model, the bin is then coded in the BAC. The value of the coded bin is fed back to the context modeler to update the used context model and to select the context model for the next to-be-coded bin. This procedure finally yields a bit stream for the whole sequence of input symbols.

2.4.1.1 Binarization

As mentioned earlier, the core block of CABAC, the BAC, takes only binary signals as input. Therefore, as a first step, the integer symbols g with $1 \leq g \leq N_\mathrm{q}$ have to be binarized using prefix-free codes $C(\cdot)$. The resulting *bin-strings* $\mathbf{b}$

$$\mathbf{b} = C(g) = \left(b_1, \ldots, b_n, \ldots, b_{N_\mathrm{C}}\right)^\top, \tag{2.29}$$

[4] `https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/`

g	$C_{\mathrm{FL(3)}}(g)$	$C_{\mathrm{TU}}(g)$	$C_{\mathrm{EG(0)}}(g)$	$C_{\mathrm{EG(1)}}(g)$	$C_{\mathrm{EG(2)}}(g)$
1	000	0	0	0 0	0 00
2	001	10	10 0	0 1	0 01
3	010	110	10 1	10 00	0 10
4	011	1110	110 00	10 01	0 11
5	100	11110	$\underbrace{110}_{\text{Prefix}}\ \underbrace{01}_{\text{Suffix}}$	10 10	10 000

Table 2.1 Fixed length (with length 3), Truncated unary (TU) and Exponential Golomb (EG) binarizations.

have variable length N_{C} and each *bin* b_n at position n is binary, $b_n \in \{0,1\}$. In this thesis, two different codes are considered for this purpose, namely Truncated unary (TU) and Exponential Golomb (EG) which are explained in the following. For more detailed information on applicable variable-length codes refer to e.g. [Wie14].

Truncated unary (TU) encodes $g \geq 1$ with $N_{\mathrm{C}} = g$ bins as

$$C_{\mathrm{TU}}(g) = \underbrace{11\ldots1}_{g-1 \text{ times}}0, \tag{2.30}$$

consisting of a sequence of '1's with length $g-1$ and a terminating '0' at position $N_{\mathrm{C}} = g$. For the maximum value of $g = N_{\mathrm{q}}$, the terminal '0' is omitted.

An Exponential Golomb (EG) code of kth order is constructed out of a concatenation of a unary code of variable length for the *prefix* and a fixed-length code for the *suffix*

$$C_{\mathrm{EG}(k)}(g) = \underbrace{11\ldots10}_{\text{Prefix}}\ \underbrace{b_1\ldots b_{N_{\mathrm{suf}}}}_{\text{Suffix}}.$$

The number of prefix bins N_{pre} is depending on the symbol g and order k,

$$2^k\left(2^{N_{\mathrm{pre}}-1}-1\right)+1 \leq g \leq 2^k\left(2^{N_{\mathrm{pre}}}-1\right) \tag{2.31}$$

which leads to $N_{\mathrm{pre}} = \left\lfloor \log_2\left(\frac{g-1}{2^k}+1\right)\right\rfloor + 1$. The prefix indicates the number of suffix bins $N_{\mathrm{suf}} = N_{\mathrm{pre}} + k - 1$. The suffix encodes the value $g-1-2^k\left(2^{N_{\mathrm{pre}}-1}-1\right)$ with a fixed-length binary code of length N_{suf} [Wie14]. EG codes are optimum prefix-free codes for geometrically distributed sources [MSW03]. Exemplary bin-strings for binarization of $g \in [1,5]$ are shown for TU and EG with $k \in \{0,1,2\}$ in Table 2.1.

2.4.1.2 Context Modeling

As mentioned above, CABAC is able to adapt to local statistics within the signal. This is achieved by modeling certain states of information with context models, each context model mapping to one particular state. For each bin b_n to be encoded or decoded, one particular context model is chosen. Given the selected context model ctx, the probability for the value of b_n is estimated by the *conditional probability* of the current bin b_n given the chosen

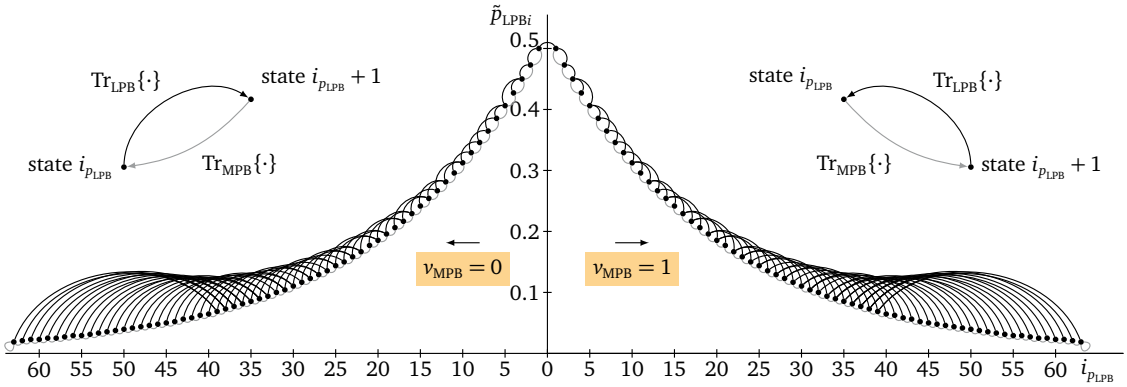


Figure 2.13 Probability state transitions [Wie14].

context model $p(b_n \mid \mathrm{ctx})$. This probability is then used in the BAC for performing the core steps of arithmetic coding such as interval subdivision. After encoding or decoding b_n, the corresponding probability model of ctx is updated given the current value of b_n which enables CABAC to adapt at run-time to the signal's statistics, thus increasing the compression performance [Ohm15].

Typically, a set of context models is available to be chosen from, modeling different probabilistic beliefs about the input symbols. The context models may be designed based on the value of previously coded, neighboring bins $b_{n'}$ with $n' < n$ or the bin position n of the current bin b_n. However, one single context model has to be chosen for each bin in the actual coding step [MSW03]. Note that the context selection process is the same for both CABAC encoding and decoding process as the encoder matches exactly the context model selection of the decoder, given previously decoded data available at the decoder.

2.4.1.3 Probability Estimation

In the following, the internal process of adapting to source symbol probabilities for each context model is briefly summarized. It will be used in Chapter 4 to evaluate the proposed novel context models. To keep CABAC multiplication-free, it was proposed in [MSW03] to represent the conditional probabilities $p(b_n \mid \mathrm{ctx})$ given context model ctx by 63 predefined probability values. Internally, the backward-adaption to the bin probabilities $p(b_n = v_{\mathrm{LPB}} \mid \mathrm{ctx})$ is achieved by a finite state machine with 63 probability states, each state modeling the probability of the value v_{LPB} of the least probable bin (LPB) [Wie14]. Therefore, to completely parametrize the state of a given bin, the binary value of the most probable bin (MPB), $v_{\mathrm{MPB}} \in \{0, 1\}$ must also be specified. The two values $p_{\mathrm{LPB}} = p(v_{\mathrm{LPB}} \mid \mathrm{ctx})$ and v_{MPB} are sufficient to describe the current state of the context model: The coding of an MPB naturally results in the decrease of p_{LPB} while the coding of an LPB causes an increase and potentially even a flip of v_{MPB} if $p_{\mathrm{LPB}} > 0.5$. In CABAC, this adaption is handled by a state machine associated to each context model which updates v_{MPB} and p_{LPB} depending on the coded bin value. The 63 states directly correspond to quantized symbol probabilities p_{LPB}, simplifying the probability estimation process significantly. Internally, CABAC uses a 64th state for signaling the termination of the coding process [Wie14].

The probability estimation process is shown in Figure 2.13. All unique 63 states of the state machine are depicted twice, for $v_{\mathrm{MPB}} = 0$ on the left and for $v_{\mathrm{MPB}} = 1$ on the right to

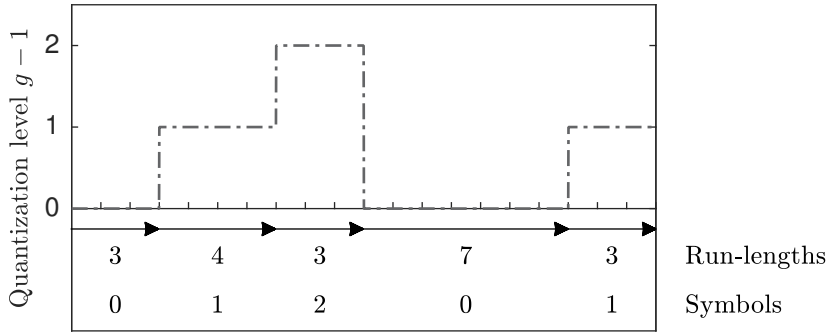


Figure 2.14 Example for run-length coding.

display all possible combinations of v_{MPB} and the probability state indexed with $i_{p_{\mathrm{LPB}}}$ at the same time. Coding an MPB yields a transition $\mathrm{Tr}_{\mathrm{MPB}}$ to a state with lower value p_{LPB} while coding an LPB yields $\mathrm{Tr}_{\mathrm{LPB}}$, a transition to a state with higher value for p_{LPB}. As depicted in Figure 2.13, a faster adaptation is achieved in case of coding an LPB than an MPB. For the state with ID $i_{p_{\mathrm{LPB}}} = 1$, which corresponds to $p_{\mathrm{LPB}} = 0.5$, coding an LPB yields in flipping v_{MPB}. The state with $i_{p_{\mathrm{LPB}}} = 63$ on the other hand corresponds to the lowest value of $p\left(v_{\mathrm{LPB}} \mid \mathrm{ctx}\right)$ where v_{MPB} is most probable. In case of coding another MPB, the state machine remains in this state. The state machine for each context model has to be initialized with starting values for p_{LPB} and v_{MPB}.

Another coding engine, the faster bypass engine, may be used as well. This engine is used in e.g. HEVC to encode bins with nearly equiprobable distribution such as sign flags. The bypass engine is also not explained any further, refer again to [MSW03].

2.4.2 Run-length Coding

Run-length Coding (RLC) is a lossless, reversible coding scheme which describes a sequence of symbols, which all have the same value, with two numbers, namely the *value* and the *run-length* of the sequence. If the input is binary ($N_{\mathrm{q}} = 2$), it is sufficient to store only the value corresponding to the first sequence, since the values between two adjacent sequences toggle. This procedure becomes more efficient with longer sequences and is summarized in the following:

1. Determine the positions of transitions between two sequences,
2. subtract the transition positions to obtain the run-lengths of each sequence and
3. encode the symbol value and the run-length of each sequence with a variable-length code (cf. Section 2.4.1.1).

The obtained run-lengths and symbol values are integer-valued and have to be encoded in a subsequent step. In this thesis, EG codes as presented in Section 2.4.1.1 are used. It is mentioned in [Ohm15] that when using a combination of RLC and variable-length codes, RLC can be interpreted to approach conditional entropy if the input symbols have Markov property. Figure 2.14 shows RLC of an exemplary sequence of symbols g. The run-lengths as well as the values of the sequences are integer-valued and need to be encoded to a bit stream in a subsequent step.

2.5 Time-Frequency Masking for Source Separation

This section deals with time-frequency masking, a core step of source separation. In the following, it is assumed that the mono mix consisting of J sources is constructed by linear instantaneous mixing. Given J TF representations of the sources $\underline{\mathbf{S}}_{\bullet,\bullet,j}$, stored in an $N_{\text{ny}} \times T \times J$ tensor, the $N_{\text{ny}} \times T$ mix spectrogram $\underline{\mathbf{X}}$ is constructed as

$$\underline{\mathbf{X}} = \sum_{j=1}^{J} \underline{\mathbf{S}}_{\bullet,\bullet,j}. \tag{2.32}$$

Note that if the MDCT is used as TF transform, the respective matrices are real-valued. Given some belief about the magnitude spectrogram of the sources which is usually generated by a precedent step to TF masking, it is possible to compute masks $\mathbf{M}_{\bullet,\bullet,j}$ for each source with the same size as $\underline{\mathbf{X}}$. Each element $0 \leq m_{f,t,j} \leq 1$ of $\mathbf{M}_{\bullet,\bullet,j}$ assigns each TF point at position (t, f) to a source j. To estimate the sources, each TF point of the mixture spectrogram $\underline{\mathbf{X}}$ is then weighted with $\mathbf{M}_{\bullet,\bullet,j}$

$$\underline{\tilde{\mathbf{S}}}_{\bullet,\bullet,j} = \underline{\mathbf{X}} \cdot \mathbf{M}_{\bullet,\bullet,j}, \tag{2.33}$$

yielding the complex source estimates $\underline{\tilde{\mathbf{S}}}_{\bullet,\bullet,j}$ which all have the mixture's phase. Furthermore, it holds that $\sum_j m_{f,t,j} = 1$ for all f and t which means that this masking process satisfies the remixing constraint, namely

$$\sum_{j=1}^{J} \underline{\tilde{\mathbf{S}}}_{\bullet,\bullet,j} = \underline{\mathbf{X}}, \tag{2.34}$$

meaning that the estimated sources sum up to the mixture again.

2.5.1 Oracle Masks

It is possible to find optimum masks $\mathbf{M}_{\text{ora}}$ for usage in (2.33) which minimize the squared difference between $\underline{\mathbf{S}}$ and $\underline{\tilde{\mathbf{S}}}$. To find these masks, the squared error between original source and Wiener estimate of each TF point was proposed in [VGP07]

$$\sum_j \left| \underline{x}_{f,t} m_{\text{ora},f,t,j} - \underline{s}_{f,t,j} \right|^2 \approx \sum_j \left| \underline{x}_{f,t} \right|^2 \left(m_{\text{ora},f,t,j} - r_{f,t,j} \right)^2 \tag{2.35}$$

with $r_{f,t,j} = \text{Re}\left\{ \underline{s}_{f,t,j} / \underline{x}_{f,t} \right\}$ denoting the real-part of the ratio between the original source and the mix. The approximation holds up to a constant which is independent of $\mathbf{M}_{\text{ora}}$. Equation (2.35) is a linear least squares problem with bound and linear equality constraints for each TF point. If $r_{f,t,j} \geq 0$ for all j, the solution is $m_{\text{ora},f,t,j} = r_{f,t,j}$. For minimizing (2.35) in the other cases, where some elements $r_{f,t,j} < 0$, the authors of [VGP07] provide MATLAB code which is used in this thesis. Using $\mathbf{M}_{\text{ora}}$ in (2.33) yields *oracle estimates* $\underline{\tilde{\mathbf{S}}}_{\text{ora}}$.

2.5.2 Masks obtained by Nonnegative Tensor Factorization

This thesis deals with source separation algorithms based on NTF which only takes nonnegative magnitude spectrograms as input. To be able to obtain estimates of the sources

in time-domain, complex spectrograms are needed prior to the inverse TF transform when using the STFT. Wiener filtering (2.33) is conducted with masks calculated from the NTF estimates $\hat{\mathbf{V}}(\Theta)$ as

$$\underline{\tilde{\mathbf{S}}}_{\bullet,\bullet,j} = \underline{\mathbf{X}} \cdot \underbrace{\frac{\hat{\mathbf{V}}_{\bullet,\bullet,j}(\Theta)}{\sum_{j'} \hat{\mathbf{V}}_{\bullet,\bullet,j'}(\Theta)}}_{=\mathbf{M}_{\bullet,\bullet,j}(\Theta)} \tag{2.36}$$

with the approximation $\hat{\mathbf{V}}(\Theta)$ dependent on the NTF parameters Θ as defined in Equation (2.14).

2.6 Phase Re-estimation

This section deals with re-estimation of phase in a source separation context. Recall that each of the J estimated source spectrograms $\underline{\tilde{\mathbf{S}}}_{\bullet,\bullet,j}$, which are obtained by the Wiener filter as described in Section 2.5, contain all the mixture's phase denoted with $\angle\underline{\mathbf{X}}$. This assumption may hold true for some TF points, since audio spectrograms are usually sparse and it can be assumed that different sources may play at the same time but not necessarily at the same frequency. However, if e.g. harmonic instruments play alongside with percussive instruments, which usually have flat spectra, the sources overlap at several TF points. In these cases, the mixture's phase is clearly the wrong estimate which leads to artifacts or even destructive interferences. The task of the algorithms to be discussed in this section is to refine the phase or amplitude and phase jointly in an iterative manner. For initialization of these algorithms, the Wiener estimates $\underline{\tilde{\mathbf{S}}}_{\bullet,\bullet,j}$ are used as given in Equation (2.36). The STFT consistency mapping $\mathcal{G}$, defined in Equation (2.8), plays an important role.

Two algorithms will be summarized, first the Griffin-Lim Algorithm (GL) in Section 2.6.1 which is solely refining the phase and second Consistent Wiener Filtering (CWF) in Section 2.6.2 which in addition to the phase also updates the amplitude values. Note that these two algorithms already yield estimates for a monaural mix. When dealing with multi-channel mixtures, sophisticated methods exist, e.g. [DT17]. More detail about phase re-estimation in general can be found in e.g. [SD11; Gna14; GKR15].

2.6.1 Griffin-Lim Algorithm

The Griffin-Lim Algorithm (GL) [GL84] makes use of the STFT consistency discussed in Section 2.1.1. Given the magnitude source spectrograms $\left|\underline{\tilde{\mathbf{S}}}\right|_{\bullet,\bullet,j}$ as obtained with the Wiener filter as given in (2.36), and the mixture's phase $\angle\underline{\mathbf{X}}$, the GL evaluates iteratively the consistency mapping $\mathcal{G}$ given in Equation (2.8), summarized in the following equation

$$\mathbf{\Phi}^{(\mathrm{it}+1)}_{\bullet,\bullet,j} \leftarrow \angle\mathcal{G}\left(\left|\underline{\tilde{\mathbf{S}}}\right|_{\bullet,\bullet,j} \exp\left(\jmath\mathbf{\Phi}^{(\mathrm{it})}_{\bullet,\bullet,j}\right)\right) \tag{2.37}$$

with initialization $\mathbf{\Phi}^{(1)}_{\bullet,\bullet,j} = \angle\underline{\tilde{\mathbf{S}}}_{\bullet,\bullet,j} = \angle\underline{\mathbf{X}}$ for all sources j and iteration $1 \leq \mathrm{it} \leq N_{\mathrm{it}}$ with number of total iterations N_{it}. The magnitude spectrograms $\left|\underline{\tilde{\mathbf{S}}}\right|_{\bullet,\bullet,j}$ obtained originally by the Wiener filter are not modified during this process. This means that the resulting complex spectrograms $\underline{\tilde{\mathbf{S}}}_{\mathrm{GL},\bullet,\bullet,j} = \left|\underline{\tilde{\mathbf{S}}}\right|_{\bullet,\bullet,j} \exp\left(\jmath\mathbf{\Phi}^{(N_{\mathrm{it}}+1)}_{\bullet,\bullet,j}\right)$ do not satisfy the remixing constraint (2.34) as they do not sum up to the mixture spectrogram $\underline{\mathbf{X}}$ anymore.

Note that the GL was not designed originally for the usage in source separation. Usually, it is initialized with zero or random phases. Extensions exist which adapt the GL to source separation, e.g. [GS10; SD11; SD12].

2.6.2 Consistent Wiener Filtering

Consistent Wiener Filtering (CWF) [RV13] jointly re-estimates both amplitude and phase information. As already done for the GL, the STFT consistency is enforced, here with a soft penalty term added to another cost term enforcing the remixing constraint (2.34). The derivation of the total cost function is based on a maximum a posteriori problem under Gaussian assumptions. The authors provide MATLAB code for the case where the number of sources is fixed to $J = 2$, namely for the task of speech-noise separation. This results in a simplified cost function which is given in the following.

The first source, also denoted as the *target* source, is set to the jth source with power spectrogram $\mathbf{V}_\mathrm{s} = \left|\underline{\tilde{\mathbf{S}}}\right|^2_{\bullet\bullet j}$. The second source, also referred to as *noise* source, is calculated as the sum of the remaining sources excluding the jth source, $\mathbf{V}_\mathrm{n} = \sum_{j' \neq j} \left|\underline{\tilde{\mathbf{S}}}\right|^2_{\bullet\bullet j'}$. Here, $\underline{\tilde{\mathbf{S}}}_{\bullet,\bullet,j}$ denotes the Wiener estimates calculated in (2.36). This means that the algorithm is run for each source j independently. The task is now to re-estimate $\underline{\hat{\mathbf{S}}}$, the complex spectrogram of the target source, given the power spectrograms of target and noise, $\mathbf{V}_\mathrm{s}$ and $\mathbf{V}_\mathrm{n}$. These power spectrograms are not modified during the process. As initialization for $\underline{\hat{\mathbf{S}}}$, the Wiener estimate for the target source is taken as

$$\underline{\hat{\mathbf{S}}}_\mathrm{Wiener} = \underline{\mathbf{X}} \cdot \frac{\mathbf{V}_\mathrm{s}}{\mathbf{V}_\mathrm{s} + \mathbf{V}_\mathrm{n}}.$$

The corresponding cost term for obtaining an estimate of the complex target spectrogram $\underline{\hat{\mathbf{S}}}$ is given as

$$\min_{\underline{\hat{\mathbf{S}}}} \sum_{f,t} \left|\underline{\hat{s}}_{f,t} - \underline{\hat{s}}_{\mathrm{Wiener},f,t}\right|^2 \left(\frac{1}{\nu_{\mathrm{s},f,t}} + \frac{1}{\nu_{\mathrm{n},f,t}}\right) + \gamma \sum_{f,t} \left\| \underline{\hat{s}}_{f,t} - \left[\mathscr{G}\left(\underline{\hat{\mathbf{S}}}\right)\right]_{f,t} \right\|^2. \tag{2.38}$$

The first term enforces that $\underline{\hat{\mathbf{S}}}$ stays close to the Wiener estimate, such that the sum of target and noise approximates the mixture. The second summand, weighted with γ, is enforcing the STFT consistency with $\mathscr{G}(\cdot)$ given in Equation (2.8). For $\gamma = 0$, the solution of (2.38) is the Wiener estimate $\underline{\hat{\mathbf{S}}} = \underline{\hat{\mathbf{S}}}_\mathrm{Wiener}$. The authors of [RV13] formulated a conjugate gradient method [She94] for solving (2.38). The initialization for $\underline{\hat{\mathbf{S}}}$ may be either the Wiener estimate $\underline{\hat{\mathbf{S}}}_\mathrm{Wiener}$ or a refined version. This process is used to estimate each of the J sources and setting $\underline{\tilde{\mathbf{S}}}_{\mathrm{CWF},\bullet,\bullet,j} = \underline{\hat{\mathbf{S}}}$.

2.7 Audio Object Coding – State of the Art

This chapter summarizes state-of-the-art algorithms for audio object coding. The different approaches for coding multi channel signals can be roughly divided into two categories, determined by the origins of the algorithm: The algorithms of the first category are designed from a *source separation* perspective, therefore denoted as Informed Source Separation (ISS)

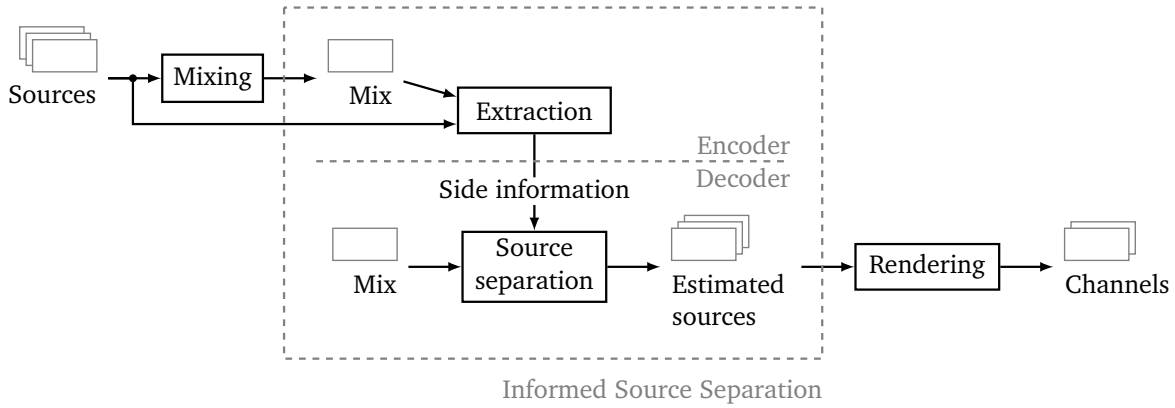


Figure 2.15 Informed Source Separation.

and summarized in Section 2.7.1. ISS unifies the two research fields of source separation and audio coding: The main principle in ISS is to use methods of source separation for the extraction of the audio objects/sources out of their mixture at the decoder, assisted by parameters extracted at the encoder with full knowledge of the sources. The second category consists of algorithms designed by the *audio coding* community. These algorithms are briefly summarized in Section 2.7.2. The algorithms of the two categories are quite similar; the task is to encode multi-channel signals or audio objects which are estimated given their downmix and parameters at the decoder.

2.7.1 Informed Source Separation

As mentioned in the introduction to this section, the name Informed Source Separation (ISS) was coined by the source separation community. In general, the problem of source separation is the estimation of sources given their mixture. In the case of BSS, the sources have to be estimated without any information about the sources but the mixture. The basic approach of ISS is shown in Figure 2.15 where ISS is embedded into a recording and rendering environment. First, the sources are assumed to be recorded and mixed by a professional sound engineer which yields a mixture. This process is typically not part of ISS. In an *encoding stage*, a compact set of side information is extracted with full knowledge of the original sources and the corresponding mix. At a *decoding stage*, this compact side information is used to estimate the sources given the mix by a *source separation* step. After that, the user has the freedom to render the sources spatially, yielding channels for playback, depending on the local loudspeaker setup. The recording and rendering procedures are not part of ISS. Note that during rendering, the sources can be modified in loudness, position or sound which enables active listening. For now, the mix is assumed to be transmitted to the decoder losslessly. The impact of coding the mix is further evaluated in Section 7.3. In the following, several ISS methods are summarized. The focus of this summary lies on methods using Wiener filtering. Other approaches are shortly summarized in the end of this section.

The algorithm proposed in [SD13] utilizes an iterative phase re-estimation technique based on a variation of the Griffin-Lim algorithm. The magnitude spectrograms of the sources have to be quantized and are used for Wiener filtering the sources. As a next step, phase re-estimation is conducted which even may alter the magnitude of the source estimates. The

additional information is the remixing constraint, meaning that all sources should add up to the mixture. In addition to that, a dual resolution TF transform is applied to cope with transients.

A variety of approaches are based on factorization methods: The methods discussed in [LBR10; Liu+11] use NTF for compressing the sources in the TF domain. The resulting NTF model is then quantized and transmitted to the decoder by embedding the data in the mixture with watermarking. The parameters are extracted from the mix and used for Wiener filtering the mixture in the TF domain. Closely related to this is the work of [Nik15] where NTF is used for audio upmixing. Here, the task is defined more generally. Objects are not necessarily assumed to be present in a single channel of the input signal whereas multiple objects can be present in a single channel of the input as well. In general, the task is to extract a signal with more channels out of a downmix with less channels, e.g. upmixing from stereo to 5.1. The NTF process is extended by a perceptually motivated weighting of each TF point. In the decoder, multi-channel Wiener filtering is used. Other extensions of these NTF-based algorithms exist, e.g. exploiting compressive sampling [BOP15] or compressive sampling of signal graphs [Puy+17]. The latter method was proposed very recently. Here, the NTF in the blind setting, with only the mixture as observation, is emulated in the encoder to calculate feature vectors which are then used for applying a compressive graph signal sampling strategy to encode ideal binary masks for the decoder Wiener filtering step. Only information needed for graph reconstruction is transmitted. At the decoder, the NTF in the blind setting is used to reconstruct the graph which then estimates binary masks.

Designed to bridge the fields of ISS and Spatial Audio Object Coding (SAOC), to be discussed in Section 2.7.2, Coding-based Informed Source Separation (CISS) [Oze+11; Liu12; Oze+13] calculates a factorization of the source spectrograms with NTF as well as the other NTF-based ISS algorithms mentioned before. In addition to that, a *residual* of the source spectrograms is calculated which is also modeled with the NTF parameters. This means that both source models and residuals are modeled jointly which is an advantage over SAOC, as pointed out in [Oze+13]. In SAOC, the calculation of the parametric model and the residuals are calculated in two different, independent blocks. Another important contribution of [Oze+13] is the derivation of a rate-distortion function which motivates the quantization of the NTF parameters in the logarithmic domain, assuming parameter transmission at high bit rates. Perceptual modeling was introduced to CISS in [Kır+14]. The perceptual model is very similar to the one proposed in [Nik15].

Besides algorithms based on Wiener filtering, the algorithm of [PGB10; PG11], which was one of the first algorithms denoted as Informed Source Separation, uses a local inversion technique in the TF domain. As another example, the algorithms [GM11; GHM13] use beamforming given a multi-channel mixture. A broader overview of these algorithms is given e.g. in the survey [Liu+12].

2.7.2 Spatial Audio (Object) Coding

In this section, several audio coding standards developed by the Moving Picture Experts Group (MPEG) are briefly summarized.

In Spatial Audio Coding (SAC) [MPE07; Her+04; Her+05], the basic objective is the same compared to ISS: A multi channel signal has to be transmitted by means of a downmix with

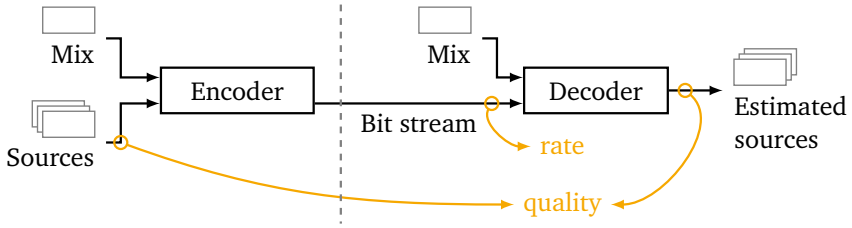


Figure 2.16 Rate and quality measurement points.

less channels and a set of parameters, extracted from the multi channel signal. In the encoder, the multi channel signal is fully known and fed into a filter bank yielding a TF representation. This representation is used to calculate spatial parameters, including channel level differences, time/phase differences, prediction coefficients and inter-channel correlation/coherence cues. At the decoder, the multi channel signal is estimated given the TF representation of the mix and the transmitted parameters. Additionally, residuals between the estimated and original multi channel input are transmitted. Backward compatibility to mono/stereo playback is guaranteed, too.

An extension of SAC is Spatial Audio Object Coding (SAOC) [MPE10; Eng+08; FTH10]. The main difference to SAC is a shift in paradigm: The signals to-be-coded are now called objects, and the transmitted parameters include object level differences, inter-object cross correlations, downmix gains and downmix channel level differences. At the decoder, a full rendering block is included which enables separate synthesis of each object. The recently defined MPEG-H 3D Audio [MPE15; Her+15] combines SAOC with Higher Order Ambisonics (HOA). The main principles of SAOC remain and are extended for the usage with HOA.

2.8 Evaluation Environment

This thesis deals with ISS which estimates the sources playing in a song given their mixture and transmitted parameters at the decoder. To assess the performance of one particular ISS algorithm, the quality of the estimated sources as well as the parameter bit rate have to be taken into account as shown in Figure 2.16.

The original sources $\underline{\mathbf{S}}$ are mixed with linear instantaneous mixing to a mono mixture $\underline{\mathbf{X}} = \sum_j \underline{\mathbf{S}}_{\bullet,\bullet,,j}$. In this thesis, a source separation algorithm in the decoder extracts the spectrograms of the estimated sources $\underline{\tilde{\mathbf{S}}}$ out of the mixture $\underline{\mathbf{X}}$. An inverse TF transform yields the estimated sources $\tilde{\mathbf{s}}'_{\bullet,j}$ in the time domain[5]. The corresponding parameters, which are assisting the source separation step, are entropy-encoded with one of the techniques discussed in Section 2.4 yielding the parameter bit rate R. This means that for evaluation of different ISS algorithms, both quality and rate have to be taken into account.

Section 2.8.1 deals with measures evaluating the separation quality. In Section 2.8.2, it is further shown how the resulting rate-quality curves are obtained, displayed and compared. Section 2.8.3 summarizes the two considered test sets in this thesis.

[5] The prime symbol is added to distinguish the time-domain from the TF domain signals.

2.8.1 Quality Assessment

2.8.1.1 Quality Measures

In [VGF06], quality measures for source separation are proposed and summarized in the following. The authors proposed to decompose the estimated source $\tilde{\mathbf{s}}'_{\bullet,j}$ into $\mathbf{s}_{\text{target}}$, a version of the original source $\mathbf{s}'_{\bullet,j}$ which may have been modified by an allowed distortion, and three error terms as

$$\tilde{\mathbf{s}}'_{\bullet,j} = \mathbf{s}_{\text{target}} + \mathbf{e}_{\text{interf}} + \mathbf{e}_{\text{noise}} + \mathbf{e}_{\text{artif}}. \tag{2.39}$$

The error terms $\mathbf{e}_{\text{interf}}$, $\mathbf{e}_{\text{noise}}$ and $\mathbf{e}_{\text{artif}}$ can be summarized as

- The interference error term $\mathbf{e}_{\text{interf}}$ takes interferences coming from other sources ($j' \neq j$) into account.
- $\mathbf{e}_{\text{noise}}$ is originating from sensor noises.
- The artifacts error term $\mathbf{e}_{\text{artif}}$ accounts for distortions of the sources caused by the separation process such as musical noise or “burbling” artifacts.

This decomposition is then used to formulate the following quality measures, all given in decibel [dB]:

The Signal-to-Distortion Ratio (SDR) measures the ratio of (modified) signal energy to distortion energy where the distortion is defined by the sum of all error terms summarized in Equation (2.39)

$$\text{SDR}_j = 10\log_{10}\frac{\sum_m s^2_{\text{target},m}}{\sum_m \left(e_{\text{interf},m} + e_{\text{noise},m} + e_{\text{artif},m}\right)^2}\,[\text{dB}].$$

The Signal-to-Interferences Ratio (SIR) measures the impact of interferences

$$\text{SIR}_j = 10\log_{10}\frac{\sum_m s^2_{\text{target},m}}{\sum_m e^2_{\text{interf},m}}\,[\text{dB}]$$

and the Signal-to-Artifacts Ratio (SAR) the impact of artifacts

$$\text{SAR}_j = 10\log_{10}\frac{\sum_m \left(s_{\text{target},m} + e_{\text{interf},m} + e_{\text{noise},m}\right)^2}{\sum_m e^2_{\text{artif},m}}\,[\text{dB}], \tag{2.40}$$

which is independent of the SIR due to the addition of $\mathbf{e}_{\text{interf}}$ in the numerator of (2.40). The fourth measure, Signal-to-Noise Ratio (SNR), takes sensor noise into account. In the case of audio source separation, all signals are recorded with a microphone which means that SNR can usually not be estimated and will therefore not be considered in this thesis.

The measures summarized above are averaged over all sources of each mixture yielding a single value for each quality measure and each mix

$$\text{SDR} = \frac{1}{J}\sum_j \text{SDR}_j, \quad \text{SIR} = \frac{1}{J}\sum_j \text{SIR}_j, \quad \text{SAR} = \frac{1}{J}\sum_j \text{SAR}_j. \tag{2.41}$$

2.8.1.2 Normalization

In this thesis, ISS algorithms are evaluated on different mixtures of a certain test set where the mixtures may consist of different number of sources. As further explained in Section 2.8.2, the quality measures and rates are examined *jointly* for all mixtures of the test set. To assess the quality obtained for mixtures consisting of different sources, it is important that the quality measures are as independent as possible of the number of sources present in each mixture. Mixtures consisting of more sources are generally harder to separate compared to mixtures with fewer sources: The possibility of overlapping TF points increases with the number of sources which makes the separation process generally more difficult. On the contrary, different types of sources exist, e.g. harmonic sources where the corresponding spectra are usually not as flat as spectra of percussive instruments which excite a lot of frequencies simultaneously. This means that depending on the *type* of sources present in the mixture, more TF points may overlap for some mixtures compared to other mixtures with the same number of sources. It can be concluded that solely averaging the scores with respect to the number of sources as done in Equation (2.41) does not lead to the scores being completely independent of the number of sources [Liu12].

As a next step, the respective averaged quality measure is therefore put into relation to the quality measure obtained by Wiener filtering with oracle masks as discussed in Section 2.5.1. Recall that the oracle masks are optimum masks calculated with full knowledge of the original sources and that they are an upper bound for algorithms using Wiener filtering. However, the oracle performance is still limited due to the fact that the mixture's phase is used for all estimated sources in Wiener filtering. This reflects the difficulty of separation based on the amount of overlapping TF points since using the mixture's phase introduces more distortion at exactly these TF points. For the example above, dealing with mixtures with the same number but not the same types of sources, different oracle scores will result, namely higher scores for mixtures with less overlapping TF points. In summary, the scores obtained for oracle filtering will account for the difficulty of separation based on different types of sources.

The averaged scores are set into relation to the oracle scores as

$$\delta\mathrm{SDR} = \mathrm{SDR} - \mathrm{SDR}_{\mathrm{ora}}, \quad \delta\mathrm{SIR} = \mathrm{SIR} - \mathrm{SIR}_{\mathrm{ora}}, \quad \delta\mathrm{SAR} = \mathrm{SAR} - \mathrm{SAR}_{\mathrm{ora}}, \tag{2.42}$$

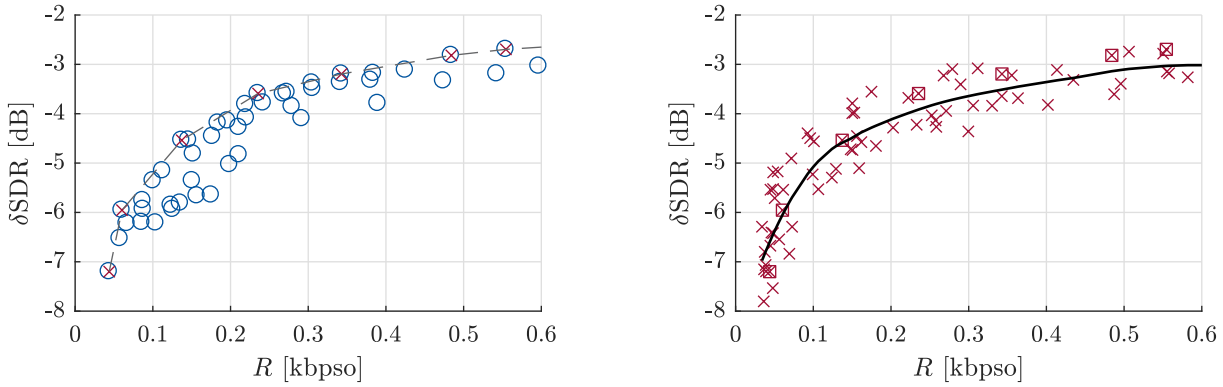
with all scores obtained for oracle Wiener filtering denoted with the subscript "ora". These *differential scores* are always negative for source separation algorithms using solely Wiener filtering for synthesis because the oracle masks are an upper bound for Wiener filtering.

In this thesis, the performance of a BSS algorithm is used as a lower bound to evaluate ISS algorithms operating at very low bit rates. The considered BSS algorithm, as further discussed in Section 3.3, is also based on NTF and uses the same TF transform as the proposed ISS schemes. Another bound simply sets all estimated source spectrograms equal to the mixture spectrogram

$$\underline{\tilde{\mathbf{S}}}_{\mathrm{LB},\bullet,\bullet,j} = \frac{1}{J}\underline{\mathbf{X}}. \tag{2.43}$$

This means that the corresponding oracle mask is equal to $m_{\mathrm{LB},f,t,j} = 1/J$ for all f,t,j in this case[6]. Given the task of extracting source estimates out of the mixture, Equation (2.43)

[6]The normalization with the number of sources simply ensures the remixing-constraint (2.32).



(a) Optimum rate-quality point selection shown for one exemplary mixture.

(b) Smoothing of all optimum points of all mixtures. Optimum points for exemplary mixture in Figure 2.17a marked with rectangles.

Figure 2.17 Parameter selection and smoothing.

estimates each source by the mixture which is the case where *no separation* is performed at all. Assuming neither any bit rate budget nor computational power, returning the mixture is the best source estimate possible in this worst case scenario. Lower bounds are further discussed in Appendix D. In summary, both the performance of a BSS algorithm as well as the estimates given by (2.43) are used as lower bounds in this thesis.

2.8.2 Rate-quality Optimization

In the previous Section 2.8.1, quality measures for evaluating the performance of source separation algorithms were discussed. In the field of ISS however, the rate to be spent for transmission of the parameters has to be taken into account, too. This leads to rate-distortion theory [Sha48; CT06] which models distortion D of the estimated sources at the decoder and rate R by rate-distortion functions $R(D)$. However, this requires statistical modeling of R, D and the sources. In practice, these assumptions can hardly be made.

In this thesis, the encoder-decoder chain is executed for each mixture of a test set (detailed in Section 2.8.3) for different combinations of parameters, e.g. number of reconstruction values, number of NTF components and so on. Given the estimated sources, a quality measure as discussed in Section 2.8.1 is obtained which is in fact the inverse measure to distortion. The resulting parameter bit rate is normalized to both the length of the mixture in seconds and the number of sources/objects, yielding values R with unit "kilo bit per second and object = kbpso". This results in a pair of rate-quality values for each mixture and each parameter combination.

For each mixture, different parameter combinations yield different rate-quality points as shown exemplary in Figure 2.17a for R, δSDR values of one particular mixture. These rate-quality points are not all optimum, as some parameter configurations do yield the same quality for too much bit rate or obtain less quality for the same rate. Therefore, optimum rate-quality points are selected which are members of the highest interconnection graph as shown in Figure 2.17a [Ohm15]. This yields optimum rate-quality points for each mixture which are then all displayed jointly in one plot, as done in Figure 2.17b.

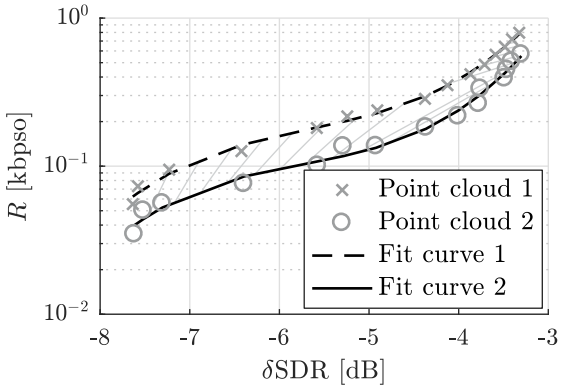


Figure 2.18 Area between exemplary R-D-curves of reference and method under evaluation.

When comparing different ISS algorithms or different configurations of one particular algorithm, the different obtained rate-quality curves have to be compared. This can get confusing if more than two different algorithms/configurations have to be compared. Therefore, two different methods for simplifying the comparison are summarized in the following.

Smoothing

Determining optimum rate-quality points per mixture yields a point cloud which can be smoothed. In [CD88], a locally weighted scatter plot smoothing method was proposed, exploiting local regression using weighted linear least squares which is used for smoothing the rate-quality point cloud in this thesis. An example point cloud and the corresponding smoothed curve are shown in Figure 2.17b.

Bjøntegaard-Delta

As proposed in [Bjø01], the differences between two rate-distortion curves (equivalently for two rate-quality curves) can be expressed by means of the Bjøntegaard Delta (BD). In this thesis, the BD with respect to the bit rate is used: Bjøntegaard Delta Bit Rate (BD-BR) measures the average percentage of rate difference between two rate-quality curves. Input to the BD-BR calculation are two rate-quality point clouds, one for each algorithm/configuration to compare. The obtained rates are converted to the logarithmic domain first. Both rate-quality clouds are then fitted each with a third order polynomial. The BD-BR is then determined by subtracting the areas under both polynomials which are obtained by integration. This process is shown exemplary in Figure 2.18. In this thesis, input to this process are two different rate-quality point clouds which were optimized for each mixture independently. The BD-BR values are then obtained for each mixture which means that point clouds for each mixture are input to the process discussed above. The resulting BD-BR values are finally averaged.

Summary

In the following, the procedure of finding optimum rate-quality points and different ways of evaluating them is summarized for SDR. The procedures for the other scores, SIR and SAR are equivalent.

1. Average the score obtained for each source, SDR_j, $\mathrm{SDR} = 1/J \sum_j \mathrm{SDR}_j$ and normalize the resulting number with respect to oracle score $\mathrm{SDR}_{\mathrm{ora}}$ as $\delta\mathrm{SDR} = \mathrm{SDR} - \mathrm{SDR}_{\mathrm{ora}}$.
2. For each mixture, calculate optimum $R, \delta\mathrm{SDR}$ points.
3. Given the optimized $R, \delta\mathrm{SDR}$ points, different ways of displaying the results may be chosen from:
   a) Display a scatter plot of all optimized $R, \delta\mathrm{SDR}$ points.
   b) Plot smoothed $R, \delta\mathrm{SDR}$ curves.
   c) For obtaining BD-BR: For each mixture, fit third-order polynomial and calculate the BD-BR for each mix. Averaging yields global BD-BR value, comparing the performances of two different algorithms/configurations.

2.8.3 Test Sets

In this thesis, two independent test sets are considered for evaluating the proposed ISS algorithms. To obtain quality measures as discussed in Section 2.8.1, the original sources are needed. The two test sets are used for evaluating source separation algorithms during the Signal Separation Evaluation Campaign (SiSEC)[7].

2.8.3.1 Test set $\mathscr{A}$

Test set $\mathscr{A}$ is composed of ten mixtures consisting of four to seven sources of the QUASI database[8]. Each recording is sampled at 44100 Hz, quantized with 32 bit per sample and is 30 s long. The musical genres are pop, electropop, rock, reggae and bossa nova. More details about each mixture, namely interpret and title, number and types of sources is given in appendix Section A.1.

2.8.3.2 Test set $\mathscr{B}$

Test set $\mathscr{B}$ is composed of 100 mixtures, consisting of four sources (bass, drums, vocals, other) of the DSD100 database[9]. 30 s long segments were cropped out of each recording, each sampled at 44100 Hz and quantized with 16 bit per sample. More detail about all songs is given in appendix Section A.2.

[7] http://sisec.inria.fr.
[8] http://www.tsi.telecom-paristech.fr/aao/en/2012/03/12/quasi.
[9] http://sisec.inria.fr/sisec-2016/2016-professionally-produced-music-recordings.

3 Reference Algorithms

This chapter deals with two reference algorithms, one ISS and one BSS algorithm, both utilizing NTF. The ISS reference algorithm is used as a baseline throughout this thesis: All contributions of this thesis are extensions of this algorithm and will be compared to it as well. The BSS algorithm will be used in Chapter 5 to extend the decoder of the reference ISS algorithm.

A detailed summary of the reference ISS algorithm is given in Section 3.1. In Section 3.2, the reference ISS algorithm is evaluated to find reasonable parameter settings for the comparison of the reference algorithm to the extensions proposed in this thesis. A short summary of an NTF-based algorithm for BSS is given in Section 3.3, as it will be used in Chapter 5. The BSS algorithm was thoroughly investigated in [Bec16].

3.1 Reference Algorithm for Informed Source Separation

This thesis deals with extensions of NTF-based informed source separation. As a reference, a modified version of the algorithm of [Liu+11] is used which was already briefly discussed in Section 2.7.1. This method is often chosen in ISS-related publications as a baseline, e.g. [Oze+13; RBW16; Puy+17]. In principle, NTF is used for compressing the sources and the resulting NTF-parameters for Wiener filtering of the mixture to extract the estimated sources. The authors of [Liu+11] also proposed another encoding scheme, namely JPEG compression of the source amplitude spectrograms. Since this thesis deals only with NTF-based methods, this approach is not considered here.

Figure 3.1 depicts the flow graph of the reference method in the TF domain. The corresponding transform blocks are omitted for conciseness. In the following, both reference encoder in Section 3.1.1 and decoder in Section 3.1.2 are summarized. Since Chapter 6 deals with residual coding in the TF domain, another reference method, also based on the algorithm of [Liu+11], is briefly discussed in Section 3.1.3.

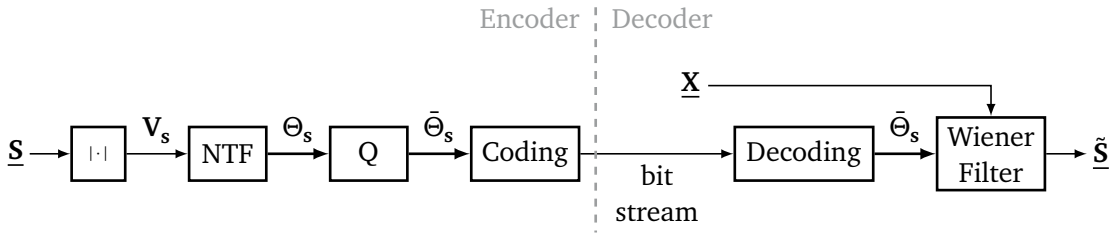


Figure 3.1 Block diagrams of reference encoder and decoder [Liu+11].

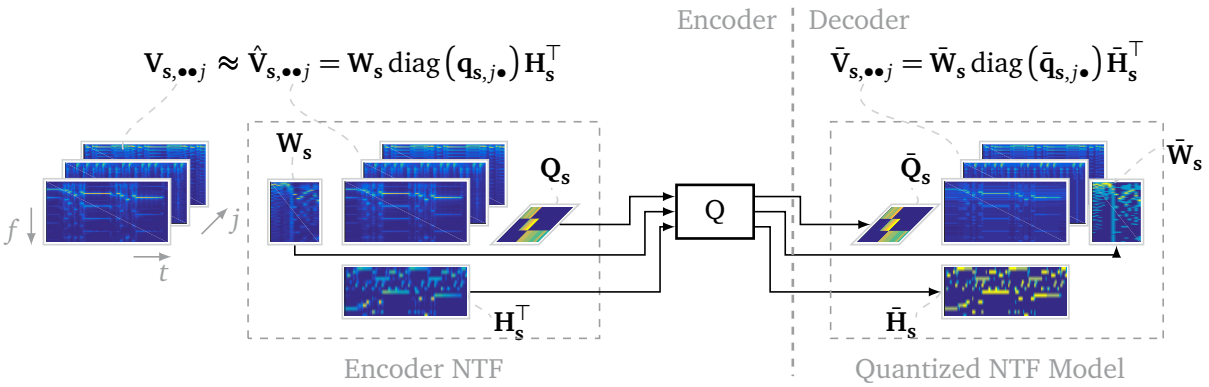


Figure 3.2 Estimation and quantization of source NTF model $\Theta_\mathbf{s}$, depicted for $J = 3$ sources, $K = 15$ components and $N_\mathrm{q} = 4$ quantization reconstruction values.

3.1.1 Encoder

Preprocessing All signals depicted in Figure 3.1 are assumed to be in the time-frequency domain: The TF coefficients of the J sources are stored in the $N_\mathrm{ny} \times T \times J$ tensor $\underline{\mathbf{S}}$. In this thesis, linear instantaneous mixing is assumed which means that the mix is calculated with (2.32) which a $N_\mathrm{ny} \times T$ complex matrix $\underline{\mathbf{X}}$. Taking the α-modulus of the sources $\underline{\mathbf{S}}$ in the TF domain yields the source amplitude spectrogram

$$\mathbf{V}_\mathbf{s} = \left|\underline{\mathbf{S}}\right|^\alpha, \tag{3.1}$$

a tensor of same size as $\underline{\mathbf{S}}$. Here, $\alpha \in \{1, 2\}$ is depending on the choice of the NTF cost function. In [Bec16; Spi12], the influence of α on the separation quality of an NTF-based blind source separation algorithm has been evaluated. A dependency on β has been proposed as

$$\alpha = \begin{cases} 1 & \text{for } \beta \neq 0 \\ 2 & \text{for } \beta = 0\,. \end{cases}$$

This dependency was also theoretically investigated in [LB15] and experimentally verified in [Bec16; Spi12]. Note that originally, $\beta = 0$ and $\alpha = 2$ were fixed in [Liu+11].

Nonnegative tensor factorization (NTF) The amplitude spectrogram $\mathbf{V}_\mathbf{s}$ of size $N_\mathrm{ny} \times T \times J$ is now input to the subsequent NTF. In this thesis, the β-divergence is used as the NTF cost function as described in Section 2.2

$$\min\ d_\beta\left(\mathbf{V}_\mathbf{s} \,|\, \hat{\mathbf{V}}_\mathbf{s}(\Theta_\mathbf{s})\right) \tag{3.2}$$

with the approximation of the source spectrograms $\hat{\mathbf{V}}_{\mathbf{s},\bullet\bullet j}(\Theta_\mathbf{s}) = \mathbf{W}_\mathbf{s}\,\mathrm{diag}\left(\mathbf{q}_{\mathbf{s},j\bullet}\right)\mathbf{H}_\mathbf{s}^\top$ which is also given in (2.13). The resulting NTF parameters, $\mathbf{W}_\mathbf{s}$, $\mathbf{H}_\mathbf{s}$, and $\mathbf{Q}_\mathbf{s}$, are gathered under $\Theta_\mathbf{s} = \{\mathbf{W}_\mathbf{s}, \mathbf{H}_\mathbf{s}, \mathbf{Q}_\mathbf{s}\}$ and quantized in the following step. The subscript $\mathbf{s}$ is used to discriminate this NTF-model from other models introduced in Chapter 5.

Quantization The grouping matrix $\mathbf{Q}_\mathbf{s}$ only has few elements compared to $\mathbf{W}_\mathbf{s}$ and $\mathbf{H}_\mathbf{s}$. In this thesis, the focus lies only on the costly coding of $\mathbf{W}_\mathbf{s}$ and $\mathbf{H}_\mathbf{s}$. The smaller matrix $\mathbf{Q}_\mathbf{s}$

is always quantized uniformly with a high number of reconstruction values. The authors of [Oze+13] propose to quantize $\mathbf{W}_\mathbf{s}$ and $\mathbf{H}_\mathbf{s}$ in the logarithmic domain,

$$\bar{\mathbf{W}}_\mathbf{s} = \exp\left(q\left(\log \mathbf{W}_\mathbf{s}\right)\right), \quad \bar{\mathbf{H}}_\mathbf{s} = \exp\left(q\left(\log \mathbf{H}_\mathbf{s}\right)\right). \tag{3.3}$$

In [Oze+13], this choice is proven to be optimum for high rates when minimizing the Itakura-Saito distance ($\beta = 0$). All quantized matrices are gathered under $\bar{\Theta}_\mathbf{s} = \left\{\bar{\mathbf{W}}_\mathbf{s}, \bar{\mathbf{H}}_\mathbf{s}, \bar{\mathbf{Q}}_\mathbf{s}\right\}$.

Encoding The reconstruction values and indices are encoded with GZIP. Other encoding schemes are further investigated in Chapter 4.

Note that since T spectra for each source have to be gathered to form $\mathbf{V}_\mathbf{s}$ prior to NTF, the delay of the encoder is at least as large as T.

3.1.2 Decoder

It is assumed that the mixture is encoded with high quality prior to transmission to the decoder. This constraint is further discussed in Chapter 7. As a first step, the mixture in the time-domain has to be transformed to the TF domain with the STFT yielding the complex mixture spectrogram $\underline{\mathbf{X}}$. The transmitted parameters $\bar{\Theta}_\mathbf{s}$ are extracted from the bit stream and used for Wiener filtering $\underline{\mathbf{X}}$ with Equation (2.36): First, the quantized NTF model is calculated as

$$\bar{\mathbf{V}}_{\mathbf{s},\bullet\bullet_j}\left(\bar{\Theta}_\mathbf{s}\right) = \bar{\mathbf{W}}_\mathbf{s} \operatorname{diag}\left(\bar{\mathbf{q}}_{\mathbf{s},j\bullet}\right) \bar{\mathbf{H}}_\mathbf{s}^\top \tag{3.4}$$

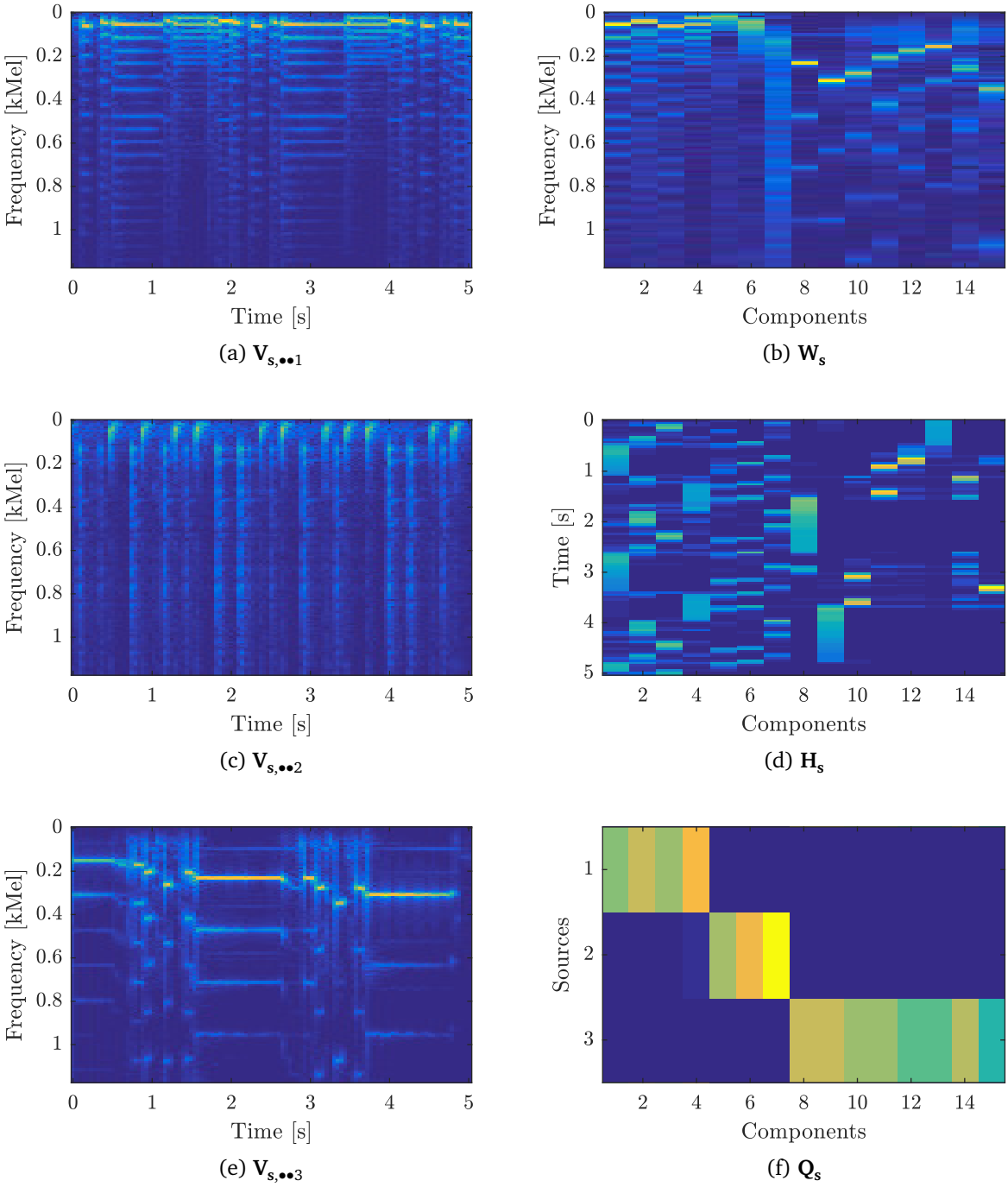
which is then used secondly for filtering $\underline{\mathbf{X}}$ yielding the estimated source spectrograms $\underline{\tilde{\mathbf{S}}}_{\bullet\bullet_j}$ with Equation (2.36). The subsequent ISTFT yields the estimated sources in the time-domain. Figure 3.2 compares the quantized NTF model $\bar{\mathbf{V}}_{\mathbf{s},\bullet\bullet_j}$ for $J = 3$ exemplary sources. In Figures 3.3 and 3.4, the exemplary NTF results of Figure 3.2 are shown with more detail.

3.1.3 Residual Transmission: Coding-based Informed Source Separation

This section briefly summarizes an extension to the reference NTF-based ISS algorithm, Coding-based Informed Source Separation (CISS) [Oze+13] which was already mentioned in Section 2.7.1. The process of estimating the source magnitude spectra is the same for CISS and the encoder discussed in Section 3.1.1. However, the calculated NTF model is not only used for calculating Wiener filters but also for determining a residual between original and estimated sources in the Karhunen–Loève Transform (KLT) domain:

For each TF point, a posterior covariance matrix can be estimated given $\hat{\mathbf{V}}_\mathbf{s}$ as computed with (2.13). This $J \times J$ matrix, denoted with $\mathbf{\Sigma}_{f,t,\bullet,\bullet}$, models the covariance of the J sources with the mix as observation at TF point (f,t). The underlying probabilistic framework is out of scope of this thesis, refer to [Oze+13; Liu12] for more detail. In [Oze+13], it is then proposed to decompose $\mathbf{\Sigma}_{f,t,\bullet,\bullet} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ with the eigenvalue decomposition. The residual is then calculated in the encoder in the KLT domain as

$$\mathbf{z}_{f,t,\bullet} = \mathbf{U}^\top\left(\underline{\mathbf{S}}_{f,t,\bullet} - \underline{\tilde{\mathbf{S}}}_{f,t,\bullet}\right),$$

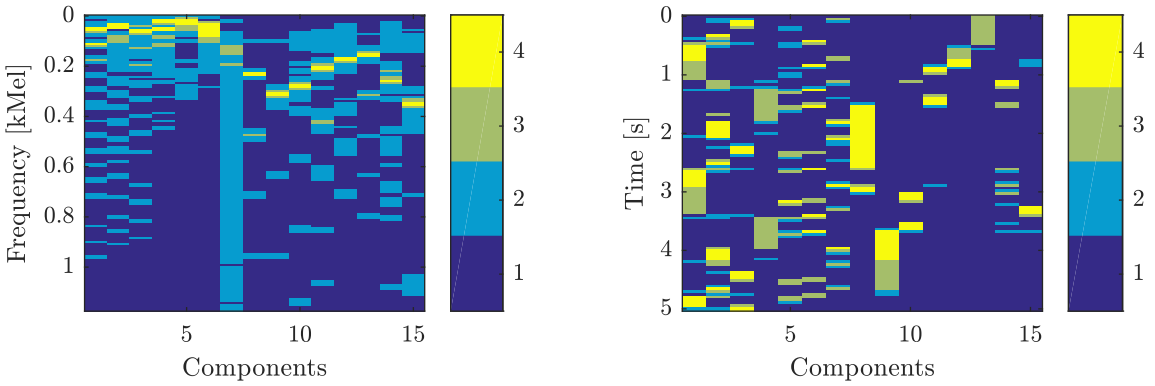


(a) $\mathbf{V}_{s,\bullet\bullet 1}$ (b) $\mathbf{W}_s$

(c) $\mathbf{V}_{s,\bullet\bullet 2}$ (d) $\mathbf{H}_s$

(e) $\mathbf{V}_{s,\bullet\bullet 3}$ (f) $\mathbf{Q}_s$

Figure 3.3 Input and parameters of the NTF in the encoder with $J = 3$ sources and $K = 15$ components.



(a) $\mathbf{G}_{\mathbf{W}_s}$ (b) $\mathbf{G}_{\mathbf{H}_s}$

Figure 3.4 Quantization indices corresponding to $\mathbf{W_s}$ and $\mathbf{H_s}$.

quantized with uniform scalar quantization and encoded with arithmetic coding. At the decoder, $\mathbf{\Sigma}_{f,t,\bullet,\bullet}$ is calculated given the transmitted NTF parameters and the eigenvalue decomposition of $\mathbf{\Sigma}_{f,t,\bullet,\bullet}$ has to be calculated again for each TF point. The Wiener estimates $\underline{\tilde{\mathbf{S}}}$ are then refined given $\mathbf{U}$ and the transmitted residual $\bar{\mathbf{z}}_{f,t,\bullet}$ as

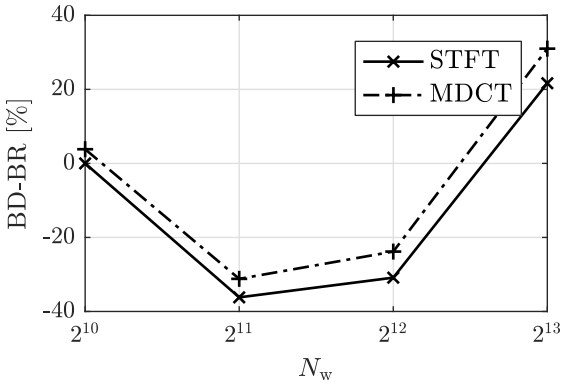
$$\underline{\tilde{\mathbf{S}}}_{\mathrm{CISS},f,t,\bullet} = \underline{\tilde{\mathbf{S}}}_{f,t,\bullet} + \mathbf{U}\bar{\mathbf{z}}_{f,t,\bullet}.$$

In both encoder and decoder, the eigenvalue decomposition of the covariance matrix has to be calculated for each TF point. On the one hand, this procedure enables separation qualities which are not bounded by the oracle estimators anymore, but it introduces more complexity on the other hand.

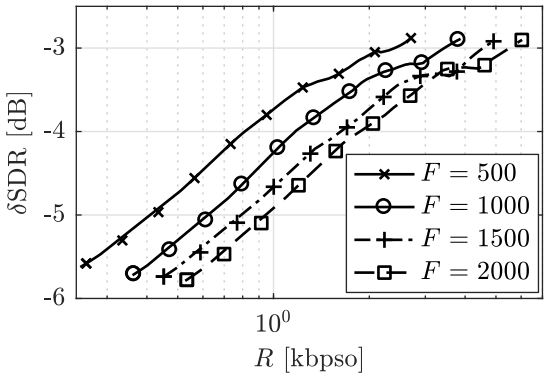
3.2 Preliminary Experiments

The reference algorithm [Liu+11] was not optimized for low bit rates but for a more general rate range. The only parameter influencing the bit rate is the number of NTF components per source K/J; the number of reconstruction values is fixed to $N_\mathrm{q} = 2^8$. In this chapter, three variations of the reference algorithm are proposed: First, it is shown in Section 3.2.1 that compressing the frequency dimension of the STFT matrices yields significantly lower bit rates while preserving quality. Second, the Kullback–Leibler divergence ($\beta = 1$) instead of Itakura-Saito ($\beta = 0$) as used in [Liu+11] is chosen. In [Bec16; RBW16] it is reported that $\beta = 1$ outperforms $\beta = 0$ significantly in both the blind and in the informed source separation scenario. In Section 3.2.2, results are given for the ISS scenario. Third, different quantizer settings are compared in Section 3.2.3 by allowing for different numbers of reconstruction values N_q, showing that the bit rate is reduced even further.

Unless mentioned otherwise, the number of NTF components per source is set to $K/J \in \{1, 2, \ldots, 10\}$ in this section. GZIP is used for encoding the quantized parameters $\bar{\Theta}_\mathbf{s}$.



(a) Comparison of STFT and MDCT. BD-BR values for four different window sizes N_w.



(b) Rate-quality curves for mel filtering of frequency axis for STFT with $N_\mathrm{w} = 2^{12}$ and $N_\mathrm{h} = 2^{11}$.

Figure 3.5 Evaluation of the TF transform.

3.2.1 Time-frequency Transform

As time-frequency transform, the algorithm of [Liu+11] uses the MDCT as TF transform. In [Oze+13], the MDCT was compared to the STFT which outperforms the MDCT. As discussed in Section 2.1.3, mel filtering the spectral dimension of the TF matrices is applied as a subsequent step to the STFT/MDCT. The mel filter bank $\mathbf{H}$, an $N_\mathrm{ny} \times F$ matrix storing F triangular filters, is applied to the modulus of the complex sources $\underline{\mathbf{S}}_{\bullet\bullet j}$

$$\mathbf{S}_{\mathrm{mel},\bullet\bullet j} = \mathbf{H}_\mathrm{mel}^\top \left|\underline{\mathbf{S}}\right|_{\bullet\bullet j} \tag{3.5}$$

yielding the $F \times T \times J$ tensor $\mathbf{S}_\mathrm{mel}$. $\mathbf{S}_\mathrm{mel}$ then replaces $\left|\underline{\mathbf{S}}\right|$ in Equation (3.1) to yield $\mathbf{V}_\mathbf{s}$. A similar approach was used in e.g. [SD13] for ISS with iterative phase-estimation. Here, the complete source spectrograms are quantized, but first, the spectrograms are transformed to the Equivalent Rectangular Bandwidth (ERB) scale, another logarithmic scaling similar to the mel scale.

Figure 3.5 shows results for a preliminary evaluation of the TF transform. The NTF parameters are quantized as proposed in [Liu+11] with $N_\mathrm{q} = 2^8$ reconstruction values in the logarithmic domain. Figure 3.5a compares the performances of the STFT and the MDCT by means of BD-BR savings, calculated with respect to the STFT with $N_\mathrm{w} = 2^{10}$. As a score, the SDR is chosen. Both the STFT and the MDCT are evaluated with different values for the window length $N_\mathrm{w} = \{2^{10}, \ldots, 2^{13}\}$ with the corresponding hop size set to $N_\mathrm{h} = \frac{N_\mathrm{w}}{2}$, yielding windows with 50 % overlap.

- For all values of N_w, the STFT outperforms the MDCT. As already discussed in Section 2.1.2, the MDCT needs the same number of TF coefficients as coefficients in the time-domain as it is critically sampled. This is an advantage over the STFT which yields twice the amount of TF coefficients with a 50 % overlap. However, it was also pointed out that the MDCT is not invariant to time-shifts which apparently has negative impact on the factorization process. Although being critically sampled, the MDCT yields slightly worse results than the STFT. This behavior was also observed in [Oze+13]. In the following, the STFT is used as TF transform.

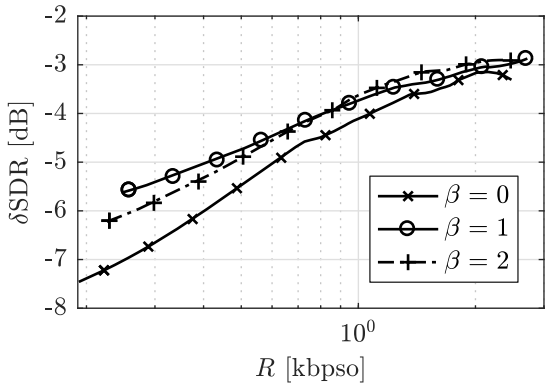


Figure 3.6 Rate-quality curves for β.

- For the STFT, $N_\mathrm{w} = 2^{11}$ and $N_\mathrm{w} = 2^{12}$ yield similar values for BD-BR, namely $-36\,\%$ savings for $N_\mathrm{w} = 2^{11}$ and $-31\,\%$ for $N_\mathrm{w} = 2^{12}$. Although $N_\mathrm{w} = 2^{11}$ yields slightly better BD-BR results, $N_\mathrm{w} = 2^{12}$ is selected in the following: One of the main contributions of this thesis is the introduction of an NTF-based BSS algorithm to the ISS decoder. This BSS algorithm was thoroughly evaluated in [Bec16] where it is reported that $N_\mathrm{w} = 2^{12}$ yields better separation results than $N_\mathrm{w} = 2^{11}$. Therefore, not only the performance of the reference ISS method but also the performance of the BSS algorithm to be introduced to the decoder in Chapter 5 is accounted for by choosing $N_\mathrm{w} = 2^{12}$. The other two options, $N_\mathrm{w} = 2^{10}$ and $N_\mathrm{w} = 2^{13}$ are increasing the rate as they result in either too many frames or frames which are too large.

In summary, the STFT is used with window size $N_\mathrm{w} = 2^{12}$ and hop size $N_\mathrm{h} = 2^{11}$ from now on, corresponding to 93 ms and 46.5 ms.

Figure 3.5b shows the effect of applying mel filtering subsequently to the STFT. Rate-quality curves are shown for different numbers of mel filters F. As a score, δSDR is used here as discussed in Section 2.8.1 and the oracle performance is calculated with the STFT and $N_\mathrm{w} = 2^{12}$ with 50 % overlap. In the following, $F = 500$ is chosen as it gives the highest bit rate savings without loosing any quality. This value also coincides with the findings of [Bec16] for BSS.

3.2.2 Factorization

For sake of completeness, the factorization process is briefly evaluated in this section. Several parameters have an impact on the factorization results, for example the choice of initialization or the number of iterations of the multiplicative update process. Here, the impact of the cost function on the factorization process is evaluated. As discussed in Section 2.2.2.1, the results of a complex SVD on $\underline{\mathbf{S}}$ is used as initialization. The number of NTF iterations is fixed to $N_\mathrm{it} = 200$. For a more thorough evaluation of the other NTF parameters, refer to e.g. [Bec16].

Figure 3.6 shows the corresponding results. $\beta = 0$ (IS distance) is outperformed by both $\beta = 1$ (KL divergence) and $\beta = 2$ (Euclidean distance). Also found by [Bec16], $\beta = 1$ is chosen from now on as it yields better results for lower bit rates than $\beta = 2$.

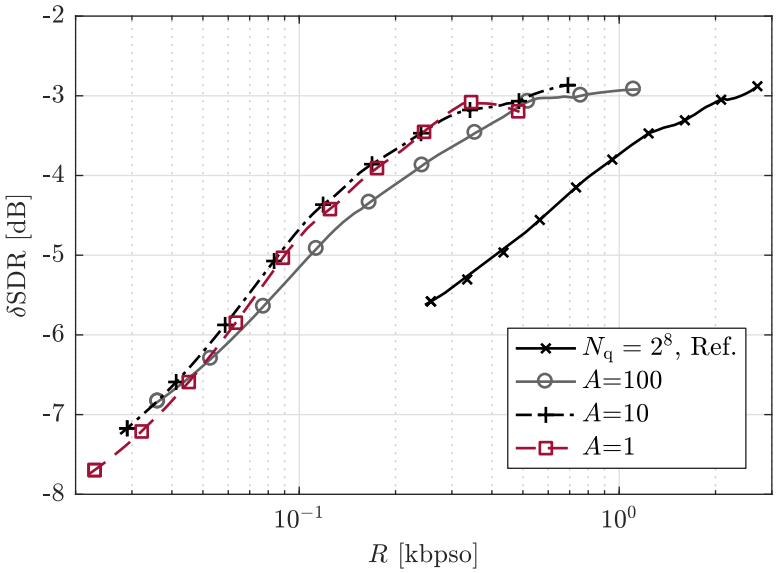


Figure 3.7 Rate-quality curves of quantization with $F = 500$.

3.2.3 Quantization

In [Liu+11], the NTF parameters $\mathbf{W_s}$, $\mathbf{H_s}$ and $\mathbf{Q_s}$ were quantized with $N_q = 2^8$ reconstruction values in the logarithmic domain as given in Equation (3.3). Since the low bit rate scenario is of interest in this thesis, the quantizer has to be adapted to this particular setting. First, less reconstruction values $N_q \in \{2, 3, 4, 8, 16\}$ are used. This parameter is included in the rate-quality optimization process as discussed in Section 2.8.2. For each mix, optimum rate-quality points are found for K/J and N_q. Second, A-law companding instead of the logarithm is used with the A-law companding curve given in Equation (2.23). The main difference to using the logarithm directly is the linear interval for lower amplitudes and the compression parameter A which models both uniform ($A = 1$) and non-uniform quantization ($A > 1$). Equation (3.3) is thus replaced by

$$\bar{\mathbf{W}}_\mathbf{s} = C_A^{-1}\left(q\left[C_A(\mathbf{W_s})\right]\right), \quad \bar{\mathbf{H}}_\mathbf{s} = C_A^{-1}\left(q\left[C_A(\mathbf{H_s})\right]\right) \tag{3.6}$$

using companding and expanding functions $C_A(\cdot)$ and $C_A^{-1}(\cdot)$ as given in Equations (2.23) and (2.24). $\mathbf{Q_s}$ is still quantized with high precision with $A = 1$ and 2^8 reconstruction values.

As a first evaluation, the different settings of N_q are compared, namely $N_q = 2^8$ with logarithmic quantization given in (3.3) to the proposed quantization scheme with $N_q \in \{2, 3, 4, 8, 16\}$ and $A \in \{1, 10, 100\}$. The corresponding δSDR results are shown in Figure 3.7. The curve for $N_q = 2^8$ obtained for $F = 500$ (—×—) is identical in Figures 3.5b and 3.7. Using $A = 100$ with $N_q = 2^8$ gives comparable results to the reference (—×—) using Equation (3.3) and is not shown in Figure 3.7. The results can be summarized as follows:

- When comparing the results of reference with $N_q = 2^8$ (—×—) to the performance of A-law companding with $N_q < 8$, it becomes clear that when lowering N_q and including it into the rate-quality optimization (Section 2.8.2) decreases the bit rate significantly.
- Using $A = 100$ (—○—) increases the bit rate compared to $A = 1$ (—□—) and $A = 10$ (–+–·). It seems that too much precision for higher amplitudes is lost when companding with $A = 100$.

- The results achieved with $A = 1$ (−□−) and $A = 10$ (−+−·) are comparable with two differences: Although $A = 1$ enables slightly smaller bit rates than $A = 10$, it is not able to yield as high δSDR results as $A = 10$ for rates around 0.7 kbpso.

$A = 10$ is chosen as a compromise between $A = 1$ and $A = 100$ as it enables lower rates than $A = 100$ and yields better quality for high rates compared to $A = 1$.

As a second evaluation, the performance of companding with $A = 10$ and using LM (cf. Section 2.3.3) with $A = 1$ are compared to estimate non-uniform reconstruction values depending on the input data. Note that in this case, the estimated reconstruction values have to be transmitted in addition to the group indices. A-law companding with $A = 10$ and LM ($A = 1$) are evaluated with two configurations: The unmodified version of LM is compared to a version of LM where the first reconstruction value is fixed to a small nonnegative value. The reconstruction value is replaced in each LM iteration. Figure 3.8 shows R,δSIR curves measuring the interference per rate. Regarding the obtained distortion measured with δSDR, all three quantizers yield very similar results which are therefore not shown here.

- Examining the obtained interferences between the sources, the reason of the modification of LM becomes clear. The unmodified version (✳) yields noticeably worse δSIR results than the modified version (□) for all mixtures. This can be explained by the fact that LM often chooses a value greater than zero for the lowest quantization value, for both $\bar{\mathbf{W}}_\mathbf{s}$ and $\bar{\mathbf{H}}_\mathbf{s}$. This then leads to Wiener masks $\mathbf{M}$ in Equation (2.36) which are never zero and thus introduce interference from the other sources. Fixing the first reconstruction value to a small nonnegative number[1], set here to 10^{-6}, solves this problem.

- $A = 10$ (+) yields slightly better δSIR results compared to the modified LM (□) for most mixtures. $A = 10$ is therefore chosen for most of the evaluations in this thesis.

3.3 Reference Algorithm for Blind Source Separation

In Chapter 5, the source separation step of the reference decoder, the Wiener filter, will be replaced by a more complex algorithm for BSS which is briefly summarized here. As mentioned in Section 2.2.2, many BSS algorithms exist using factorization methods such as NTF. The BSS algorithm considered here uses NTF on the mixture and is evaluated in detail in e.g. [Bec16]. Most of the building blocks are already used in the ISS encoder and do not need further explanation. The flow graph is shown in Figure 3.9:

Factorization The mixture spectrogram $\underline{\mathbf{X}}$ in the TF domain is transformed to the mel domain with mel filter bank $\mathbf{H}_\text{mel}$ as $\mathbf{V}_\mathbf{x} = \left(\mathbf{H}_\text{mel}^\top \left|\underline{\mathbf{X}}\right|\right)^\alpha$ which was already proposed for the sources in Section 3.2.1. The resulting mixture spectrogram $\mathbf{V}_\mathbf{x}$ of size $F \times T$ is then factorized by NTF with the grouping information set to $q_{\mathbf{x},1,k} = 1$ for all k as already discussed in Section 2.2. The NTF minimizes the β-divergence between mix $\mathbf{V}_\mathbf{x}$ and its approximation $\hat{\mathbf{V}}_\mathbf{x}(\Theta_\mathbf{x}) = \mathbf{W}_\mathbf{x}\mathbf{H}_\mathbf{x}^\top$

$$\min\ d_\beta\left(\mathbf{V}_\mathbf{x} \,|\, \hat{\mathbf{V}}_\mathbf{x}(\Theta_\mathbf{x})\right) \tag{3.7}$$

[1] Setting the lowest reconstruction value to zero could lead to division by zero in (2.36).

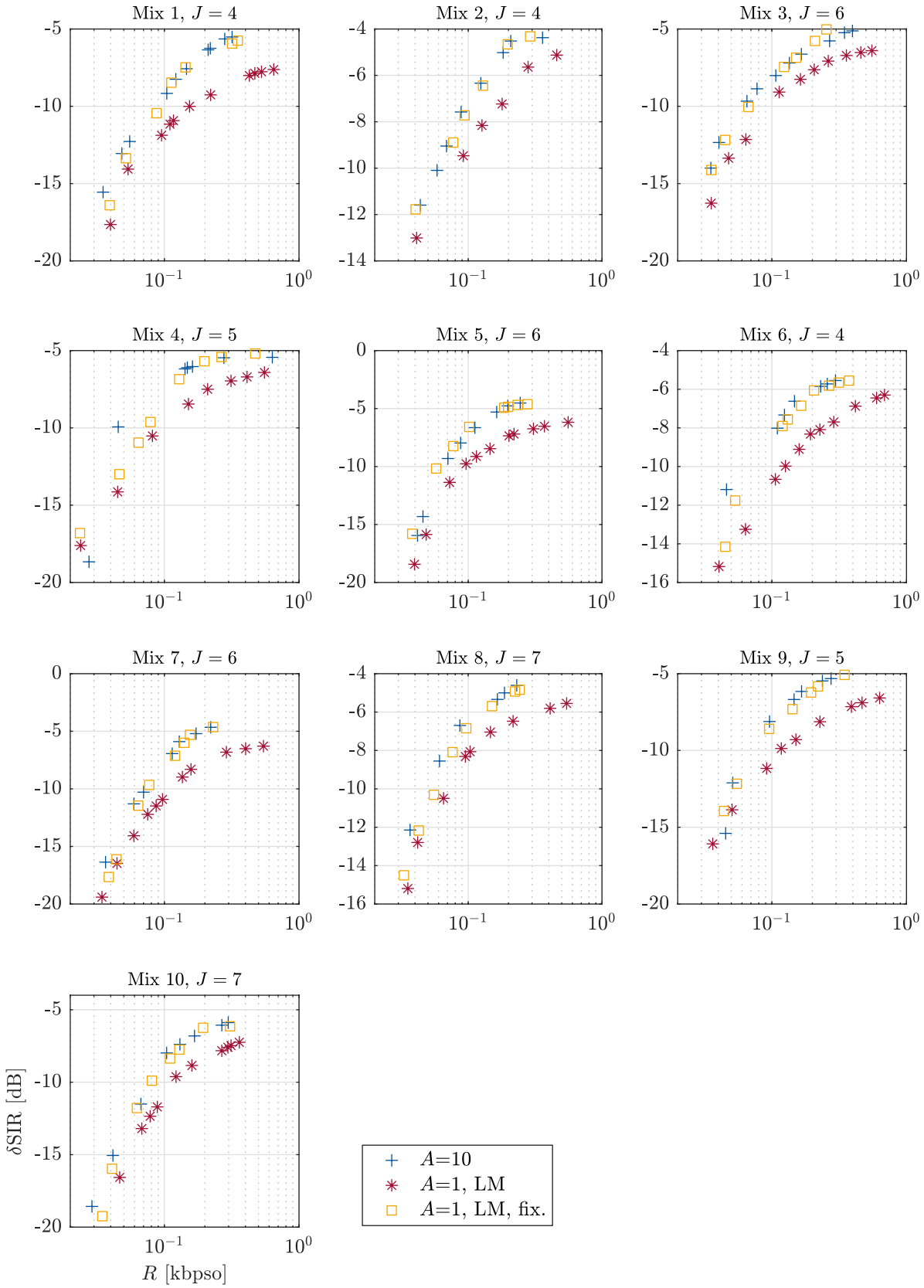


Figure 3.8 R,δSIR curves comparing the performance of A-law companding to Lloyd-Max algorithm.

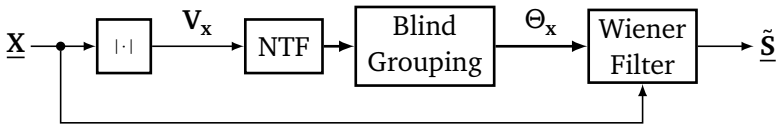


Figure 3.9 Block diagrams of reference BSS algorithm [Spi12; Bec16].

with the mix model

$$\Theta_{\mathbf{x}} = \{\mathbf{W}_{\mathbf{x}}, \mathbf{H}_{\mathbf{x}}, \mathbf{1}\} \tag{3.8}$$

where $\mathbf{W}_{\mathbf{x}}$, $\mathbf{H}_{\mathbf{x}}$ are of sizes as $F \times K$ and $T \times K$ respectively and $\mathbf{1}$ denotes a vector consisting of K ones. Regarding the initialization of $\Theta_{\mathbf{x}}$, different choices are available as already discussed in Section 2.2.2.1.

Blind grouping To estimate the J sources by Wiener filtering the mix $\underline{\mathbf{X}}$, the information about the mapping of the K components obtained by the NTF to the J sources is needed. This information, stored in $J \times K$ matrix $\mathbf{Q}_{\mathbf{x}}$, is estimated blindly in the blind grouping block which is thoroughly investigated in [Spi12]: The mapping of components to sources can be conducted in a blind manner by calculating meaningful features on $\mathbf{W}_{\mathbf{x}}$ and $\mathbf{H}_{\mathbf{x}}$ which belong to the family of Mel Frequency Cepstral Coefficients (MFCC) features. These features are then fed into a clustering algorithm, e.g. fuzzy c-means. During clustering, the K features calculated on each column of $\mathbf{W}_{\mathbf{x}}$ and/or $\mathbf{H}_{\mathbf{x}}$ are mapped to J centroids, each corresponding to one of the sources. This means that the only information the blind algorithm of [Spi12] needs is the number of sources J.

As an upper bound for the blind grouping algorithm, the author of [Vir07] proposes a reference grouping algorithm which calculates the mapping from components to sources with knowledge of the original sources. A hill-climbing approach is used here as proposed in [Spi12] to fasten up the calculation. For Wiener filtering with Equation (2.36), $\mathbf{Q}_{\mathbf{x}}$ replaces then the $\mathbf{1}$-vector in Equation (3.8), yielding the NTF parameters $\Theta_{\mathbf{x}} \leftarrow \{\mathbf{W}_{\mathbf{x}}, \mathbf{H}_{\mathbf{x}}, \mathbf{Q}_{\mathbf{x}}\}$ which can be used for Wiener filtering with (2.36) to estimate the sources.

3.4 Summary

In this chapter, the reference algorithm for ISS was summarized and some aspects were evaluated: Regarding the time frequency transform, logarithmic frequency scaling was added as already used in [Spi12] for BSS. This scaling is achieved by conducting mel filtering on the spectral dimension of the STFT/MDCT spectrograms which leads to significant bit rate savings. The quantizer is adapted to yield lower bit rates. It is operated with different numbers of reconstruction values, the selection of the optimum parameter is done during the rate-quality optimization as discussed in Section 2.8.2. It was proposed to use A-law companding as summarized in Section 2.3.2. The corresponding parameters are summarized in Table 3.1.

In addition to that, a BSS algorithm, also based on NTF, was briefly summarized. It will be used in Chapter 5 as reference BSS algorithm.

Method	Parameter	Symbol and Value
STFT	Window size	$N_\mathrm{w} = 2^{12}$
	Hop size	$N_\mathrm{h} = 2^{11}$ (50 % overlap)
	Number of mel filters	$F = 500$
NTF	Cost function	$\beta = 1$ (KL divergence)
	Initialization	Complex SVD [BMR15]
	Number of components per source	$K/J \in \{1, 2, \ldots, 10\}$
Quantization	A-law companding factor	$A = 10$
	Number of reconstruction values	$N_\mathrm{q} \in \{2, 3, 4, 8, 16\}$

Table 3.1 Parameters chosen for evaluation.

4 Efficient Parameter Encoding

This chapter deals with an extension of the reference ISS encoder as summarized in Chapter 3. Context-based Adaptive Binary Arithmetic Coding (CABAC) as described in Section 2.4.1 is adapted to the task of encoding the quantized NTF parameters. Figure 4.1 depicts the proposed ISS encoder which is a modified version of the reference encoder shown in Figure 3.1. As indicated by the highlighted blocks, CABAC is proposed for encoding the quantization indices $\mathbf{G}_{\mathbf{W}_s}$ and $\mathbf{G}_{\mathbf{H}_s}$ corresponding to the NTF parameters $\mathbf{W}_s$ and $\mathbf{H}_s$. As mentioned in Section 3.1, the grouping matrix $\mathbf{Q}_s$ has only few elements compared to $\mathbf{W}_s$ and $\mathbf{H}_s$. Therefore, $\mathbf{Q}_s$ is still encoded with GZIP since it can be assumed that using CABAC instead will not significantly improve bit rate savings. In addition to that, GZIP is also used for coding the reconstruction values $\mathbf{c}_{\mathbf{W}_s}$ and $\mathbf{c}_{\mathbf{H}_s}$. The bit streams for all parameters are finally concatenated.

As a motivation for using CABAC, the typical structure of the NTF matrices or rather their quantized versions is summarized in the following. Figure 4.2 depicts the quantization indices for exemplary NTF parameters $\mathbf{W}_s$ and $\mathbf{H}_s$. Both matrices are strongly structured:

1. As already discussed, the NTF parameters modeling audio spectrograms are usually sparse. Index 1 corresponding to reconstruction value 0 is by far the most frequent value.

2. Long runs of 0 and other values are also common for each component k. This structure inspired the design of NTF constraints in the literature, e.g. sparseness or continuity [Vir07] as discussed in Section 2.2.2.2. Here, these properties are exploited to adapt CABAC for the usage in ISS.

Recall that CABAC is able to approach conditional entropy by exploiting local statistics of the data. The margin, up to which the conditional entropy is reached, is dependent on the used context models. In Section 4.1, statistics of the structured quantization indices discussed above are investigated in more detail. Based on these findings, suitable binarization schemes and novel context models are proposed in Section 4.2. Evaluations of different binarization

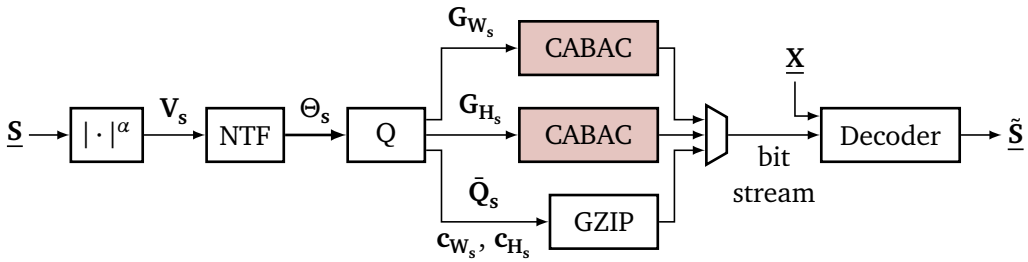
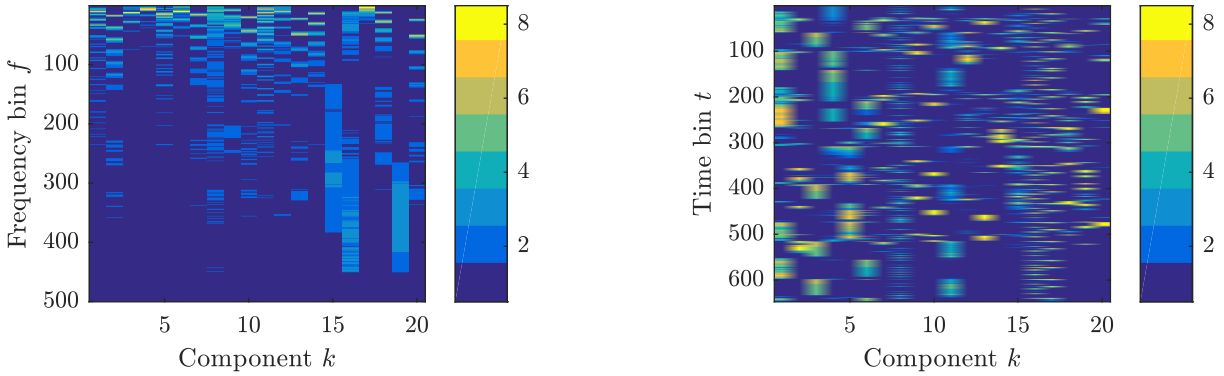


Figure 4.1 Proposed encoder using CABAC.



(a) Quantization indices $\mathbf{G}$ for $\mathbf{W}_\mathrm{s}$.

(b) Quantization indices for $\mathbf{H}_\mathrm{s}$.

Figure 4.2 Quantization indices for exemplary mix with $N_\mathrm{q} = 8$ and $K/J = 5$.

	$p\left(g_{d,k} = 1 \mid \mathscr{C}\right)$			$p\left(g_{d,k} > 1 \mid \mathscr{C}\right)$		
$\mathscr{C}$	–	$g_{d-1,k} = 1$	$g_{d-1,k} = 1,\ g_{d-2,k} = 1$	–	$g_{d,k} > 1$	$g_{d-1,k} > 1,\ g_{d-2,k} > 1$
$\mathbf{G}_{\mathbf{W}_\mathrm{s}}$	0.81	0.96	0.97	0.19	0.83	0.84
$\mathbf{G}_{\mathbf{H}_\mathrm{s}}$	0.60	0.87	0.89	0.40	0.81	0.83

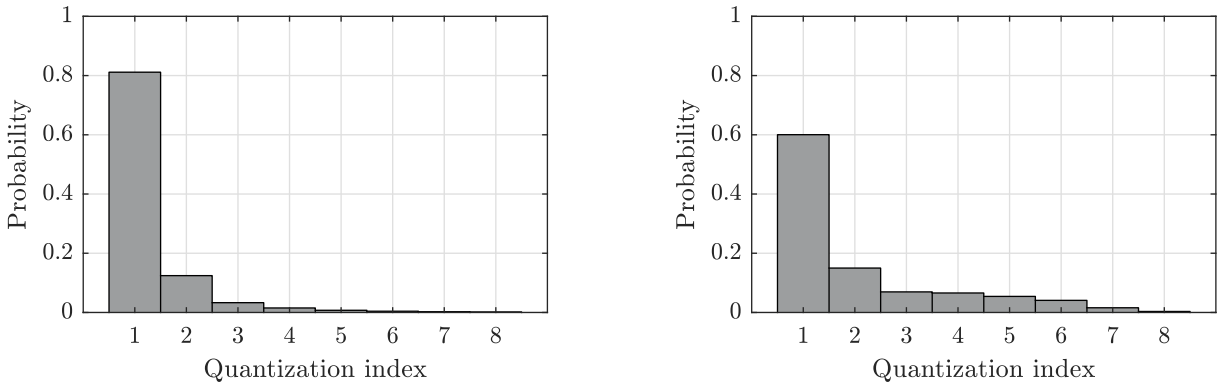
Table 4.1 Probabilities for two different sequences in the data: Either runs of zeros or runs of values greater than 0. Element $g_{d,k}$ being either $g_{d,k} = g_{\mathbf{W}_\mathrm{s},f,k}$ with $d = f$ or $g_{d,k} = g_{\mathbf{H}_\mathrm{s},t,k}$ with $d = t$.

and context model sets for CABAC as well as a comparison of CABAC to other encoding schemes are conducted in Section 4.3. Finally, all findings are summarized in Section 4.4. The usage of CABAC for NTF-based ISS was originally evaluated in [Gao17] and summarized in [Blä+18].

4.1 Preliminary Evaluation

In this section, local statistics of each component of the quantized NTF parameters to be coded with CABAC are investigated. Table 4.1 shows probabilities for two different sequences in the quantization indices $\mathbf{G}_{\mathbf{W}_\mathrm{s}}$ of $\mathbf{W}_\mathrm{s}$ and $\mathbf{G}_{\mathbf{H}_\mathrm{s}}$ of $\mathbf{H}_\mathrm{s}$, namely sequences of ones or sequences of values greater than one. Note that quantization index $g = 1$ maps to the lowest reconstruction value $c_1 = 0$. To measure probabilities of these sequences, all sources of test set $\mathscr{A}$ are factorized by the encoder NTF with $K/J = 5$. The resulting NTF parameters are quantized with $A = 10$ and $N_\mathrm{q} = 8$ fixed.

Probabilities of the aforementioned sequences of lengths 1 to 3 are taken into consideration and averaged over all components in $\mathbf{G}_{\mathbf{W}_\mathrm{s}}$ or $\mathbf{G}_{\mathbf{H}_\mathrm{s}}$. The corresponding probabilities are given with respect to the quantization index $g_{d,k}$ at position (d,k) with d being either $d = f$ for $\mathbf{G}_{\mathbf{W}_\mathrm{s}}$ or $d = t$ for $\mathbf{G}_{\mathbf{H}_\mathrm{s}}$. The probabilities of the one-valued sequences can be expressed as conditional probabilities with a certain condition $\mathscr{C}$, $p\left(g_{d,k} = 1 \mid \mathscr{C}\right)$. These conditional probabilities are obtained with either no condition, thus counting the occurrence of $g_{d,k}$ being one, the previous index being one with condition $\mathscr{C}$: $g_{d-1,k} = 1$, or the last two levels



(a) $\mathbf{G}_{\mathbf{W}_s}$ (b) $\mathbf{G}_{\mathbf{H}_s}$

Figure 4.3 Histograms of $\mathbf{G}_{\mathbf{W}_s}$ and $\mathbf{G}_{\mathbf{H}_s}$ for $N_\mathrm{q} = 8$ and $K/J = 5$.

equal to one $\mathscr{C}: \; g_{d-1,k} = 1, g_{d-2,k} = 1$. Runs of indices greater than one can be modeled in the same manner: $p\left(g_{d,k} > 1 \mid \mathscr{C}\right)$ with again either no condition, $\mathscr{C}: \; g_{d-1,k} > 1$ or $\mathscr{C}: \; g_{d-1,k} > 1, g_{d-2,k} > 1$. These values are given in Table 4.1 and lead to the following conclusions:

- For both $\mathbf{G}_{\mathbf{W}_s}$ and $\mathbf{G}_{\mathbf{H}_s}$, the probabilities $p\left(g_{d,k} = 1\right)$ are quite high which confirms that both matrices are sparse. The reconstruction values of the frequency basis $\mathbf{W}_s$ are more probable to be zero than the values for the time activations $\mathbf{H}_s$.

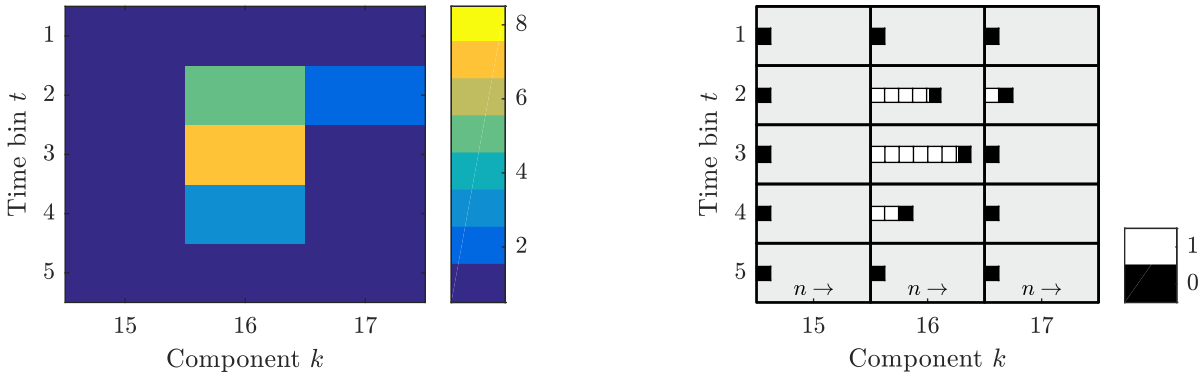
- For both parameters, the sequences defined above are also quite likely: Sequences of ones (mapping to zero-valued sequences of reconstruction values) are more probable than the sequences of values greater than one. Sequences of ones are more probable to appear in $\mathbf{G}_{\mathbf{W}_s}$ than in $\mathbf{G}_{\mathbf{H}_s}$ and sequences with values larger than one are equiprobable in both parameters.

- Comparing the probabilities of the same sequence but different lengths, it becomes clear that conditioning on $g_{d-2,k}$ in addition to $g_{d-1,k}$ gives similar probabilities as only taking $g_{d-1,k}$ into account. This shows that the Markov property for the group indices is fulfilled, meaning that current value $g_{d,k}$ is only dependent on the previous value $g_{d-1,k}$. Therefore, only the previously decoded value $g_{d-1,k}$, or its binarized version respectively, will be considered for the context design in the next section.

For sake of completeness, histograms of $\mathbf{G}_{\mathbf{W}_s}$ and $\mathbf{G}_{\mathbf{H}_s}$ are shown in Figure 4.3.

4.2 Context Design

In the following, the encoding process for $\mathbf{W}_s$ is shown. $\mathbf{H}_s$ is encoded in the same way and the bit streams for both parameters are concatenated. The same context design is used for $\mathbf{H}_s$ as well. Recall that quantization of $\mathbf{W}_s$ yields integer-valued grouping indices, stored in an $F \times K$ matrix $\mathbf{G}_{\mathbf{W}_s}$ which is abbreviated as $\mathbf{G}$ in the following with $1 \leq g_{f,k} \leq N_\mathrm{q}$. Each element of $\mathbf{G}$ has to be binarized as CABAC only takes binary sources.



(a) Quantization indices $\mathbf{G}_{\mathbf{H}_s}$ of $\mathbf{H}_s$ as given in Figure 4.2b for $1 \leq t \leq 5$ and $15 \leq k \leq 17$.

(b) Corresponding bin-strings $\{\mathbf{b}^{t,k}\}$ after TU coding with elements $b_n^{t,k}$.

Figure 4.4 Binarization of quantization indices $\mathbf{G}_{\mathbf{H}_s}$ with TU coding.

For binarization of the integer-valued grouping indices $\mathbf{G}$, the information about the position of each element $g_{f,k}$ in $\mathbf{G}$ is kept by adding the exponent (f,k) to each bin-string, composed of bins $b_n^{f,k}$. Thus, (2.29) in Section 2.4.1.1 becomes

$$\mathbf{b}^{f,k} = C\left(g_{f,k}\right) = \left(b_1^{f,k}, \ldots, b_n^{f,k}, \ldots, b_{N_C}^{f,k}\right)^\top \tag{4.1}$$

with either truncated unary or exponential Golomb codes $C(\cdot)$ of variable length N_C (refer to Section 2.4.1.1). Figure 4.4 illustrates the arrangement of bin-strings for a subset of quantization indices of $\mathbf{H}_s$ as given originally in Figure 4.2.

In the following, the proposed context models are presented, each modeling a particular structure in $\mathbf{G}$. The decoder can choose a specific context model for the current to-be-decoded bin $b_n^{f,k}$ given the following conditions:

- The value of previously decoded data. Two options are investigated in this thesis, namely:
  - The value of bin $b_n^{f-1,k}$ at the same position n within the previously decoded bin-string $\mathbf{b}^{f-1,k}$ in the same component (column) k.
  - The value of the previously decoded integer-valued symbol $g_{f-1,k}$ in the same component k.
- The position n of $b_n^{f,k}$ within the current bin-string.
- A combination of the previously decoded values and bin position n.

As previously mentioned, two different context model sets will be proposed. For both sets, the context model for the current to-be-coded bin is selected based on previously coded data: In the first approach, the context model for the current bin is selected depending on the previously coded *bin* $b_n^{f-1,k}$ at same position n in the previous bin-string $\mathbf{b}^{f-1,k}$. This procedure is proposed in Section 4.2.1. In the second approach, the context model is chosen based on the previously coded *integer value* $g_{f-1,k}$ instead and is further discussed in Section 4.2.2.

Conditions	$n \leq N_{\mathrm{LBP}}$		$n > N_{\mathrm{LBP}}$
	–	$b_n^{f-1,k} = v,\ v \in \{0,1\}$	–
Selection	$\mathrm{ctx}_{n,\mathrm{na}}$	$\mathrm{ctx}_{n,\mathrm{up}v}$	$\mathrm{ctx}_{\mathrm{rst}}$
Init.	$p\left(b_n^{f,k} = 0\right)$	$p\left(b_n^{f,k} = 0 \mid b_n^{f-1,k} = v\right)$	$p\left(b_n^{f,k} = 0\right)$

Table 4.2 Bin-value based context model selection and initialization for bin $b_n^{f,k}$ at position n of bin-string $\mathbf{b}^{f,k}$ with $n \leq N_{\mathrm{LBP}}$. For $n > N_{\mathrm{LBP}}$, a common context model $\mathrm{ctx}_{\mathrm{rst}}$ is used.

To distinguish the two proposed context model sets, the sets are denoted as bin-value or integer-value based context models, respectively.

Note that internally, CABAC addresses each context model with an integer-valued *context id*. For sake of completeness, the context ids for the two different context model sets are summarized in Appendix C.1.

4.2.1 Context Modeling Based on Bin Values

In this section, bin-value based context models are proposed. Note that the context model selection process is dependent on bin position n of the to-be-coded bin $b_n^{f,k}$. To limit the total number of context models, different context models are only used for bin positions up to a number N_{LBP} indicating the last bin position for context modeling. All bins at position $n > N_{\mathrm{LBP}}$ are modeled with a common "rest" context model $\mathrm{ctx}_{\mathrm{rst}}$. For bins at position $n \leq N_{\mathrm{LBP}}$, the following context models may be selected:

- If bin $b_n^{f-1,k}$ in the previously coded bin-string $\mathbf{b}^{f-1,k}$ is available and has value $v \in \{0,1\}$, the *conditional* context model $\mathrm{ctx}_{n,\mathrm{up}v}$ is chosen. This context model corresponds to the conditional probability $p\left(b_n^{f,k} \mid b_n^{f-1,k} = v\right)$.
- If the length N_{C} of the previously decoded bin-string is smaller than n, which means that $b_n^{f-1,k}$ is not available, the *default* context model $\mathrm{ctx}_{n,\mathrm{na}}$ is selected. In this case, the probability $p\left(b_n^{f,k} \mid b_n^{f-1,k}\ \text{n.a.}\right)$ is modeled. It is possible to deactivate the conditional context model $\mathrm{ctx}_{n,\mathrm{up}v}$. In this case, $\mathrm{ctx}_{n,\mathrm{na}}$ operates as a default context model, modeling all bins at position n which are not subject to conditional context modeling.

Table 4.2 gives an overview of these context models which are applied for TU codes and the *prefix* of EG codes. For the suffix of EG codes however, a second default context model $\mathrm{ctx}_{\mathrm{suf},n,\mathrm{na}}$ is used modeling the global probability $p\left(b_n^{f,k}\right)$ of suffix bins $b_n^{f,k}$ with $n > N_{\mathrm{pre}}$ and the number of prefix bins N_{pre} given in Equation (2.31). This is motivated by the fact that suffix bins are representing the least significant bits and hence do not show strong conditional probabilities within the same bin-string. Such bins could also be coded in bypass mode for higher throughput, assuming an equiprobable distribution. The number of context models for the suffix bins are limited as well. Bins at position $n - N_{\mathrm{pre}} > N_{\mathrm{LBP}}$ are modeled with $\mathrm{ctx}_{\mathrm{suf,rst,na}}$. Table 4.2 also shows the corresponding probability values for initializing the corresponding context models. These values must be transmitted to the decoder.

In total, the bin-value based context model set comprises $3N_{\mathrm{LBP}} + 1$ context models for TU binarization, namely N_{LBP} context models for each $\mathrm{ctx}_{n,\mathrm{up0}}$, $\mathrm{ctx}_{n,\mathrm{up1}}$ and $\mathrm{ctx}_{n,\mathrm{na}}$ and one context

t	$g_{t,16}$	$C\left(g_{t,16}\right)$	Selected context models for bin $b_n^{t,16}$			
			$n=1$	$n=2$	$n=3$	$n>3$
1	1	0	$\text{ctx}_{1,\text{na}}$	–	–	–
2	5	11110	$\text{ctx}_{1,\text{up0}}$	$\text{ctx}_{2,\text{na}}$	$\text{ctx}_{3,\text{na}}$	ctx_{rst}
3	7	1111110	$\text{ctx}_{1,\text{up1}}$	$\text{ctx}_{2,\text{up1}}$	$\text{ctx}_{3,\text{up1}}$	ctx_{rst}
4	3	110	$\text{ctx}_{1,\text{up1}}$	$\text{ctx}_{2,\text{up1}}$	$\text{ctx}_{3,\text{up1}}$	–
5	1	0	$\text{ctx}_{1,\text{up1}}$	–	–	–

Table 4.3 Bin-value based context model selection for each bin $b_n^{t,k}$ of bin-strings $\mathbf{b}^{t,k} = C\left(g_{t,k}\right)$ given the 16th component ($k = 16$) of exemplary quantization indices of $\mathbf{H_s}$ shown in Fig. 4.2 with $1 \leq t \leq 5$, and $N_{\text{LBP}} = 3$.

Binarization	Context Model	
	$\text{ctx}_{n,\text{up0}}$	$\text{ctx}_{n,\text{up1}}$
TU	$g_{f-1,k} = n$	$g_{f-1,k} \geq n+1$
EG0	$2^{n-1} \leq g_{f-1,k} \leq 2^n - 1$	$g_{f-1,k} \geq 2^n$
EGl	$2^l\left(2^{n-1}-1\right)+1 \leq g_{f-1,k} \leq 2^l\left(2^n-1\right)$	$g_{f-1,k} \geq 2^l\left(2^n-1\right)+1$

Table 4.4 Integer-level interpretation of context models defined on bin-level.

model for the rest, ctx_{rst}. For EG binarization, this number increases to $4N_{\text{LBP}}+2$, adding N_{LBP} context models for $\text{ctx}_{\text{suf},n,\text{na}}$ and one for $\text{ctx}_{\text{suf,rst,na}}$. Table 4.3 shows an exemplary context model selection for bin-strings belonging to component $k = 16$ as shown in Figure 4.4b with TU binarization.

Note that other conditional context designs, for example context models describing the behavior across components (between $b_n^{f,k}$ and $b_n^{f,k-1}$) or across more than one preceding bin-string ($b_n^{f,k}$, $b_n^{f-1,k}$ and $b_n^{f-2,k}$), were evaluated in [Gao17] and did not improve the performance significantly compared to the context models shown in Table 4.2.

An interpretation with respect to the integer-valued symbols is given in Section 4.2.1.1 and a preliminary evaluation comparing the proposed context models is conducted in Sections 4.2.1.2 and 4.2.1.3. Note that these are qualitative evaluations. Quantitative results are shown in Section 4.3.

4.2.1.1 Interpretation of Bin-value Based Context Models

The conditional bin-value based context models are evaluated in the following. An interpretation of the bin-value based context design on the integer-valued input symbols is provided.

Since the proposed context models are selected for coding the data after binarization, an interpretation of the context models on the integer-valued symbols is given in the following. A more detailed explanation is given in Appendix C.2. In Table 4.4, these findings are summarized. Although TU yields longer code words for larger values $g_{f,k}$, the context models

Bin position	Previous bin value $b_n^{f-1,k}$	Previous integer value $g_{f-1,k}$	Selected context model for bin $b_n^{f,k}$	
			"up0"	"up1"
$n=1$	0	1	$\text{ctx}_{1,\text{up0}}$	$\text{ctx}_{1,\text{na}}$
	1	≥ 2	$\text{ctx}_{1,\text{na}}$	$\text{ctx}_{1,\text{up1}}$
$n=2$	not available	1	$\text{ctx}_{2,\text{na}}$	$\text{ctx}_{2,\text{na}}$
	0	2	$\text{ctx}_{2,\text{up0}}$	$\text{ctx}_{2,\text{na}}$
	1	≥ 3	$\text{ctx}_{2,\text{na}}$	$\text{ctx}_{2,\text{up1}}$

Table 4.5 All possible conditions for bin-value based context selection for to-be-coded bin $b_n^{f,k}$ with $n \in \{1,2\}$ with $N_{\text{LBP}} = 2$. For bins with $n > N_{\text{LBP}} = 2$, ctx_{rst} is chosen.

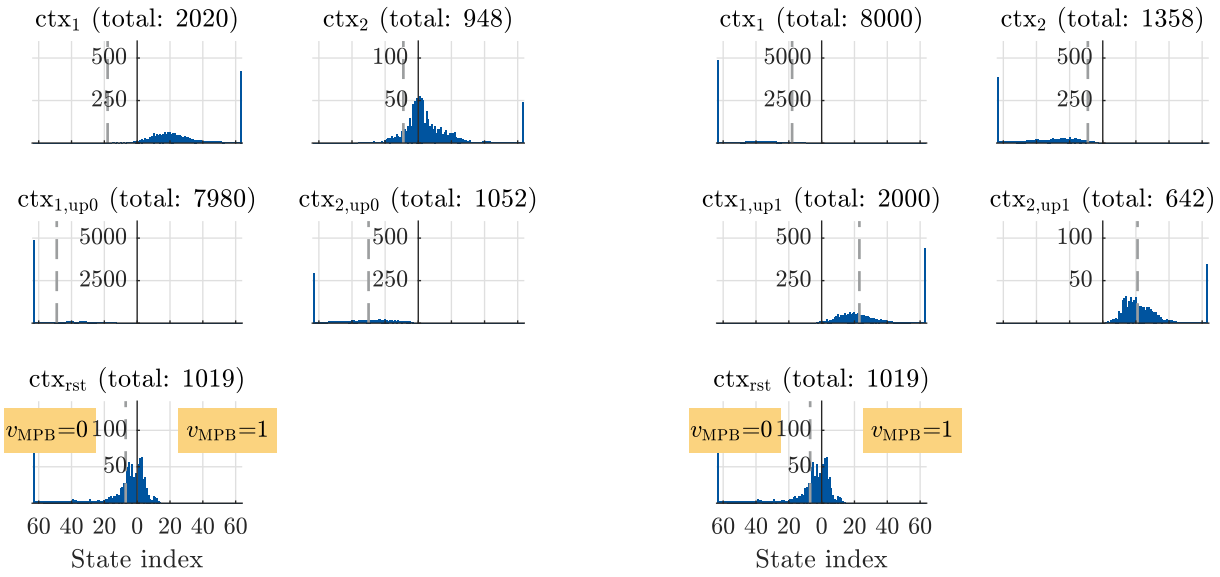
$\text{ctx}_{n,\text{up}v}$ are modeling sequences of integer values with finer granularity compared to EG binarizations: Context model $\text{ctx}_{n,\text{up0}}$ models runs of identical values for TU binarization. When using EG, $\text{ctx}_{n,\text{up0}}$ models runs of values lying in an interval as given Table 4.4. $\text{ctx}_{n,\text{up1}}$ models sequences of values greater or equal than a threshold for both TU and EG binarizations, although the different thresholds depending on n are more precise for TU than for EG.

4.2.1.2 Comparison of Bin-value Based Context Models

In the following, the conditional context models $\text{ctx}_{n,\text{up0}}$ and $\text{ctx}_{n,\text{up1}}$ for TU binarization are compared in more detail assuming $N_{\text{LBP}} = 2$. For context modeling, either $\text{ctx}_{n,\text{up0}}$ ("up0") or $\text{ctx}_{n,\text{up1}}$ ("up1") are active in addition to default context model $\text{ctx}_{n,\text{na}}$ and rest context model ctx_{rst} which are active in both cases. Table 4.5 shows the corresponding general context selection process. Bin position n, all cases for possible states of previous bin $b_n^{f-1,k}$ and the corresponding range of integer values $g_{f-1,k}$ are shown in the first three columns. The context selection depending on these previous values is shown in the last two columns for either $\text{ctx}_{n,\text{up0}}$ ("up0") or $\text{ctx}_{n,\text{up1}}$ ("up1") active.

- It becomes clear that $\text{ctx}_{n,\text{na}}$ is used for representing the *non*-active conditional context model: When $\text{ctx}_{n,\text{up0}}$ is active, $\text{ctx}_{n,\text{na}}$ will be used if $b_n^{f-1,k} = 1$. In contrast to this, it is used for activated $\text{ctx}_{n,\text{up1}}$ for bins where $b_n^{f-1,k} = 0$. In both cases, $\text{ctx}_{n,\text{na}}$ is also used if the previous bin $b_n^{f-1,k}$ is not available (in this example for $n = 2$).

- In the case when $\text{ctx}_{n,\text{up0}}$ is active, the conditions for selecting $\text{ctx}_{2,\text{na}}$ are contradicting: This context model is chosen if the previous bin is either not available or has value 1 which corresponds to previous integer-value being either $g_{f,k-1} = 1$ *or* $g_{f,k-1} \geq 3$. Recall that the current to-be-coded bin is at position $n = 2$ which means that the current integer value is $g_{f,k} \geq 2$. Therefore, $\text{ctx}_{2,\text{na}}$ models transitions from $g_{f,k-1} = 1$ (corresponding to reconstruction value zero) to $g_{f,k} \geq 2$ and transitions from $g_{f,k-1} \geq 3$ to $g_{f,k} \geq 2$. In other words, $\text{ctx}_{2,\text{na}}$ models rising and falling quantization indices at the same time.



(a) Histograms for context model $\text{ctx}_{n,\text{up0}}$ activated ("up0").

(b) Histograms for context model $\text{ctx}_{n,\text{up1}}$ activated ("up1").

Figure 4.5 State index histograms for exemplary quantization indices in Figure 4.2a for $N_{\text{LBP}} = 2$ and either conditional context model $\text{ctx}_{n,\text{up0}}$ or $\text{ctx}_{n,\text{up1}}$ activated in addition to $\text{ctx}_{n,\text{na}}$ and ctx_{rst} which are always active here.

- For the other conditional context model $\text{ctx}_{n,\text{up1}}$, the conditions for selecting $\text{ctx}_{2,\text{na}}$ are more useful. It models transitions from $g_{f-1,k} \in \{1,2\}$ to $g_{f,k} \geq 2$ which translates to either a sequence of values equal two or increasing values.

This exemplary context selection process makes clear that the default context model $\text{ctx}_{n,\text{na}}$ is selected in contradicting cases when used in combination with the conditional context model $\text{ctx}_{n,\text{up0}}$. Recall that $\text{ctx}_{n,\text{up0}}$ is selected for $b_n^{f-1,k} = 0$ which corresponds to the previous integer symbol being $g_{f-1,k} = n$ (refer to Table 4.4). $\text{ctx}_{n,\text{na}}$ is selected in all other, contradicting cases which correspond to either $g_{f-1,k} < n$ or $g_{f-1,k} > n$.

4.2.1.3 Preliminary Evaluation of Bin-value Based Context Models

In Section 4.2.1.2, the two bin-value based context models are compared. In the following, it is evaluated if the contradicting conditions for selecting $\text{ctx}_{2,\text{na}}$ for "cond0" have impact on the coding performance. CABAC is used to encode exemplary **G** as shown in Figure 4.2a with the same setup as before, namely $N_{\text{LBP}} = 2$ and TU binarization. To provide a more detailed evaluation, the state machines of each context model are considered here: As already explained in Section 2.4.1.2, CABAC uses state machines to model the probability $p_{\text{LPB}} = p\,(v_{\text{LPB}} \mid \text{ctx})$, the probability of the least probable bin v_{LPB} for each context model ctx. The state with index $i_{p_{\text{LPB}}} = 63$ (either for $v_{\text{MPB}} = 0$ or for $v_{\text{MPB}} = 1$) denotes the state with the lowest value p_{LPB}. If the state machine frequently remains in this state, it can be concluded that the occurrence of v_{MPB} is very likely. In this case, CABAC is able to model the underlying data quite well, limited by the precision of the discrete probability values. In contrast to that, the value v_{MPB} may flip for a context model if $p_{\text{LPB}} = 0.5$, even multiple times. This means

Conditions	$n \leq N_{\mathrm{LBP}}$	$n > N_{\mathrm{LBP}}$
	$g_{f-1,k} = v,\ v \in \{1,\ldots,N_{\mathrm{q}}\}$	–
Selection	$\mathrm{ctx}_{n,\mathrm{iup}v}$	$\mathrm{ctx}_{\mathrm{rst}}$
Init.	$p\left(b_n^{f,k} = 0 \mid g_{f-1,k} = v\right)$	$p\left(b_n^{f,k} = 0\right)$

Table 4.6 Integer-value based context model selection and initialization for bin $b_n^{f,k}$ at position n of bin-string $\mathbf{b}^{f,k}$ with $n \leq N_{\mathrm{LBP}}$. For $n > N_{\mathrm{LBP}}$, a common context model $\mathrm{ctx}_{\mathrm{rst}}$ is used.

that the context model is not able to adapt efficiently which automatically results in more bits written out.

Figure 4.5 shows histograms of state indices for either $\mathrm{ctx}_{n,\mathrm{up0}}$ (“up0”) and $\mathrm{ctx}_{n,\mathrm{up1}}$ (“up1”) active. The initial state is marked with a dashed gray line as well.

- Figure 4.5a shows the histograms for “up0”. As expected, v_{MPB} toggles between ’0’ and ’1’ for $\mathrm{ctx}_{2,\mathrm{na}}$ quite frequently which means that the state machine is not able to adapt well. $\mathrm{ctx}_{2,\mathrm{na}}$ is chosen in contradicting cases as mentioned before. For context models $\mathrm{ctx}_{1,\mathrm{na}}$, $\mathrm{ctx}_{1,\mathrm{up0}}$ and $\mathrm{ctx}_{2,\mathrm{up0}}$, the state machines are remaining frequently in state $i_{p_{\mathrm{LPB}}} = 63$. This means that the state machine adapts well and CABAC works efficiently if these context models are chosen.

- The histograms of state indices for “up1” are depicted in Figure 4.5b. v_{MPB} is not toggling in all state machines but the one for $\mathrm{ctx}_{\mathrm{rst}}$ which is discussed below. Comparing the performance of “up1” and “up0”, using the “up1” configuration yields a BD-BR reduction of 7% for this example compared to “up0”. This shows the disadvantage of the sub-optimal usage of $\mathrm{ctx}_{2,\mathrm{na}}$ for “up0” as discussed above.

- For $\mathrm{ctx}_{\mathrm{rst}}$, v_{MPB} is toggling as well. Recall that $\mathrm{ctx}_{\mathrm{rst}}$ is selected for coding all bins at positions $n > 2$. These bins are more often equal to ’0’ than ’1’, as the state machine is remaining quite often in the corresponding state. This can be explained by the fact that smaller quantization indices are more probable than larger indices, refer to Figure 4.3. Recall that only $N_{\mathrm{LBP}} = 2$ bins are modeled with (conditional) contexts here. By increasing N_{LBP}, this effect can be diminished.

4.2.2 Context Modeling Based on Integer Values

In Section 4.2.1, the context model for the to-be-coded bin $b_n^{f,k}$ was chosen based on the binary value of the previously coded bin $b_n^{f-1,k}$. In this section, it is proposed to choose the context model based on the value of the previously coded integer value $1 \leq g_{f-1,k} \leq N_{\mathrm{q}}$ instead. For each bin $b_n^{f,k}$, the previously coded symbol $g_{f-1,k}$ is used for choosing the appropriate context model. As already done for the bin-level context model design, the number of bins for which context models are used is limited to N_{LBP}. All bins at position $n > N_{\mathrm{LBP}}$ are modeled with common context model $\mathrm{ctx}_{\mathrm{rst}}$. For all other bins at bin position $n \leq N_{\mathrm{LBP}}$, the value of the previously coded *integer symbol* $g_{f-1,k}$, abbreviated with $v \in \left[1, N_{\mathrm{q}}\right]$, is chosen to select the conditional context model $\mathrm{ctx}_{n,\mathrm{iup}v}$. This procedure models the conditional probability $p\left(b_n^{f,k} \mid g_{f-1,k} = v\right)$.

t	$g_{t,16}$	$C\left(g_{t,16}\right)$	Selected context models for bin $b_n^{t,16}$			
			$n=1$	$n=2$	$n=3$	$n>3$
1	1	0	$\text{ctx}_{1,\text{iup1}}$	–	–	–
2	5	11110	$\text{ctx}_{1,\text{iup1}}$	$\text{ctx}_{2,\text{iup1}}$	$\text{ctx}_{3,\text{iup1}}$	ctx_{rst}
3	7	1111110	$\text{ctx}_{1,\text{iup5}}$	$\text{ctx}_{2,\text{iup5}}$	$\text{ctx}_{3,\text{iup5}}$	ctx_{rst}
4	3	110	$\text{ctx}_{1,\text{iup7}}$	$\text{ctx}_{2,\text{iup7}}$	$\text{ctx}_{3,\text{iup7}}$	–
5	1	0	$\text{ctx}_{1,\text{iup3}}$	–	–	–

Table 4.7 Integer-value based context model selection for each bin $b_n^{t,k}$ of bin-strings $\mathbf{b}^{t,k} = C\left(g_{t,k}\right)$ given for the same exemplary data already chosen for Table 4.3.

Note that with this general approach, no default context model ($\text{ctx}_{n,\text{na}}$ as defined in Section 4.2.1 for the bin-level context models) is needed. For $f = 1$ at the beginning of a component, $g_{f-1,k}$ is not available. It is simply assumed that $v = 1$ as it corresponds to the zero-valued reconstruction value. The conditional context model $\text{ctx}_{n,\text{iup}v}$ is only applicable for TU codes and the prefix of EG codes. As already proposed for the bin-level context modeling in Section 4.2.1, the context models $\text{ctx}_{\text{suf},n,\text{na}}$ for the first N_{LBP} suffix bins and $\text{ctx}_{\text{suf,rst,na}}$ for the rest are used for modeling the suffix of EG codes.

Compared to the bin-level context design proposed in Section 4.2.1, the integer-value based design is more complex as it comprises in total $N_{\text{q}}N_{\text{LBP}} + 1$ different context models for TU binarization. For each combination of bin $n \leq N_{\text{LBP}}$ and integer value $v \in \left[1, N_{\text{q}}\right]$ a context model exists. Recall that for the bin-level design only $3N_{\text{LBP}} + 1$ context models are needed. Table 4.7 shows an exemplary context model selection process for the same input data as already used in Table 4.3, this time for the integer-value based context models.

4.3 Experimental Results

First, the different binarization methods and context designs are evaluated on test set $\mathscr{A}$ (refer to Section A.1). For the next experiments, the choices of the binarization method and the context models is fixed. CABAC is then compared to reference coding methods on test set $\mathscr{B}$ (refer to Section A.2). The number of NTF components per source is set to $K/J \in \{1, \ldots 20\}$ and the NTF minimizes the Kullback-Leibler divergence ($\beta = 1$) throughout all experiments. The resulting parameters are companded with $A = 10$ and quantized with $N_{\text{q}} \in \{2, 3, 4, 8, 16\}$ reconstruction values.

As a quality measure, BD-BR as summarized in Section 2.8.2 is calculated with GZIP as baseline. Note that in this chapter, GZIP is used for encoding $\mathbf{G}_{\mathbf{W}_{\text{s}}}$ and $\mathbf{G}_{\mathbf{H}_{\text{s}}}$ *separately* as CABAC and all other reference methods encode the parameters separately as well. In all other chapters, GZIP is used for jointly encoding the parameters. Results with this variant of GZIP as baseline are given in Appendix C.3. The joint GZIP variant yields a BD-BR reduction of −12.38 % compared to encoding the parameters separately with GZIP.

	TU	EG0	EG1
$N_{\mathrm{LBP}} = 0$	−10.71	−9.03	−16.84
$N_{\mathrm{LBP}} = 1$	−19.97	−19.71	−18.38
$N_{\mathrm{LBP}} = 5$	−21.44	−20.46	−18.57
$N_{\mathrm{LBP}} = 10$	**−21.59**	–	–

Table 4.8 BD-BR in % with respect to GZIP. All conditional context models were deactivated. For $N_{\mathrm{LBP}} = 0$, only $\mathrm{ctx}_{\mathrm{rst}}$ is active.

Method	GBAC	CABAC			
Cond. Ctx.	–	–	$\mathrm{ctx}_{n,\mathrm{up0}}$	$\mathrm{ctx}_{n,\mathrm{up1}}$	$\mathrm{ctx}_{n,\mathrm{up0}}$, $\mathrm{ctx}_{n,\mathrm{up1}}$
BD-BR, %	−10.71	−21.44	−31.81	**−33.23**	−32.81
Mean saving, %	−11.76	−18.34	−28.01	**−29.45**	−29.31
Std. saving, %	10.18	8.57	7.32	6.57	6.49

Table 4.9 Results for bin-level context models. BD-BR with respect to GZIP for GBAC ($N_{\mathrm{LBP}} = 0$) and CABAC ($N_{\mathrm{LBP}} = 5$) for test set $\mathscr{A}$.

Binarization

The first step of CABAC is binarization. To evaluate the chosen binarization methods TU, EG0 and EG1 coding, all conditional context models are deactivated. Said binarization methods are evaluated for different values of N_{LBP}, indicating the last bin position which is modeled with $\mathrm{ctx}_{n,\mathrm{na}}$. For the highest number of quantization centroids, $N_{\mathrm{q}} = 16$, the highest code length is equal to 15. The prefixes of EG0 and EG1 codes are smaller for this case, namely five for EG0 and four for EG1. The number of bins modeled individually is set to $N_{\mathrm{LBP}} \in \{0, 1, 5, 10\}$. Note that $N_{\mathrm{LBP}} = 10$ is only useful for TU.

Table 4.8 shows rate reductions with GZIP as reference for different values of N_{LBP}. With only the rest context model activated (with $N_{\mathrm{LBP}} = 0$), EG1 outperforms TU and EG0. EG0 yields better results when activating $\mathrm{ctx}_{1,\mathrm{na}}$ additionally with $N_{\mathrm{LBP}} = 1$. But with the first $N_{\mathrm{LBP}} = 5$ bins modeled with $\mathrm{ctx}_{n,\mathrm{na}}$, TU gives better results than EG0 and outperforms EG1. Setting $N_{\mathrm{LBP}} = 10$ does not increase the performance noticeably. In the following, TU is chosen as binarization method and the number of modeled bins is set to $N_{\mathrm{LBP}} = 5$.

Conditional bin-value based context models

In the following, the performance of CABAC is evaluated with the bin-level conditional context models $\mathrm{ctx}_{n,\mathrm{up0}}$ and $\mathrm{ctx}_{n,\mathrm{up1}}$ are activated in addition to ctx_n. As a comparison, the performance of the BAC with only one *global* activated context model $\mathrm{ctx}_{\mathrm{rst}}$ is evaluated as well and abbreviated with GBAC. Table 4.9 shows the corresponding rate savings:

- Activating only $\mathrm{ctx}_{n,\mathrm{na}}$ yields a noticeable decrease of rate compared to GBAC where only $\mathrm{ctx}_{\mathrm{rst}}$ is active from −10.71 % to −21.44 %. Modeling already the probability

Method	CABAC	
Cond. Ctx.	Bin-level $\text{ctx}_{n,\text{up1}}$	Integer-level $\text{ctx}_{n,\text{iup}v}$
BD-BR, %	−33.23	**−33.68**
Mean saving, %	−29.45	**−30.27**
Std. saving, %	6.57	6.24

Table 4.10 Results for integer-level context models. BD-BR with respect to GZIP evaluated on test set $\mathscr{A}$.

$p\left(b_n^{f,k} \mid b_n^{f-1,k} \text{ n.a.}\right)$ of the bins at position n with $\text{ctx}_{n,\text{na}}$ increases the performance compared to one globally activated context.

- Regarding $\text{ctx}_{n,\text{up}v}$, $\text{ctx}_{n,\text{up1}}$ gives the highest rate decrease of −33.23 % compared to GZIP, closely followed by $\text{ctx}_{n,\text{up0}}$ at −31.81 %. Compared to the performance of only activating $\text{ctx}_{n,\text{na}}$, the rate is further decreased when modeling conditional statistics with $\text{ctx}_{n,\text{up}v}$. The worse performance of $\text{ctx}_{n,\text{up0}}$ is explained in Section 4.2.1.2. $\text{ctx}_{n,\text{up0}}$ is selected in contradicting cases whereas $\text{ctx}_{n,\text{up1}}$ is modeling the underlying structure more efficiently.

- Activating both $\text{ctx}_{n,\text{up0}}$ and $\text{ctx}_{n,\text{up1}}$ does not decrease the rate any further compared to $\text{ctx}_{n,\text{up1}}$. This can also be explained by the findings shown in Section 4.2.1.2.

Conditional integer-value based context models

Table 4.10 shows results comparing the bin-level context models proposed in Section 4.2.1 to the integer-value based context models summarized in Section 4.2.2. The latter context design uses context models for each combination of bin position n and previously coded integer value v. The integer-value based context design yields slightly better results, decreasing BD-BR by only 0.5%. Compared to the increased number of total context models, the gain is not very large. Therefore, in the next section, the bin-value based context model $\text{ctx}_{n,\text{up1}}$ is preferred over $\text{ctx}_{n,\text{iup}v}$.

Comparison against reference methods

In the previous sections, the binarization method and context model settings for CABAC were chosen. In this section, CABAC is compared to reference methods which are briefly discussed in Section 2.4:

- GZIP which was used in the reference ISS method as discussed in Section 3.1.

- Arithmetic Coding (AC) as described in Section 2.4. AC was used in e.g. CISS [Oze+13].

- Huffman Coding (HC) which was e.g. used in [NV10].

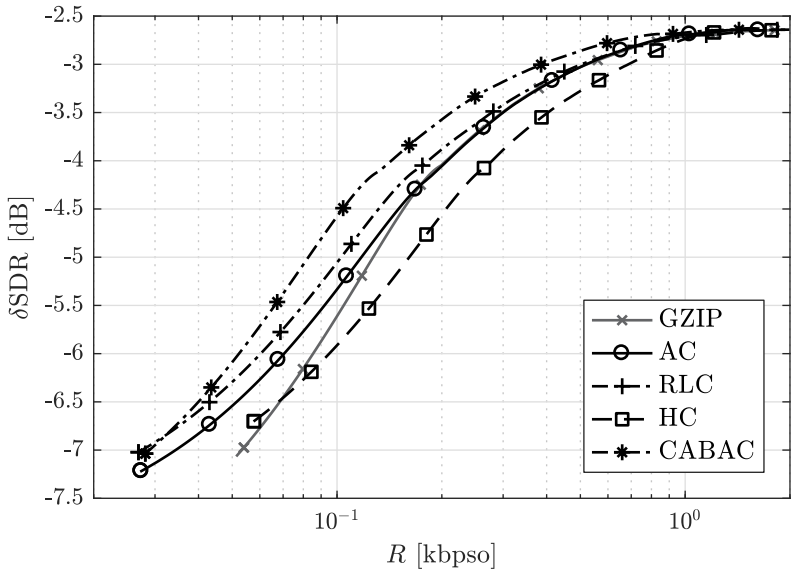


Figure 4.6 Comparison of proposed CABAC scheme with reference coding methods on test set $\mathscr{B}$.

- Run-length Coding (RLC) as discussed in Section 2.4.2 which encodes sequences of the same values. RLC is also approaching conditional entropy [Ohm15] if the Markov property of the input data is fulfilled. Here, the run-lengths and their corresponding values are each encoded with EG0. RLC was used in the NMF-based ISS method of [RBW16] for encoding the NTF parameters which were quantized with only $N_\mathrm{q} = 2$ reconstruction values.

These coding methods are used in the following for coding the quantization indices $\mathbf{G}_{\mathbf{W}_\mathbf{s}}$ and $\mathbf{G}_{\mathbf{H}_\mathbf{s}}$. The grouping matrix $\bar{\mathbf{Q}}_\mathbf{s}$ is encoded independently with GZIP.

The optimum CABAC settings were already determined while evaluating CABAC on test set $\mathscr{A}$ in the previous experiments. To give results for an independent test set, all methods and CABAC with the same optimum settings are evaluated on test set $\mathscr{B}$. Figure 4.6 shows the corresponding results:

- CABAC (−✱−·) outperforms all reference methods noticeably. The highest rate reductions are yielded in the medium rate range. At lower rates, the grouping matrix $\bar{\mathbf{Q}}_\mathbf{s}$, which is always quantized with high precision, has the biggest portion of the bit rate. The total rate reduction in this case with respect to GZIP (—×—) is −34.44 %.

- RLC (−+−·) is the method with the second highest gain by around −20.37 %. Since the data considered here has the Markov property, as shown experimentally in Section 4.1, RLC is able to approach conditional entropy as well.

- AC (—○—) outperforms GZIP at lower rates and reaches the same performance at higher bit rates.

- HC (—□—) yields worse results than GZIP and needs about 18.37 % more rate. This can be explained by the fact that HC is the only method not exploiting any conditional statistics in the data or using fractional number of bits per symbol as discussed in Section 2.4.

CABAC and RLC are the two methods with the highest bit rate savings. RLC only approaches conditional entropy for data having the Markov property. The proposed context model design for CABAC is also based on the Markov property of the NTF parameters. In general, CABAC is able to model more complex statistics. The implementation of RLC however is less complex than CABAC.

4.4 Summary

In this chapter, it was proposed to use CABAC for efficiently coding the quantized NTF matrices. After a short evaluation in Section 4.1 where it was shown that the quantization indices have Markov property, two different sets of context models were proposed in Section 4.2: The more simplistic set of context models describes the data on a bin-level. The context model selection is based on the bin's value in the previously coded *bin-string* at the same bin position as the to-be-coded bin. The other design uses the previously coded *integer* value for selecting a particular context model. This approach is more complex as it needs context models for each combination of bin index and integer value. Contrarily, the simplistic model offers only two choices per bin index since the value of the previously coded bin is binary.

In Section 4.3, several experiments were conducted. In the first part, the proposed CABAC context models were evaluated. The integer-value based context design yielded only slightly better rate reductions than the bin-value based design. Therefore, the latter context model set was chosen for comparing CABAC to other entropy coding schemes in the second part. Evaluated on a large test set, it became clear that CABAC outperformed all other reference methods. RLC, modeling sequences of the same value in the data, was the method with the second highest bit rate savings. Both CABAC and RLC are approaching conditional entropy which is reflected in the results. It can be concluded that adapting the coding method to the structure of the NTF enables a more efficient compression.

5 Parameter Re-estimation at Decoder

This chapter deals with an extension of the reference decoder for lower and very low bit rates. A second NTF prior to the Wiener filter with the mixture instead of the sources as observation may be used to

1. refine coarsely quantized parameters,

2. estimate parameters which were not transmitted,

3. operate blindly without any parameter transmission.

The encoder is able to simulate the decoding process and chooses one of the scenarios listed above. However, the parameter estimation in the encoder is not modified in this scenario, the sources are still factorized with the encoder NTF as summarized in Section 3.1. The main modification of the decoder is to use an algorithm originally designed for blind source separation in [Spi12] instead of solely Wiener filtering in the decoder. In [Spi12], the mixture spectrogram is separated by NMF into K components, each of them modeling a particular acoustical event. In the blind setting, the components have to be mapped to the sources using a blind grouping algorithm. This procedure is summarized in Section 3.3. In the ISS case, this information is stored in the grouping matrix $\mathbf{Q_s}$. The estimated sources are then obtained by Wiener filtering.

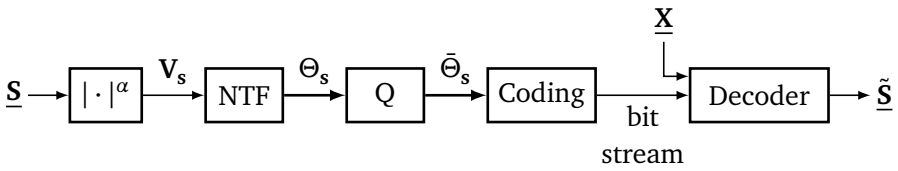
In Section 5.1, the necessary changes to the algorithm of [Spi12] for usage in an ISS context are summarized and the resulting decoder is evaluated. Furthermore, the decoder NTF is constrained with the objective to prevent deviations from the source NTF model in Section 5.2. Section 5.3 summarizes the findings of this chapter.

The usage of the decoder NTF was originally proposed in [RBW16] and constraining the decoder NTF in [RLB17].

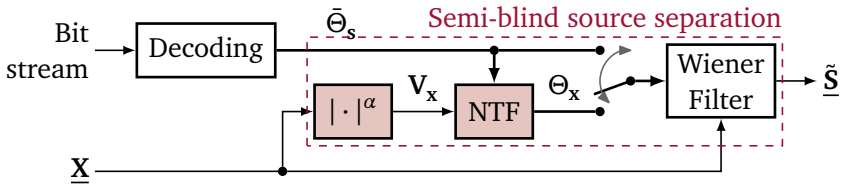
5.1 Nonnegative Factorization of the Mixture

In this chapter, the proposed ISS decoder is investigated in detail. The contribution can be interpreted in two different ways: On the one hand, it is an extension of the decoder of the ISS scheme proposed in [Liu+11] which uses NTF for parameter encoding but solely Wiener filtering for decoding. On the other hand, the blind source separation algorithm of [Spi12] is used in the decoder to enhance the transmitted parameters. This allows a second interpretation: This contribution also evaluates how this algorithm, originally used for BSS, performs with knowledge of the sources under a quantization constraint.

Figure 5.1 shows block diagrams of the encoder and the proposed decoder which are summarized in the following:



(a) Reference encoder. Consists of decoder.



(b) Decoder with semi-blind source separation (SBSS) extension.

Figure 5.1 Block diagrams of encoder and proposed ISS decoder in the TF domain.

Parameter estimation at the encoder The encoder is mainly structured as described in Chapter 4 and shown in Figure 5.1a. It calculates an NTF model of the source amplitude spectrograms $\mathbf{V}_s$ denoted with Θ_s. In a (subsequent) quantization step, the NTF model is quantized yielding $\bar{\Theta}_s$, the quantized source model. This model is sent to the decoder which the encoder is able to simulate to choose optimum working points for each mixture. These points are chosen by rate-quality optimization as discussed in Section 2.8.2.

Parameter re-estimation at the decoder The decoder proposed in [Liu+11] uses Wiener filtering of the mix $\underline{\mathbf{X}}$ with the quantized source model $\bar{\Theta}_s$ to reconstruct the estimated sources $\underline{\tilde{\mathbf{S}}}$. In [RBW16], a more advanced decoder was proposed using an NTF-based algorithm originally designed in [Spi12] for the task of blind source separation as summarized in Section 3.3. The main building blocks are already used in the encoder and shown in Figure 5.1b: The mixture amplitude spectrogram $\mathbf{V}_x$ is constructed in the same way as the source spectrogram in Equation (3.5) as

$$\mathbf{V}_x = \left(\mathbf{H}_{\text{mel}}^{\top} \left|\underline{\mathbf{X}}\right|\right)^{\alpha}. \tag{5.1}$$

$\mathbf{V}_x$ is now fed into the decoder NTF which computes the so-called *mix model* denoted with Θ_x given $\mathbf{V}_x$ as observation by minimizing

$$\min d_{\beta}\left(\mathbf{V}_x \mid \hat{\mathbf{V}}_x(\Theta_x)\right). \tag{5.2}$$

This model is used for Wiener filtering the complex mixture $\underline{\mathbf{X}}$ instead of $\bar{\Theta}_s$ in a subsequent step. The main difference to the algorithm proposed by [Spi12] is the non-blind initialization with the quantized source model: The mixture model Θ_x may be initialized with $\bar{\Theta}_s$ such that the decoder NTF is able to refine the quantized parameters given the mix. Therefore, the procedure in the decoder is denoted as Semi-blind Source Separation (SBSS). Note that other initializations are also possible: In the extreme case, the decoder is able to run without any transmitted parameters. In this case, the algorithm falls back to the blind source separation algorithm of [Spi12].

The factorization process of the mixture in the decoder is compared to the factorization of the sources in the encoder in Section 5.1.1. The decoder can be operated with different

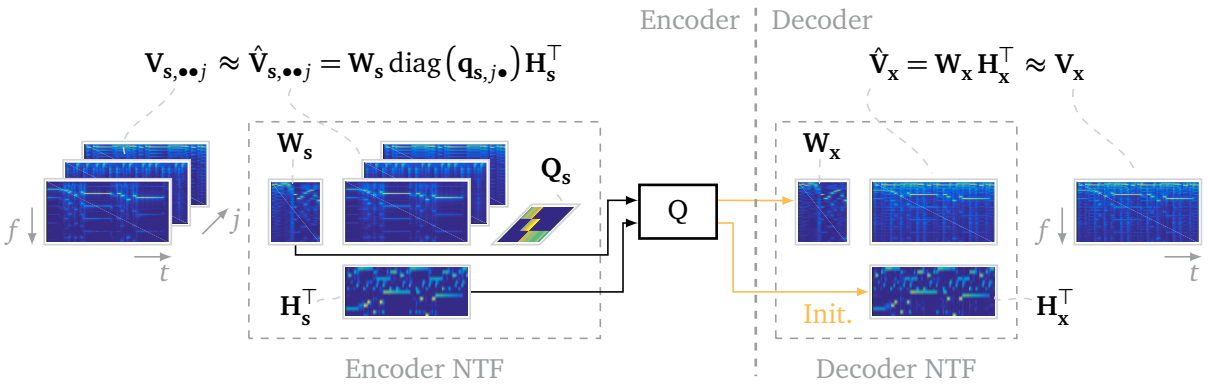


Figure 5.2 Comparison of encoder and decoder NTF models $\Theta_\mathbf{s}$ and $\Theta_\mathbf{x}$.

configurations, depending on the transmitted parameters. The encoder consists of the decoder as shown in Figure 5.1a which enables the encoder to simulate the decoder and choose the best configuration under a rate-quality-constraint. These configurations are presented in Section 5.1.2. In Section 5.1.3, the decoder is evaluated experimentally and compared to both the reference ISS and reference BSS methods.

5.1.1 Decoder Factorization Model

The basic procedure of factorization of the mixture was already subject in Section 3.3. Here, the two NTF models, one describing the sources and the other one describing the mixture, are compared. The encoder NTF model and the decoder NTF model are considered for an exemplary mixture with $J = 3$ sources in Figure 5.2. The encoder estimates $\Theta_\mathbf{s}$ given the original source spectrograms $\mathbf{V}_{\mathbf{s},\bullet\bullet j}$. To obtain $\Theta_\mathbf{s}$, the encoder NTF minimizes the β-divergence between the *original sources* and their estimation $\hat{\mathbf{V}}_\mathbf{s}(\Theta_\mathbf{s})$, $d_\beta\left(\mathbf{V}_\mathbf{s} \mid \hat{\mathbf{V}}_\mathbf{s}(\Theta_\mathbf{s})\right)$. The decoder NTF minimizes Equation (5.2), the β-divergence between mixture amplitude spectrogram and its approximation $\hat{\mathbf{V}}_\mathbf{x}(\Theta_\mathbf{x})$. The decoder model $\Theta_\mathbf{x}$ is initialized in this example with the quantized source model $\bar{\Theta}_\mathbf{s}$. Other initializations are also possible, refer to Section 5.1.2. The decoder NTF calculates only $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$. As mentioned above, the grouping information $\mathbf{Q}_\mathbf{x}$ has to be either provided by the encoder or estimated blindly in the decoder as stated in Section 5.1.2. This information is needed for Wiener filtering which is not shown in Figure 5.2. It is assumed that the mix $\underline{\mathbf{X}}$ is constructed as the sum of all sources for each TF point as shown in (2.32). This operation introduces *overlap* in the TF points of $\underline{\mathbf{X}}$ and in the TF points of the corresponding amplitude spectrogram $\mathbf{V}_\mathbf{x}$. These interferences limit the separation quality obtained by the decoder NTF model compared to the encoder NTF model. Figure 5.3 shows the exemplary NTF results of Figure 5.2 with more detail.

The NTF implementation of the decoder NTF is the same as the one used in the encoder. Multiplicative update rules as explained in Section 2.2 are used in both cases. The proposed NTF introduces an increase of computational complexity in the decoder: The NTF estimates $\mathbf{V}_\mathbf{x}$ of size $F \times T$ with K components with N_it iterations of the multiplicative update rules. It is shown in [Lin07] that the complexity of NMF using multiplicative update rules is $N_\mathrm{it} \times \mathcal{O}(FTK)$. In this thesis, F is rather small as it denotes the number of mel filters used in Equation (5.1) for computing $\mathbf{V}_\mathbf{x}$. Furthermore, only small numbers of iterations N_it are considered. The complexity is linearly increasing with both increasing number of time bins

T and number of components K. As already briefly discussed in Section 3.1.1, the delay is also increasing with T.

5.1.2 Decoder Configurations

Originally, it was proposed in [RBW16] to refine coarsely quantized parameters as already mentioned before. In addition to that, it is possible to omit the transmission of $\bar{\mathbf{W}}_\mathbf{s}$ and/or $\bar{\mathbf{H}}_\mathbf{s}$ and estimate them at the decoder given the mix (and the transmitted parameter). This procedure was already proposed in [REL17] to estimate missing Higher-Order SVD parameters describing the sources. This approach is adapted here for estimating the NTF parameters. It is also possible to use the SBSS algorithm in a blind setting if no parameters are transmitted at all. In the following, these different scenarios are summarized:

1. *All* quantized source parameters $\bar{\Theta}_\mathbf{s} = \left\{\bar{\mathbf{W}}_\mathbf{s}, \bar{\mathbf{H}}_\mathbf{s}, \bar{\mathbf{Q}}_\mathbf{s}\right\}$ are transmitted. The decoder NTF is initialized with $\bar{\mathbf{W}}_\mathbf{s}$ and $\bar{\mathbf{H}}_\mathbf{s}$ and re-estimates them given the mix spectrogram $\mathbf{V}_\mathbf{x}$, as evaluated in Section 5.1.3.1. $\bar{\mathbf{Q}}_\mathbf{s}$ is then used as grouping information for Wiener filtering with Equation (2.36).

2. Either $\bar{\mathbf{W}}_\mathbf{s}$ or $\bar{\mathbf{H}}_\mathbf{s}$ is not transmitted. The decoder initializes the missing parameter with random values[1]. The transmitted parameter is fixed while the missing parameter is estimated by the decoder NTF. Excluding $\bar{\mathbf{H}}_\mathbf{s}$ from transmission means that the transmitted parameters and therefore the bit rate are independent of time. This "transmit two" configuration is further evaluated in Section 5.1.3.2.

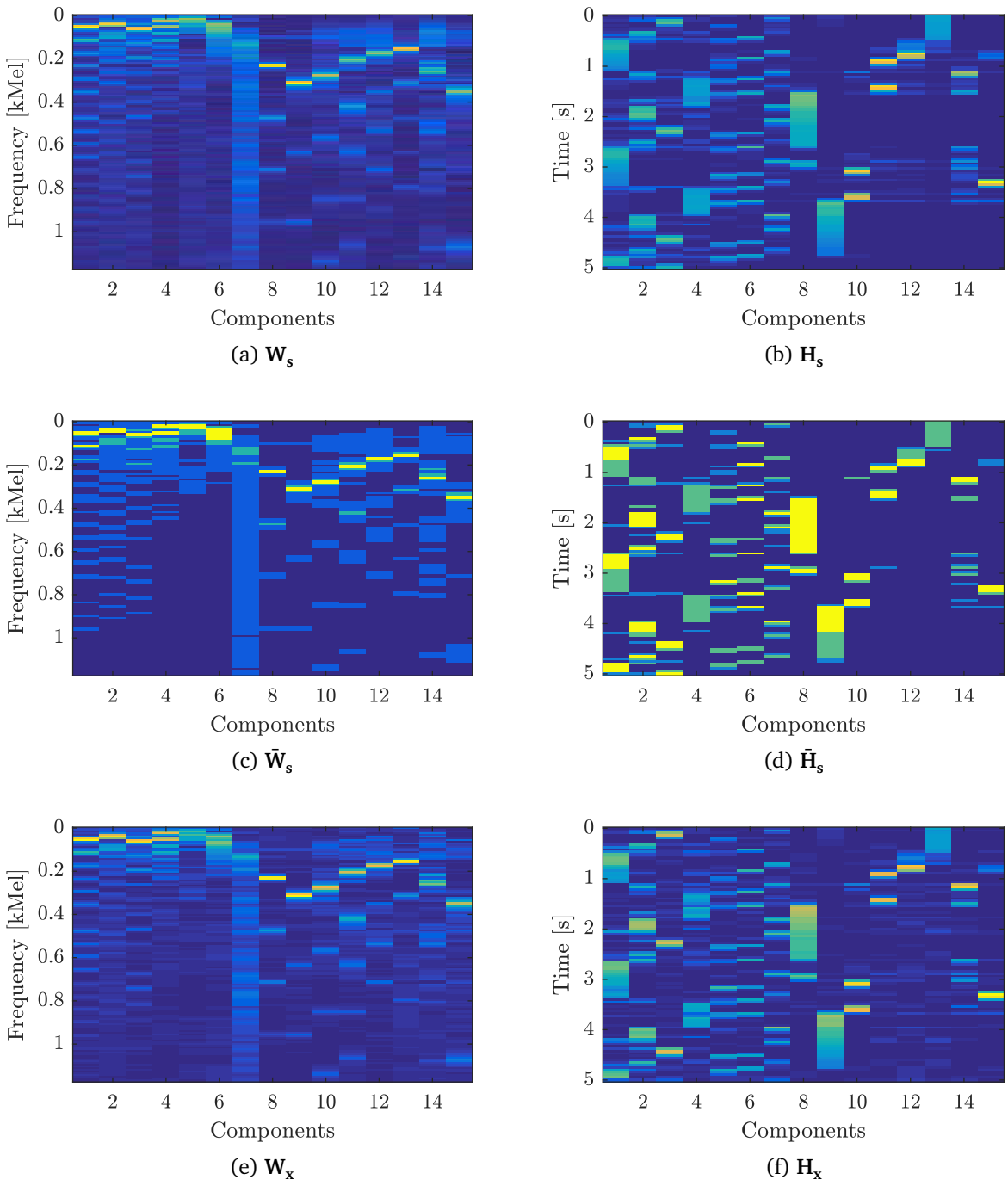
3. If neither $\bar{\mathbf{W}}_\mathbf{s}$ nor $\bar{\mathbf{H}}_\mathbf{s}$ are transmitted, both parameters have to be estimated. They are initialized with estimates given by the CSVD initialization [BMR15] already used in the encoder, now with input $\underline{\mathbf{X}}$ instead of $\underline{\mathbf{S}}$. In this case, the encoder runs the decoder NTF and uses the resulting $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$ to estimate $\mathbf{Q}_\mathbf{x}$ as shown in Eq. (5.3). $\mathbf{Q}_\mathbf{x}$ is then quantized with high quality yielding $\bar{\mathbf{Q}}_\mathbf{x}$ and transmitted back to the decoder where it is used in addition to $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$ for Wiener filtering. This configuration is evaluated in Section 5.1.3.2 and denoted as "transmit one".

4. If even the grouping information is not transmitted, the decoder falls back to the *blind setting*. In addition to initialization of $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$ with the CSVD, $\mathbf{Q}_\mathbf{x}$ is estimated by a blind grouping algorithm after the estimation of $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$ by the decoder NTF as proposed in [Spi12]. This case is investigated in Section 5.1.3.3.

5. The decoder NTF is *skipped*. This is indicated by setting the decision switch ( ) in Figure 5.1b to the upper position. The transmitted parameters are directly used for Wiener filtering which means that the proposed decoder falls back to the reference decoder.

All configurations are summarized again in Table 5.1. For case 3., the "transmit one" configuration, only $\bar{\mathbf{Q}}_\mathbf{x}$ is transmitted. The decoder NTF is initialized with the CSVD evaluated

[1] The "multi-start" initialization proposed by [Cic+09] is used. NTF with few iterations is applied on randomly initialized parameters. This is repeated several times and the parameters corresponding to the lowest cost function value are used as initialization of the subsequent decoder NTF. The random value generators of both encoder and decoder are assumed to be identical.



(a) $\mathbf{W}_s$ (b) $\mathbf{H}_s$

(c) $\bar{\mathbf{W}}_s$ (d) $\bar{\mathbf{H}}_s$

(e) $\mathbf{W}_x$ (f) $\mathbf{H}_x$

Figure 5.3 NTF factorizations of encoder and decoder NTF.

Configuration	Transmitted parameters	Initialization of decoder NTF / Comment
Transmit all	$\bar{\mathbf{Q}}_\mathbf{s}$, $\bar{\mathbf{W}}_\mathbf{s}$ and $\bar{\mathbf{H}}_\mathbf{s}$	Transmitted parameters.
Transmit two	$\bar{\mathbf{Q}}_\mathbf{s}$ and either $\bar{\mathbf{W}}_\mathbf{s}$ or $\bar{\mathbf{H}}_\mathbf{s}$	Transmitted parameter and random values for the missing parameter.
Transmit one	$\bar{\mathbf{Q}}_\mathbf{x}$	$\mathbf{W}_\mathbf{x}$, $\mathbf{H}_\mathbf{x}$ with CSVD on $\underline{\mathbf{X}}$. $\mathbf{Q}_\mathbf{x}$ estimated in encoder given the decoder NTF's output and transmitted as $\bar{\mathbf{Q}}_\mathbf{x}$ to decoder for Wiener filtering.
Blind	–	$\mathbf{W}_\mathbf{x}$, $\mathbf{H}_\mathbf{x}$ with CSVD on $\underline{\mathbf{X}}$, blind estimation of $\mathbf{Q}_\mathbf{x}$.
Skip	$\bar{\mathbf{Q}}_\mathbf{s}$, $\bar{\mathbf{W}}_\mathbf{s}$ and $\bar{\mathbf{H}}_\mathbf{s}$	None, the decoder NTF is skipped. Transmitted parameters directly used for Wiener filtering.

Table 5.1 Summary of decoder configurations.

on $\underline{\mathbf{X}}$ and estimates $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$ given $\mathbf{V}_\mathbf{x}$. Note that sending $\bar{\mathbf{Q}}_\mathbf{s}$ for Wiener filtering would lead to a mismatch: $\bar{\mathbf{Q}}_\mathbf{s}$ maps the components found by the encoder NTF to the J sources but the NTF in the decoder calculates $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$ blindly without any information provided by the encoder. Therefore, $\mathbf{Q}_\mathbf{x}$ is estimated in the encoder. First, the encoder runs the decoder which estimates $\Theta_\mathbf{x}$. Afterwards, the encoder is able to estimate $\mathbf{Q}_\mathbf{x}$ by minimizing

$$\min_{\mathbf{Q}_\mathbf{x}} d_\beta\left(\mathbf{V}_\mathbf{s} \mid \mathbf{W}_\mathbf{x}, \mathbf{H}_\mathbf{x}, \mathbf{Q}_\mathbf{x}\right) \tag{5.3}$$

while keeping $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$ fixed. The resulting parameter $\mathbf{Q}_\mathbf{x}$ is quantized with high quality yielding $\bar{\mathbf{Q}}_\mathbf{x}$ and sent to the decoder for Wiener filtering subsequently to the decoder NTF. For only one missing parameter ($\bar{\mathbf{W}}_\mathbf{s}$ or $\bar{\mathbf{H}}_\mathbf{s}$, configuration "transmit two"), this procedure was also tested but did not improve the separation quality.

5.1.3 Experimental Results

In this section, the proposed decoder including the SBSS algorithm is evaluated. The basic SBSS algorithm was already thoroughly evaluated in [Bec16] for the blind scenario. In this thesis, optimum parameters for the TF transform and the β-divergence are chosen in Section 3.2 which coincide with the findings in [Bec16]: The STFT window size is set to $N_\mathrm{w} = 2^{12}$ and the hop size to $N_\mathrm{h} = 2^{11}$ as well as $F = 500$ for mel filtering the spectral dimension of the resulting STFT matrices. The Kullback-Leibler divergence ($\beta = 1$) as NTF cost function is also selected which is minimized in 200 iterations. The encoder NTF runs with $K/J \in \{1, \ldots, 10\}$ components per source. $\bar{\Theta}_\mathbf{s}$ is obtained by quantizing $\Theta_\mathbf{s}$ with $N_\mathrm{q} \in \{2, 3, 4, 8, 16\}$ reconstruction values obtained by the dead-zone quantizer shown in Section 3.2.3.

The experiments are outlined as follows. In the first experiments, each decoder configuration listed in Table 5.1 is evaluated separately. In Section 5.1.3.1, the "transmit all" con-

figuration is evaluated for different numbers of decoder NTF iterations. The configurations "transmit two" and "transmit one", where either one or two parameters are omitted from transmission, are compared in Section 5.1.3.2. In Section 5.1.3.3, the proposed decoder is evaluated in the "blind" setting, meaning that no parameters are transmitted and the SBSS algorithm falls back to the BSS algorithm. Finally, in Section 5.1.3.4, the encoder is enabled to select the optimum decoder configuration. This scheme is compared to the reference ISS algorithm detailed in Section 3.1.

5.1.3.1 Parameter Re-estimation

In [RBW16], it was proposed to use the SBSS algorithm for refining coarsely quantized parameters. The decoder NTF for this scenario, denoted as the "transmit all" configuration, is investigated in this section. In the following, the impact of the number of decoder NTF iterations N_{it} on the separation quality is investigated.

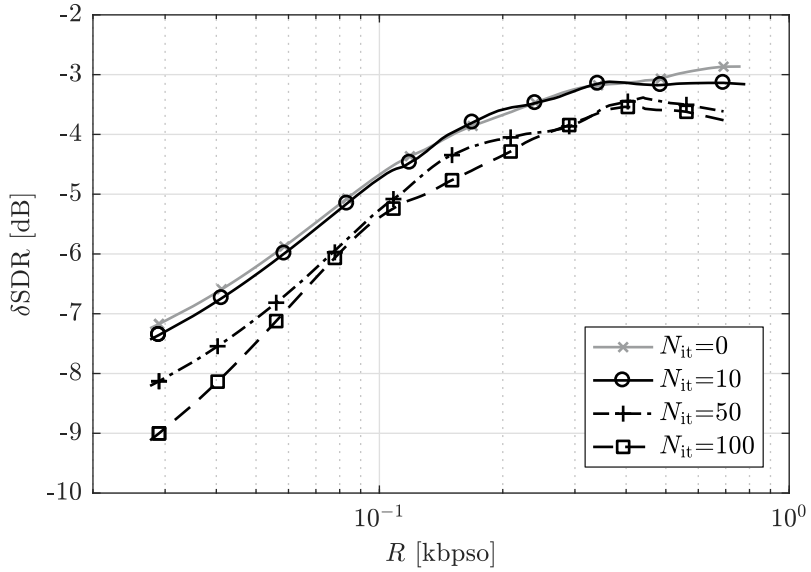
Figure 5.4 depicts rate-quality curves for different numbers of decoder NTF iterations $N_{\text{it}} \in \{10, 50, 100\}$ in comparison to the reference decoder using only Wiener filtering, denoted with $N_{\text{it}} = 0$. As quality measures, δSDR between original sources and estimated sources in Figure 5.4a as well as different NTF cost functions are given: The β-divergence between sources $\mathbf{V}_\mathbf{s}$ and quantized NTF source model $\bar{\Theta}_\mathbf{s}$ is taken into account, namely $d_\beta\left(\mathbf{V}_\mathbf{s} \mid \hat{\mathbf{V}}_\mathbf{s}\left(\bar{\Theta}_\mathbf{s}\right)\right)$ which is abbreviated with $d_\beta\left(\mathbf{V}_\mathbf{s} \mid \bar{\Theta}_\mathbf{s}\right)$. Figure 5.4b also shows $d_\beta\left(\mathbf{V}_\mathbf{s} \mid \Theta_\mathbf{x}\right)$, the β-divergence between the sources and the mix model $\Theta_\mathbf{x}$ which is obtained by the decoder NTF. Note that the decoder NTF does *not* minimize the aforementioned cost function but $d_\beta\left(\mathbf{V}_\mathbf{x} \mid \Theta_\mathbf{x}\right)$, the β-divergence between mix $\mathbf{V}_\mathbf{x}$ and $\Theta_\mathbf{x}$. $d_\beta\left(\mathbf{V}_\mathbf{s} \mid \Theta_\mathbf{x}\right)$ is evaluated after N_{it} iterations of the decoder NTF.

The results shown in Figure 5.4 can be summarized as follows:

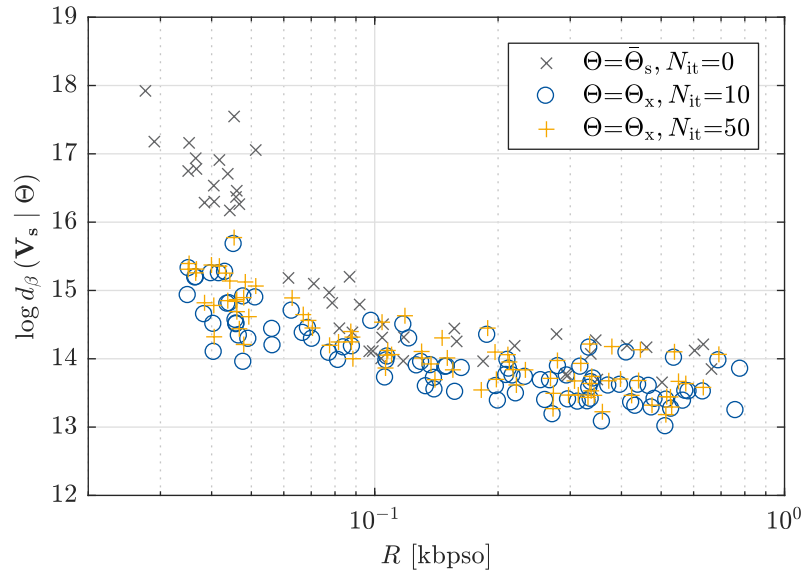
- The δSDR values obtained for the decoder NTF decrease with increasing number of iterations N_{it}. This can be explained by interferences introduced by the TF overlap in the mixture which is constructed by summation over the sources for each TF point (cf. (2.32)): Given the mixture as observation and the quantized source parameters $\bar{\Theta}_\mathbf{s}$ as initialization, the parameters obtained by the decoder NTF deviate from the initial values such that the distortion between original and estimated sources increases. Recall that the initial values stored in $\bar{\Theta}_\mathbf{s}$ are quantized versions of the *interference-free* source parameters $\Theta_\mathbf{s}$. The decoder NTF introduces more interferences by learning its parameters on the mixture. These interferences prevent a useful re-estimation of the quantized parameters. The better the approximation of the mixture with a larger number of iterations N_{it}, the higher the deviation from the sources becomes.

- When comparing the reference ($N_{\text{it}} = 0$, —×—) to the proposed decoder, using the decoder NTF achieves slightly worse δSDR results with $N_{\text{it}} = 10$ (—○—). The proposed method is not able to outperform the reference because of the previously mentioned deviations of the decoder NTF. For larger numbers of N_{it}, the distortion increases.

- $d_\beta\left(\mathbf{V}_\mathbf{s} \mid \bar{\Theta}_\mathbf{s}\right)$ evaluates the quality of the quantized parameters obtained by the reference decoder (×) with $N_{\text{it}} = 0$ as shown in Figure 5.4b. Here, this cost is compared to $d_\beta\left(\mathbf{V}_\mathbf{s} \mid \Theta_\mathbf{x}\right)$, measuring the cost between original sources and the parameters calculated by the proposed decoder NTF with $N_{\text{it}} = 10$ (○). It becomes clear that the decoder



(a) δSDR.



(b) Cost function with respect to sources $\mathbf{V_s}$.

Figure 5.4 Rate-quality curves for disabled decoder NTF ($N_{\mathrm{it}} = 0$) and enabled decoder NTF with different number of decoder NTF iterations $N_{\mathrm{it}} \in \{10, 50, 100\}$ in the "transmit all" configuration. Quality measures either δSDR or NTF cost functions $d_\beta\left(\mathbf{V_s} \mid \hat{\mathbf{V}}_\mathbf{s}(\Theta)\right) = d_\beta\left(\mathbf{V_s} \mid \Theta\right)$ with $\Theta = \bar{\Theta}_\mathbf{s}$ or $\Theta = \Theta_\mathbf{x}$.

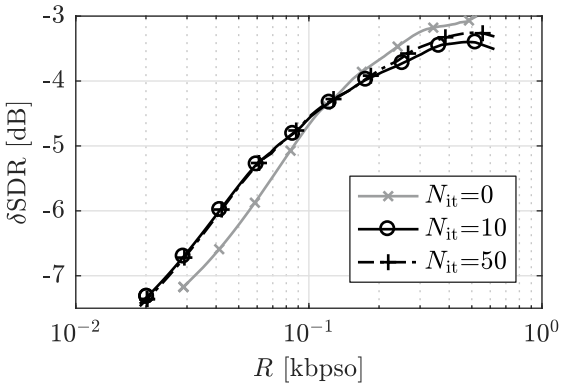


Figure 5.5 Configuration "transmit two": $\bar{\mathbf{Q}}_\mathbf{s}$ and either $\bar{\mathbf{W}}_\mathbf{s}$ or $\bar{\mathbf{H}}_\mathbf{s}$ transmitted.

NTF is able to increase the parameter quality (lower β-divergence value). Again, the actual cost function of the decoder NTF is $d_\beta\left(\mathbf{V}_\mathbf{x} \mid \Theta_\mathbf{x}\right)$ which is not shown here. Thus it is an interesting fact that activating the decoder NTF with $N_{\text{it}} = 10$ yields values for $d_\beta\left(\mathbf{V}_\mathbf{s} \mid \Theta_\mathbf{x}\right)$ smaller than the cost function $d_\beta\left(\mathbf{V}_\mathbf{s} \mid \bar{\Theta}_\mathbf{s}\right)$ obtained with the quantized model $\bar{\Theta}_\mathbf{s}$ without activating the decoder NTF ($N_{\text{it}} = 0$). Increasing the number of iterations to $N_{\text{it}} = 50$ (+) does not enhance the quality any further.

The cost function $d_\beta\left(\mathbf{V}_\mathbf{s} \mid \Theta_\mathbf{x}\right)$ is decreased for $N_{\text{it}} = 10$ compared to $d_\beta\left(\mathbf{V}_\mathbf{s} \mid \bar{\Theta}_\mathbf{s}\right)$ which means that $\Theta_\mathbf{x}$ models $\mathbf{V}_\mathbf{s}$ better than $\bar{\Theta}_\mathbf{s}$. However, this behavior is not reflected in the δSDR values. The small decrease of the cost function does not translate into a noticeable increase of the separation quality. Raising the number of iterations to $N_{\text{it}} = 50$ yields an even higher deviation from the optimum interference-free sources which is both reflected in the cost function as well as the δSDR scores. As already pointed out in Section 5.1.1, the mixing process (2.32) introduces overlap in the TF domain. With only the mixture $\mathbf{V}_\mathbf{x}$ as observation, the decoder NTF is not able to refine $\bar{\Theta}_\mathbf{s}$, which are quantized versions of the NTF parameters $\Theta_\mathbf{s}$ learned on the interference-free sources $\mathbf{V}_\mathbf{s}$.

5.1.3.2 Parameter Estimation

In this section, the decoder is evaluated in the "transmit two" and "transmit one" configurations as defined in Table 5.1 and compared to the reference ($N_{\text{it}} = 0$) which uses solely Wiener filtering.

Figure 5.5 gives rate-quality curves for the "transmit two"-configuration. In this scenario, only one parameter, either $\bar{\mathbf{W}}_\mathbf{s}$ or $\bar{\mathbf{H}}_\mathbf{s}$ is transmitted. A higher number of decoder iterations, $N_{\text{it}} = 50$ (−+·), gives slightly better results for higher rates than $N_{\text{it}} = 10$ (—o—). Note that the transmitted parameter is fixed during the update rules of the decoder NTF. Already few iterations are sufficient to estimate the missing parameter. Fixing the transmitted parameter also prevents the deviation from the source model for lower bit rates as it is the case in the "transmit all" configuration (refer to Section 5.1.3.1). For higher rates, transmitting both parameters as done by the reference (—×—) yields better results. Here, both $\bar{\mathbf{W}}_\mathbf{s}$ and $\bar{\mathbf{H}}_\mathbf{s}$ are transmitted and contain information about both frequency and temporal behavior which is beneficial for estimating the sources. However, the reference (—×—) is outperformed at lower rates by the proposed configuration.

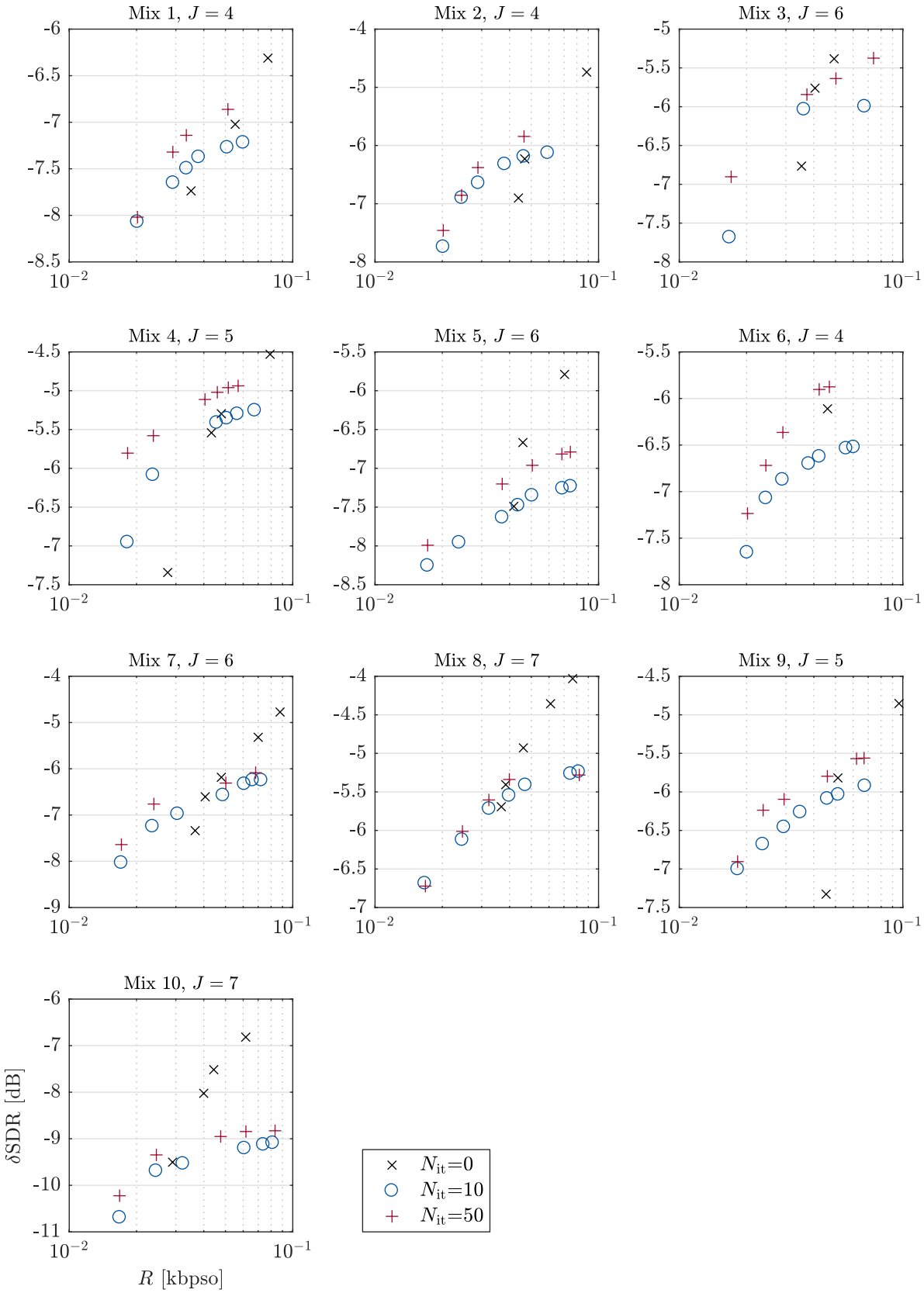


Figure 5.6 Configuration "transmit one": Only the grouping matrix, either $\bar{\mathbf{Q}}_{\mathbf{x}}$ or $\bar{\mathbf{Q}}_{\mathbf{s}}$ is transmitted. $\mathbf{Q}_{\mathbf{x}}$ is estimated with Equation (5.3).

Figure 5.6 shows results for the "transmit one"-configuration. Here, only the grouping information $\bar{\mathbf{Q}}_\mathbf{x}$ is transmitted and $\mathbf{W}_\mathbf{x}$ as well as $\mathbf{H}_\mathbf{x}$ are estimated given the mixture in the decoder without any guidance provided by the encoder. $\bar{\mathbf{Q}}_\mathbf{x}$ is solely used for Wiener filtering. Only the number of components per source K/J has impact on the bit rate R. The range of the x-axis corresponding to the rate R is limited to the interval $\left[10^{-2}, 10^{-1}\right]$.

- When comparing the number of iterations, it becomes clear that $N_\text{it} = 50$ (+) clearly outperforms $N_\text{it} = 10$ (○). In this configuration, the decoder NTF operates blindly. Using more iterations yields a better reconstruction as already found in [Bec16].
- Compared to the reference (×), $N_\text{it} = 50$ (+) is able to operate at very low rates and gives better results than the reference at rates in the range of $[10, 40]$ bpso (0.5 dB δSDR gain). Already at rates higher than 40 bpso, the reference outperforms the decoder in the "transmit one" configuration.

In Section 5.1.3.3, the "transmit one" configuration is compared to the BSS algorithm summarized in Section 3.3.

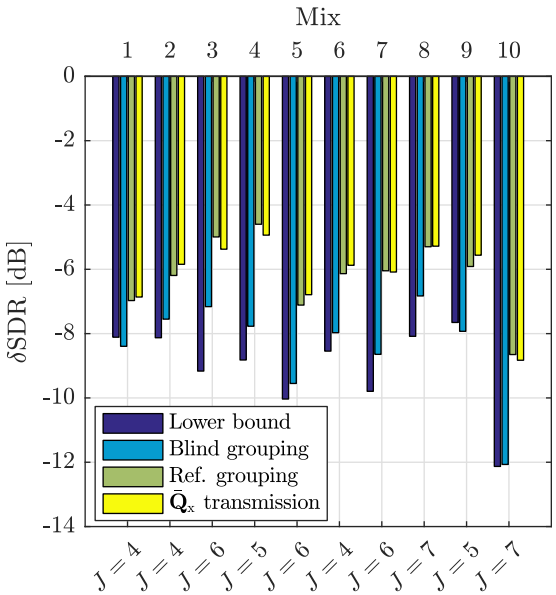
Comparing the lowest bit rates achieved in both configurations, it becomes clear that "transmit one" and "transmit two" yield both comparable low bit rates. This is due to the fact that the majority of the bit rate for "transmit two" is consumed by $\bar{\mathbf{Q}}_\mathbf{x}$ which is transmitted always with high precision. At lower rates, the transmitted parameter is quantized coarsely and needs only a small additional number of bits for transmission. Refer to Section 5.1.3.4 for a more detailed comparison of these configurations.

5.1.3.3 Comparison to Blind Source Separation

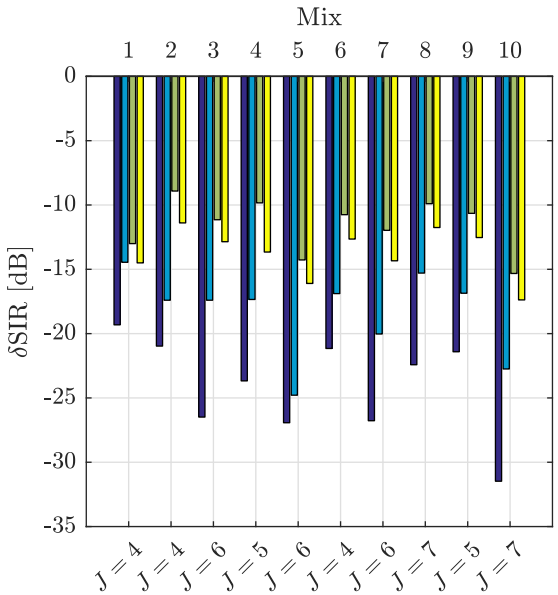
In the previous sections, the performance of the proposed ISS decoder for the first three configurations shown in Table 5.1 were evaluated. In this section, the configuration "transmit one", where only $\bar{\mathbf{Q}}_\mathbf{x}$ is transmitted, is compared to the "blind" configuration which estimates not only $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$ given the mix but also $\mathbf{Q}_\mathbf{x}$ mapping the NTF components to the sources. As already mentioned before in Section 3.3, the blind source separation algorithm [Spi12] is used in this case which utilizes a feature-based algorithm to group the K components in $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$ to the J sources. Recall that an upper bound for the blind grouping can be calculated as well given the original sources. This bound is also called reference grouping.

Usually, the NTF performance is enhanced with increasing number of components K. However, it was reported [Bec16] that the number of permutations of component-to-source-mappings increase depending on K which makes it harder to find a useful grouping blindly. The highest number of components/source is therefore not used. For each mix, the value of K/J is chosen which yields the highest quality measure for the blind grouping algorithm. This choice has no impact on the bit rate since in the "blind" setting, no NTF parameters are transmitted. For the other two methods, the reference grouping and the proposed decoder in the "transmit one" setting, optimum K/J values are determined per mix as well. For the latter case, the choice has impact on the bit rate. Since only $\bar{\mathbf{Q}}_\mathbf{x}$ has to be transmitted which is small compared to $\bar{\mathbf{W}}_\mathbf{s}$ and $\bar{\mathbf{H}}_\mathbf{s}$, the bit rate is negligible[2].

[2] The bit rates necessary for transmission of $\mathbf{Q}_\mathbf{x}$ are given as average and standard deviation in the captions of Figures 5.7a and 5.7b. Note that these values are different when measuring the performance with either δSDR (Figure 5.7a) or δSIR (Figure 5.7b) since the optimum K/J values were found yielding the highest corresponding quality measure per mixture.



(a) δSDR results. The transmission of $\bar{\mathbf{Q}}_\mathbf{x}$ needs an average rate of $R = 65.10(\pm 13.91)$ bits per second and object.



(b) δSIR results. The transmission of $\bar{\mathbf{Q}}_\mathbf{x}$ needs an average rate of $R = 65.86(\pm 13.12)$ bits per second and object.

Figure 5.7 δSDR and δSIR for lower bound, blind grouping, reference grouping and transmission of $\bar{\mathbf{Q}}_\mathbf{x}$ for each mix of test set $\mathscr{A}$.

In summary, the following methods will be compared:

- Estimation of each source by the mix with Equation (2.43).
- NTF on mix and blind grouping [Spi12].
- NTF on mix and reference grouping [Vir07].
- Proposed decoder in "transmit one" setting: NTF on mix, $\bar{\mathbf{Q}}_\mathbf{x}$ estimated in encoder given the output of the decoder NTF.

Figure 5.7 shows the results comparing these methods measured in δSDR and δSIR. The following observations can be made:

- The blind grouping algorithm gives better δSDR results than the estimates given by (2.43) for all but two mixtures (1 and 9). In terms of δSIR, the blind algorithm is able to perform better than the lower bound for all mixtures. It should be mentioned that taking the mix as estimate, as done for yielding the lower bound, is worse in terms of δSIR, measuring source interference, than in terms of δSDR. Therefore, it can be expected that it is easier to outperform the lower bound by means of SIR than for SDR. Note that in [Spi12] only mixtures with $J = 2$ sources were used to optimize the blind grouping algorithm and its parameters used here. The parameters of the blind grouping algorithm were not optimized for the usage in ISS but used as provided in [Spi12] for the BSS case with $J = 2$. This means that even with this sub-optimum parameter choice, the grouping algorithm as used in the proposed decoder in the "blind" setting

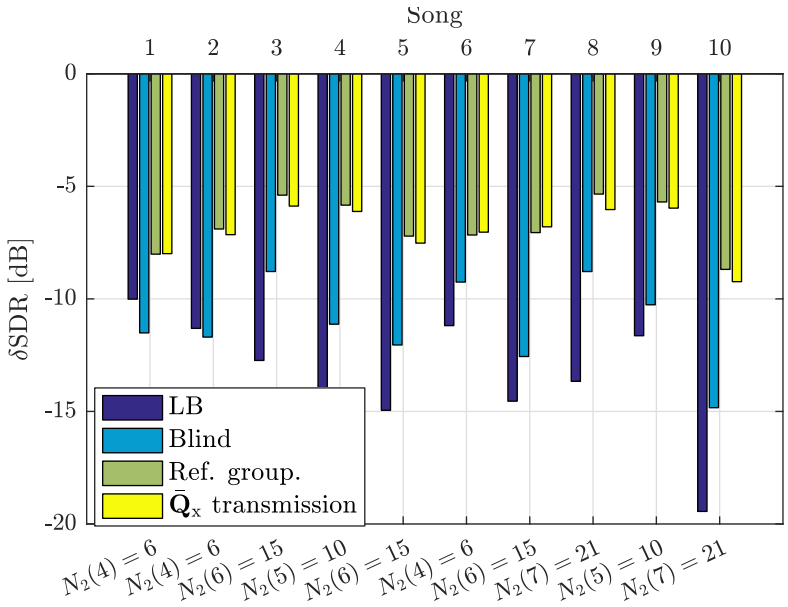


Figure 5.8 δSDR for lower bound, blind grouping, reference grouping and transmission of $\bar{\mathbf{Q}}_\mathbf{x}$ for mixes consisting of $J = 2$ sources.

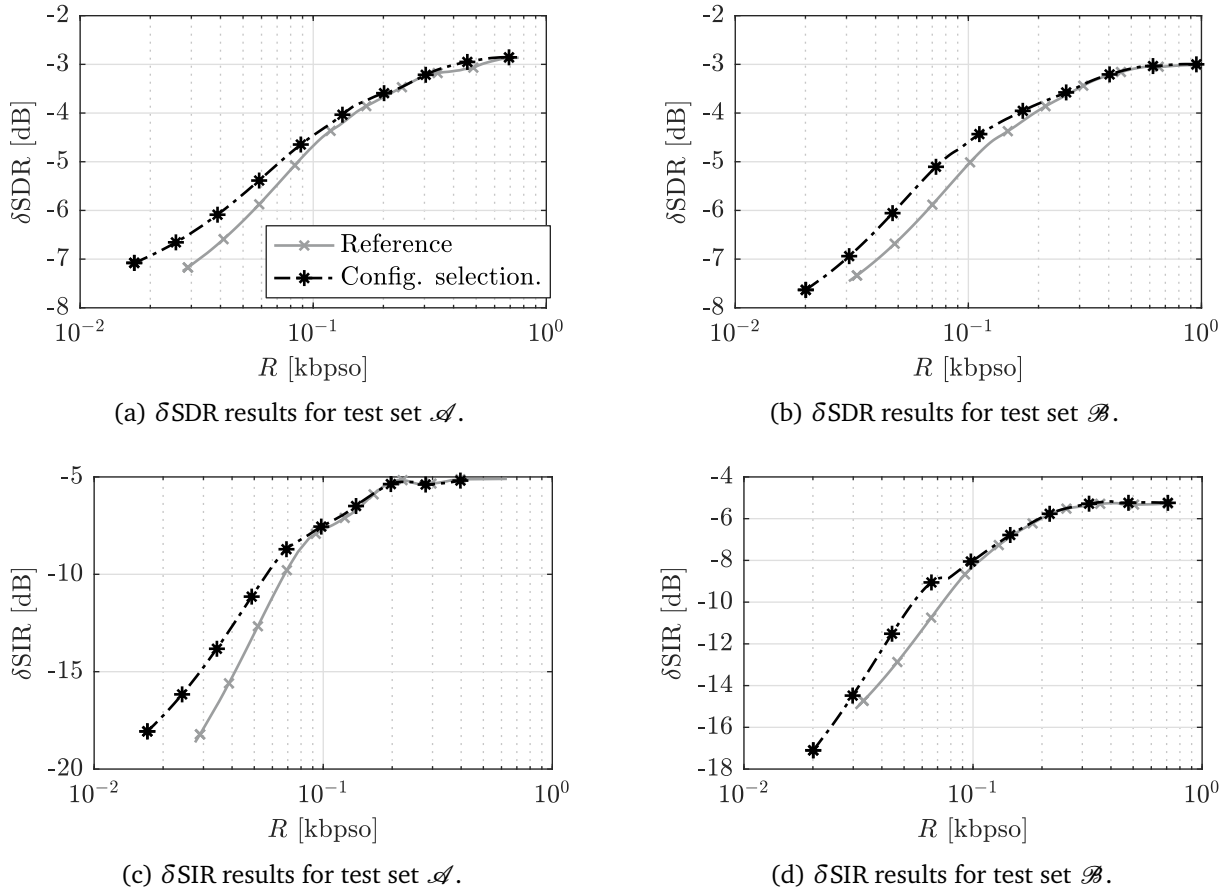
is able to obtain separation results higher than the lower bound (2.43) in many cases, with no transmission of parameters. To make the comparison more fair to the blind algorithm, results for separating mixtures of only $J = 2$ sources will be given further below.

- Comparing the reference grouping to the "transmit one"-performance, it becomes clear that estimating $\bar{\Theta}_\mathbf{x}$ with Equation (5.3) gives similar results. This is very interesting for the evaluation of blind grouping algorithms in general as the hill climbing approach which is used for obtaining the reference grouping is more time consuming than the 10 NTF iterations performed to obtain $\bar{\mathbf{Q}}_\mathbf{x}$.

- The bit rate needed for the transmission of $\bar{\mathbf{Q}}_\mathbf{x}$ amounts for both scores to $R \approx 65$ bpso (K/J is optimized for both scores independently). As found in Section 5.1.3.2, the proposed decoder in the "transmit one" configuration is only competitive to the reference ISS algorithm for rates smaller than 40 bpso. In this section, the optimum number of components per source K/J is chosen for the "transmit one" configuration based only the quality score (and thus neglecting the rate) to make a fair comparison to the reference grouping and the blind grouping. However, limiting K/J to yield bit rates around 40 bpso yields in a decrease for δSDR of only -0.31 ± 0.16 dB.

Since the parameters of the blind grouping algorithm were optimized for $J = 2$ sources per mix in [Spi12], another evaluation is conducted to make the comparison more fair: For each of the ten songs of test set $\mathscr{A}$, mixtures are created consisting of only two sources. Only sources of the same song are mixed. The number of mixtures per song originally consisting of J sources is $N_2(J) = \frac{J(J-1)}{2}$. This results in a total number of 125 mixtures.

Figure 5.8 shows δSDR values averaged over all mixtures of $J = 2$ sources of each song. It becomes clear that the blind algorithm is able to perform better in this scenario than



(a) δSDR results for test set $\mathscr{A}$.

(b) δSDR results for test set $\mathscr{B}$.

(c) δSIR results for test set $\mathscr{A}$.

(d) δSIR results for test set $\mathscr{B}$.

Figure 5.9 Configuration selection for test sets $\mathscr{A}$ and $\mathscr{B}$. If neither $\tilde{\mathbf{W}}_\mathbf{s}$ nor $\tilde{\mathbf{H}}_\mathbf{s}$ transmitted, estimate $\mathbf{W}_\mathbf{x}$, $\mathbf{H}_\mathbf{x}$ with $N_\text{it} = 50$ iterations and refine $\mathbf{Q}_\mathbf{x}$ afterwards. In all other cases, use $N_\text{it} = 10$.

in the scenario with more than two sources. Again, the reference grouping method and transmitting $\bar{\mathbf{Q}}_\mathbf{x}$ obtain similar values and outperform the blind algorithm.

5.1.3.4 Configuration Selection

Since the encoder includes the decoder, as shown in Figure 5.1a, the encoder may also decide which configuration the decoder should use. The configuration selection is taken into account for rate-quality optimization as summarized in Section 2.8.2. The encoder can decide between the configurations “transmit one”, “transmit two”, “transmit all” and “skip”. For the first two configurations, $\mathbf{W}_\mathbf{x}$ and/or $\mathbf{H}_\mathbf{x}$ are estimated with $N_\text{it} = 50$ iterations. For “transmit all”, the number is lowered to $N_\text{it} = 10$ and for “skip”, no decoder NTF iterations are performed, hence $N_\text{it} = 0$.

Figure 5.9 shows the corresponding results, evaluated on both test sets $\mathscr{A}$ and $\mathscr{B}$. As quality measures, δSDR and δSIR are shown on the first and the second row, respectively. As already shown in Figure 5.6 for the “transmit one”-configuration, the decoder NTF (—◇—) is operating at lower bit rates than the reference decoder (—×—). It also outperforms the reference at rates up to 10^{-2} kbpso. This results in gains for δSDR of more than 0.5 dB for test sets $\mathscr{A}$ and $\mathscr{B}$. Regarding δSIR, gains of up to 3 dB can be achieved. At higher ranges,

the reference encoder yields slightly better results, as already explained in Section 5.1.3.1. This means that the encoder decides to skip the SBSS algorithm in the decoder in these cases.

Figures 5.10 and 5.11 show the same results for test set $\mathscr{A}$ as Figure 5.9, the encoder can still decide which configuration to use. Here, each optimum rate-quality-point is shown and marked with respect to the chosen configuration. In Figure 5.10 δSDR and in Figure 5.11 δSIR is chosen as quality measure. The plots are combining rate-quality points for three configurations, namely "transmit one" (+), "transmit two" (○), "transmit all" (◇), and "skip" (□) in comparison to the reference encoder without the decoder NTF (×). In addition to that, quality measures for both lower bound, calculated with Equation (2.43), and for the "blind" estimation are given. As done for yielding the results shown in Section 5.1.3.3, the BSS algorithm is evaluated for each mix and for $K/J \in \{1,\ldots,10\}$. For each mix, the maximum quality measure is selected. This means that the encoder has to send only the optimum number for K/J.

The "transmit one" (+) and "transmit two" (○) configurations are chosen at lower bit rates, enabling lower bit rates than the reference (×) and increasing the separation quality significantly. When optimizing δSDR, the "transmit one" (+) configuration is often chosen at very low bit rates and outperforms the blind grouping (−✱−·) for most mixtures. This is not the case when choosing the configuration by means of δSIR. Here, "transmit one" is chosen less and is outperformed by the blind grouping in most cases. Although only the grouping information $\bar{\mathbf{Q}}_\mathbf{x}$ has to be transmitted in this case, the "transmit two" (○) configuration performs better at low rates for δSIR than for δSDR and is hence preferred in the configuration selection: In this configuration, $\bar{\mathbf{Q}}_\mathbf{s}$ and either $\bar{\mathbf{W}}_\mathbf{s}$ or $\bar{\mathbf{H}}_\mathbf{s}$ is transmitted. To increase δSIR, measuring interference of the other sources, extra information of either frequency or temporal behavior is highly beneficial. Assume e.g. the transmission of $\bar{\mathbf{Q}}_\mathbf{s}$ and a binary temporal activation $\bar{\mathbf{H}}_\mathbf{s}$ indicating if a source is active at some time bin or not. This information directly prevents the activity of other sources in time bins where the current source is active. This is the reason why "transmit two" (○) outperforms the blind grouping (−✱−·) already at very low rates and makes the "transmit one" (+) configuration unnecessary.

In only some cases, the "transmit all" configuration (◇) is chosen, yielding higher quality around rates of 10^{-1} kbpso. The reason, why this configuration is not chosen more often, is the deviation of the decoder NTF as already pointed out in Section 5.1.3.1. This is the same reason why the decoder NTF is often skipped (□) at rates towards 1 kbpso which means that falling back to solely Wiener filtering gives better source estimates than refining the quantized parameters with the decoder NTF.

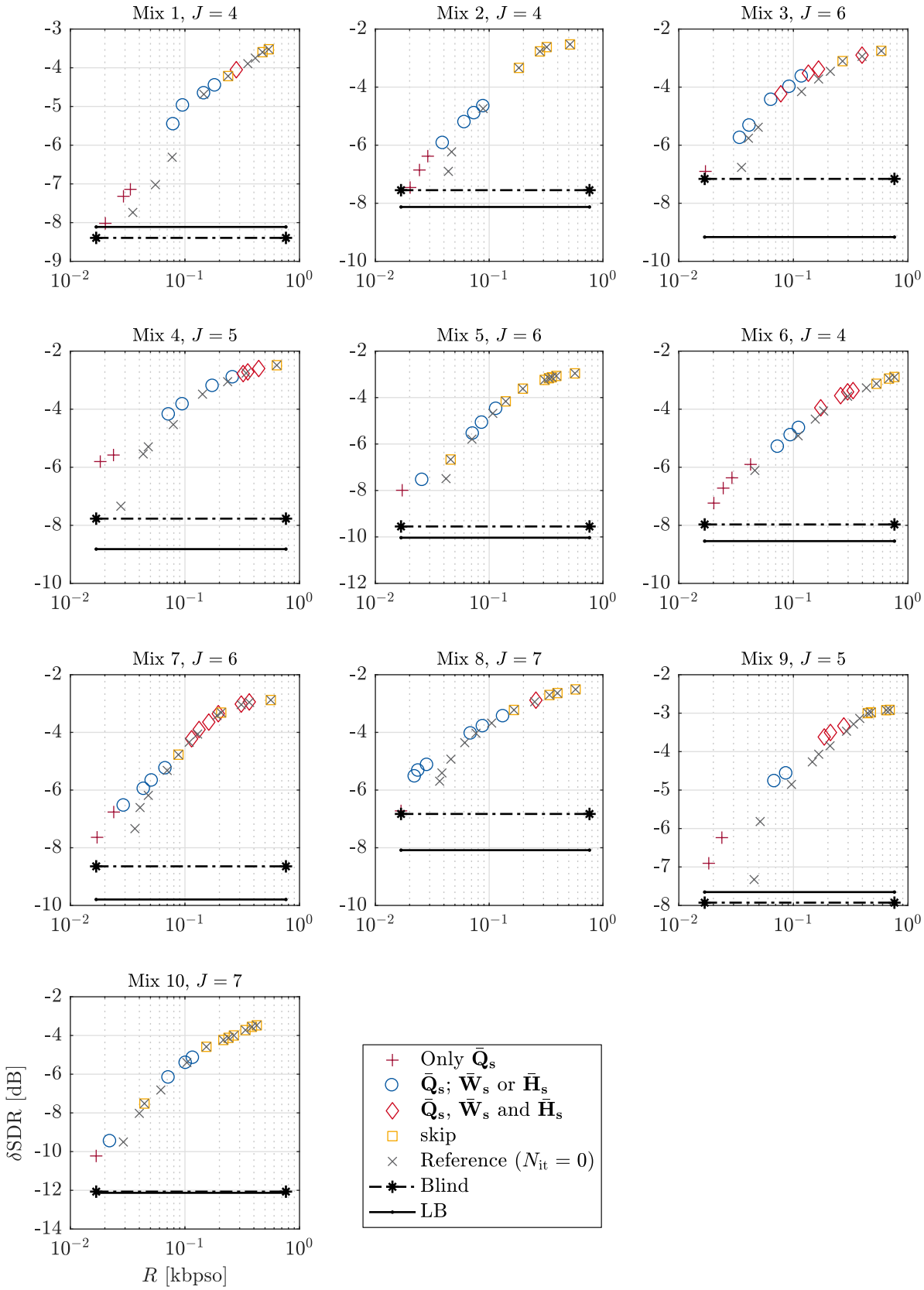


Figure 5.10 δSDR results per mixture of test set $\mathscr{A}$.

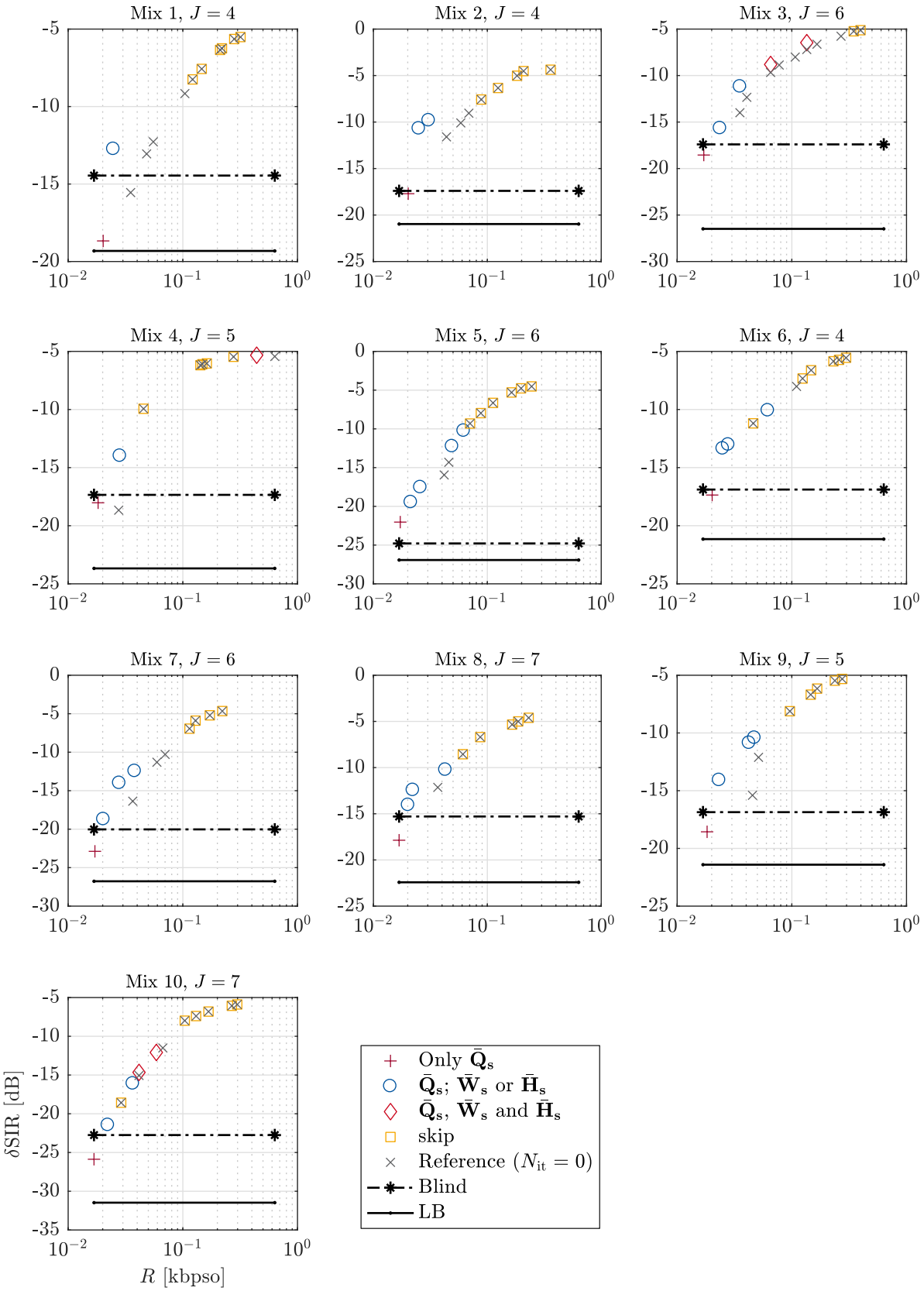


Figure 5.11 δSIR results per mixture of test set $\mathscr{A}$.

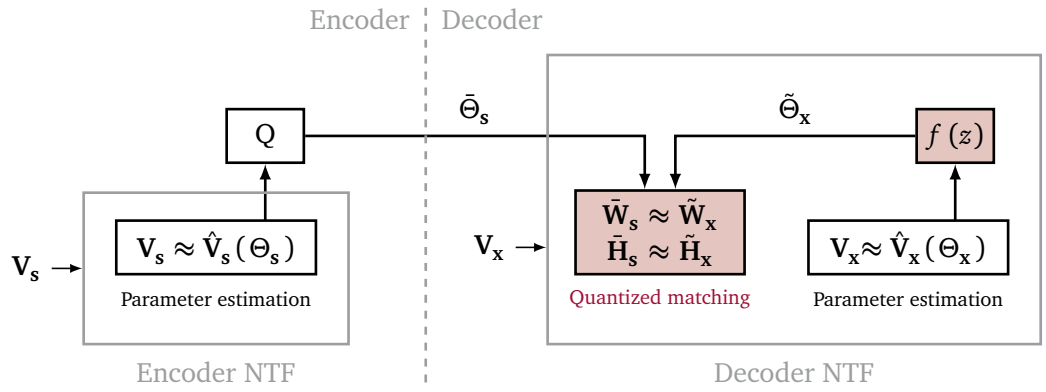


Figure 5.12 Quantized-matching. The decoder NTF model Θ_x is constrained in the quantization domain.

5.2 Decoder NTF Constraints

In Section 5.1, an NTF was introduced to the ISS decoder, enabling estimation of missing parameters or operating blindly. In the case where all NTF parameters are transmitted to the decoder, the decoder NTF is not able to refine the quantized parameters any further. Instead, the decoder NTF model deviates from the quantized source NTF model and is therefore not able to yield higher separation results compared to the quantized model as observed in Section 5.1.3.1. The higher the number of decoder NTF iterations, the higher the deviation. This can be explained by the fact that the decoder NTF estimates its parameter given only the mix, whereas the encoder NTF is given the interference-free sources as input instead.

In this section, a constraint on the decoder NTF preventing this deviation as proposed in [RLB17] is discussed. The quantized encoder NTF parameters $\bar{\Theta}_s$ are not only used for initialization but also for constraining the decoder NTF during its parameter updates. The main idea here is that $\bar{\Theta}_s$ is the result of quantizing the interference-free source model Θ_s which the decoder NTF tries to recover, given the mix and $\bar{\Theta}_s$. When quantized, the decoder NTF parameters Θ_x should match $\bar{\Theta}_s$ as much as possible. Directly constraining Θ_x being close to the quantized encoder parameters $\bar{\Theta}_s$ would yield in an unnecessary quantization of Θ_x. Instead, a quantized version of Θ_x is constrained to match $\bar{\Theta}_s$ as much as possible. This can be interpreted as a constraint in the quantization domain, providing as much freedom of learning Θ_x as possible while constraining it only when it is quantized to match $\bar{\Theta}_s$. This procedure is called *quantized-matching* in the following and is depicted in Figure 5.12:

In the encoder, Θ_s is learned with the source spectrogram $\mathbf{V}_s$ as observation and quantized in a subsequent step yielding $\bar{\Theta}_s$ which is transmitted to the decoder. Here, Θ_x is re-estimated with $\bar{\Theta}_s$ as initialization. To yield multiplicative update rules for the proposed constraint, the quantization characteristic has to be approximated by a differentiable function $f(z)$. This function is further explained in Section 5.2.1. Applying $f(z)$ on the parameters to-be-updated yields *soft-quantized parameters* $\tilde{\mathbf{W}}_x$ and $\tilde{\mathbf{H}}_x$ which are differentiable with respect to $\mathbf{W}_x$ and $\mathbf{H}_x$ to yield constraint update rules. This derivation is outlined in Section 5.2.2. In Section 5.2.3, a preliminary evaluation of the proposed quantized matching constraint is conducted. The constraint is finally evaluated experimentally in Section 5.2.4.

Note that other constraints on the decoder NTF steering the NTF at run-time were proposed in [RB16]. In addition to the quantized source NTF model, some prior information

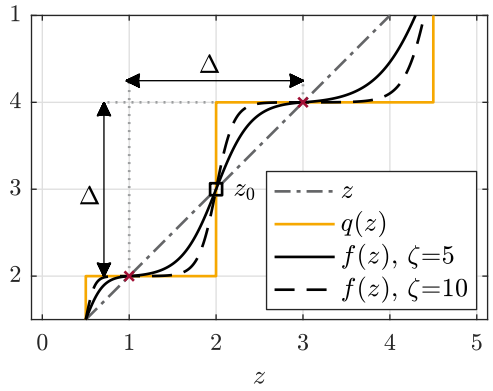


Figure 5.13 Soft quantization. The quantization characteristic $q(z)$ is approximated by differentiable soft quantization curve $f(z)$.

about the NTF components is transmitted to the decoder. It is shown that this variation is increasing the performance of the decoder NTF. The separation quality of the reference decoder, using solely Wiener filtering, is not met however. Therefore, these constraints are not further discussed here.

5.2.1 Approximation of Quantization Characteristic

This section deals with an approximation of the quantization characteristic which itself is not differentiable[3]. In [RLB17], it was proposed to approximate the quantization characteristic by a differentiable function, namely the logistic function. The quantization characteristic is approximated piecewise in the interval between two reconstruction values c_{g-1} and c_g with midpoint centered at the quantization boundary between the two reconstruction values $z_0 = e_g = \frac{c_{g-1}+c_g}{2}$. The so-called *soft quantization curve* is given as

$$f(z) = z_0 + \frac{\Delta}{d}\left[\underbrace{\left[1+\exp\left(-\zeta\frac{2}{\Delta}(z-z_0)\right)\right]^{-1}}_{=f_0(z)} - \frac{1}{2}\right] \tag{5.4}$$

with midpoint of the logistic function centered between two reconstruction values $z_0 = q(z) + \frac{\Delta}{2}\mathrm{sgn}(z-q(z))$, quantization characteristic $q(z)$ given in Equation (2.22) and steepness parameter ζ. The difference between the two reconstruction values is denoted with $\Delta = c_g - c_{g-1}$. The factor $d = [1+\exp(-\zeta)]^{-1} - [1+\exp(\zeta)]^{-1}$ scales $f_0(z)$ to ensure continuity at the corner points (lower and upper reconstruction value). The soft quantization curve can approximate all scalar quantizers discussed in Section 2.3. The derivative of $f(z)$ can be expressed as

$$\frac{\partial f(z)}{\partial z} = \frac{2\zeta}{d} f_0(z)[1-f_0(z)], \tag{5.5}$$

with the unscaled version $f_0(z)$ of $f(z)$ given in Equation (5.4). The derivative is further used in Section 5.2.2 to yield multiplicative update rules for the constraint.

[3] Constraining the NTF parameters to be close to their quantized version given directly by the quantization characteristic $q(z)$ would yield in degenerated update rules since the derivative yields either 0 or ∞.

Figure 5.13 shows an exemplary quantization characteristic $q(z)$ (——) as well as the proposed approximation $f(z)$ with $\zeta = 5$ and $\zeta = 10$. The soft quantization curve is approximating the quantization characteristic better with larger steepness factor $\zeta = 10$ (− −) than with $\zeta = 5$ (——).

5.2.2 Constraint Formulation

In the following, multiplicative update rules for the quantized-matching constraints are derived. As a first step, the parameter quantization process in the encoder (3.6) with A-law companding is given again as

$$\bar{\mathbf{W}}_\mathbf{s} = C_A^{-1}\left(q\left[C_A(\mathbf{W}_\mathbf{s})\right]\right), \quad \bar{\mathbf{H}}_\mathbf{s} = C_A^{-1}\left(q\left[C_A(\mathbf{H}_\mathbf{s})\right]\right).$$

Note that $q(z)$ is either a uniform scalar quantizer with $A \geq 1$ or an LM quantizer with $A = 1$. In [RLB17], the companding was carried out by taking the logarithm whereas the expanding was obtained by using the exponential function. The corresponding equations are given for the sake of completeness in Appendix B. The quantization procedure is emulated in the decoder where $q(z)$ is replaced by the soft quantization curve $f(z)$ as defined in Equation (5.4)

$$\tilde{\mathbf{W}}_\mathbf{x} = C_A^{-1}\left(f\left[C_A(\mathbf{W}_\mathbf{x})\right]\right), \quad \tilde{\mathbf{H}}_\mathbf{x} = C_A^{-1}\left(f\left[C_A(\mathbf{H}_\mathbf{x})\right]\right) \tag{5.6}$$

yielding *soft-quantized* parameters $\tilde{\mathbf{W}}_\mathbf{x}$ and $\tilde{\mathbf{H}}_\mathbf{x}$, gathered under $\tilde{\Theta}_\mathbf{x}$.

The cost function of the decoder NTF as given in Equation (5.2) is extended by the proposed quantized matching constraint

$$\min_{\Theta_\mathbf{x}} \; d_\beta\left(\mathbf{V}_\mathbf{x} \,|\, \hat{\mathbf{V}}_\mathbf{x}(\Theta_\mathbf{x})\right) + \gamma_{\text{qm}}\left[d_{\beta'}\left(\bar{\mathbf{W}}_\mathbf{s} \,|\, \tilde{\mathbf{W}}_\mathbf{x}\right) + d_{\beta'}\left(\bar{\mathbf{H}}_\mathbf{s} \,|\, \tilde{\mathbf{H}}_\mathbf{x}\right)\right] \tag{5.7}$$

penalizing the differences between the quantized source parameters $\bar{\mathbf{W}}_\mathbf{s}$ and $\bar{\mathbf{H}}_\mathbf{s}$ and the soft-quantized parameters $\tilde{\mathbf{W}}_\mathbf{x}$ and $\tilde{\mathbf{H}}_\mathbf{x}$ with the β-divergence. The corresponding parameter is denoted as β' not to be confused with the value β of the reconstruction term $d_\beta\left(\mathbf{V}_\mathbf{x} \,|\, \hat{\mathbf{V}}_\mathbf{x}(\Theta_\mathbf{x})\right)$. The scalar $\gamma_{\text{qm}} \geq 0$ weighs the influence of the constraint on the total cost term with $\gamma_{\text{qm}} = 0$ deactivating the constraint completely. Due to complexity reasons in the encoder, the cost functions for both parameters $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$ are weighted jointly with γ_{qm}. To formulate update rules minimizing (5.7), it is necessary to derive the soft-quantized parameters (5.6) with respect to the NTF-parameters $\mathbf{W}_\mathbf{x}$ and $\mathbf{H}_\mathbf{x}$ in a first step. The derivative of (5.6) is dependent on the derivative of $f(z)$ as given in Equation (5.5) and the derivatives of the companding and expanding functions $C_A(z)$ and $C_A^{-1}(z)$ as given in Equations (2.25) and (2.26). In the following, $\tilde{\mathbf{W}}_\mathbf{x}$ is differentiated with respect to $\mathbf{W}_\mathbf{x}$. The derivatives for $\tilde{\mathbf{H}}_\mathbf{x}$ can be obtained in the same manner. Using the chain rule yields

$$\nabla_{\mathbf{W}_\mathbf{x}} \tilde{\mathbf{W}}_\mathbf{x} = \nabla_{f[C_A(\mathbf{W}_\mathbf{x})]} C_A^{-1}\left(f\left[C\left(\mathbf{W}_\mathbf{x}\right)\right]\right) \cdot \nabla_{C_A(\mathbf{W}_\mathbf{x})} f\left[C_A(\mathbf{W}_\mathbf{x})\right] \cdot \nabla_{\mathbf{W}_\mathbf{x}} C_A(\mathbf{W}_\mathbf{x})$$

which is then split up into positive and negative gradient terms as

$$\begin{aligned}
\nabla^+_{\mathbf{W}_\mathbf{x}} \tilde{\mathbf{W}}_\mathbf{x} &= \frac{2\zeta}{d} f_0\left(C_A(\mathbf{W}_\mathbf{x})\right) \cdot \nabla_{f[C_A(\mathbf{W}_\mathbf{x})]} C_A^{-1}\left(f\left[C_A(\mathbf{W}_\mathbf{x})\right]\right) \cdot \nabla_{\mathbf{W}_\mathbf{x}} C_A(\mathbf{W}_\mathbf{x}), \\
\nabla^-_{\mathbf{W}_\mathbf{x}} \tilde{\mathbf{W}}_\mathbf{x} &= \frac{2\zeta}{d} f_0^2\left(C_A(\mathbf{W}_\mathbf{x})\right) \cdot \nabla_{f[C_A(\mathbf{W}_\mathbf{x})]} C_A^{-1}\left(f\left[C_A(\mathbf{W}_\mathbf{x})\right]\right) \cdot \nabla_{\mathbf{W}_\mathbf{x}} C_A(\mathbf{W}_\mathbf{x}).
\end{aligned} \tag{5.8}$$

$f_0(\mathbf{W}_\mathbf{x})$ is the unscaled version of $f(\mathbf{W}_\mathbf{x})$ as defined in Equation (5.4) and the derivative of $f(z)$ is given in Equation (5.5). Using the gradient terms of the soft quantized parameters (5.8) and the gradient of the β-divergence as given in Equation (2.17) yields finally positive and negative gradient terms for the proposed quantized-matching constraint

$$\begin{aligned}\nabla^+_{\mathbf{W}_\mathbf{x}} d_{\beta'}\left(\bar{\mathbf{W}}_\mathbf{s} \mid \tilde{\mathbf{W}}_\mathbf{x}\right) &= \nabla^+_{\mathbf{W}_\mathbf{x}} \tilde{\mathbf{W}}_\mathbf{x} \cdot \tilde{\mathbf{W}}_\mathbf{x}^{\beta'-1} + \nabla^-_{\mathbf{W}_\mathbf{x}} \tilde{\mathbf{W}}_\mathbf{x} \cdot \bar{\mathbf{W}}_\mathbf{x} \cdot \tilde{\mathbf{W}}_\mathbf{x}^{\beta'-2} \\ \nabla^-_{\mathbf{W}_\mathbf{x}} d_{\beta'}\left(\bar{\mathbf{W}}_\mathbf{s} \mid \tilde{\mathbf{W}}_\mathbf{x}\right) &= \nabla^-_{\mathbf{W}_\mathbf{x}} \tilde{\mathbf{W}}_\mathbf{x} \cdot \tilde{\mathbf{W}}_\mathbf{x}^{\beta'-1} + \nabla^+_{\mathbf{W}_\mathbf{x}} \tilde{\mathbf{W}}_\mathbf{x} \cdot \bar{\mathbf{W}}_\mathbf{x} \cdot \tilde{\mathbf{W}}_\mathbf{x}^{\beta'-2}\end{aligned}$$

which are used for updating $\mathbf{W}_\mathbf{x}$ in a multiplicative manner as already shown for the NTF reconstruction cost function in Section 2.2.1. As previously mentioned, the gradient terms for updating $\mathbf{H}_\mathbf{x}$ are derived in the same way. The multiplicative update rules minimizing (5.7) are extending Equation (2.20) where only the reconstruction cost (2.18) between input $\mathbf{V}_\mathbf{x}$ and the NTF approximation $\hat{\mathbf{V}}_\mathbf{x}(\Theta_\mathbf{x})$ is minimized

$$\begin{aligned}\mathbf{W}_\mathbf{x} &\leftarrow \mathbf{W}_\mathbf{x} \cdot \frac{\nabla^-_{\mathbf{W}_\mathbf{x}} d_\beta\left(\mathbf{V}_\mathbf{x} \mid \hat{\mathbf{V}}_\mathbf{x}\right) + \gamma_{\mathrm{qm}} \nabla^-_{\mathbf{W}_\mathbf{x}} d_{\beta'}\left(\bar{\mathbf{W}}_\mathbf{s} \mid \tilde{\mathbf{W}}_\mathbf{x}\right)}{\nabla^+_{\mathbf{W}_\mathbf{x}} d_\beta\left(\mathbf{V}_\mathbf{x} \mid \hat{\mathbf{V}}_\mathbf{x}\right) + \gamma_{\mathrm{qm}} \nabla^+_{\mathbf{W}_\mathbf{x}} d_{\beta'}\left(\bar{\mathbf{W}}_\mathbf{s} \mid \tilde{\mathbf{W}}_\mathbf{x}\right)} \\ \mathbf{H}_\mathbf{x} &\leftarrow \mathbf{H}_\mathbf{x} \cdot \frac{\nabla^-_{\mathbf{H}_\mathbf{x}} d_\beta\left(\mathbf{V}_\mathbf{x} \mid \hat{\mathbf{V}}_\mathbf{x}\right) + \gamma_{\mathrm{qm}} \nabla^-_{\mathbf{H}_\mathbf{x}} d_{\beta'}\left(\bar{\mathbf{H}}_\mathbf{s} \mid \tilde{\mathbf{H}}_\mathbf{x}\right)}{\nabla^+_{\mathbf{H}_\mathbf{x}} d_\beta\left(\mathbf{V}_\mathbf{x} \mid \hat{\mathbf{V}}_\mathbf{x}\right) + \gamma_{\mathrm{qm}} \nabla^+_{\mathbf{H}_\mathbf{x}} d_{\beta'}\left(\bar{\mathbf{H}}_\mathbf{s} \mid \tilde{\mathbf{H}}_\mathbf{x}\right)}.\end{aligned} \tag{5.9}$$

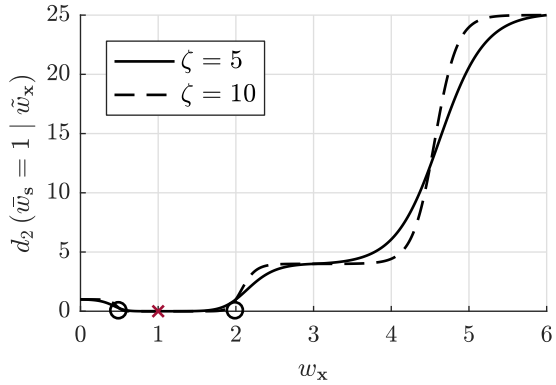
5.2.3 Preliminary Evaluation

In this section, the influence of the steepness parameter ζ of the soft quantization curve as given in Equation (5.4) on the constraint cost function (5.7) is evaluated. Here, the constraint is not yet implemented in the decoder NTF but evaluated for scalar values of $w_\mathbf{x}$ and $\bar{w}_\mathbf{s}$ as $d_2(\bar{w}_\mathbf{s} \mid \tilde{w}_\mathbf{x})$ with soft approximation $\tilde{w}_\mathbf{x} = f(w_\mathbf{x})$ given in (5.6).

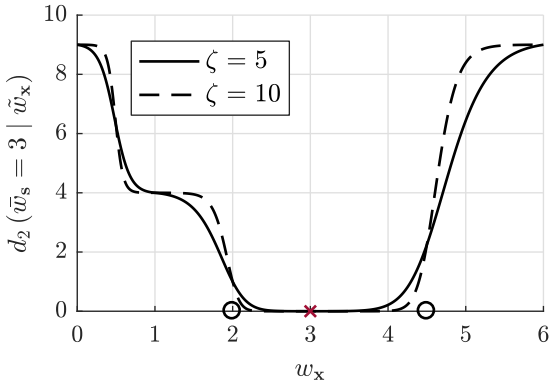
Figure 5.14 shows the constraint cost function $d_2(\bar{w}_\mathbf{s} \mid \tilde{w}_\mathbf{x})$ (upper row) and its gradient $\frac{\partial}{\partial w_\mathbf{x}} d_2(\bar{w}_\mathbf{s} \mid \tilde{w}_\mathbf{x})$ (lower row) for fixed scalar $\bar{w}_\mathbf{s} = 1$ (left column) and $\bar{w}_\mathbf{s} = 3$ (right column) and for two different steepness values $\zeta \in \{5, 10\}$ as already used in Figure 5.13.

It becomes clear that the cost function $d_2(\bar{w}_\mathbf{s} \mid \tilde{w}_\mathbf{x})$ has more obvious plateaus around all reconstruction values for $\zeta = 10$ (− −) than for $\zeta = 5$ (——) as shown in Figures 5.14a and 5.14b. These plateaus translate into the cost function gradient being close to zero in these regions as depicted in Figures 5.14d and 5.14c. This behavior is desired for the region around the *target* reconstruction value $\bar{w}_\mathbf{s}$: Once the parameter $w_\mathbf{x}$ is lying in the quantization interval, the constraint is deactivated by design. This ensures freedom of the parameter re-estimation process of the decoder NTF. For all other reconstruction values, these plateaus yield in the constraint being ineffective, especially for $\zeta = 10$: Values $w_\mathbf{x}$ lying in a quantization interval corresponding to *another* reconstruction value are not forced towards the target $\bar{w}_\mathbf{s}$ because the constraint is deactivated here as well. In contrast to this, the gradient still points towards the true reconstruction value $\bar{w}_\mathbf{s}$ for $\zeta = 5$: It can be seen in Figures 5.14c and 5.14d that the gradient is negative for $w_\mathbf{x} < \bar{w}_\mathbf{s}$ and positive for $w_\mathbf{x} > \bar{w}_\mathbf{s}$ for $\zeta = 5$ (——). For $\zeta = 10$ (− −), this behavior is only visible close to the quantization edges since the previously mentioned plateaus are wider.

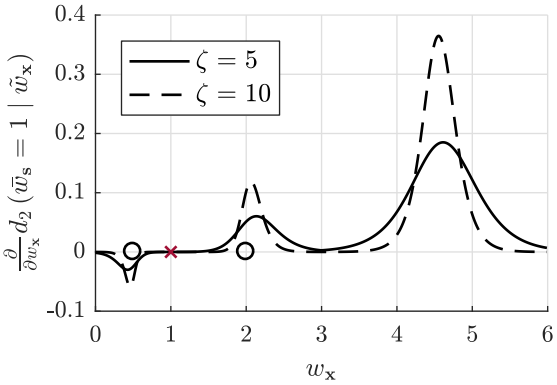
The cost function and its gradient are depicted for two different target values $\bar{w}_\mathbf{s} \in \{1, 3\}$ to show that the plateau is only present around the target reconstruction value $\bar{w}_\mathbf{s}$ for $\zeta = 5$.



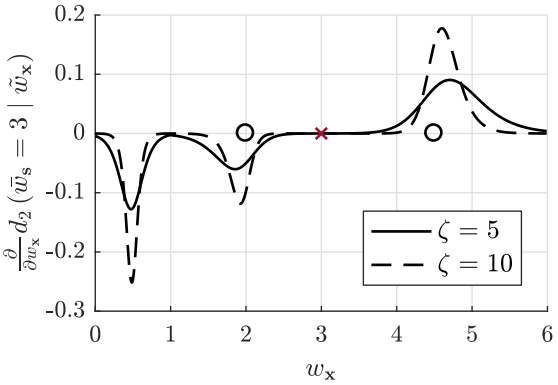
(a) Euclidean cost function for $\bar{w}_\mathbf{s} = 1$.



(b) Euclidean cost function for $\bar{w}_\mathbf{s} = 3$.



(c) Gradient of cost function for $\bar{w}_\mathbf{s} = 1$.



(d) Gradient of cost function for $\bar{w}_\mathbf{s} = 3$.

Figure 5.14 Quantized matching cost functions for soft quantization curve in Figure 5.13.

β'	δSDR		δSIR	
	$N_{\text{it}} = 50$	$N_{\text{it}} = 50$, QM	$N_{\text{it}} = 50$	$N_{\text{it}} = 50$, QM
0		−1.81		−22.49
1	51.86	−11.35	32.52	−11.34
2		−13.60		−6.24

Table 5.2 BD-BR reduction in % with respect to the reference decoder with $N_{\text{it}} = 0$.

E.g. for target $\bar{w}_{\mathbf{s}} = 3$, there is no clear plateau of the gradient around $w_{\mathbf{x}} = 1$ as shown in Figure 5.14d. For $\bar{w}_{\mathbf{s}} = 1$ as depicted in Figure 5.14c, there is no plateau near $w_{\mathbf{x}} = 3$.

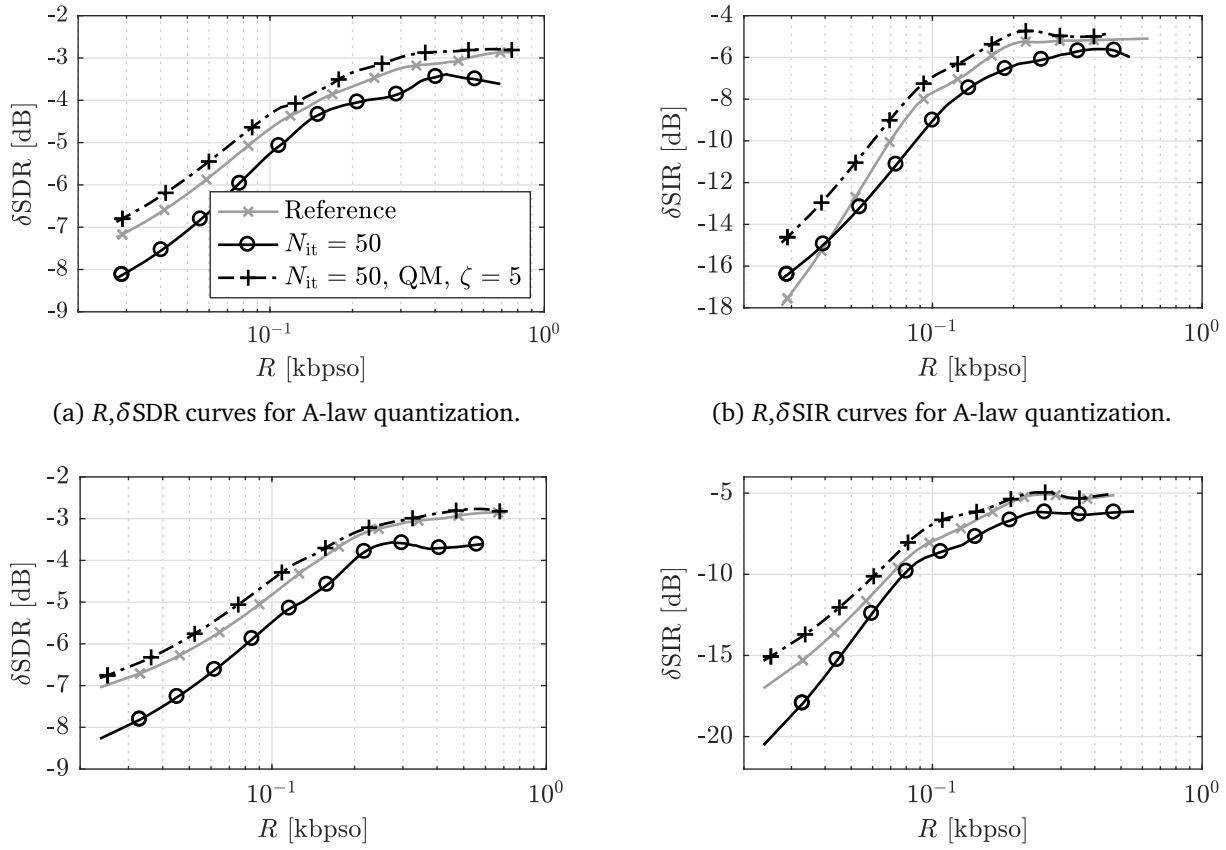
In summary, even though the soft quantization curve yields a more accurate approximation of the quantization characteristic for larger values of ζ, the corresponding constraint cost function may not prevent some entries of $\mathbf{W}_{\mathbf{x}}$ being in the wrong quantization interval in this case. In the following, $\zeta = 5$ is chosen.

5.2.4 Experimental Results

In this section, the proposed quantized-matching constraint is evaluated. The parameter values for K/J and N_{q} are chosen as given in Section 5.1.3. The decoder NTF is tested in the "transmit all" configuration. As already discussed in the beginning of Section 5.2 and shown experimentally in Section 5.1.3.1, this configuration, with no additional constraints, is not able to refine the transmitted parameters. Here, the impact of the proposed constraint is evaluated with constraint weights chosen as $\gamma_{\text{qm}} \in \left\{0, 10^{-1}, \ldots, 10^{5}\right\}$ and steepness factor $\zeta = 5$ as discussed in Section 5.2.3.

In a first experiment, the value of β' is evaluated selecting the corresponding cost function (5.7), namely for $\beta' \in \{0, 1, 2\}$. A-law quantization with $A = 10$ is chosen. Table 5.2 displays BD-BR values calculated with respect to the reference with $N_{\text{it}} = 0$. When using δSDR as score, it becomes clear that the Euclidean distance ($\beta' = 2$) as cost function performs best. The Euclidean distance is the only symmetric distance considered here. As shown in Section 5.2.3, this cost function punishes values which diverge to the left or to the right of the target reconstruction value to the same amount. This property yields less distortion between original and estimated sources as measured by δSDR. As shown in Figure 2.6, the other two considered cost functions, especially the IS distance ($\beta' = 0$), do not have this property and lead to a higher distortion.

Contrarily, when measuring the interferences coming from the other sources with δSIR, the IS distance ($\beta' = 0$) yields the highest bit rate reduction while the Euclidean distance yields the lowest one. This can be interpreted as follows. It is important that zero entries in $\bar{\mathbf{W}}_{\mathbf{s}}$ and $\bar{\mathbf{H}}_{\mathbf{s}}$ stay zero during the update rules of the decoder NTF to prevent the subsequent Wiener filter from introducing extra interferences. These entries indicate inactivity of the component k at the corresponding frequency bin f or time bin t. Whenever the target reconstruction value is equal to zero (in either $\bar{\mathbf{W}}_{\mathbf{s}}$ or $\bar{\mathbf{H}}_{\mathbf{s}}$), the IS distance ensures that the parameter values estimated by the decoder NTF stay close to zero: The term $-\log\frac{x}{y}$ in the β-divergence for $\beta = 0$ (refer to Equation (2.16)) already punishes small deviations of y (corresponding to



(a) R,δSDR curves for A-law quantization.

(b) R,δSIR curves for A-law quantization.

(c) R,δSDR curves for LM.

(d) R,δSIR curves for LM.

Figure 5.15 Results for quantized-matching.

the parameters $\mathbf{W}_{\mathbf{x}}$ or $\mathbf{H}_{\mathbf{x}}$) from $x \approx 0$ (corresponding here to the target reconstruction values in either $\bar{\mathbf{W}}_{\mathbf{s}}$ or $\bar{\mathbf{H}}_{\mathbf{s}}$). Note that for $\beta' = 1$ (KL divergence), the previously considered term is also present but scaled by x, $-x \log \frac{x}{y}$, which yields lower cost function values for small values of x.

In summary, $\beta' = 2$ yields the best results measured with δSDR because of the Euclidean distance's symmetry. In contrast to this, when using δSIR as score assessing interferences, $\beta' = 0$ performs best because the IS distance punishes deviations from zero entries in the transmitted parameters $\bar{\mathbf{W}}_{\mathbf{s}}$ and $\bar{\mathbf{H}}_{\mathbf{s}}$. In the following, $\beta' = 1$ is chosen as a trade-off as it leads to comparable rate reductions to $\beta' = 2$ for δSDR and slightly less reductions than $\beta' = 0$ for δSIR.

In the next experiment, two different quantization scenarios are considered to show that the constraint does not only work with A-law companding. In the following, either A-law quantization with $A = 10$ or LM is used. For both quantizers, rate-quality curves are given for the reference decoder, using solely Wiener filtering, and the unconstrained decoder NTF ($\gamma_{\mathrm{qm}} = 0$) with $N_{\mathrm{it}} = 50$ iterations. These methods are compared to the decoder NTF with $N_{\mathrm{it}} = 50$ iterations and the quantized-matching constraint activated ($\gamma_{\mathrm{qm}} \geq 0$).

Figure 5.15 shows results with both δSDR (left column) and δSIR (right column) as scores for A-law companding with $A = 10$ (upper row) and LM quantization (lower row).

- As already discussed in Section 5.1.3.1, the decoder NTF with $N_{\text{it}} = 50$ iterations (—o—) yields worse separation results than the reference decoder (—×—).
- Regarding δSDR, activating the proposed constraint (—+·) yields gains of about 1 dB compared to the decoder NTF with deactivated constraint (—o—) for both quantization methods. This means that the proposed constraint is able to force the factorization in the right direction and prevent the aforementioned deviations. The reference decoder (—×—) is outperformed as well: The gains are smaller in this case compared to the gains with respect to the decoder NTF. This translates to a BD-BR reduction with respect to the reference of about −11.35 % for A-law companding (as shown in Table 5.2) and −12.17 % for LM.
- When measuring the performance with δSIR, the constraint (—+·) also yields better results compared to the unconstrained decoder NTF (—o—). The reference is outperformed for very low bit rates (—×—) more clearly than for higher rates. In the latter case, the inactivity information provided by the quantized NTF matrices, which are directly used for Wiener filtering by the reference, is sufficient to prevent interferences.

In terms of bit rate, the proposed constraint comes for free since no extra information needs to be transmitted. The computational complexity of the decoder is however slightly increased since the constraint needs additional computations for the multiplicative update rules given in (5.9). For both considered quality scores and both quantization methods, the proposed constraint is able to enhance the separation quality of the reference. Regarding δSDR, the gains are rather small. However, the constraint is able to prevent interferences better than the reference method at rates smaller than 10^{-1} kbpso.

5.3 Summary

In this chapter, the reference decoder was extended to comprise a full BSS algorithm which also uses NTF, this time with the mixture as observation. This scheme was called Semi-blind Source Separation (SBSS). This decoder NTF was compared to the encoder NTF, taking the original sources as input. Adding the NTF to the decoder enables different configurations which are selected based on the amount of transmitted data.

In the “transmit all” configuration, all parameters are transmitted to be refined by the decoder. During evaluation, it became clear that this configuration does not yield better results than directly using Wiener filtering because the decoder NTF with mixture as observation deviates from the (optimum) encoder NTF results. This shortcoming is addressed by the quantized-matching constraint summarized further below. One or two NTF parameters are omitted from transmission in the “transmit two/both” configuration. The decoder NTF is able to estimate the missing parameter(s) and yields better separation results at very low bit rates. It became clear that the considered BSS algorithm, which is used in the case where no parameters are transmitted at all, is not performing well enough for mixtures with more than $J = 2$ sources. This problem influences the performance of the proposed SBSS algorithm as well as it is a modified version of the BSS algorithm. Although very low bit rates are achieved and the reference method is outperformed at low bit rates, the corresponding quality at these rates should be evaluated more thoroughly.

Additionally, a constraint was proposed in Section 5.2, constraining the mixture NTF model in the quantization domain. Since the transmitted parameters are quantized versions of the optimum encoder parameters, the quantized versions of the decoder parameters are constraint to be as similar to the transmitted parameters as possible. It was shown that this quantized-matching constraint works for different quantizers and is able to prevent interferences from other sources better than the reference at lower rates.

6 Residual Parameter Coding

Wiener filtering, as used in the reference decoder and the decoder proposed in Section 5.1, introduces artifacts because all resulting source estimates contain the mixture's phase. Re-estimation techniques, as discussed in Section 2.6, can refine the source estimates, even without transmitting any extra bit rate. However, the basic idea in this chapter is to transmit residuals, indicating the TF positions where the error between original and estimated sources is especially large. It is important to note that the re-estimation techniques work well only if the source magnitude spectrograms are estimated with a sufficiently high quality. Therefore, the NTF model should be estimated and quantized with high precision.

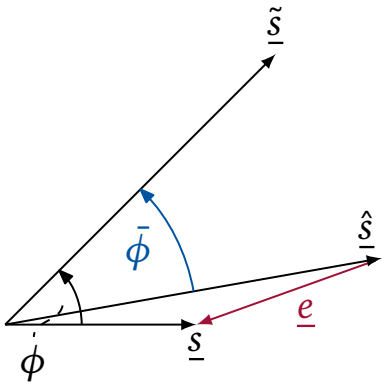
In Section 6.1, the calculation of phase errors between the original sources and the Wiener filter estimated sources is discussed. This error can be interpreted as a phase residual between the original phases in the encoder and the mixture's phase used in the decoder. It is furthermore proposed in Section 6.2 to quantize this error information, transmit it to the decoder and use it for refining the phases obtained by Wiener filtering in the decoder. As mentioned before, the magnitude was assumed to be estimated with sufficient quality. This assumption can not be met completely in practice. The transmission of a complex residual, containing both magnitude and phase information, is therefore considered in Section 6.3. These procedures are finally evaluated in Section 6.4 and summarized in Section 6.5.

6.1 Phase Residual

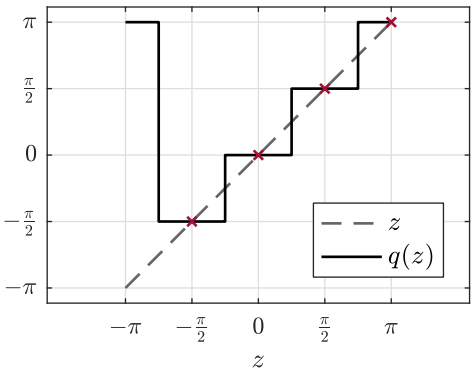
Given the original sources $\underline{\mathbf{S}}$ and the Wiener filter estimates $\underline{\tilde{\mathbf{S}}}$, obtained in general with (2.33) and in case of the NTF with (2.36), the phase error between $\underline{\mathbf{S}}$ and $\underline{\tilde{\mathbf{S}}}$ can be calculated as

$$\phi_{f,t,j} = \angle \frac{\exp\left(\jmath \angle \underline{s}_{f,t,j}\right)}{\exp\left(\jmath \angle \underline{\tilde{s}}_{f,t,j}\right)} \tag{6.1}$$

with phase of the original source $\angle \underline{s}_{f,t,j}$ and phase of the Wiener estimate $\angle \underline{\tilde{s}}_{f,t,j} = \angle \underline{x}_{f,t}$ for all j. This error $\mathbf{\Phi}$ is an $N_{\text{ny}} \times T \times J$ real-valued tensor with elements $-\pi \leq \phi_{f,t,j} \leq \pi$. In general, residuals in the TF domain, as holds true for $\mathbf{\Phi}$, are of size $N_{\text{ny}} \times T \times J$ which is comparable to the size of the original sources in time domain as discussed in Section 2.1. This means that transmitting this data has to be handled with care to avoid large bit rates. Here, two steps are considered: First, recall that TF representations are usually sparse. This allows for masking the magnitude spectrograms $\underline{\mathbf{S}}$ and $\underline{\tilde{\mathbf{S}}}$ prior to calculating $\mathbf{\Phi}$ with (6.1). Values below a certain magnitude threshold are ignored. Phase at TF points with a magnitude close to zero behaves completely random and the influence of these TF points on the separation quality is very small. Second, a very efficient quantization method, Rate-distortion Optimized Quantization (RDOQ) as summarized in Section 2.3.4, is used for quantization which minimizes not only the quantization distortion but also the bit rate. The application



(a) Phase residual between original source $\underline{s}_{f,t,j}$ and Wiener estimate $\tilde{\underline{s}}_{f,t,j}$ at a given TF point and source j. Subscripts omitted.



(b) Uniform phase quantization with $N_{\text{q,res}} = 4$ reconstruction values.

Figure 6.1 Phase residual and corresponding quantization characteristic.

of RDOQ on $\mathbf{\Phi}$ is detailed in Section 6.2. For now, it is assumed that it is possible to quantize $\mathbf{\Phi}$ efficiently which yields a quantized version of $\mathbf{\Phi}$ denoted as $\bar{\mathbf{\Phi}}$. At the decoder, the phase of the Wiener estimate $\tilde{\underline{\mathbf{S}}}$, which is by design of the Wiener filter the phase of the mixture $\underline{\mathbf{X}}$, can be refined as

$$\hat{\underline{s}}_{f,t,j} = \tilde{\underline{s}}_{f,t,j} \exp\left(\jmath \bar{\phi}_{f,t,j}\right) \tag{6.2}$$

with magnitude and phase of Wiener estimate as $\tilde{s}_{f,t,j} = \left|\tilde{\underline{s}}\right|_{f,t,j}$ and $\angle\tilde{\underline{s}}_{f,t,j} = \angle\underline{x}_{f,t}$. This refined estimation may be then subject to a subsequent re-estimation step as evaluated in Section 6.4, replacing $\tilde{\underline{\mathbf{S}}}$ with $\hat{\underline{\mathbf{S}}}$ as input of the re-estimation algorithms. Figure 6.1a shows these complex values for one (f,t,j) point.

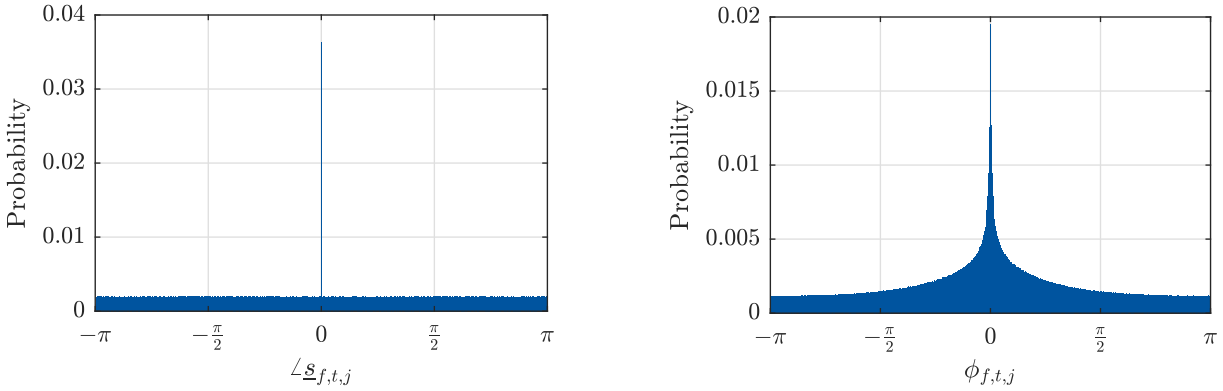
It is possible that the phase error (6.1) is large for more than one source at a given TF bin. Exploiting this redundancy is left for future work.

6.2 Quantization of Phase Residual

As mentioned in Section 6.1, the quantization of $\mathbf{\Phi}$ has to be handled with care, since $\mathbf{\Phi}$ holds $N_{\text{ny}} \times T \times J$ elements. In the following, only cases with small numbers of reconstruction values, denoted as $N_{\text{q,res}}$, are considered. Additionally, it is proposed to use Rate-distortion Optimized Quantization (RDOQ), as summarized in Section 2.3.4, thus minimizing the quantization distortion and the rate to transmit the corresponding quantized data jointly. The criterion (2.28) is given again as

$$\text{crit} = D + \lambda R. \tag{6.3}$$

Equation (6.3) is dependent on the Lagrangian multiplier λ which adjusts the impact of the rate on the total criterion. The proposed approach to minimize this criterion is quite simplistic. The criterion (6.3) is minimized locally for each TF point and each source: For each element $\phi_{f,t,j}$ of $\mathbf{\Phi}$, each of the $N_{\text{q,res}}$ reconstruction values c_g is considered with $1 \leq g \leq N_{\text{q,res}}$. The criterion (6.3) is evaluated for each combination of $\phi_{f,t,j}$ and c_g and the reconstruction value which yields the lowest value for the criterion is chosen for representing the current



(a) Histogram of $\angle\underline{\mathbf{S}}$.

(b) Histogram of $\mathbf{\Phi}$.

Figure 6.2 Histograms of both phase and phase error for an exemplary mixture.

element $\phi_{f,t,j}$. The corresponding quantization index is stored in the $N_{\text{ny}} \times T \times J$ tensor $\mathbf{G}$ with each element $1 \leq g_{f,t,j} \leq N_{\text{q,res}}$. The reconstruction values c_g are predetermined as uniformly spaced in the interval $(-\pi, \pi]$ where the values 0 and π are always represented as shown in Figure 6.1b. During the RDOQ process, the reconstruction values are not modified. In the following, the considered measures for D and R are discussed to calculate (6.3).

For deriving a measure for the distortion, the error between original source $\underline{s}_{f,t,j}$ and the refined estimate $\hat{\underline{s}}_{f,t,j}$ as calculated with (6.2) in the decoder is considered[1], namely $\underline{e}_{f,t,j} = \underline{s}_{f,t,j} - \hat{\underline{s}}_{f,t,j}$ as depicted in Figure 6.1a. The distortion is then evaluated as the squared magnitude of this error term,

$$D = \left|\underline{e}_{f,t,j}\right|^2 = \underline{e}_{f,t,j}\underline{e}^*_{f,t,j} = s^2_{f,t,j} + \tilde{s}^2_{f,t,j} - 2s_{f,t,j}\tilde{s}_{f,t,j}\cos\left(\phi_{f,t,j} - c_g\right) \tag{6.4}$$

with $s_{f,t,j} = \left|\underline{s}\right|_{f,t,j}$ and $\tilde{s}_{f,t,j} = \left|\tilde{\underline{s}}\right|_{f,t,j}$ denoting the magnitude of the original source and the Wiener estimate at TF point (t, f) and c_g the currently evaluated reconstruction value. The phase error $\phi_{f,t,j}$ is calculated according to Equation (6.1). Note that the magnitudes of the original and estimated sources are taken into account in the distortion term (6.4) as well, yielding a masking of TF points depending on the value of the Lagrangian multiplier λ.

The rate R is estimated with the entropy of the quantization indices

$$R = -\sum_{g=1}^{N_{\text{q,res}}} p_g \log_2 p_g \tag{6.5}$$

where the probability p_g is estimated at run-time with a histogram of previously chosen reconstruction values $1 \leq g_{f,t,j} \leq N_{\text{q,res}}$. To yield a better estimate, $\mathbf{\Phi}$ is quantized with a

[1] Another derivation is obtained by integrating the squared error between two cos-signals in time-domain with amplitudes $a, b \geq 0$ which have a phase shift ϕ

$$\int_0^1 \left[a\cos(2\pi t) - b\cos(2\pi t + \phi)\right]^2 \,\mathrm{d}t = a^2 + b^2 - 2ab\cos(\phi).$$

Algorithm 6.1 Rate-distortion optimized phase error quantization.

Input: phase error $\mathbf{\Phi}$, magnitudes of original sources $\mathbf{S} = \left|\underline{\mathbf{S}}\right|$ and Wiener estimated sources $\tilde{\mathbf{S}} = \left|\underline{\tilde{\mathbf{S}}}\right|$, number of reconstruction values $N_{\text{q,res}}$, Lagrangian multiplier λ

Output: quantization indices $\mathbf{G}$, reconstruction values $\mathbf{c}$

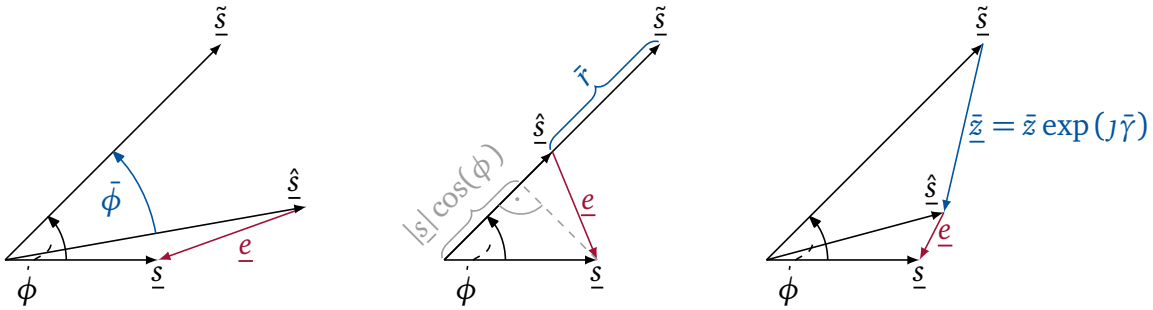
choose $N_{\text{q,res}}$ reconstruction values c_g equally spaced in $(-\pi, \pi]$, quantize $\mathbf{\Phi}$ with uniform quantization yielding initial quantization indices $\mathbf{G}_{\text{init}}$
calculate initial histogram $\mathbf{h}$ of values stored in $\mathbf{G}_{\text{init}}$
for each element $\phi_{f,t,j}$ of phase error $\mathbf{\Phi}$ **do**
  $\text{crit}_{\text{opt}} = \infty$
  for each quantization index g in $\left\{1, g_{\text{init},f,t,j} - 1, g_{\text{init},f,t,j}, g_{\text{init},f,t,j} + 1\right\}$ **do**
    calculate rate R with Equation (6.5) where probabilities $p_g = \frac{h'_g}{\sum_{g'} h'_{g'}}$ and modified local copy of histogram $\mathbf{h}' = \mathbf{h}$ with $h'_g = h_g + 1$
    calculate distortion D with Equation (6.4)
    $\text{crit} = D + \lambda R$
    if $\text{crit} < \text{crit}_{\text{opt}}$ **then**
      save $\text{crit}_{\text{opt}} = \text{crit}$ and $g_{\text{opt}} = g$
    end if
  end for
  update histogram as $h_{g_{\text{opt}}} = h_{g_{\text{opt}}} + 1$ and store quantization index $g_{f,t,j} = g_{\text{opt}}$
end for
return reconstruction values $\mathbf{c}$ and quantization indices $\mathbf{G}$

uniform quantizer prior to RDOQ with reconstruction values equally spaced in $(-\pi, \pi]$ as depicted in Figure 6.1b. The corresponding quantization indices are used for initializing the histograms estimating p_g[2]. Exemplary histograms of both $\angle\underline{\mathbf{S}}$ and $\mathbf{\Phi}$ are shown in Figure 6.2. The nonzero elements of $\angle\underline{\mathbf{S}}$ are uniformly distributed whereas the elements of $\mathbf{\Phi}$ are more likely to be closer to zero.

Evaluating (6.3) for each TF point and each source yields a decision which reconstruction value to use. The histogram is then updated accordingly and the chosen quantization index is stored in $g_{f,t,j}$. The proposed RDOQ process for quantizing $\mathbf{\Phi}$ is summarized in Algorithm 6.1. The resulting quantization indices, stored in tensor $\mathbf{G}$ of same size as $\mathbf{\Phi}$, are finally coded with an adaptive arithmetic coder (cf. Section 2.4).

Note that not all $N_{\text{q,res}}$ quantization indices are tested for minimizing (6.3). To limit the computational complexity, only the quantization indices are tested which correspond to reconstruction value zero ($g = 1$), the quantization index obtained by the uniform quantizer (which functions here as an initial guess), and quantization indices corresponding to the two neighboring quantization cells to the left and to the right of the quantization cell found by

[2]Note that CABAC was tested for encoding $\mathbf{G}$ and for estimating the rate R instead of (6.5). For very sparse $\mathbf{G}$, at lower bit rates, entries $g_{f,t,j} = 1$ corresponding to reconstruction value zero are very probable. CABAC adapts to this structure and the corresponding state machine frequently remains in the state with best adaption. However, CABAC is limited by the precision of the discrete probability values and needs to write out fractional bits each time the state machine remains in this state. Therefore, Equation (6.5) is preferred for estimating R. An adaptive arithmetic coder is used for coding the resulting quantization indices $\mathbf{G}$. Using CABAC for encoding $\mathbf{G}$ is still feasible if very long runs of $g_{f,t,j} = 1$ would be signaled beforehand.



(a) Phase residual. (b) Magnitude residual. (c) Complex residual.

Figure 6.3 Error terms between original source $\underline{s}_{f,t,j}$ and Wiener estimate $\tilde{\underline{s}}_{f,t,j}$ at a given TF point and source j. Error $\underline{e}_{f,t,j}$ between $\underline{s}_{f,t,j}$ and refined estimate $\hat{\underline{s}}_{f,t,j}$ marked in red, transmitted residuals in blue. Subscripts (f,t,j) omitted.

the uniform quantizer. This scheme limits the computational complexity and yields the same quantization results as obtained with a full search over all $N_{\text{q,res}}$ indices.

6.3 Generalized Complex Residual

Sections 6.1 and 6.2 dealt with phase errors, implicitly assuming that the magnitude of a source at a given TF point was estimated well enough. Contrarily, this section deals with a more general case, taking also magnitude errors into account. This leads to the transmission of a complex residual. For quantizing the residual, RDOQ is considered again as previously used for the phase residual in Section 6.2. In the following, three different cases are considered, namely the transmission of only a phase residual as already discussed in Section 6.1, of only a magnitude residual, or of a complex residual, consisting of both magnitude and phase. In all cases, the distortion for one TF point and one source j to be used for RDOQ is calculated as $D = \left|\underline{e}_{f,t,j}\right|^2 = \left|\underline{s}_{f,t,j} - \hat{\underline{s}}_{f,t,j}\right|^2$ with refinement $\hat{\underline{s}}_{f,t,j}$ depending on the residual and given below. The rate R is estimated with the entropy of the quantization indices as already discussed in Section 6.2. Figure 6.3 shows all three cases which are also summarized in the following:

1. Only the phase residuum is transmitted as already discussed in Section 6.1 and shown in Figure 6.3a. At the decoder, the phase of the Wiener estimate $\tilde{\underline{s}}_{f,t,j}$ is refined as

$$\hat{\underline{s}}_{f,t,j} = \tilde{s}_{f,t,j} \exp\left(\jmath\left[\angle\tilde{\underline{s}}_{f,t,j} + \bar{\phi}_{f,t,j}\right]\right)$$

   which yields the distortion term (6.4) given again as

$$D = \left|\underline{s}_{f,t,j} - \hat{\underline{s}}_{f,t,j}\right|^2 = s_{f,t,j}^2 + \tilde{s}_{f,t,j}^2 - 2s_{f,t,j}\tilde{s}_{f,t,j}\cos\left(\phi_{f,t,j} - \bar{\phi}_{f,t,j}\right).$$

   The reconstruction values are equally spaced in $(-\pi, \pi]$ and not modified during the RDOQ process. The histogram for estimating R with the entropy of the grouping indices is initialized given the grouping indices of scalar quantization of $\phi_{f,t,j}$ as given in Equation (6.1).

2. Only the magnitude residuum is transmitted. In this case, the phase of the estimated sources is not changed. The magnitude residual can be calculated as

$$r_{f,t,j} = s_{f,t,j}\cos\left(\phi_{f,t,j}\right) - \tilde{s}_{f,t,j}$$

where the first term, $s_{f,t,j}\cos\left(\phi_{f,t,j}\right)$, is the projection of $\underline{s}_{f,t,j}$ on $\underline{\tilde{s}}_{f,t,j}$ as shown in Figure 6.3b and $\phi_{f,t,j}$ the angle between $\underline{s}_{f,t,j}$ and $\underline{\tilde{s}}_{f,t,j}$, cf. (6.1). This residual is quantized with RDOQ and transmitted to the decoder where the refinement is obtained given the quantized residual $\bar{r}_{f,t,j}$ with

$$\underline{\hat{s}}_{f,t,j} = \left(\tilde{s}_{f,t,j} + \bar{r}_{f,t,j}\right)\exp\left(\jmath\angle\underline{\tilde{s}}_{f,t,j}\right).$$

For quantization of $r_{f,t,j}$ with RDOQ, the distortion is measured as

$$D = \left|\underline{s}_{f,t,j} - \underline{\hat{s}}_{f,t,j}\right|^2 = s_{f,t,j}^2 + \left(\tilde{s}_{f,t,j} + \bar{r}_{f,t,j}\right)^2 - 2s_{f,t,j}\left(\tilde{s}_{f,t,j} + \bar{r}_{f,t,j}\right)\cos\left(\phi_{f,t,j}\right).$$

The reconstruction values are obtained by quantizing $r_{f,t,j}$ with LM which also yields quantization indices for initialization of R.

3. Both magnitude and phase residuals are transmitted. The complex residual is calculated as $\underline{z}_{f,t,j} = \underline{s}_{f,t,j} - \underline{\tilde{s}}_{f,t,j}$ as shown in Figure 6.3c. At the decoder, the Wiener estimate $\underline{\tilde{s}}_{f,t,j}$ is refined with the quantized version $\underline{\bar{z}}_{f,t,j}$ of $\underline{z}_{f,t,j}$

$$\underline{\hat{s}}_{f,t,j} = \underline{\tilde{s}}_{f,t,j} + \underline{\bar{z}}_{f,t,j}. \tag{6.6}$$

For quantization, the complex residual is expressed as $\underline{z}_{f,t,j} = z_{f,t,j}\exp\left(\jmath\gamma_{f,t,j}\right)$ with magnitude $z_{f,t,j}$ and phase $\gamma_{f,t,j}$ (not to be confused with the angle between $\underline{s}_{f,t,j}$ and $\underline{\tilde{s}}_{f,t,j}$, $\phi_{f,t,j}$). Quantization then yields quantized versions of magnitude and phase, denoted as $\bar{z}_{f,t,j}$ and $\bar{\gamma}_{f,t,j}$. For a joint quantization of magnitude and phase with RDOQ, the distortion between original source $\underline{s}_{f,t,j}$ and refinement $\underline{\hat{s}}_{f,t,j}$ is taken as

$$D = \left|\underline{s}_{f,t,j} - \Big[\,\underline{\tilde{s}}_{f,t,j} + \underbrace{\bar{z}_{f,t,j}\exp\left(\jmath\bar{\gamma}_{f,t,j}\right)}_{=\underline{\bar{z}}_{f,t,j}}\,\Big]\right|^2. \tag{6.7}$$

In this thesis, the same number of reconstruction values for $\bar{z}_{f,t,j}$ and $\bar{\gamma}_{f,t,j}$ are used and denoted as $N_{\text{q,res}}$. The reconstruction values corresponding to $\bar{z}_{f,t,j}$ are obtained with LM with $\left|\underline{s}_{f,t,j} - \underline{\tilde{s}}_{f,t,j}\right|$ as input. This also yields quantization indices used for initializing the histogram for the rate term. The reconstruction values for $\bar{\gamma}_{f,t,j}$ are uniformly spaced in $(-\pi, \pi]$. The rate portion spent on the transmission of $\bar{\gamma}_{f,t,j}$ is initialized with quantization indices which are calculated with uniform quantization of $\angle\left(\underline{s}_{f,t,j} - \underline{\tilde{s}}_{f,t,j}\right)$.

During the proposed procedure, especially case 3, a complex residual for each source and TF point is transmitted. As previously mentioned in Section 6.1, coherence of complex errors is not considered in this thesis. However, other more sophisticated methods exist: In

CISS [Oze+13], a residual based on transform coding is transmitted as summarized in Section 3.1.3. For each TF point, all sources are considered jointly and transformed by means of the Karhunen–Loève Transform (KLT) where the transformation matrix is calculated given the same NTF model already used for Wiener filtering. This approach exploits the dependencies of the sources given the mixture as observation. Note that an SVD computation for each TF point in both encoder *and* decoder are required here. In contrast to CISS, no further computations are needed in the decoder of the proposed method. The residual is simply added to the Wiener estimate as shown in Equation (6.6). This means that compared to the costly SVD computation for each TF point, the computational complexity of the proposed refinement in the decoder is negligible. The authors of [Oze+13] provide MATLAB code which is used in the Section 6.4. The bit rate for the residual however is not measured with an actual coding step but rather estimated given by the underlying theoretical framework assuming scalar quantization and an arithmetic coding step. It is important to note that the proposed RDOQ-based method can be extended to exploit correlation between the errors and is also applicable in the KLT domain.

6.4 Experimental Results

In this section, the proposed residual calculation and quantization methods are evaluated. Regarding the chosen parameter configuration, it has to be mentioned again that the NTF parameters modeling the source magnitude spectrograms have to be transmitted with high precision, therefore $N_\mathrm{q} = 2^8$ reconstruction values are used throughout this section. The parameters are quantized in the logarithmic domain again, cf. (3.3). For the same reason, the number of components per source is evaluated in a wider range $K/J \in [1, 2, \ldots, 30]$. For RDOQ, the Lagrangian multiplier is set to $\lambda \in \left[10^2, 10^3, \ldots, 10^8\right]$. Test set $\mathscr{A}$ is used throughout this section.

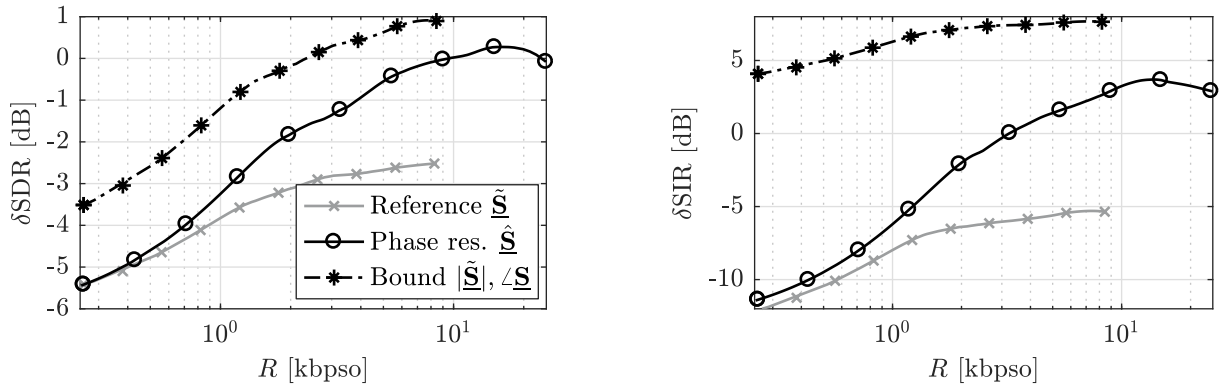
The experiments are structured as follows. In Section 6.4.1, the transmission of the phase residual is evaluated. In addition to that, the impact of initializing an exemplary re-estimation algorithm with the refined source estimates is evaluated, too. In Section 6.4.2, a complex residual is transmitted and compared both to the transmission of a phase residual and to CISS.

6.4.1 Phase Residual

This section evaluates the transmission of a phase residual as discussed in Sections 6.1 and 6.2. The phase residual is quantized with λ defined above and $N_{\mathrm{q,res}} \in [2, 4, \ldots, 10]$. Several configurations are investigated:

- The phase residual $\boldsymbol{\Phi}$ is quantized with RDOQ in the encoder, transmitted and added to the phase of the Wiener filter estimates in the decoder.

- Refining magnitude and phase with CWF as summarized in Section 2.6.2 is activated in the decoder as a posterior step after Wiener filtering. Note that this step comes for free in terms of bit rate.



(a) R, δSDR curves.

(b) R, δSIR curves

Figure 6.4 Phase residual and theoretical upper bound with Wiener-estimated magnitude $\left|\underline{\tilde{\mathbf{S}}}\right|$ and phase of original sources $\angle\underline{\mathbf{S}}$, shown at rate points R obtained for reference.

- The Wiener filter estimates are refined with $\bar{\mathbf{\Phi}}$ and the resulting spectrograms are fed into CWF as initialization which is a combination of the two other cases above.

First, the phase residual is evaluated. In Figure 6.4, the performance of Wiener estimates $\underline{\tilde{\mathbf{S}}}$ is compared to the refined estimates $\underline{\hat{\mathbf{S}}}$ with the phase residual $\bar{\mathbf{\Phi}}$ and a theoretical upper bound: The magnitude of the Wiener estimates $\tilde{\mathbf{S}} = \left|\underline{\tilde{\mathbf{S}}}\right|$ is combined with the phase of the original sources $\angle\underline{\mathbf{S}}$. These are the optimum estimates for solely transmitting phase residuals. To construct hypothetical rate-quality curves, it is assumed that the corresponding quality measure is yielded at rates spent on the transmission of solely $\tilde{\mathbf{S}}$[3]. Two different quality measures are considered in Figure 6.4, namely δSDR and δSIR, measuring distortion and interference.

- When comparing the results for Wiener estimates $\underline{\tilde{\mathbf{S}}}$ (—×—) and refined estimates $\underline{\hat{\mathbf{S}}}$ (—○—), it becomes clear that transmitting the phase residual yields gains up to 2 dB in δSDR and more than 7 dB in δSIR at higher rates. For high bit rates, saturation is reached which could be overcome by increasing the number of reconstruction values $N_{\text{q,res}}$. At lower rates, the same quality is reached for both estimates.

- The refined estimates $\underline{\hat{\mathbf{S}}}$ (—○—) reach the quality of the upper bound (–✱·) for higher rates up to a margin of 1 dB for δSDR. For δSIR, the margin is larger, about 5 dB.

Figure 6.5 shows results for phase residual transmission in combination with the re-estimation algorithm CWF as detailed in Section 2.6.2.

- Enabling CWF (–+·) with the mixture's phase as initialization yields gains up to 1 dB in comparison to the Wiener estimates $\underline{\tilde{\mathbf{S}}}$ (—×—), with no extra parameters transmitted. The computational complexity of the decoder, however, is increased in this case.

- CWF initialized with the refined estimates (—□—) yields again gains up to 1 dB compared to the refined estimates (—○—) with the phase residual and outperforms the oracle estimator at high rates.

[3]Recall that the rate is spent on transmitting the model $\bar{\Theta}_{\mathbf{s}}$ which is used for Wiener filtering yielding $\tilde{\mathbf{S}}$.

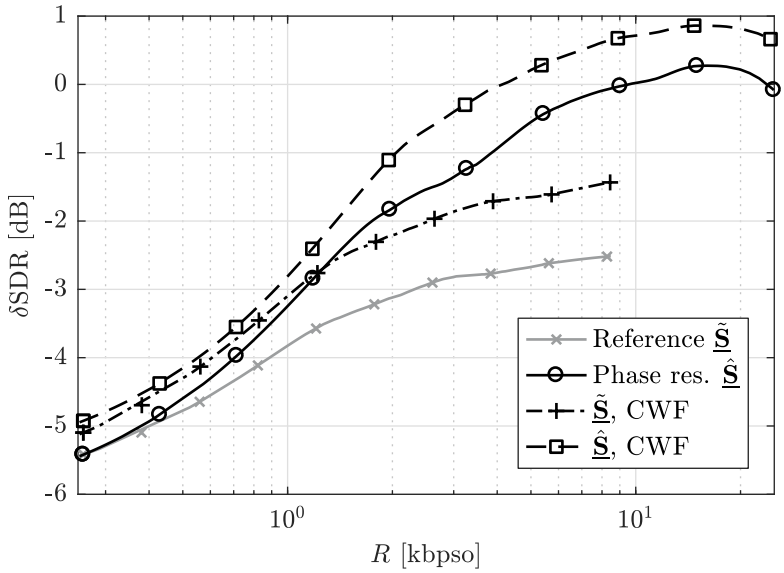


Figure 6.5 Phase residual transmission and re-estimation with CWF.

In summary, the transmission of phase residuals is able to enhance the separation quality noticeably. For δSIR, the residual transmission yields values which are even higher than the oracle performance ($\delta\text{SIR} > 0$). It is also shown that initializing with the refined estimates increases the quality of an exemplary reference re-estimation algorithm, CWF, further.

6.4.2 Complex Residual

In this section, transmitting the phase residual as already evaluated in Section 6.4.1 is compared to the transmission of complex or magnitude residuals as discussed in Section 6.3. Additionally, the complex residual is also compared to another baseline method, CISS, which is designed to operate at higher rates. For quantizing the CISS residual, the quantizer step size is evaluated as $\Delta \in \{0.5, \ldots, 10^4\}$. In fact, when evaluating CISS, the proposed complex residual calculation is replaced by the CISS residual calculation. Both residuals, the proposed and the CISS residual, refine the Wiener estimates $\underline{\tilde{\mathbf{S}}}$. To calculate the KLT matrices, the quantized source NTF parameters $\bar{\Theta}_{\mathbf{s}}$ are fed into CISS. $\bar{\Theta}_{\mathbf{s}}$ is also used for calculating the Wiener filter masks yielding $\underline{\tilde{\mathbf{S}}}$. Note that the IS distance ($\beta = 0$) is selected for CISS which is required by the probabilistic framework. All residuals are quantized with $N_{\text{q,res}} \in [2, 4, \ldots, 10]$.

Figure 6.6 shows rate-quality curves comparing the three different residual cases, transmission of a complex, a phase or a magnitude residual.

- Using the Wiener filter output as estimate (—×—) is clearly outperformed by refining the Wiener estimate with all considered residual types.

- The magnitude residual (—□—) outperforms the phase residual (–+–·) up to 1 dB for mid-range rates. For high rates, it is also outperforming the oracle estimator, again up to 1 dB.

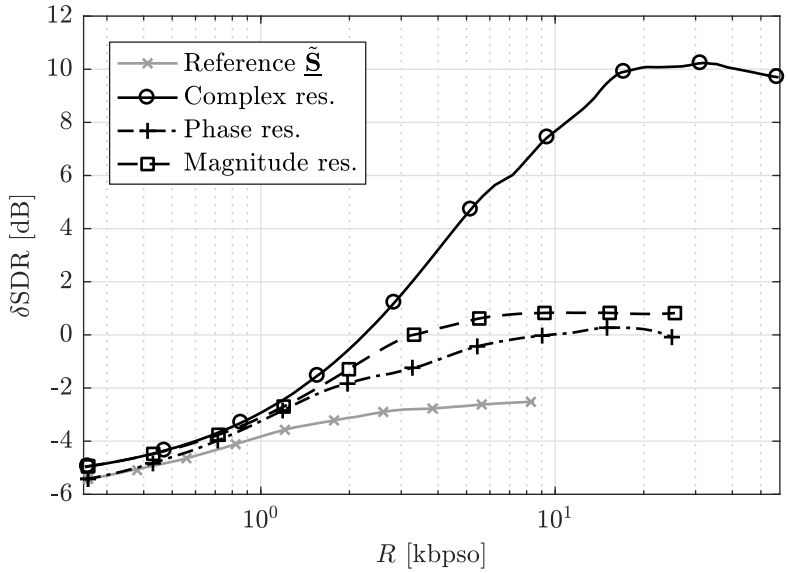


Figure 6.6 Comparison of different residuals.

- The complex residual (—○—) outperforms all other residuals significantly. Quantization of both residual magnitude and phase, using Equation (6.7) as joint distance measure, yields higher quality at same rates, although both magnitude and phase information needs to be transmitted in this case. The quantizer reaches saturation for rates higher than 20 kbpso which are not realistic for using ISS (cf. Section 7.2).

In the following experiment, the complex residual is compared to CISS. Here, the number of reconstruction values for the complex residual is extended to $N_{\text{q,res}} \in [2, 4, \ldots, 16]$. In Figure 6.7, the performances of transmitting the complex residual (○) or transmitting a transform-coded residual with CISS (✳) are compared.

- The proposed RDOQ method for quantizing the complex residual (○) is clearly outperforming the reference using solely Wiener filtering (×) as observed in the previous experiment.

- Setting $\lambda = 0$ (•), which corresponds to the case where only the distortion has impact on the overall criterion, yields worse results than using different values of λ. A BD-BR reduction of about -10% is yielded when jointly minimizing rate and distortion with Equation (6.3).

- Compared to CISS (✳), the proposed method is competitive at rates around 1 kpbso. However, CISS outperforms the proposed complex residual transmission at higher rates for all mixtures. This is expected as CISS exploits correlation of the source contributions to each TF point using the KLT. The proposed RDOQ-based method quantizes each TF point for each source independently and does not exploit statistical dependencies. The decoder complexity however is significantly higher for CISS compared to the proposed method since for each TF point a complex SVD has to be computed.

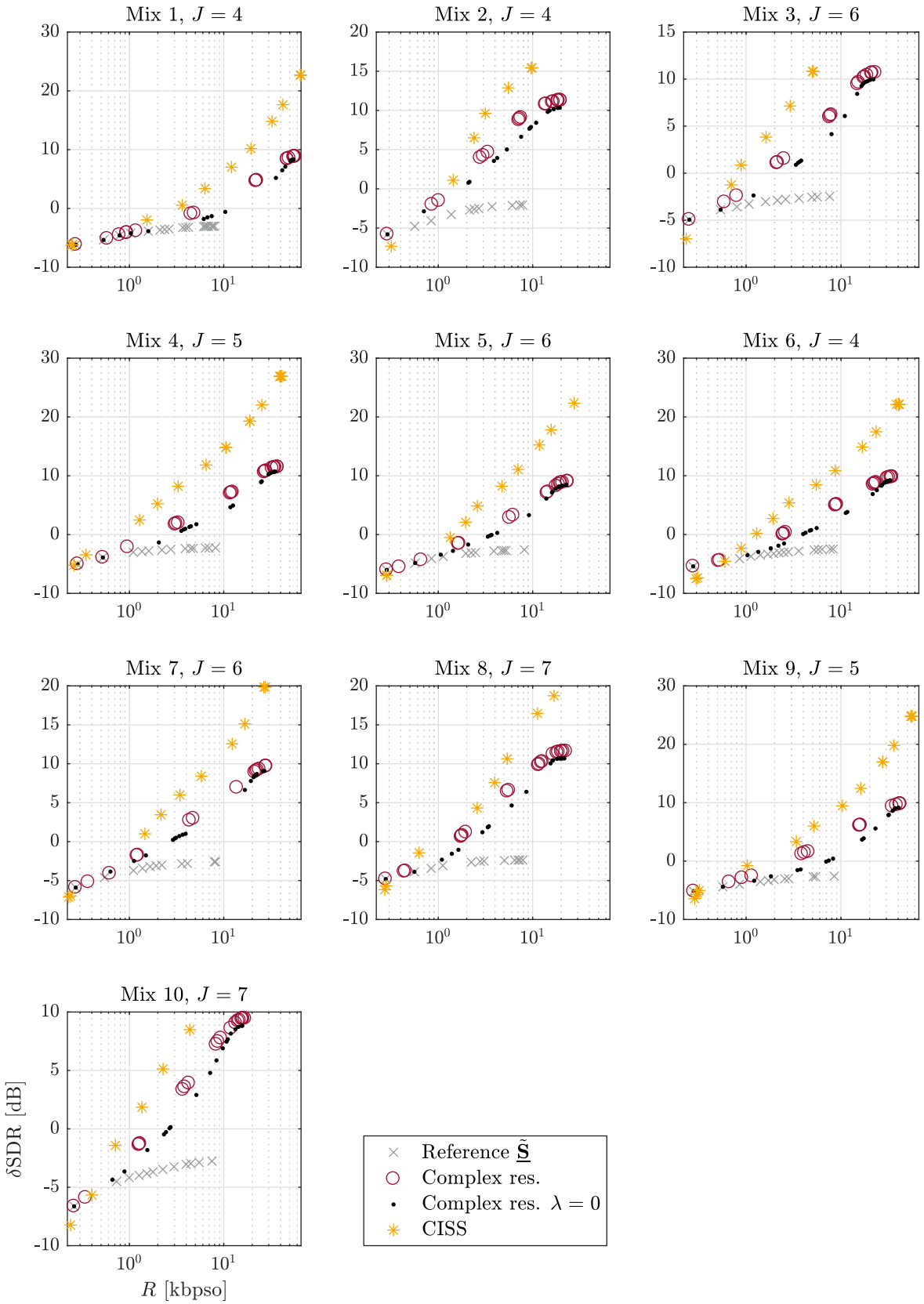


Figure 6.7 Comparison of complex residual with CISS.

6.5 Summary

In this chapter, residuals between the original sources available at the encoder and their estimates yielded by the decoder were examined. Since the Wiener filter always yields estimates containing the mixture's phase, the original idea was to indicate TF points with large deviations from the original source's phase to improve the quality of the Wiener estimates directly or enhance the performance of a re-estimation technique.

This phase residual was discussed in Section 6.1. Prior to transmission of the residual to the decoder, quantization is needed. Since this residual is in the TF domain, the corresponding bit rate has to be handled with care. Therefore, RDOQ was adapted for quantizing the residuals in Section 6.2. However, this procedure assumes a very good magnitude estimation which in practice is not the case. It was therefore proposed in Section 6.3 to calculate a complex residual instead, jointly accounting for phase and magnitude errors. The proposed residuals were finally evaluated in Section 6.4.

During evaluation, it became clear that transmitting phase residuals already enhances the quality. A generalization of the classical Wiener filter, CWF, was initialized with these refined source estimates. It was shown that the initialization has a strong influence on the performance of CWF as the refined initialization yielded better results. It became clear in the following experiment that transmitting not only phase residuals but also information about magnitude errors significantly increases the quality again.

This complex residual transmission was finally compared to the performance of CISS, also calculating residuals in the TF domain. It was shown that the proposed method yields similar performance at rates around 1 kbpso. For higher rates, CISS yielded higher quality by exploiting correlation between the sources. However, KLT computations for each TF point are needed in the decoder for this scheme. In contrast to CISS, the proposed method introduces no significant extra computational complexity to the decoder. It should be also mentioned here that the bit rates obtained by CISS are estimated by the underlying framework. To encode the proposed residuals, an implementation of an adaptive arithmetic coder was used yielding the actual bit rate needed for transmission.

7 Complete Algorithm

In this chapter, all contributions of this thesis are summarized and jointly evaluated. The contributions of Chapters 4 and 5, using CABAC for entropy coding and the SBSS algorithm for estimating parameters in the decoder, both operate at rather low bit rates and are therefore considered jointly as a low bit rate configuration. The proposed residual quantization and transmission process of Chapter 6 is considered as a high bit rate configuration instead. The ISS algorithm in this configuration is furthermore compared to SAOC, an MPEG standard for audio object coding. Finally, the influence of encoding the mixture with AAC on the subsequent ISS algorithm is evaluated and compared to the case where all sources are independently encoded with AAC.

In Section 7.1, an overview over all contributions is given. The proposed algorithm is compared to SAOC in Section 7.2. In Section 7.3, the influence of encoding the mixture with AAC is evaluated. All findings are summarized in Section 7.4.

7.1 Overview

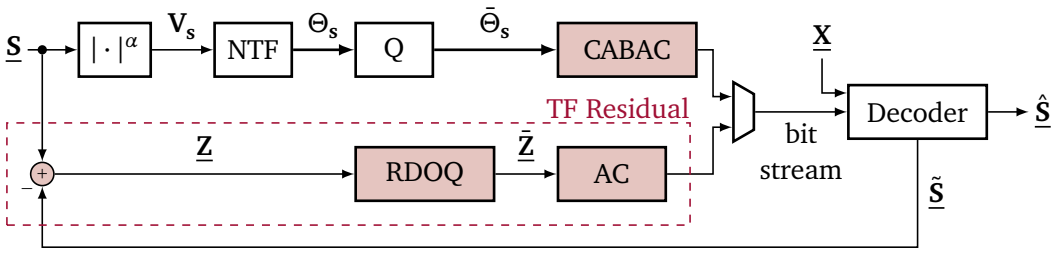
In Figure 7.1, the block diagrams of the proposed ISS encoder and decoder are displayed. The following modifications of the reference ISS algorithm, as discussed in Section 3.1, are proposed:

1. CABAC is used to code the quantized NTF source model $\bar{\Theta}_s$ losslessly as discussed in Chapter 4. CABAC is adapted to exploit local dependencies in the quantization symbols depending on the proposed context design.

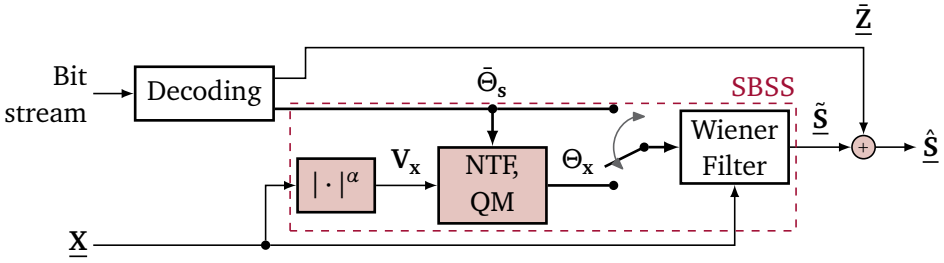
2. To enable lower bit rates, an NTF with the mixture as observation is added to the decoder. Depending on the configuration chosen by the encoder, the decoder NTF re-estimates missing parameters in the quantized model $\bar{\Theta}_s$. This procedure is denoted as Semi-blind Source Separation (SBSS) and summarized in Chapter 5.

3. A residual in the TF domain is calculated to correct possible errors in the source estimates given by the decoder as summarized in Chapter 6. It is proposed to use RDOQ for quantizing these residuals to constrain the rate necessary for transmitting the residuals. This modification does not introduce noticeable computational complexity to the decoder.

To summarize all contributions in experiments, cases 1 and 2 are considered jointly as a low bit rate configuration and evaluated in Section 7.1.1. Case 3 is evaluated as a higher rate configuration in Section 7.1.2.



(a) Proposed encoder. Consists of decoder.



(b) Proposed decoder.

Figure 7.1 Block diagrams of proposed ISS encoder and decoder in the TF domain.

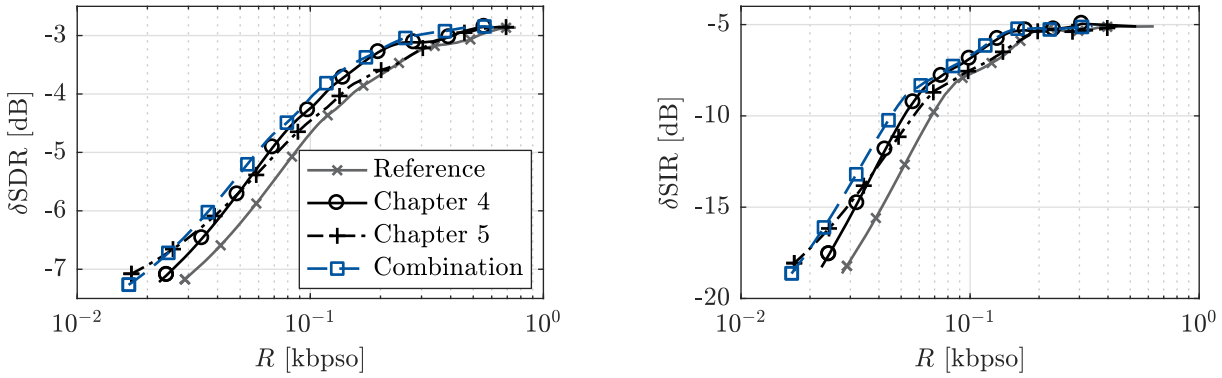
7.1.1 Low Bit Rate Scenario

First, the low bit rate configuration is evaluated in this section. Four different cases are evaluated:

- The reference algorithm which uses GZIP for entropy coding and solely Wiener filtering for source separation.
- The proposed variant of CABAC for entropy coding in combination with Wiener filtering as discussed in Chapter 4. CABAC is used with the bin-value based context $\mathrm{ctx}_{n,\mathrm{up1}}$; refer to Section 4.2.1 for more detail.
- The proposed SBSS algorithm in the decoder utilizing an NTF of the mixture. This extension is detailed in Chapter 5. The SBSS algorithm is tested in all configurations listed in Table 5.1 excluding the "blind" case. This means that the decoder NTF is either used for refining or re-estimating parameters or is skipped in the first place. GZIP is used for entropy coding in this case.
- A combination of both contributions: CABAC is used for coding the parameters which are refined/estimated by the SBSS algorithm in the decoder.

Figure 7.2 shows results for this lower bit rate configuration evaluated on all mixtures of test set $\mathscr{A}$ with both δSDR and δSIR as scores.

- As already found in Chapter 4, using CABAC (—o—) as entropy coder, which exploits the structure of the NTF parameters, consistently outperforms the reference (—×—) which uses GZIP as coder. GZIP was not specifically adapted to code the quantization indices of the NTF parameters.



(a) R,δSDR curves.

(b) R,δSIR curves.

Figure 7.2 Rate-quality curves for reference and proposed decoder with either GZIP or CABAC as entropy coder.

- The SBSS algorithm in the decoder (−+−·) as proposed in Chapter 5 enables lower bit rates than the reference. Excluding certain parameters from transmission and estimating them at the decoder also yields better performance at low bit rates smaller than 10^{-1} kbpso. For higher rates, the SBSS algorithm is skipped which means that only Wiener filtering is used in the decoder. Therefore, the proposed method yields the same performance as the reference towards rates of 1 kbpso.

- Combining the two contributions (−□−), namely CABAC for entropy coding and the SBSS algorithm in the decoder, yields the best performance. For lower rates, a similar performance as the SBSS algorithm detailed in Chapter 5 (−+−·) is reached. Only the grouping matrix has to be transmitted in both cases which is encoded with GZIP. At rates around 10^{-1} kbpso, the combination performs better than both contributions evaluated separately. At rates towards 1 kbpso, it reaches the same separation quality as using CABAC without the SBSS algorithm (—o—) because in this rate range, the SBSS algorithm is skipped (cf. Section 5.1.3.4).

7.1.2 High Bit Rate Extension

Chapter 6 dealt with the efficient quantization of complex residuals in the TF domain. In Section 6.4.2, the proposed extension was compared to another baseline, CISS, which also calculates residuals. Here, this comparison is not repeated but the best configuration of the low bit scenario in Section 7.1.1 is compared to enabling residual transmission. To limit the complexity of the experiments, the residual is calculated between original and estimated sources obtained by Wiener filtering using GZIP-encoded parameters.

In contrast to the experiments in Section 6.4.2 evaluating the residual, the quantizer setting for quantizing the NTF parameters is modified to match the low bit rate scenario: A-law companding is used with $A = 10$ and $N_\mathrm{q} = 16$ reconstruction values instead of quantizing with $N_\mathrm{q} = 2^8$ levels in the logarithmic domain. The number of components is set to $K/J \in \{1, 2, \ldots, 30\}$ in this case. Figure 7.3 displays the corresponding results for test set $\mathscr{A}$.

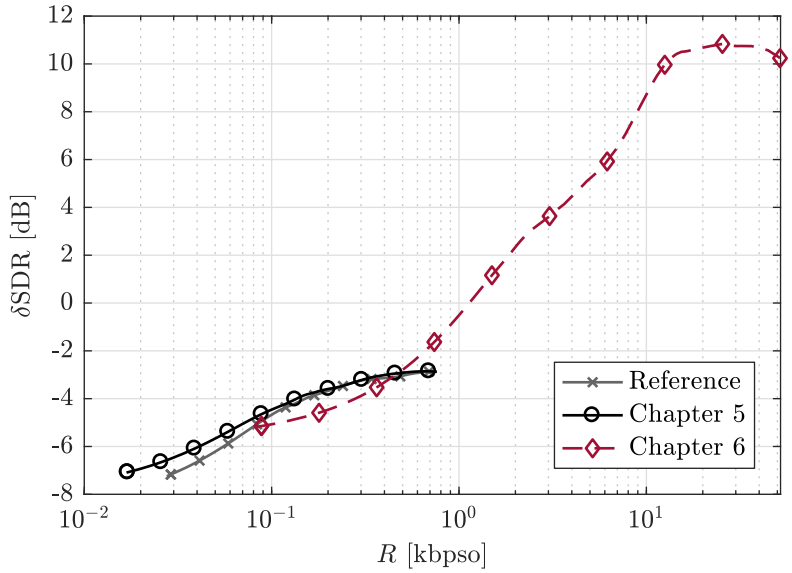


Figure 7.3 R,δSDR curves comparing the reference, the proposed SBSS without and with residual transmission.

- Transmitting the complex residual (—◇—) extends the rate range towards higher rates compared to the low bit scenario. For rates higher than 10 kbpso, the quantizer reaches saturation. This could potentially be resolved by using more reconstruction values. It should be noted however that the range for rates larger than 10 kbpso is not useful anymore for audio object coding. Separate coding of the sources with AAC yields better results as shown in Section 7.3.

- Note that both reference (—×—) and SBSS (—○—) are evaluated with different numbers of reconstruction values $N_q \in \{2, 3, 4, 8, 16\}$ than the residual transmission (—◇—), cf. Table 3.1. This is the reason why the residual transmission does not outperform the other two methods for rates smaller than 1 kbpso as the number of reconstruction values is fixed to $N_q = 16$ here.

In the following sections, the encoder is enabled to decide between the and the high bit rate configurations choosing the configuration following rate-quality optimization (cf. Section 2.8.2).

7.2 Comparison to SAOC

In this section, the proposed ISS method with enabled residual transmission is compared to SAOC [MPE10] which is briefly introduced in Section 2.7.2 and AAC. SAOC itself is able to transmit residuals, in this context called enhanced audio objects (EAO). In the implementation at hand, a maximum of $N_{eao} = 4$ objects can be assigned as EAO. The corresponding residuals are encoded using AAC-based techniques for waveform coding [Her+12].

SAOC is used in two parameter configurations aiming either at low bit rates with 32 time slots and 14 parameter bands or high bit rates with 16 time slots and 28 parameter bands. Additionally, EAO transmission is activated for a high bit rate scenario with $N_{eao} \in \{1, 2, 3, 4\}$.

All possible source-to-EAO assignments are tested for each mix yielding $\binom{J}{N_{\text{eao}}}$ combinations without repetition. For each mixture of test set $\mathscr{A}$, each combination of parameter configuration and EAO assignment is evaluated in the following. As another method to compare with, the sources of each mixture are encoded separately with AAC. Here, the sources are encoded with variable bit rate (VBR) using the quality parameter $q \in \{0.05, 0.15, \ldots, 0.95\}$. The two configurations of the proposed ISS method as detailed in Section 7.1 aim at either low bit rates or high bit rates. Here, they are jointly evaluated. The encoder can choose the configuration yielding the best rate-quality scenario. As already done in Section 7.1.2, CABAC is disabled.

Following the principles of Section 2.8.2, optimum rate-quality points are selected for both methods and displayed on Figures 7.4 and 7.5 with either SDR or SIR as score. Comparing SAOC to ISS, the following observations can be made:

- SAOC operates in a rate range of approximately $R \in [1\,\text{kbpso}, 10\,\text{kbpso}]$. The proposed ISS method yields smaller rates and outperforms SAOC at rates around 1 kbpso.

- For rates around 10 kbpso, SAOC and the proposed ISS method yield similar results. However, ISS outperforms SAOC for all rates for the mixtures with the highest number of $J = 7$ sources. The disadvantage of the SAOC implementation is the limited number of $N_{\text{eao}} = 4$ residuals whereas the proposed ISS method is able to transmit residuals for all J sources. The question remains if the user has need for more than four enhanced objects in the first place.

- At rates higher than 10 kbpso, SAOC yields better SDR and SIR than the proposed ISS method for mixtures with $J < 7$ sources. For mixtures with four or five sources, the resulting gains are especially large.

In the following, coding the sources independently with AAC is compared to SAOC and the proposed ISS method:

- AAC ($\cdot$) needs at least 10 kbpso for coding the sources. Both SAOC and the proposed ISS method enable lower rates. Note that the considered rate R only accounts for the necessary parameters to extract the source estimates out of the mixture which is assumed to be already present at the decoder in the cases of SAOC and ISS. In Section 7.3, this assumption is dropped and the ISS reference method is evaluated with an AAC-encoded mixture while measuring the rate which is necessary for both parameter and mixture transmission.

- AAC introduces less interference, measured by SIR, compared to both SAOC and the proposed ISS method at rates around 10 kbpso (and shown in Figure 7.5). The source separation step of the proposed ISS decoder or the upmixing process in the SAOC decoder could be responsible for this behavior. The influence of coding artifacts on the interference is not as high as the impact of the previously mentioned source extraction methods.

Note that the experiment conducted in this section is of preliminary nature. It can e.g. be expected that the performance of SAOC increases when handling stereo mixtures. The delays of both SAOC and the proposed ISS method should also be compared.

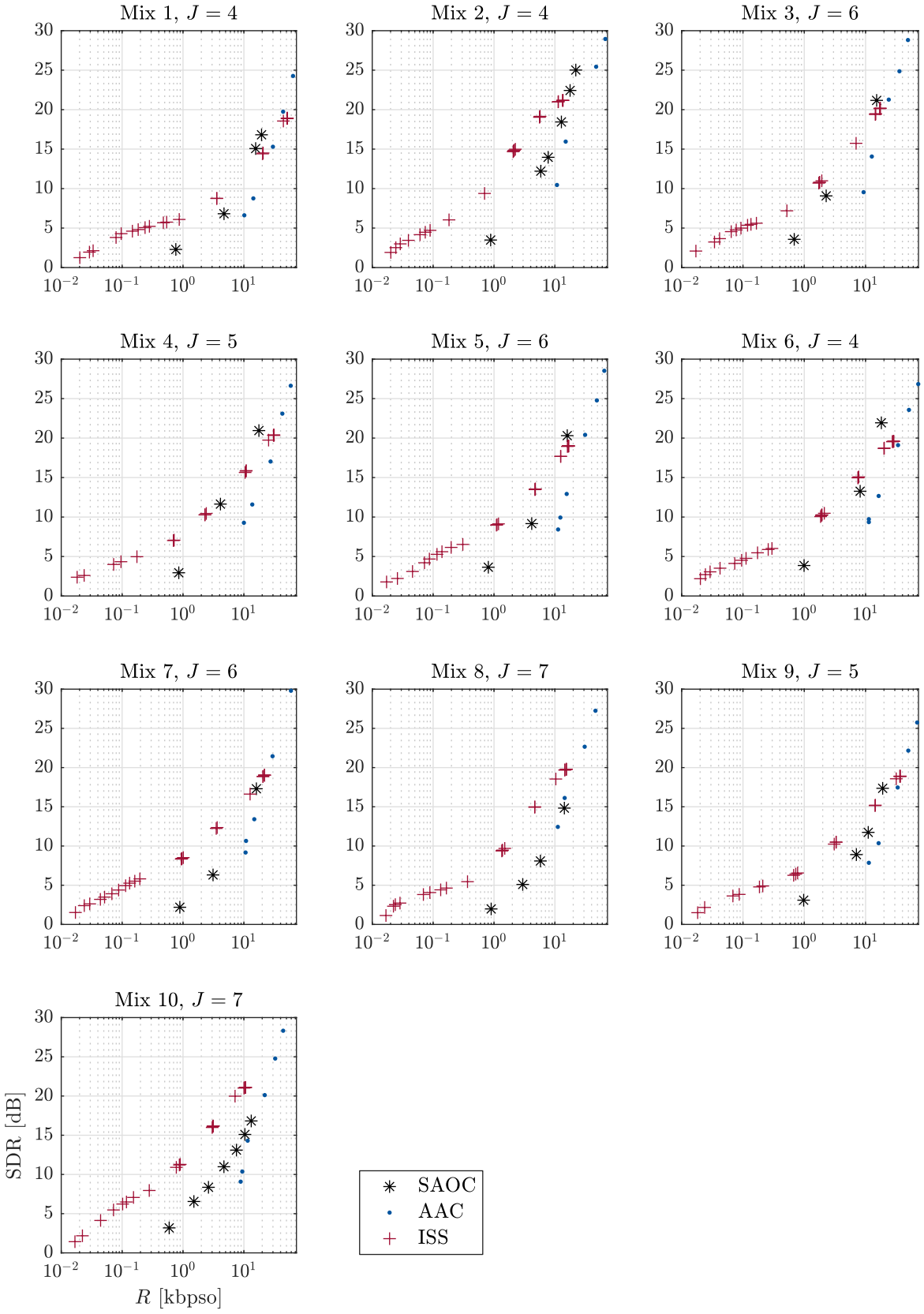


Figure 7.4 Comparison of SAOC, AAC and the proposed ISS method with SDR as score.

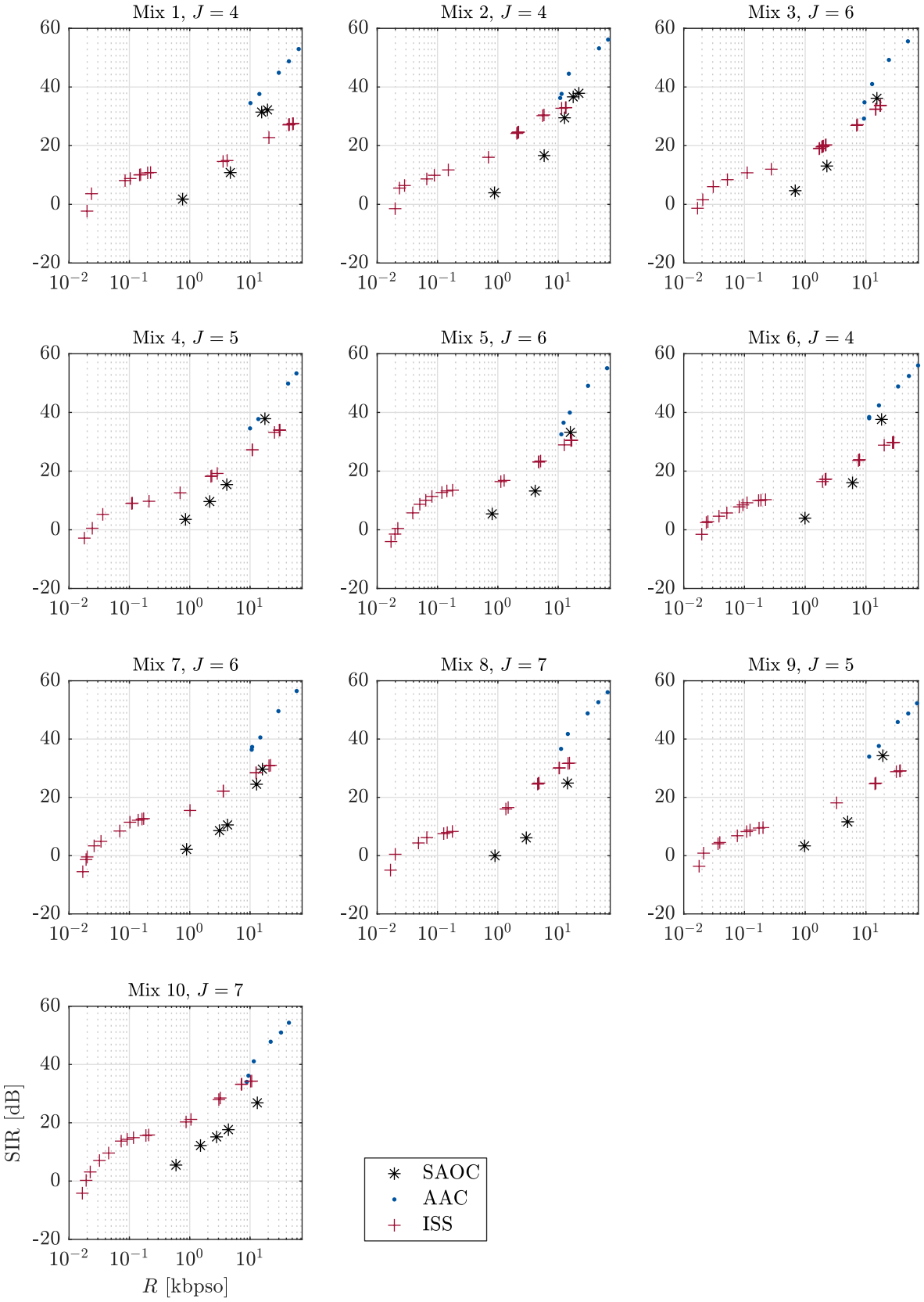


Figure 7.5 Comparison of SAOC, AAC and the proposed ISS method with SIR as score.

7.3 Influence of Lossy-Encoded Mixture

Throughout this thesis, all evaluations were computed under the assumption that the mix is available at high precision at the decoder. Embedding the ISS parameters in a PCM-encoded mix with data hiding techniques as assumed in [Liu+11] is a more realistic scenario. As the bit rates for the proposed extensions for both ISS encoder and decoder are even lower than the bit rates achieved for the reference method, the influence of the data hiding technique on the quality of the mix should decrease. In this section however, the influence of a lossy coded mix on the separation quality with respect to the estimated sources is evaluated. The mix is encoded with AAC and the ISS parameters are stored in the AAC metadata in this scenario. This means that backward compatibility is guaranteed as a standard AAC decoder is still able to decode the mix. In comparison to the data hiding scenario, the storage of the ISS parameters does not perturb the mix additionally.

In this section, the influence of coding the mixture on the ISS process is evaluated. In the following, two scenarios are considered.

1. It is assumed that the mixture was already transmitted to the decoder. Therefore, it is evaluated how much *extra rate* is necessary for extracting the sources out of the mixture to enable applications such as remixing. This is the most common scenario in audio object coding which was also assumed throughout this thesis.

2. For sake of completeness, it is evaluated how much *rate in total* is needed for transmitting the sources from the encoder to the decoder. This means that the rate considered here accounts for both coding the mixture and the ISS parameters.

The ISS performance is compared to the case where the sources are independently coded with AAC. The quality is independent of the mix rate as the information available in the mixture is not used for transmitting the sources in this case.

In the following, the reference ISS method discussed in Section 3.1 is considered. The chosen parameters are summarized in Table 3.1. In the decoder, solely Wiener filtering is used. Here, the only difference to the reference method is that the mixture is encoded with AAC prior to the TF transform. The mix is transmitted with rates $R_{\text{mix}} \in \{11\,\text{kbps}, 35\,\text{kbps}, 192\,\text{kbps}\}$. Note that $R_{\text{mix}} = 192\,\text{kbps}$ yields very similar ISS performance compared to the lossless case which is used as a reference throughout this thesis.

The corresponding results for case 1 are shown in Figure 7.6 for test set $\mathscr{A}$. Here, the parameter rate R is considered which is necessary to transmit the NTF parameters $\bar{\Theta}_{\mathbf{s}}$ for the ISS method or for direct transmission of the sources for AAC. This case is assumed in the other chapters of this thesis and most of the ISS literature. The rate is normalized per object, resulting in kbps/object. In the following, SDR is chosen as score. Since the performance of the ISS reference method is compared to AAC, which does not use Wiener filtering, the SDR values are not set into relation to the oracle scores here.

- Encoding the sources independently with AAC ($\cdot$) needs more bit rate than the ISS method, assuming that the mixture is already at hand at the decoder. The lowest rate achievable for AAC is around 10 kbpso.

- Increasing the mixture bit rate R_{mix} improves the quality. $R_{\text{mix}} = 35\,\text{kbps}$ ($\circ$) yields similar results as $R_{\text{mix}} = 192\,\text{kbps}$ ($\times$). As mentioned above, the latter bit rate yields the same results as the case assuming lossless coding of the mixture.

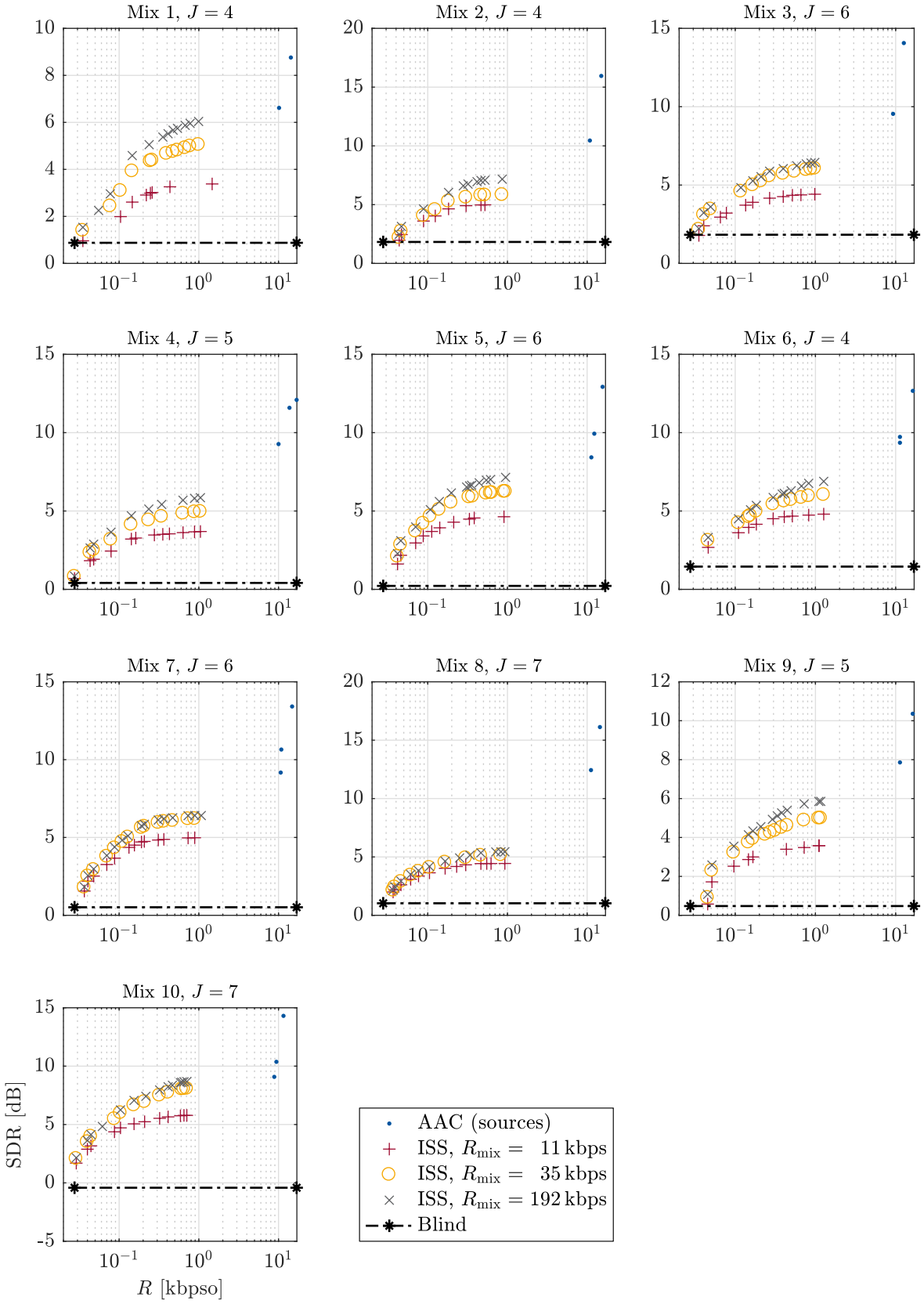
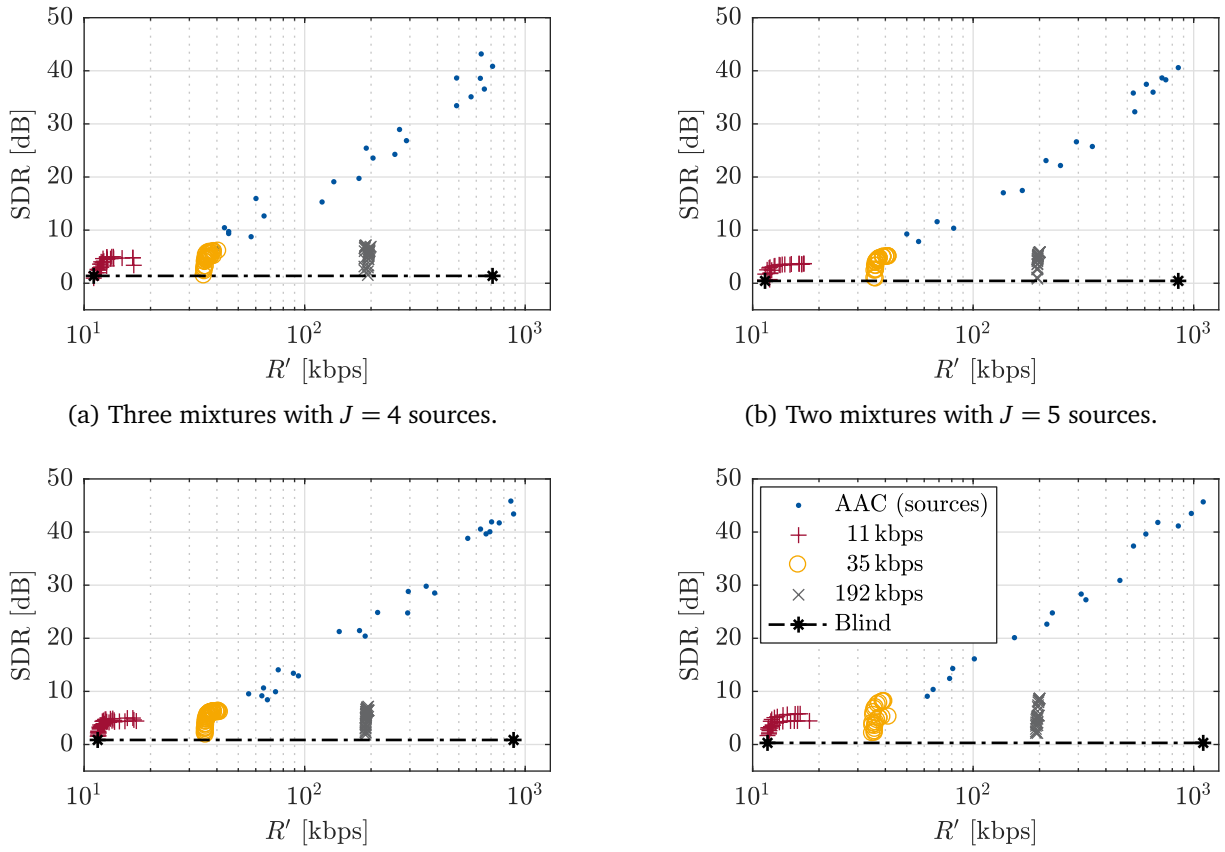


Figure 7.6 SDR and parameter rate R.



(a) Three mixtures with $J = 4$ sources.

(b) Two mixtures with $J = 5$ sources.

(c) Three mixtures with $J = 6$ sources.

(d) Two mixtures with $J = 7$ sources.

Figure 7.7 SDR and R' for encoding the sources. The sources were either independently encoded with AAC or with ISS. In the latter case, the mix has to be transmitted which is added to the parameter bit rate R as $R' = R_{\text{mix}} + JR$. AAC does not require the transmission of the mixture.

In the following, case 2 is considered. Again, sources encoded separately with AAC are compared to ISS. In contrast to case 1, it is evaluated here how much rate in total is needed to fully transmit the sources, accounting for the transmission of both the mixture and the parameters. This means in this case that the total rate for ISS is calculated as $R' = R_{\text{mix}} + JR$ since the parameter bit rate R is normalized with respect to the number of sources J. For AAC, R' amounts to the rate for encoding the sources *excluding* the mix rate R_{mix}.

Figure 7.7 shows the corresponding results which are given for mixtures with the same number of J sources of test set $\mathscr{A}$. For ISS, the mixture is again transmitted with rate $R_{\text{mix}} \in \{11\,\text{kbps}, 35\,\text{kbps}, 192\,\text{kbps}\}$.

- The highest mixture rate $R_{\text{mix}} = 192\,\text{kbps}$ ($\times$) is consistently outperformed by AAC encoding the sources ($\cdot$). The transmission of the mix yields in a constant shift of rate by R_{mix}: The highest value of the parameter bit rate which is reached for the reference ISS method is $R \approx 1\,\text{kbpso}$ as depicted in Figure 7.6. For J sources, the maximum rate consumed by the parameters for extracting the sources therefore yields J kbpso. Compared to $R_{\text{mix}} = 192\,\text{kbps}$, this amount is almost negligible for mixtures with up to $J = 7$ sources. The resulting separation quality is worse than coding the

sources independently with AAC where for the same rate an increase of SDR by 10 dB is yielded. However, it can be expected that transmitting additional residuals in the KLT domain, as done by CISS, could improve the quality of the reference ISS method (cf. the results in Section 6.4.2).

- For $R_{\text{mix}} = 35$ kbps (○), the rate range of AAC is extended towards lower values while obtaining the same or slightly worse quality. The obtained rate-quality points extend the rate-quality curve of AAC for lower rates.
- Using $R_{\text{mix}} = 11$ kbps (+) for compressing the mixture yields the lowest bit rates for ISS. As a lower bound, the performance of the BSS algorithm (−✱−·) is given as well. As already discussed in the summary of Chapter 5, it should be evaluated whether the qualities obtained at these low rates are useful in practice.

7.4 Summary

In this section, all contributions of this thesis detailed in Chapters 4, 5 and 6 were jointly evaluated. The contributions of Chapters 4 and 5 both aim at lower bit rates and were combined to a low bit rate configuration. It was shown that using CABAC as entropy coder in combination with the decoder SBSS algorithm in the decoder yielded the best separation results at rates lower than 1 kbpso. However it is still an open question, if operating at rates around 10^{-1} kbpso is useful in practice.

Next, the complex residual transmission as proposed in Chapter 6 was considered as a high bit rate configuration. It became clear in experiments that this configuration operates at rates between 1 kbpso and 10 kpbso. In another experiment, this configuration was compared to SAOC and AAC. The proposed method outperformed SAOC at rates around 1 kbpso. At rates towards 10 kbpso, both methods yielded similar qualities. Only for mixtures with a high number of sources, the limitation of the SAOC implementation of transmitting residuals accounting for a maximum of four objects resulted in reduced qualities compared to ISS for all bit rates. At rates higher than 10 kbpso, SAOC yielded better separation quality than the proposed method. However, encoding the sources independently with AAC was performing better than the other two schemes in this rate range. It should be noted that this experiment was of preliminary nature as only mono mixtures and objective quality scores were considered.

Finally, the influence of encoding the mixture with AAC on the subsequent ISS process was evaluated. It was shown that transmitting the mono mix at rates around 35 kbps yielded a performance comparable to the cases where the mixture was either encoded losslessly with PCM or with AAC at 192 kbpso. When accounting for the total rate needed for the transmission of both mixture and ISS parameters, transmitting the mixture with 35 kbpso enabled lower rates while preserving the quality compared to the separate encoding of the sources with AAC.

8 Conclusions and Outlook

8.1 Conclusions

Audio object coding has recently raised interest in both audio coding and audio source separation communities. In the field of source separation, Informed Source Separation (ISS) extracts side information from the original sources in the encoder. At the decoder, the side information assists a source separation step separating the sources out of the mixture. Many state-of-the-art algorithms use Nonnegative Tensor Factorization (NTF) as an extraction step allowing efficient compression of the sources and Wiener filtering as the source separation step in the decoder. In addition to this basic setup, methods of e.g. source coding or compressive sensing are used in the literature. This thesis deals with three extensions of a reference ISS algorithm using NTF at the encoder.

The first contribution introduces an efficient entropy coding method to ISS. Context-based Adaptive Binary Arithmetic Coding (CABAC), usually used in the field of video coding, is adapted to code the NTF parameters. Two context models are proposed describing local statistics within these parameters. These sets are evaluated both theoretically and experimentally. Using the set which both gives good coding performance while being reasonably compact, it is shown that CABAC is able to outperform all previously considered entropy coders in experiments over a large test set. The highest gains were obtained at rates between 10^{-1} kbps/object and 1 kbps/object. It can be concluded that exploiting the structure of the NTF parameters is beneficial for coding their quantized versions.

The second contribution deals with the fact that state-of-the-art ISS methods solely use Wiener filtering as the source separation step in the decoder. It is proposed to use an algorithm originally designed for the problem of Blind Source Separation (BSS) and adapt it for usage in the ISS decoder. It is shown that the proposed approach, denoted as Semi-blind Source Separation (SBSS), enables lower bit rates while introducing only small additional computational complexity to the decoder. Several configurations are proposed coping with different scenarios with respect to the amount of transmitted parameters. While testing the decoder, the encoder can omit certain parameters which are then estimated by the SBSS algorithm in the decoder. These different configurations were thoroughly evaluated and compared to the reference algorithm. The corresponding results confirm that the separation quality for low bit rates (smaller than 10^{-1} kbps/object) is increased. The SBSS algorithm is also able to operate completely blind as the original structure of the BSS algorithm is preserved. However, it could not be finally concluded if the separation quality obtained by both the proposed SBSS algorithm and the reference at very low rates is sufficiently high for a practical scenario. In summary, it was shown that replacing the Wiener filter by a full source separation algorithm enhances the separation quality at lower rates if the decoder has some available computing resources without causing noticeably higher delay.

The third contribution tackles possible errors between original sources available at the encoder and estimated sources obtained by the decoder by calculating residuals in the TF

domain. Starting with the idea to only deal with phase errors of the Wiener filter, it was proposed to signal TF points where the phase estimate significantly diverges from the original source's phase. Rate-distortion Optimized Quantization (RDOQ) is used for quantizing these residuals to constrain the bit rate. It was shown that this procedure already yields gains in quality compared to the reference method. As a generalization, the transmission of complex residuals was considered which were quantized with RDOQ as well. It was shown in experiments that these residuals perform better than the phase residuals at medium and high rate ranges. When compared to Coding-based Informed Source Separation (CISS), which uses source coding methods for calculating complex residuals, it is shown that the proposed method is competitive at rates around 1 kbps/object. For higher rates towards 10 kbps/object, CISS performs better as it exploits correlation of the sources. However, CISS introduces high computational complexity to the decoder whereas the additional complexity of the proposed RDOQ-based method is negligibly small.

Finally, these contributions were considered jointly. While the combination of CABAC as entropy coder and the SBSS algorithm in the decoder enable lower bit rates, the proposed residual transmission scheme was compared to an implementation of Spatial Audio Object Coding (SAOC), an MPEG standard for audio object coding. It became clear that in the experimental setup at hand, the proposed ISS method yields better results at lower rates towards 1 kbps/object. Both schemes yield similar results at rates around 10 kbps/object and SAOC is able to outperform the proposed method at rates higher than 10 kbps/object.

In summary, by exploiting the typical structure of the NTF parameters for efficiently coding them to a bit stream in the encoder and for estimating missing parameters at the decoder, the methods proposed in this thesis enable very low bit rates for audio object coding. By introducing rate-distortion optimization, transmission of residuals between the original sources and their estimation at the decoder can further improve the compression towards higher rates while introducing no extra computational complexity to the decoder.

8.2 Outlook

Novel extensions for NTF-based ISS were proposed in this thesis. Possible future work on each extension is motivated in the following.

With respect to enhanced entropy coding with CABAC, not only the NTF parameters but also other information at hand at the decoder, for example gathered from the mixture, could be considered for the context model selection. The information whether certain frequency bands or time frames are completely inactive could already be used for designing more advanced context models. Another interesting extension would be the usage of linear prediction methods prior to CABAC to increase its performance at higher bit rates. CABAC could also be used for encoding the residuals in the TF domain.

Regarding the SBSS algorithm in the decoder, an open problem is the performance at very low rates which does not yield sufficient quality for all types of sources and for all foreseen applications of audio object coding. Part of the problem is the performance of the BSS algorithm which was originally evaluated on mixtures consisting of only two sources. Preliminary experiments in this thesis show that the algorithm is not able to estimate more sources very well. For example, enabling additional constraints such as sparseness or temporal continuity could already improve the separation. Other BSS algorithms could be adapted in the same

manner, e.g. variants of the NTF describing each acoustical event with matrices instead of vectors. The proposed quantized-matching constraint only considers quantized NTF parameters. The constraint could be adapted for the case where RDOQ is used for quantizing the NTF parameters. RDOQ is minimizing a criterion jointly accounting for distortion and rate which leads to cases where it is beneficial to assign another, neighboring reconstruction value to reduce the rate necessary for transmission. In these cases, the quantized-matching constraint would force the parameters towards these reconstruction values which are sub-optimal by means of distortion. To overcome this problem, the RDOQ process could be matched in the decoder by constraining the entropy of the estimated parameters. Furthermore, it is still an open question if additional transmitted information, such as features calculated on the NTF parameters describing the sources in the encoder, could steer the decoder NTF towards better factorization results.

Regarding the residual transmission, RDOQ could be used for quantizing residuals in the KLT domain. In this case, the dependencies of the sources given the mixture as observation would be exploited in the same way as done by CISS. However, the resulting quantized residuals could be coded with an actual arithmetic coding step as done in this thesis instead of estimating the bit rate. However, redundancies of residuals per TF-point could be exploited with simpler methods as well. Another possible extension could be the usage of more sophisticated quantizers designed especially for quantizing complex values, e.g. the methods proposed in [RD07].

The basic ISS algorithm considered here uses Wiener filtering at the decoder which yields source estimates containing the mixture's phase. Apart from residual transmission, other methods could be used which extend Wiener filtering: Recent work [LRD18] shows promising results when using a novel probabilistic model describing the sources in the TF domain which could be adapted as well. In this thesis, the mixtures were assumed to be monaural. Algorithms for adapting the used methods for stereo, e.g. [LBR13], could be implemented and adapted to exploit spatial information. Here, a professional mixing process, allowing for time delays and effects, is modeled by a probabilistic framework for NTF-based ISS. The sources are assumed to have spatial spread which may vary over frequency. It could be interesting to investigate if methods like binaural cue coding [BF03] could be used for this task as well. Extensions of the NTF including perceptual modeling were used in e.g. [Kır+14; Nik15]. A parameter weighting each TF point of the sources by perceptually motivated scores was additionally used for NTF which could also be included in the proposed framework.

In this thesis, SAOC was compared to the proposed ISS method in a limited setting. For a more thorough evaluation, SAOC should be tested on mono and stereo mixtures as SAOC is able to exploit spatial information present in the two stereo channels. The complexity of these methods should also be compared. With applications such as active listening and spatialization in mind, subjective listening tests could be carried out as well to evaluate the quality of both schemes.

A Test Sets

A.1 Test Set $\mathscr{A}$

Test set $\mathscr{A}$ consists of ten mixtures each composed of four to seven sources of the QUASI database[1]. Each recording is sampled at 44100 Hz, quantized with 32 bit per sample and is 30 s long. More detail, namely interpret, song title, segment information (start and end time), number and names of sources, as well as the oracle performance measured in SDR is given in Table A.1.

A.2 Test Set $\mathscr{B}$

Test set $\mathscr{B}$ consists of 100 mixtures, each composed of four sources (bass, drums, vocals, other) of the DSD100 database[2]. We cropped 30 s long segments out of each recording, each sampled at 44100 Hz and quantized with 16 bit per sample. The segment start and end times for each song is given in Table A.2.

[1] `http://www.tsi.telecom-paristech.fr/aao/en/2012/03/12/quasi/`.
[2] "MUS 2016" task, `http://sisec.inria.fr`.

ID	Interpret – Song title	Start – End [s]	J	Sources	SDR_{ora} [dB]
1	Alexq – Carol of the bells	0 – 30	4	bass, drums, lead gtr, rhythm gtr	9.27
2	Another Dreamer – One we love	69 – 99	4	bass, drums, gtr, voc	9.38
3	Carl Leth – The world is under attack	44 – 74	6	drums, e-gtr, piano, piano loop, speech, synth	8.99
4	Fort Minor – Remember the name	54 – 84	5	bass, drums, samples, strings, voc	8.18
5	Glen Philips – The spirit of shackleton	163 – 193	6	ac. gtr, bass, drums, voc, organ, pads	9.77
6	Jims Big Ego – Mix tape	22 – 55	4	bass, drums, gtr, voc	9.42
7	Nine Inch Nails – Good soldier	104 – 134	6	bass, drums, gtr, keys, lead voc, vibes	9.16
8	Shannon Hurley - Sunrise	62 – 92	7	ac. gtr, bass, cell, drums, e-gtr, piano, voc	7.86
9	Ultimate NZ Tour	43 – 73	5	bass, drums, gtr, synth, voc	8.40
10	Vieux Farka - Ana	120 – 150	7	bass, claves, drums, gtr, organ, voc, wind	11.65

Table A.1 Test set $\mathscr{A}$.

ID	Interpret – Song title	Start – End [s]	ID	Interpret – Song title	Start – End [s]
1	ANiMAL – Clinic A	51 – 81	51	AM Contra – Heart Peripheral	165 – 195
2	ANiMAL – Rockshow	34 – 64	52	ANiMAL – Easy Tiger	140 – 170
3	Actions – One Minute Smile	82 – 112	53	Actions – Devil's Words	162 – 192
4	Al James – Schoolboy Facination	144 – 174	54	Actions – South Of The Water	89 – 119
5	Angela Thomas Wade – Milk Cow Blues	40 – 70	55	Angels In Amplifiers – I'm Alright	144 – 174
6	Atlantis Bound – It Was My Fault For Waiting	216 – 246	56	Arise – Run Run Run	86 – 116
7	BKS – Too Much	152 – 182	57	BKS – Bulldozer	82 – 112
8	Bill Chudziak – Children Of No–one	88 – 118	58	Ben Carrigan – We'll Talk About It All Tonight	59 – 89
9	Bobby Nobody – Stitch Up	165 – 195	59	Black Bloc – If You Want Success	295 – 325
10	Carlos Gonzalez – A Place For Us	183 – 213	60	Buitraker – Revo X	181 – 211
11	Cnoc An Tursa – Bannockburn	40 – 70	61	Chris Durban – Celebrate	247 – 277
12	Dark Ride – Burning Bridges	193 – 223	62	Cristina Vane – So Easy	155 – 185
13	Drumtracks – Ghost Bitch	94 – 124	63	Detsky Sad – Walkie Talkie	107 – 137
14	Fergessen – Back From The Start	33 – 63	64	Enda Reilly – Cur An Long Ag Seol	132 – 162
15	Fergessen – The Wind	139 – 169	65	Fergessen – Nos Palpitants	90 – 120
16	Forkupines – Semantics	104 – 134	66	Flags – 54	107 – 137
17	Girls Under Glass – We Feel Alright	164 – 194	67	Georgia Wonder – Siren	170 – 200
18	Hollow Ground – Ill Fate	66 – 96	68	Giselle – Moss	149 – 179
19	James Elder & Mark M Thompson – The English Actor	148 – 178	69	Hollow Ground – Left Blind	113 – 143
20	James May – Dont Let Go	52 – 82	70	James May – All Souls Moon	106 – 136
21	James May – On The Line	61 – 91	71	James May – If You Say	67 – 97
22	Johnny Lokke – Promises & Lies	183 – 213	72	Jay Menon – Through My Eyes	93 – 123
23	Jokers, Jacks & Kings – Sea Of Leaves	150 – 180	73	Johnny Lokke – Whisper To A Scream	94 – 124
24	Leaf – Come Around	191 – 221	74	Juliet's Rescue – Heartbeats	106 – 136
25	Leaf – Wicked	109 – 139	75	Leaf – Summerghost	190 – 220
26	Louis Cressy Band – Good Time	188 – 218	76	Little Chicago's Finest – My Own	241 – 271
27	M.E.R.C. Music – Knockout	212 – 242	77	Lyndsey Ollard – Catching Up	134 – 164
28	Motor Tapes – Shore	176 – 206	78	Moosmusic – Big Dummy Shake	29 – 59
29	Nerve 9 – Pray For The Rain	120 – 150	79	Mu – Too Bright	19 – 49
30	Patrick Talbot – A Reason To Leave	167 – 197	80	North To Alaska – All The Same	98 – 128
31	Phre The Eon – Everybody's Falling Apart	102 – 132	81	Patrick Talbot – Set Me Free	159 – 189
32	Raft Monk – Tiring	147 – 177	82	Punkdisco – Oral Hygiene	121 – 151
33	Sambasevam Shanmugam – Kaathaadi	114 – 144	83	Remember December – C U Next Time	187 – 217
34	Secretariat – Over The Top	91 – 121	84	Secretariat – Borderline	114 – 144
35	Signe Jakobsen – What Have You Done To Me	46 – 76	85	Side Effects Project – Sing With Me	116 – 146
36	Skelpolu – Resurrection	298 – 328	86	Skelpolu – Human Mistakes	264 – 294
37	Speak Softly – Broken Man	39 – 69	87	Skelpolu – Together Alone	67 – 97
38	Spike Mullings – Mike's Sulking	94 – 124	88	Speak Softly – Like Horses	260 – 290
39	Swinging Steaks – Lost My Way	116 – 146	89	St Vitus – Word Gets Around	140 – 170
40	The Long Wait – Back Home To Blue	161 – 191	90	The Doppler Shift – Atrophy	79 – 109
41	The Mountaineering Club – Mallory	187 – 217	91	The Long Wait – Dark Horses	45 – 75
42	The Wrong'Uns – Rothko	71 – 101	92	The Sunshine Garcia Band – For I Am The Moon	45 – 75
43	Timboz – Pony	54 – 84	93	Tim Taler – Stalker	42 – 72
44	Tom McKenzie – Directions	83 – 113	94	Titanium – Haunted Age	124 – 154
45	Traffic Experiment – Sirens	206 – 236	95	Traffic Experiment – Once More (With Feeling)	189 – 219
46	Triviul – Dorothy	73 – 103	96	Triviul – Angelsaint	192 – 222
47	Voelund – Comfort Lives In Belief	95 – 125	97	Triviul feat. The Fiend – Widow	185 – 215
48	We Fell From The Sky – Not You	30 – 60	98	Wall Of Death – Femme	13 – 43
49	Young Griffo – Facade	106 – 136	99	Young Griffo – Blood To Bone	184 – 214
50	Zeno – Signs	62 – 92	100	Young Griffo – Pennies	146 – 176

Table A.2 Test set $\mathscr{B}$ with $J = 4$ (bass, drums, vocals, other).

B Quantized Matching Derivations

The quantized matching constraint as discussed in Section 5.2 uses the A-law companding algorithm or LM for quantization. In [RLB17], Equation (3.3) was originally used for quantization. In this section, the constraint gradient terms of the corresponding soft quantization parameters are given for this case. The constraint cost function is denoted as

$$d_{\beta'}\left(\bar{\mathbf{W}}_\mathbf{s} \mid \tilde{\mathbf{W}}_\mathbf{x}\right) = d_{\beta'}\left(\bar{\mathbf{W}}_\mathbf{s} \mid \exp\left(f\left[\log\left(\mathbf{W}_\mathbf{x}\right)\right]\right)\right).$$

Here, the gradient of the soft quantization parameter is written as

$$\begin{aligned}\nabla_{\mathbf{W}_\mathbf{x}} \tilde{\mathbf{W}}_\mathbf{x} &= \underbrace{\exp\left(f\left[\log \mathbf{W}_\mathbf{x}\right]\right)}_{=\tilde{\mathbf{W}}_\mathbf{x}} \cdot \frac{2\zeta}{d} f_0\left(\log \mathbf{W}_\mathbf{x}\right)\left[1 - f_0\left(\log \mathbf{W}_\mathbf{x}\right)\right] \cdot \frac{1}{\mathbf{W}_\mathbf{x}} \\ &= \frac{2\zeta}{d} f_0\left(\log \mathbf{W}_\mathbf{x}\right)\left[1 - f_0\left(\log \mathbf{W}_\mathbf{x}\right)\right] \cdot \frac{\tilde{\mathbf{W}}_\mathbf{x}}{\mathbf{W}_\mathbf{x}}\end{aligned}$$

with $f_0(z)$ defined in Equation (5.4). This yields the positive and negative gradient terms

$$\nabla^+_{\mathbf{W}_\mathbf{x}} \tilde{\mathbf{W}}_\mathbf{x} = \frac{2\zeta}{d} f_0\left(\log \mathbf{W}_\mathbf{x}\right) \cdot \frac{\tilde{\mathbf{W}}_\mathbf{x}}{\mathbf{W}_\mathbf{x}}, \qquad \nabla^-_{\mathbf{W}_\mathbf{x}} \tilde{\mathbf{W}}_\mathbf{x} = \frac{2\zeta}{d} f_0^2\left(\log \mathbf{W}_\mathbf{x}\right) \cdot \frac{\tilde{\mathbf{W}}_\mathbf{x}}{\mathbf{W}_\mathbf{x}}. \tag{B.1}$$

C CABAC

C.1 CABAC Context Identifiers

The CABAC engine discriminates the different context models by identifiers. These identifiers are given in Tables C.1a and C.1b for sake of completeness for the context models discussed in Chapter 4. Table C.1a shows the context IDs for the bin-level context models proposed in Section 4.2.1 and Table C.1b for the integer-level context models proposed in Section 4.2.2.

Prefix				Suffix	
$\mathrm{ctx}_{n,\mathrm{na}}$	$\mathrm{ctx}_{n,\mathrm{up0}}$	$\mathrm{ctx}_{n,\mathrm{up1}}$	$\mathrm{ctx}_{\mathrm{rst}}$	$\mathrm{ctx}_{n,\mathrm{na}}$	$\mathrm{ctx}_{\mathrm{rst}}$
n	$N+n$	$2N+n$	$3N+1$	$3N+1+n$	$4N+2$

(a) Context model IDs for bin-level design with $n \leq N_{\mathrm{LBP}}$ and abbreviation $N = N_{\mathrm{LBP}}$.

Prefix		Suffix	
$\mathrm{ctx}_{n,\mathrm{iup}\nu}$	$\mathrm{ctx}_{\mathrm{rst}}$	$\mathrm{ctx}_{n,\mathrm{na}}$	$\mathrm{ctx}_{\mathrm{rst}}$
$(\nu-1)N_{\mathrm{LBP}}+n$	$N_{\mathrm{LBP}}N_{\mathrm{q}}+1$	$N_{\mathrm{LBP}}N_{\mathrm{q}}+1+n$	$\big(N_{\mathrm{q}}+1\big)N_{\mathrm{LBP}}+2$

(b) Context model IDs for integer-level design with $1 \leq n \leq N_{\mathrm{LBP}}$ and value of previously coded quantization index $\nu \in \big[1, N_{\mathrm{q}}\big]$.

Table C.1 Context model IDs for the proposed context designs.

C.2 Bin-value based Context Model Interpretation

The context models in Section 4.2.1 were derived on a bin-level. In this section, an interpretation of the conditional context model $\mathrm{ctx}_{n,\mathrm{up}\nu}$ on the integer-level is given, depending on the chosen binarization method.

Truncated Unary Binarization

When using TU coding as binarization, the resulting bin-strings consist of ones and are terminated with a single '0'. This means that conditional context model $\mathrm{ctx}_{n,\mathrm{up0}}$, which is chosen if previous bin was equal to '0', models *runs of identical values* in column $\mathbf{g}_{\bullet,k}$ of $\mathbf{G}$ which were all terminated with '0' at position n. Since TU uses $N_{\mathrm{C}} = g$ bins for encoding integer g and the last bin is always '0', the value of these sequences is given as

$$g_{f-1,k} = n. \tag{C.1}$$

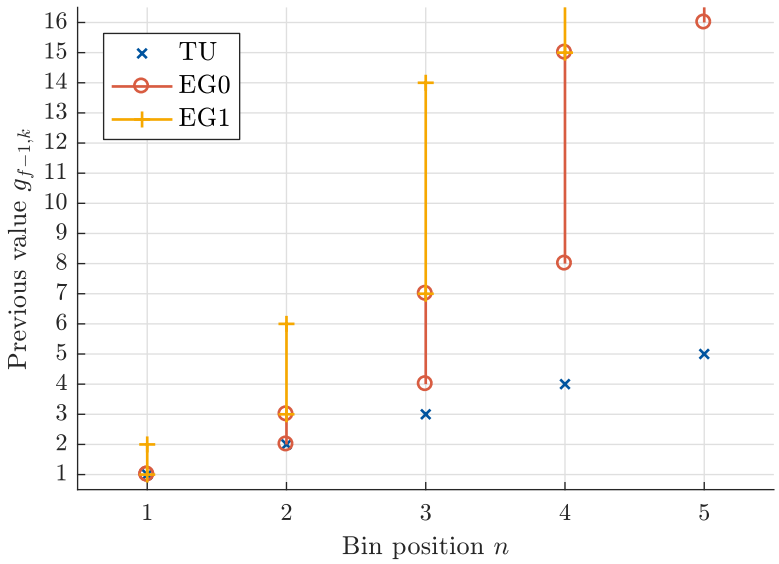


Figure C.1 Intervals of $g_{f-1,k}$ indicated by $\text{ctx}_{n,\text{up0}}$ depending on bin position n for TU and EG binarizations.

Especially if $n = 1$, $\text{ctx}_{1,\text{up0}}$ models a sequence of quantization indices being one (which maps to reconstruction value zero).

$\text{ctx}_{n,\text{up1}}$ can be interpreted as follows: At position $n = 1$, the context model indicates that the quantization index $g_{f-1,k}$ corresponding to the previously decoded bin-string $\mathbf{b}^{f-1,k}$ is greater than 1 because 1 is mapped to a single '0' and all values larger than 1 have a '1' at position $n = 1$. For $n = 2$, context model $\text{ctx}_{2,\text{up1}}$ indicates that $g_{f-1,k}$ is greater than two and so on. In summary, $\text{ctx}_{n,\text{up1}}$ models a sequence of values in $\mathbf{g}_{\bullet,k}$ with

$$g_{f-1,k} \geq n + 1.$$

Exponential Golomb Binarization

For EG binarization, a similar interpretation to the one given for TU binarization is possible, since the prefix of an EG code is a unary code. This code now gives information about the number of suffix bins. Using the interpretation for TU binarization above, $\text{ctx}_{n,\text{up0}}$ gives now information about the *number of suffix bins* which then leads to information about $g_{f-1,k}$. First, if $b_n^{f-1,k} = 0$, the number of prefix bins for coding $g_{f-1,k}$ has to be $N_{\text{pre}} = n$ (cf. Equation (C.1)). Inserting this into Equation (2.31) yields

$$2^l \left(2^{n-1} - 1\right) + 1 \leq g_{f-1,k} \leq 2^l \left(2^n - 1\right) \tag{C.2}$$

which means that $\text{ctx}_{n,\text{up0}}$ models runs of values lying in the interval defined above. Figure C.1 shows these intervals for EG0 and EG1 for bin positions $n \in [1, 5]$ compared to the values indicated by $\text{ctx}_{n,\text{up0}}$ when using TU binarization.

$\text{ctx}_{n,\text{up1}}$ still models runs of values being greater or equal than a certain value, namely the upper boundary of the interval defined in Equation (C.2),

$$g_{f-1,k} \geq 2^l \left(2^n - 1\right) + 1.$$

Method	GBAC	CABAC			
Cond. Ctx.	–	–	$\text{ctx}_{n,\text{up0}}$	$\text{ctx}_{n,\text{up1}}$	$\text{ctx}_{n,\text{up0}}$, $\text{ctx}_{n,\text{up1}}$
BD-BR, %	3.39	−8.99	−20.99	**−22.62**	−22.17
Mean saving, %	−4.42	−10.95	−21.47	**−23.21**	−22.96
Std. saving, %	10.76	4.33	3.55	3.73	3.84

Table C.2 Results for bin-level context models. BD-BR with respect to GZIP encoding parameters jointly for GBAC ($N_{\text{LBP}} = 0$) and CABAC ($N_{\text{LBP}} = 5$) for test set $\mathscr{A}$.

Method	CABAC	
Cond. Ctx.	Bin-level $\text{ctx}_{n,\text{up1}}$	Integer-level $\text{ctx}_{n,\text{iup}v}$
BD-BR, %	−22.62	**−23.14**
Mean saving, %	−23.21	**−24.01**
Std. saving, %	3.73	3.79

Table C.3 Results for integer-level context models. BD-BR with respect to GZIP evaluated on test set $\mathscr{A}$.

Note that for $l = 0$ and $n = 1$, $\text{ctx}_{n,\text{up1}}$ models sequences of values $g_{f-1,k} \geq 2$ as already found for TU binarization.

C.3 Results with GZIP Jointly Encoding Parameters as Baseline

In Section 4.3, the BD-BR improvements of CABAC and the other baseline coding methods are calculated with respect to the case, where GZIP is used for encoding the NTF parameters independently. This procedure is chosen because CABAC and the other baseline methods encode the parameters jointly as well. Throughout this thesis, the parameters are usually coded jointly with GZIP. In this section, some tables listed in Section 4.3 are given again. Here, the BD-BR savings are calculated with GZIP coding the parameters jointly as baseline.

Table C.2 is the equivalent of Table 4.9, evaluating the bin-value based context models. Table C.3 shows results equivalent to the ones shown in Table 4.10.

D Lower Bound

To derive the lowest bound for Wiener filters assuming monaural mixtures, the squared error term of the SDR denominator is chosen as cost function as already done for deriving the oracle Wiener filters in [VGP07], refer to Section 2.5 : $\sum_{f,t,j}\left(\underline{s}_{f,t,j}-m_{f,t,j}\underline{x}_{f,t}\right)^2$. This cost can be maximized for each TF point independently to find mask $m_{f,t,j}$ yielding the lowest SDR value. Omitting (f,t), the problem can be written as

$$\max \sum_j \left|\underline{s}_j - m_j\underline{x}\right|^2 \quad \text{subject to } \sum_j m_j = 1 \text{ and } m_j \geq 0$$

which also takes the remixing constraint and the non-negativity of Wiener masks into account. This problem is inverse to the one formulated for the oracle filters where the cost function is minimized under the same constraints. The same simplifications to the cost function can be done, namely

$$\begin{aligned}\left|\underline{s}_j - m_j\underline{x}\right|^2 &= m_j^2\left|\underline{x}\right|^2 - m_j\underline{s}_j^*\underline{x} - m_j\underline{s}_j\underline{x}^* + \left|\underline{s}_j\right|^2 \\ &= \left|\underline{x}\right|^2\left[m_j^2 - m_j\left(\left(\frac{\underline{s}_j}{\underline{x}}\right)^* + \frac{\underline{s}_j}{\underline{x}}\right)\right] + \left|\underline{s}_j\right|^2 \\ &= \left|\underline{x}\right|^2\left(m_j - \mathrm{Re}\left\{\frac{\underline{s}_j}{\underline{x}}\right\}\right)^2 \\ &\quad + \underbrace{\left|\underline{s}_j\right|^2 - \left|\underline{x}\right|^2\mathrm{Re}^2\left\{\frac{\underline{s}_j}{\underline{x}}\right\}}_{=\text{const}}\end{aligned}$$

which leads to a real-valued maximization problem

$$\max \sum_j \left(m_j - \underbrace{\mathrm{Re}\left\{\frac{\underline{s}_j}{\underline{x}}\right\}}_{=r_j}\right)^2 = \max \sum_j \left(m_j - r_j\right)^2 \qquad \text{(D.1)}$$

$$\text{subject to } \sum_j m_j = 1 \text{ and } m_j \geq 0.$$

The question remains which m_j is fulfilling (D.1). For the oracle estimators, the problem is solved for each TF point with quadratic programming. The inverse problem (D.1) under the same constraints considered here has a more intuitive solution. First, it should be acknowledged that $\sum_j r_j = \mathrm{Re}\left\{\sum_j \frac{\underline{s}_j}{\underline{x}}\right\} = 1$ with $\sum_j \underline{s}_j = \underline{x}$. This constrains the values of r_j to a hyperplane going through all unit vectors of $\mathbb{R}^J$. Second, since $\sum_j m_j = 1$ *and* $m_j \geq 0$,

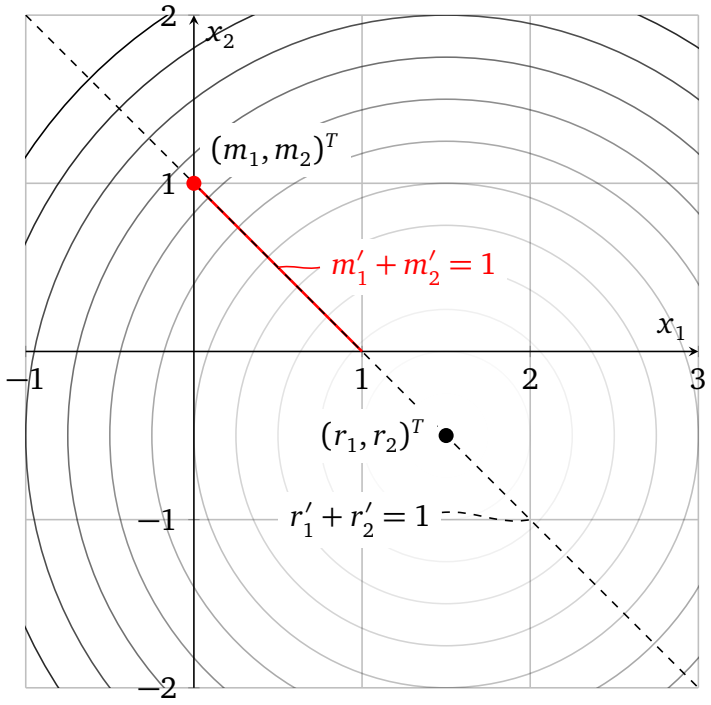


Figure D.1 Example for finding lower bound masks for $J = 2$ sources.

the values for m_j are lying in the same hyper-plane but are constrained further by $m_j \geq 0$ to the positive orthant of $\mathbb{R}^J$ at the same time. This region has the unit vectors of $\mathbb{R}^J$ as corner points. Since the cost is symmetric and monotonic increasing in $\mathbb{R}^J$ and its minimum given by r_j lies on the hyper-plane through all unit vectors, the cost should be still monotonic increasing and symmetric for points lying on this hyperplane. As the solution of problem (D.1), select the nonnegative vector on this hyper-plane which is farthest away from the minimum of the cost: Therefore, set all $m_{j'} = 0$ with $j' \neq j$ and set $m_j = 1$, with j being the index for which r_j is the smallest value, such that $r_j < r_{j'}$, $j' \neq j$. This means that the solution is always one of the unit vectors, namely the unit vector of $\mathbb{R}^J$ which is farthest away from the vector with elements r_j.

This solution is shown in Figure D.1 for $J = 2$. The cost function to be minimized can be written as $(m_1 - r_1)^2 + (m_2 - r_2)^2$. First, the dashed black line shows the set of all valid values of $\left(r'_1, r'_2\right)^\top$ for which $r'_1 + r'_2 = 1$. Then, in red, all values for $\left(m'_1, m'_2\right)^\top$ are marked which are constrained by $m'_1, m'_2 \geq 0$ and $m'_1 + m'_2 = 1$. In the background of Figure D.1, the function values of $(x_1 - r_1)^2 + (x_2 - r_2)^2$ for $(r_1, r_2)^\top = (1.5, -0.5)^\top$ are indicated by circles whose colors correspond to the function values, white mapping to zero. Finally, the point $(m_1, m_2)^\top = (0, 1)^\top$ is shown which solves (D.1). This is the point which is farthest away from $(r_1, r_2)^T$, in the set of possible values defined by $m'_1, m'_2 \geq 0$ and $m'_1 + m'_2 = 1$.

For sake of completeness, Figure D.2 shows δSDR and δSIR values for Equation (2.43) and the solution of Equation (D.1), denoted with "Lowest bound" for each mix of test set $\mathscr{A}$. When comparing the amplitude range of δSDR and δSIR values, it becomes clear that both measures yield lower δSIR values. Similar to the lower bound obtained by returning the mixture, the solution of Equation (D.1) estimates each source with a signal which mainly consists of the mixture without the source to-be-estimated present.

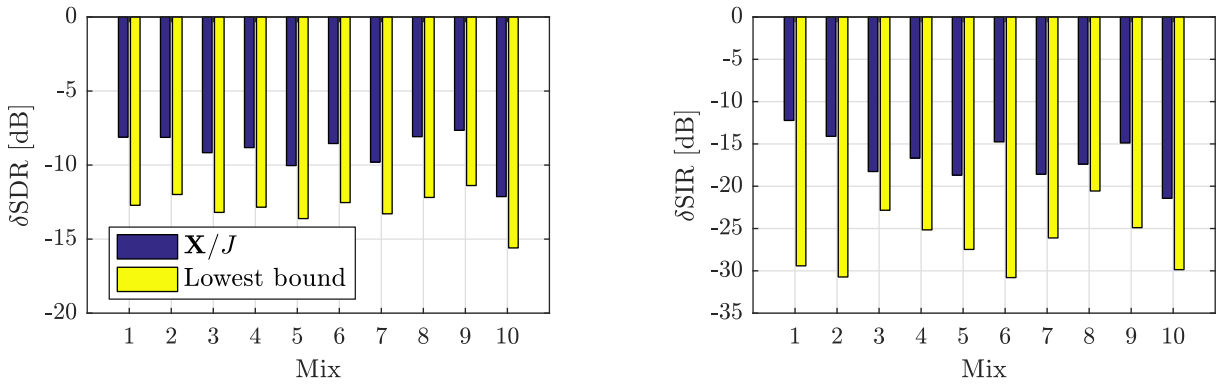


Figure D.2 Quality of lower bounds for test set $\mathscr{A}$.

Bibliography

[AR77] J. B. Allen and L. R. Rabiner. “A Unified Approach to Short-Time Fourier Analysis and Synthesis.” In: *Proceedings of the IEEE* 65.11 (Nov. 1977), pages 1558–1564. DOI: 10.1109/PROC.1977.10770 (cited on pages 5, 6).

[Bec16] J. M. Becker. *Nonnegative Matrix Factorization with Adaptive Elements for Monaural Audio Source Separation*. Volume 16. Aachen Series on Multimedia and Communications Engineering. Aachen: Shaker Verlag, Oct. 2016. ISBN: 978-3-8440-4814-8 (cited on pages 7, 17, 18, 39, 40, 43, 45, 47, 49, 70, 75).

[BF03] F. Baumgarte and C. Faller. “Binaural cue coding-Part I: psychoacoustic fundamentals and design principles.” In: *IEEE Transactions on Speech and Audio Processing* 11.6 (Nov. 2003), pages 509–519. ISSN: 1063-6676. DOI: 10.1109/TSA.2003.818109 (cited on page 117).

[BG03] M. Bosi and R. E. Goldberg. *Introduction to digital audio coding and standards*. Volume 721. Springer Science & Business Media New York, 2003. DOI: 10.1007/978-1-4615-0327-9 (cited on pages 8, 9).

[BGL16] C. Brauer, T. Gerkmann, and D. Lorenz. “Sparse reconstruction of quantized speech signals.” In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Piscataway, Mar. 2016, pages 5940–5944. DOI: 10.1109/ICASSP.2016.7472817 (cited on page 18).

[Bjø01] G. Bjøntegaard. *Calculation of average PSNR differences between RD curves*. Technical report VCEG-M33. Austin, Texas, USA: ITU-T SG16/Q6 VCEG, Apr. 2001 (cited on page 37).

[Blä+18] M. Bläser, **C. Rohlfing**, Y. Gao, and M. Wien. “Adaptive Coding of Non-negative Factorization Parameters with Application to Informed Source Separation.” In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, Canada: IEEE, Piscataway, Apr. 2018 (cited on page 52).

[BMR15] J. M. Becker, M. Menzel, and **C. Rohlfing**. “Complex SVD Initialization for NMF Source Separation on Audio Spectrograms.” In: *Fortschritte der Akustik DAGA ’15*. Nürnberg, Germany, Mar. 2015. URL: http://www.ient.rwth-aachen.de/services/bib2web/pdf/BeMeRo15.pdf (cited on pages 14, 16, 17, 50, 68).

[BOP15] Ç. Bilen, A. Ozerov, and P. Pérez. “Compressive sampling-based informed source separation.” In: *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, Piscataway, Oct. 2015, pages 1–5. DOI: 10.1109/WASPAA.2015.7336953 (cited on page 32).

[BR14] J. M. Becker and **C. Rohlfing**. “Custom Sized Non-Negative Matrix Factor Deconvolution for Sound Source Separation.” In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Florence, Italy: IEEE, Piscataway, May 2014, pages 2124–2128. DOI: 10.1109/ICASSP.2014.6853974 (cited on page 15).

[BR15] J. M. Becker and **C. Rohlfing**. “Component-Adaptive Priors for NMF.” In: *Latent Variable Analysis and Signal Separation*. Springer, Heidelberg/Berlin, Aug. 2015 (cited on page 18).

[Bro91] J. C. Brown. “Calculation of a constant Q spectral transform.” In: *Journal of the Acoustical Society of America* 89.1 (1991), pages 425–434. DOI: 10.1121/1.400476 (cited on page 10).

[BRR15] J. M. Becker, M. Rohbeck, and **C. Rohlfing**. “Adaptive Weights for NMF with Additional Priors.” In: *2015 IEEE International Workshop on Intelligent Signal Processing and Communication Systems (ISPACS)*. Nusa Dua, Bali, Indonesia: IEEE, Piscataway, Nov. 2015, pages 89–94. DOI: 10.1109/ISPACS.2015.7432744 (cited on page 18).

[BSH08] J. Benesty, M. M. Sondhi, and Y. Huang. *Springer Handbook of Speech Processing*. Volume 1. Springer-Verlag Berlin Heidelberg, 2008. DOI: 10.1007/978-3-540-49127-9 (cited on page 10).

[CD88] W. Cleveland and S. Devlin. “Locally weighted regression: an approach to regression analysis by local fitting.” In: *Journal of the American Statistical Association* 83.403 (1988), pages 596–610 (cited on page 37).

[Cic+09] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, Sept. 2009. DOI: 10.1002/9780470747278 (cited on pages 11, 13, 14, 68).

[CT06] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. 2nd edition. Wiley Publishing, Sept. 2006. ISBN: 0471241954 (cited on pages 22, 23, 36).

[Deu+96] L. P. Deutsch, J.-L. Gailly, M. Adler, and G. Randers-Pehrson. *GZIP file format specification version 4.3*. RFC 1952. RFC Editor, May 1996. URL: http://www.rfc-editor.org/rfc/rfc1952.txt (cited on page 23).

[DT17] A. Deleforge and Y. Traonmilin. “Phase unmixing: Multichannel source separation with magnitude constraints.” In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans, USA: IEEE, Piscataway, Mar. 2017, pages 161–165. DOI: 10.1109/ICASSP.2017.7952138 (cited on page 29).

[DVG10] N. Q. K. Duong, E. Vincent, and R. Gribonval. “Under-Determined Reverberant Audio Source Separation Using a Full-Rank Spatial Covariance Model.” In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.7 (Sept. 2010), pages 1830–1840. ISSN: 1558-7916. DOI: 10.1109/TASL.2010.2050716 (cited on page 2).

[Eng+08] J. Engdegård et al. “Spatial Audio Object Coding (SAOC) - The Upcoming MPEG Standard on Parametric Object Based Audio Coding.” In: *Audio Engineering Society Convention 124*. May 2008. URL: http://www.aes.org/e-lib/browse.cfm?elib=14507 (cited on page 33).

[FBD09] C. Févotte, N. Bertin, and J.-L. Durrieu. “Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis.” In: *Neural Computation* 21.3 (Mar. 2009), pages 793–830. DOI: http://dx.doi.org/10.1162/neco.2008.04-08-771 (cited on page 13).

[FCC05] D. FitzGerald, M. Cranitch, and E. Coyle. “Shifted non-negative matrix factorisation for sound source separation.” In: *2005 IEEE/SP Workshop on Statistical Signal Processing*. IEEE, Piscataway, July 2005, pages 1132–1137. DOI: 10.1109/SSP.2005.1628765 (cited on page 15).

[FCC08] D. FitzGerald, M. Cranitch, and E. Coyle. “Extended Nonnegative Tensor Factorisation Models for Musical Sound Source Separation.” In: *Computational Intelligence and Neuroscience* (2008). DOI: 10.1155/2008/872425 (cited on page 10).

[FI11] C. Févotte and J. Idier. “Algorithms for nonnegative matrix factorization with the β-divergence.” In: *Neural computation* 23.9 (2011), pages 2421–2456 (cited on pages 13, 14).

[FTH10] C. Falch, L. Terentiev, and J. Herre. “Spatial audio object coding with enhanced audio object separation.” In: *2010 International Conference on Digital Audio Effects (DAFx)*. Graz, Austria, Sept. 2010 (cited on page 33).

[Gao17] M. Gao. “Advanced Entropy Coding Methods for Informed Source Separation.” Master’s Thesis. Institut für Nachrichtentechnik, RWTH Aachen University, May 2017 (cited on pages 52, 56).

[GHM13] S. Gorlow, E. A. P. Habets, and S. Marchand. “Multichannel Object-Based Audio Coding with Controllable Quality.” In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vancouver, Canada: IEEE, Piscataway, May 2013, pages 561–565. URL: https://hal.archives-ouvertes.fr/hal-00806382 (cited on page 32).

[GKR15] T. Gerkmann, M. Krawczyk-Becker, and J. Le Roux. “Phase Processing for Single-Channel Speech Enhancement: History and recent advances.” In: *IEEE Signal Processing Magazine* 32.2 (Mar. 2015), pages 55–66. DOI: 10.1109/MSP.2014.2369251 (cited on pages 8, 29).

[GL84] D. Griffin and J. Lim. “Signal estimation from modified short-time Fourier transform.” In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32.2 (Apr. 1984), pages 236–243. ISSN: 0096-3518. DOI: 10.1109/TASSP.1984.1164317 (cited on page 29).

[GM11] S. Gorlow and S. Marchand. “Informed source separation: Underdetermined source signal recovery from an instantaneous stereo mixture.” In: *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, Piscataway, Oct. 2011, pages 309–312. DOI: 10.1109/ASPAA.2011.6082312 (cited on page 32).

[GN98] R. M. Gray and D. L. Neuhoff. "Quantization." In: *IEEE Transactions on Information Theory* 44.6 (Oct. 1998), pages 2325–2383. DOI: 10.1109/18.720541 (cited on page 18).

[Gna14] V. Gnann. *Signal Reconstruction from Multiresolution Magnitude Spectrograms for Audio Signal Processing*. Volume 13. Aachen Series on Multimedia and Communications Engineering. Aachen: Shaker Verlag, Feb. 2014. ISBN: 978-3-8440-2500-2 (cited on page 29).

[GS10] D. Gunawan and D. Sen. "Iterative Phase Estimation for the Synthesis of Separated Sources From Single-Channel Mixtures." In: *IEEE Signal Processing Letters* 17.5 (May 2010), pages 421–424. DOI: 10.1109/LSP.2010.2042530 (cited on page 30).

[Hen11] Romain Hennequin. "Décomposition de spectrogrammes musicaux informée par des modèles de synthèse spectrale. Modélisation des variations temporelles dans les éléments sonores." PhD thesis. Télécom ParisTech, 2011 (cited on page 15).

[Her+04] J. Herre, C. Faller, S. Disch, C. Ertel, J. Hilpert, A. Hoelzer, K. Linzmeier, C. Spenger, and P. Kroon. "Spatial Audio Coding: Next-Generation Efficient and Compatible Coding of Multi-Channel Audio." In: *Audio Engineering Society Convention 117*. Oct. 2004. URL: http://www.aes.org/e-lib/browse.cfm?elib=12843 (cited on page 32).

[Her+05] J. Herre, H. Purnhagen, J. Breebaart, C. Faller, S. Disch, K. Kjörling, E. Schuijers, J. Hilper, and F. Myburg. "The Reference Model Architecture for MPEG Spatial Audio Coding." In: *Audio Engineering Society Convention 118*. May 2005. URL: http://www.aes.org/e-lib/browse.cfm?elib=13163 (cited on page 32).

[Her+12] J. Herre et al. "MPEG Spatial Audio Object Coding—The ISO/MPEG Standard for Efficient Coding of Interactive Audio Scenes." In: *Audio Engineering Society Convention 129* 60.9 (Nov. 2012), pages 655–673. URL: http://www.aes.org/e-lib/browse.cfm?elib=16371 (cited on page 106).

[Her+15] J. Herre, J. Hilpert, A. Kuntz, and J. Plogsties. "MPEG-H audio – the new standard for universal spatial/3D audio coding." In: *Journal of the Audio Engineering Society* 62.12 (2015), pages 821–830 (cited on page 33).

[Kam+09] H. Kameoka, N. Ono, K. Kashino, and S. Sagayama. "Complex NMF: A new sparse representation for acoustic signals." In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Taipei, Taiwan: IEEE, Piscataway, Apr. 2009, pages 3437–3440. DOI: 10.1109/ICASSP.2009.4960364 (cited on page 15).

[Kır+14] S. Kırbız, A. Ozerov, A. Liutkus, and L. Girin. "Perceptual coding-based Informed Source Separation." In: *2014 European Signal Processing Conference (EUSIPCO)*. Lisbon, Portugal, Sept. 2014, pages 959–963 (cited on pages 32, 117).

[LB15] A. Liutkus and R. Badeau. “Generalized Wiener filtering with fractional power spectrograms.” In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brisbane, Australia: IEEE, Piscataway, Apr. 2015, pages 266–270. DOI: 10.1109/ICASSP.2015.7177973 (cited on page 40).

[LBR10] A. Liutkus, R. Badeau, and G. Richard. “Informed Source Separation Using Latent Components.” In: *Latent Variable Analysis and Signal Separation*. Edited by Vincent Vigneron, Vicente Zarzoso, Eric Moreau, Rémi Gribonval, and Emmanuel Vincent. Volume 6365. Lecture Notes in Computer Science. Saint Malo, France: Springer, 2010, pages 498–505. DOI: 10.1007/978-3-642-15995-4_62 (cited on page 32).

[LBR13] A. Liutkus, R. Badeau, and G. Richard. “Low bitrate informed source separation of realistic mixtures.” In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vancouver, Canada: IEEE, Piscataway, May 2013, pages 66–70. DOI: 10.1109/ICASSP.2013.6637610 (cited on pages 2, 117).

[Lin07] C.-J. Lin. “Projected gradient methods for nonnegative matrix factorization.” In: *Neural computation* 19.10 (2007), pages 2756–2779 (cited on page 67).

[Liu+11] A. Liutkus, J. Pinel, R. Badeau, L. Girin, and G. Richard. “Informed source separation through spectrogram coding and data embedding.” In: *Signal Processing* 92.8 (2011), pages 1937–1949. DOI: 10.1016/j.sigpro.2011.09.016 (cited on pages 15, 32, 39, 40, 43, 44, 46, 65, 66, 110).

[Liu+12] A. Liutkus, S. Gorlow, N. Sturmel, S. Zhang, L. Girin, R. Badeau, L. Daudet, S. Marchand, and G. Richard. “Informed Source Separation : a Comparative Study.” In: *2012 IEEE European Signal Processing Conference (EUSIPCO)*. Bucarest, Romania, Aug. 2012, pages 2397–2401 (cited on page 32).

[Liu12] Antoine Liutkus. “Processus gaussiens pour la séparation de sources et le codage informé.” PhD thesis. Télécom ParisTech, 2012. URL: https://pastel.archives-ouvertes.fr/pastel-00790841 (cited on pages 14, 15, 32, 35, 41).

[Llo82] S. Lloyd. “Least squares quantization in PCM.” In: *IEEE Transactions on Information Theory* 28.2 (Mar. 1982), pages 129–137. DOI: 10.1109/TIT.1982.1056489 (cited on page 21).

[LRD18] A. Liutkus, **C. Rohlfing**, and A. Deleforge. “Audio Source separation with magnitude priors: the BEADS model.” In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, Canada: IEEE, Piscataway, Apr. 2018 (cited on page 117).

[LS01] D. D. Lee and H. S. Seung. “Algorithms for Non-negative Matrix Factorization.” In: *Advances in Neural Information Processing Systems (NIPS)*. Volume 13. The MIT Press, Apr. 2001, pages 556–562 (cited on pages 11–14).

[LS99] D. D. Lee and H. S. Seung. “Learning the parts of objects by non-negative matrix factorization.” In: *Nature* 401.6755 (1999), page 788. DOI: 10.1038/44565 (cited on page 11).

[Mal08] S. Mallat. *A Wavelet Tour of Signal Processing*. 3rd edition. Elsevier Science, Dec. 2008. ISBN: 9780080922027 (cited on pages 6, 7).

[Max60] J. Max. "Quantizing for minimum distortion." In: *IRE Transactions on Information Theory* 6.1 (Mar. 1960), pages 7–12. DOI: 10.1109/TIT.1960.1057548 (cited on page 21).

[MPE07] MPEG-D Part 1. *Information technology – MPEG audio technologies – Part 1: MPEG Surround*. ISO/IEC 23003-1:2007. ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information, Feb. 2007 (cited on page 32).

[MPE10] MPEG-D Part 2. *Information technology – MPEG audio technologies – Part 2: Spatial Audio Object Coding (SAOC)*. ISO/IEC 23003-2:2010. ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information, Oct. 2010 (cited on pages 33, 106).

[MPE15] MPEG-H Part 3. *Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 3: 3D audio*. ISO/IEC 23008-3:2015. ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information, Mar. 2015 (cited on page 33).

[MPE99] MPEG-4 Part 3. *Information technology – Coding of audio-visual objects – Part 3: Audio*. ISO/IEC 14496-3:1999. ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information, Dec. 1999 (cited on page 8).

[MSW03] D. Marpe, H. Schwarz, and T. Wiegand. "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard." In: *IEEE Transactions on Circuits and Systems for Video Technology* 13.7 (July 2003), pages 620–636. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2003.815173 (cited on pages 23–27).

[Nia06] O. A. Niamut. "Rate-distortion optimal time-frequency decompositions for MDCT-based audio coding." PhD thesis. Technische Universiteit Delft, Nov. 2006. ISBN: 978-90-9021159-6 (cited on page 9).

[Nik15] J. Nikunen. *Object-based Modeling of Audio for Coding and Source Separation*. Volume 1276. Tampere University of Technology. Publication. Tampere University of Technology., Jan. 2015. ISBN: 978-952-15-3438-6. URL: https://tutcris.tut.fi/portal/files/2460357/nikunen_1276.pdf (cited on pages 15, 32, 117).

[NV10] J. Nikunen and T. Virtanen. "Object-Based Audio Coding Using Non-Negative Matrix Factorization for the Spectrogram Representation." In: *Audio Engineering Society Convention 128*. London, UK, May 2010 (cited on page 62).

[OF10] A. Ozerov and C. Févotte. "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation." In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.3 (Mar. 2010), pages 550–563. DOI: 10.1109/TASL.2009.2031510 (cited on page 15).

[Ohm15] J.-R. Ohm. *Multimedia Signal Coding and Transmission*. Signals and Communication Technology. Springer-Verlag Berlin Heidelberg, 2015. DOI: 10.1007/978-3-662-46691-9 (cited on pages 26, 27, 36, 63).

[OVB12] A. Ozerov, E. Vincent, and F. Bimbot. "A general flexible framework for the handling of prior information in audio source separation." In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.4 (May 2012), pages 1118–1133. DOI: 10.1109/TASL.2011.2172425 (cited on page 15).

[Oze+11] A. Ozerov, A. Liutkus, R. Badeau, and G. Richard. "Informed source separation: Source coding meets source separation." In: *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, Piscataway, Oct. 2011, pages 257–260. DOI: 10.1109/ASPAA.2011.6082285 (cited on page 32).

[Oze+13] A. Ozerov, A. Liutkus, R. Badeau, and G. Richard. "Coding-Based Informed Source Separation: Nonnegative Tensor Factorization Approach." In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.8 (Aug. 2013), pages 1699–1712. DOI: 10.1109/TASL.2013.2260153 (cited on pages 15, 32, 39, 41, 44, 62, 97).

[PB86] J. Princen and A. Bradley. "Analysis/Synthesis filter bank design based on time domain aliasing cancellation." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34.5 (Oct. 1986), pages 1153–1161. DOI: 10.1109/TASSP.1986.1164954 (cited on pages 5, 8).

[PG11] M. Parvaix and L. Girin. "Informed Source Separation of Linear Instantaneous Under-Determined Audio Mixtures by Source Index Embedding." In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.6 (Aug. 2011), pages 1721–1733. DOI: 10.1109/TASL.2010.2097250 (cited on page 32).

[PGB10] M. Parvaix, L. Girin, and J.-M. Brossier. "A Watermarking-Based Method for Informed Source Separation of Audio Signals With a Single Sensor." In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.6 (Aug. 2010), pages 1464–1475. DOI: 10.1109/TASL.2009.2035216 (cited on page 32).

[PGB14] J. Pinel, L. Girin, and C. Baras. "A high-rate data hiding technique for uncompressed audio signals." In: *Journal of the Audio Engineering Society* 62.6 (June 2014), pages 400–413. URL: https://hal.archives-ouvertes.fr/hal-01143294 (cited on page 2).

[PJB87] J. Princen, A. Johnson, and A. Bradley. "Subband/Transform coding using filter bank designs based on time domain aliasing cancellation." In: *1987 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Dallas, USA: IEEE, Piscataway, Apr. 1987, pages 2161–2164. DOI: 10.1109/ICASSP.1987.1169405 (cited on pages 5, 8).

[Puy+17] G. Puy, A. Ozerov, N. Q. K. Duong, and P. Pérez. "Informed source separation via compressive graph signal sampling." In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans, USA: IEEE, Piscataway, Mar. 2017, pages 1–5. DOI: 10.1109/ICASSP.2017.7951786 (cited on pages 32, 39).

[RB16] **C. Rohlfing** and J. M. Becker. "Generalized Constraints for NMF with Application to Informed Source Separation." In: *2016 European Signal Processing Conference (EUSIPCO)*. Budapest, Hungary: IEEE, Piscataway, Aug. 2016, pages 597–601. DOI: 10.1109/EUSIPCO.2016.7760318 (cited on page 82).

[RBW16] **C. Rohlfing**, J. M. Becker, and M. Wien. "NMF-based Informed Source Separation." In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Shanghai, China: IEEE, Piscataway, Mar. 2016, pages 474–478. DOI: 10.1109/ICASSP.2016.7471720 (cited on pages 39, 43, 63, 65, 66, 68, 71).

[RD07] E. Ravelli and L. Daudet. "Embedded Polar Quantization." In: *IEEE Signal Processing Letters* 14.10 (Oct. 2007), pages 657–660. DOI: 10.1109/LSP.2007.896379 (cited on page 117).

[REL17] **C. Rohlfing**, J. E.Cohen, and A. Liutkus. "Very Low Bitrate Spatial Audio Coding with Dimensionality Reduction." In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans, USA: IEEE, Piscataway, Mar. 2017, pages 741–745. DOI: 10.1109/ICASSP.2017.7952254 (cited on page 68).

[RLB17] **C. Rohlfing**, A. Liutkus, and J. M. Becker. "Quantization-aware Parameter Estimation for Audio Upmixing." In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans, USA: IEEE, Piscataway, Mar. 2017, pages 746–750. DOI: 10.1109/ICASSP.2017.7952255 (cited on pages 65, 82–84, 123).

[RV13] J. Le Roux and E. Vincent. "Consistent Wiener Filtering for Audio Source Separation." In: *IEEE Signal Processing Letters* 20.3 (Mar. 2013), pages 217–220. DOI: 10.1109/LSP.2012.2225617 (cited on page 30).

[Say05] Khalid Sayood. *Introduction to data compression*. 3rd edition. The Morgan Kaufmann Series in Multimedia Information and Systems. Morgan Kaufmann, Dec. 2005. ISBN: 9780080509259 (cited on pages 22, 23).

[SB03] P. Smaragdis and J. C. Brown. "Non-negative matrix factorization for polyphonic music transcription." In: *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, Piscataway, Oct. 2003, pages 177–180. DOI: 10.1109/ASPAA.2003.1285860 (cited on page 15).

[SD11] N. Sturmel and L. Daudet. "Signal reconstruction from STFT magnitude: A state of the art." In: *International conference on digital audio effects (DAFx)*. Sept. 2011, pages 375–386 (cited on pages 8, 29, 30).

[SD12] N. Sturmel and L. Daudet. "Iterative phase reconstruction of Wiener filtered signals." In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Kyoto, Japan: IEEE, Piscataway, Mar. 2012, pages 101–104. DOI: 10.1109/ICASSP.2012.6287827 (cited on page 30).

[SD13] N. Sturmel and L. Daudet. "Informed Source Separation Using Iterative Reconstruction." In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.1 (Jan. 2013), pages 178–185. DOI: 10.1109/TASL.2012.2215597 (cited on pages 31, 44).

[Sha48] C. E. Shannon. “A mathematical theory of communication.” In: *Bell System Technical Journal* 27 (1948), pages 379–423 (cited on page 36).

[She94] J. R. Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Technical report. Pittsburgh, PA, USA, 1994 (cited on page 30).

[Sma04] P. Smaragdis. “Non-negative Matrix Factor Deconvolution; Extraction of Multiple Sound Sources from Monophonic Inputs.” In: *2004 International Conference on Independent Component Analysis and Signal Separation (ICA)*. Berlin, Heidelberg: Springer Berlin Heidelberg, Sept. 2004, pages 494–499. DOI: 10.1007/978-3-540-30110-3_63 (cited on page 15).

[Spi12] M. Spiertz. *Underdetermined Blind Source Separation for Audio Signals*. Volume 10. Aachen Series on Multimedia and Communications Engineering. Aachen: Shaker Verlag, July 2012. ISBN: 978-3844011746. URL: http://www.ient.rwth-aachen.de/services/bib2web/pdf/Sp12.pdf (cited on pages 6, 10, 11, 15, 40, 49, 65, 66, 68, 75–77).

[Sul+12] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand. “Overview of the High Efficiency Video Coding (HEVC) Standard.” In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (Dec. 2012), pages 1649–1668. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2012.2221191 (cited on page 23).

[SVN37] S. S. Stevens, J. Volkmann, and E. B. Newman. “A scale for the measurement of the psychological magnitude pitch.” In: *Journal of the Acoustical Society of America* 8.3 (1937), pages 185–190 (cited on page 10).

[SW98] G. J. Sullivan and T. Wiegand. “Rate-distortion optimization for video compression.” In: *IEEE Signal Processing Magazine* 15.6 (Nov. 1998), pages 74–90. DOI: 10.1109/79.733497 (cited on page 22).

[VGF06] E. Vincent, R. Gribonval, and C. Févotte. “Performance measurement in blind audio source separation.” In: *IEEE Transactions on Audio, Speech and Language Processing* 14.4 (June 2006), pages 1462–1469. DOI: 10.1109/TSA.2005.858005 (cited on page 34).

[VGP07] E. Vincent, R. Gribonval, and M. D. Plumbley. “Oracle estimators for the benchmarking of source separation algorithms.” In: *Signal Processing* 87.8 (2007), pages 1933–1950. DOI: 10.1016/j.sigpro.2007.01.016 (cited on pages 28, 129).

[Vir07] T. Virtanen. “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria.” In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.3 (Mar. 2007), pages 1066–1074. DOI: 10.1109/TASL.2006.885253 (cited on pages 15, 17, 18, 49, 51, 76).

[VM06] P. Vary and R. Martin. *Digital speech transmission: Enhancement, coding and error concealment*. John Wiley & Sons, 2006. ISBN: 978-0-471-56018-0 (cited on pages 20, 21).

[VVT12] JM. Valin, K. Vos, and T. Terriberry. *Definition of the Opus Audio Codec*. RFC 6716. RFC Editor, Sept. 2012 (cited on page 8).

[Wie14] M. Wien. *High Efficiency Video Coding – Coding Tools and Specification*. Berlin, Heidelberg: Springer, Sept. 2014. DOI: 10.1007/978-3-662-44276-0 (cited on pages 24–26).

[ZBC10] A. Zymnis, S. Boyd, and E. Candès. "Compressed sensing with quantized measurements." In: *IEEE Signal Processing Letters* 17.2 (Feb. 2010), pages 149–152. DOI: 10.1109/LSP.2009.2035667 (cited on page 18).

Lebenslauf

Christian Rohlfing

15. Juli 1985	Geburt in Krefeld
1992 – 1996	Städtische Gemeinschaftsgrundschule Girmesdyk, Krefeld
1996 – 2005	Gymnasium Horkesgath, Krefeld
Juni 2005	Abitur
2005 – 2006	Zivildienst
2006 – 2012	Studium der Technischen Informatik an der RWTH Aachen University
Mai 2012	Diplom
2012 – 2018	Wissenschaftlicher Mitarbeiter am Institut für Nachrichtentechnik, RWTH Aachen University