BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS
ÚSTAV TELEKOMUNIKACÍ

MODERN OPTIMIZATION METHODS FOR INTERPOLATION OF MISSING SECTIONS IN AUDIO SIGNALS

MODERNÍ OPTIMALIZAČNÍ METODY PRO INTERPOLACI CHYBĚJÍCÍCH ÚSEKŮ V AUDIO SIGNÁLECH

DOCTORAL THESIS
DISERTAČNÍ PRÁCE

AUTHOR
AUTOR PRÁCE

Ing. Ondřej Mokrý

ADVISOR
VEDOUCÍ PRÁCE

prof. Mgr. Pavel Rajmic, Ph.D.

BRNO 2024

ABSTRACT

Damage to audio signals is in practice common, yet undesirable. Information loss can occur due to improper recording (low sample rate or dynamic range), transmission error (sample dropout), media damage, or because of noise. The removal of such disturbances is possible using inverse problems. Specifically, this work focuses on the situation where sections of an audio signal of length in the order of tens of milliseconds are completely lost, and the goal is to interpolate the missing samples based on the unimpaired context and a suitable signal model. The first part of the dissertation is devoted to convex and non-convex optimization methods, which are designed to find a solution to the interpolation problem based on the assumption of sparsity of the time-frequency spectrum. The general background and some algorithms are taken from the literature and adapted to the interpolation problem, many modifications and experimental approaches are original. The second part of the thesis focuses on the use of non-negative matrix factorization, with which a probabilistic model of the signal spectrogram can be constructed and used for the interpolation of the signal. This model is then used as the basis for a successful reconstruction algorithm, to which two alternative methods are derived in the present thesis. Finally, an extensive experimental validation of the methods on a group of musical signals is conducted. Using objective indicators of the quality of the interpolated signal, it is shown, that in each class of methods, the proposed modifications lead to a noticeable improvement in quality or convergence over the baseline methods. In particular, within the studied range of impairments, algorithms using factorization compete with the current best methods for interpolating missing sections of the audio signal.

KEYWORDS

audio inpainting, audio interpolation, inverse problems, non-negative matrix factorization, optimization, proximal algorithms, restoration, sparsity

ABSTRAKT

Poškození audio signálů je v praxi běžným, avšak nežádoucím faktem. Ke ztrátě informace může dojít nevhodným záznamem (nízký vzorkovací kmitočet či dynamický rozsah), chybou přenosu (výpadek vzorků), poškozením média či z důvodu rušení. Odstraňování takových poruch je možné pomocí inverzních úloh. Tato práce se konkrétně zaměřuje na situaci, kdy jsou úseky audio signálu o délce v řádu desítek milisekund zcela ztraceny a cílem je chybějící vzorky interpolovat na základě kontextu a vhodného modelu signálu. První část dizertační práce se věnuje metodám konvexní i nekonvexní optimalizace, které hledají řešení interpolační úlohy na základě předpokladu řídkosti časově-kmitočtového spektra. Obecný základ i některé algoritmy jsou převzaté z literatury a přizpůsobené interpolační úloze, řada modifikací a experimentálních přístupů je originální. Druhá část práce je zaměřena na využití nezáporné faktorizace matic, s níž lze sestavit pravděpodobnostní model spektrogramu signálu a tento využít pro jeho interpolaci. Z tohoto modelu pak vychází úspěšný rekonstrukční algoritmus, k němuž jsou v této práci odvozeny dvě alternativní metody. Závěr práce se věnuje rozsáhlému experimentálnímu ověření funkčnosti metod na skupině hudebních signálů. S využitím objektivních ukazatelů kvality interpolovaného signálu je ukázáno, že v jednotlivých třídách metod vedou navržené modifikace ke znatelnému zlepšení kvality či zlepšení konvergence oproti metodám základním. V rámci studovaného rozsahu poškození pak zejména algoritmy využívající faktorizace konkurují současným nejlepším metodám pro interpolaci chybějících úseků audio signálu.

KLÍČOVÁ SLOVA

audio inpainting, interpolace audio signálů, inverzní úlohy, nezáporná faktorizace matic, optimalizace, proximální algoritmy, restaurace, řídkost

Typeset by the thesis package, version 4.07; https://latex.fekt.vut.cz/

MOKRÝ, Ondřej. *Modern optimization methods for interpolation of missing sections in audio signals*. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications, 2024, 134 p. Doctoral thesis. Advised by prof. Mgr. Pavel Rajmic, Ph.D.

Author's Declaration

Author:	Ing. Ondřej Mokrý
Author's ID:	170583
Paper type:	Doctoral thesis
Academic year:	2023/24
Topic:	Modern optimization methods for interpolation of missing sections in audio signals

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno ....

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

Set theory, a fundamental branch of mathematics, has a precisely defined notion of ordering. However, it does not provide a clear answer to a very natural question: How to order all the acknowledgments that I feel are necessary? For this reason, the following list will be chaotic, unordered, and most probably incomplete.

I would like to thank professor Pavel Rajmic for being my supervisor, employer, scientific leader and colleague, source of know-how and extraordinary opportunities etc. for more than 7 years. A special acknowledgment belongs also to his contagious enthusiasm for audio signals and mathematics, which was the catalyst at the start of my signal-processing journey.

I would like to thank all the people at the Department of Telecommunications of FEEC BUT who help to create a friendly yet productive environment. In particular, I would like to thank everyone who shared the legendary room SD5.66 with me. Besides calling me “MP” only sometimes (and only because of envy), I need to thank them for providing high quality caffeine, inspiration, sports passion, and excellent atmosphere in general.

I would like to thank my past, current and future family for their support in numerous ways, including, but not limited to: acceptance, finance, genome, love, and trust. This concerns especially my beloved Daniela, who had to accept the uncanny love triangle of me, her, and the thesis.

I would like to thank the Signal & Communications group of IRIT (and Cédric Févotte in particular) for welcoming me for my Erasmus+ internship, which gave me many precious memories, valuable experience, deepened addiction to baguettes, and significant portion of this thesis.

The research which allowed me to write this thesis was supported by Czech science foundation, projects 17-33798L, 20-29009S and 23-07294S. In conducting the research, no audio signals were actually harmed.

Contents

List of Figures

List of Tables

Introduction

Interpolation of data, i.e., the addition of a number into the middle of a series, calculated based on the numbers before and after it [14], is a task that can be dated back to the very beginning of the existence of data itself. This is due to natural incompleteness of data, caused either by measurement errors, low frequency of measurement, loss of previously recorded samples or presence of impulsive noise. At the same time, incomplete information naturally complicate any further data processing, analysis or reasoning.

Such issues are commonly encountered also in audio signals. While analog signals represent the evolution of real-world phenomena over continuous time, the current center of interest lies especially in the area of digital signal processing (DSP). Digital audio signals, if not digitally synthesized from scratch, are sampled and quantized versions of the initially continuous audio signals. As such, they are frequently corrupted in many ways (or several ways combined): Samples of the signal may be missing, contaminated with noise, or highly degraded, for example by physical damage on the analog media such as LP or wax cylinder [15]. A particular case of noise is induced by clipping, which occurs during recording or playback of an audio signal that exceeds physical dynamic limits of the recording/playback device, or the A/D or D/A converter. Parts of the signal may be also missing due to transmission errors, especially packet loss [16]. Recently, methods for audio recording using optical fibers as microphones have come to the forefront of interest, especially due to applications in audio forensics and the threat of eavesdropping [17]. However, physical properties of such recording systems may introduce clicks with excessive amplitude, preventing automated processing [18].

A natural question arises, whether such a degradation can be reversed, and to what amount, which is referred to as the *inverse problem*. All the aforementioned corruptions, especially the drop-out of samples, cause irreversible loss of information. However, by supplementing prior assumptions about the original signal, inversion of the degradation process may be possible. A typical example would be the upsampling of an audio signal, i.e., interpolation of (evenly distributed) missing samples. This common procedure can be also seen as an inverse problem which has a unique solution under the assumption of band-limitedness of the original signal in the spectral domain, as stated by the famous sampling theorem [19].

This thesis concerns a particular audio inverse problem, which is the interpolation of missing audio samples, also known as *audio inpainting* [20]. In the context of numerical mathematics, interpolation is possible with Lagrange interpolators and related models, which have historically been proposed for lossy speech coding and used in pioneer audio inpainting methods [21, 22]. Similarly, very short drop-outs or random subsampling lead to rather simple interpolation tasks, where band-limitedness or smoothness may serve as sufficient priors. However, the task of particular interest and applicability is the interpolation of missing compact segments, i.e., gaps in the signal. While the *middle-length* gaps with length in the range of tens of milliseconds may appear as minor at first, their interpolation is a very challenging problem. The challenge stems also from a psychoacoustical point of view, since improper treatment of middle-length gaps introduces very disturbing audible artifacts in the signal. However, the interpolation may still aim at precise recovery of the original information. In the case of even longer gaps, methods using close neighborhoods of the gap to interpolate it are not sufficient anymore. The only reliable possibility is to analyze the signal semantically and copy or generate new information based on the retrieved knowledge about the signal.

This thesis follows on previous research in the field of audio restoration and aims at proposing novel, effective methods with the primary focus on interpolation of musical audio signals. Chapter 1 serves as a brief, theoretical basis for the rest of the thesis. It is followed by a review of state-of-the-art approaches to the problem of audio signal interpolation in chapter 2. The flaws and possible space for improvement of those methods are discussed, which allows to declare the aims and objectives of this thesis in chapter 3. In the core part of the work, we focus on two classes of optimization-based methods, which have in common the applicability in signal compression. These are methods based on spectral sparsity in chapter 4 and methods using low-rank representations and probabilistic modeling in chapter 5. Chapter 6 presents numerical experiments, aiming at objective evaluation of all the developed methods and comparison with state-of-the-art methods, followed by conclusion of the thesis and proposition of possible directions for future research.

1 Concepts and conventions

This chapter will lay the theoretical foundations for the rest of the thesis. The aim is not an exhaustive explanation, but rather a formal introduction of concepts and a theoretical basis for the more practically oriented chapters.

1.1 Spaces and operators

1.1.1 Vector spaces

Since the full axiomatic development of linear algebra is not crucial for the specific field of signal processing, we will introduce only selected definitions and conventions. For an overview of elementary terms such as vector (sub)space, linear (in)dependence, basis etc., see any introductory material to linear algebra or functional analysis, such as [23, 24].

Remark 1.1 (Notation)**.** A generic vector space over a field $\mathbb{F}$, where $\mathbb{F}$ represents real or complex numbers, will be denoted $V, W, \dots$, while the individual elements (vectors) will be denoted $u, v, w, \dots$ Of particular importance to us are vectors in the sense of N-tuples, which will be denoted in bold: $\mathbf{x}, \mathbf{y}, \mathbf{z}, \dots \in \mathbb{F}^N$. Because of the common interaction between vectors and matrices, such vectors of length N will be interchangeable with matrices of size $N \times 1$ (in other words, we consider column vectors). The individual entries of vectors and matrices will be indexed starting from 1, i.e., $\mathbf{x} = [x_1, \dots, x_N]^\top \in \mathbb{F}^N$. In some cases, we will use also brackets: $\mathbf{x}(n) = x_n$. Similarly, we will be able to select entries defined by a set of indices: $\mathbf{x}(\{1,2,4\}) = [x_1, x_2, x_4]^\top$.

Definition 1.2 (Normed vector space [24, Definition 1.63])**.** A normed vector space is a vector space V over $\mathbb{F}$, equipped with the function $\|\cdot\| \colon V \to \mathbb{F}$, called the norm on V, which satisfies for every $u, v \in V$ and $\alpha \in \mathbb{F}$ the following axioms:

1. $\|u + v\| \leq \|u\| + \|v\|$, (triangle inequality)
2. $\|\alpha \cdot u\| = |\alpha| \, \|u\|$, (absolute homogeneity)
3. $\|u\| = 0 \Leftrightarrow u = 0$. (positive definiteness)

From the above axioms, it can be directly proven that norm is non-negative:

4. $\|u\| \geq 0$.

Example 1.3 (ℓ_p norms)**.** For the spaces $\mathbb{R}^N$ or $\mathbb{C}^N$ the usual norm is the Euclidean norm: $\|\mathbf{x}\|_2 = \sqrt{|x_1|^2 + \cdots + |x_N|^2}$. Its square $\|\mathbf{x}\|_2^2$ is also called energy in the context of signals. Similarly, we can treat a matrix $\mathbf{A} \in \mathbb{F}^{M \times N}$ as a vector $[a_{11}, \dots, a_{MN}] \in \mathbb{F}^{MN}$ and compute its Euclidean norm, which is referred to as the Frobenius norm $\|\mathbf{A}\|_\mathrm{F}^2 = \sum_{i=1}^M \sum_{j=1}^N |a_{ij}|^2$.

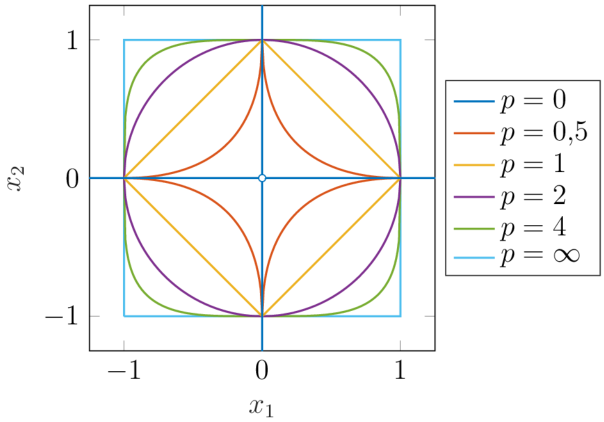


Fig. 1.1: Illustration of ℓ_p norms in example 1.3. For different choices of p we plot the unit ball, i.e., the set of vectors $\mathbf{x} \in \mathbb{R}^2$ such that $\|\mathbf{x}\|_p = 1$.

Generalization of the Euclidean norm is the ℓ_p norm, or p-norm, defined as

$$\|\mathbf{x}\|_p = \left(|x_1|^p + \cdots + |x_N|^p\right)^{\frac{1}{p}} . \tag{1.1}$$

The concept of different norms is illustrated in figure 1.1.

For $1 \leq p \leq \infty$, it is a true norm (satisfying the axioms of the definition 1.2). The limit case for $p = \infty$ is the maximum norm $\|\mathbf{x}\|_\infty = \max\{|x_1|, \ldots, |x_N|\}$. For $0 < p < 1$, the triangle inequality does not hold. The extreme case is $p = 0$, which symbolizes the vector sparsity, i.e., the number of its non-zero entries:

$$\|\mathbf{x}\|_0 = |\{n \mid x_n \neq 0, n = 1, \ldots, N\}|. \tag{1.2}$$

Although conveniently denoted as ℓ_0 norm, sparsity does not satisfy the axiom of absolute homogeneity.

Definition 1.4 (Inner product space [24, Definition 1.57])**.** Inner product space is a vector space V over $\mathbb{F}$, equipped with the inner (scalar) product $\langle\cdot,\cdot\rangle \colon V \times V \to \mathbb{F}$, which satisfies for all $u, v, w \in V$ and for all $\alpha \in \mathbb{F}$ the following axioms:

1. $\langle u + v, w\rangle = \langle u, w\rangle + \langle v, w\rangle$, (additivity in the first argument)
2. $\langle \alpha \cdot u, v\rangle = \alpha\langle u, v\rangle$, (homogeneity in the first argument)
3. $\langle u, v\rangle = \overline{\langle v, u\rangle}$, (conjugate symmetry)
4. $\langle u, u\rangle \geq 0$, while $\langle u, u\rangle = 0 \Leftrightarrow u = 0$. (positive definiteness)

In the case of a space over $\mathbb{R}$, the inner product is real, therefore the third axiom is only symmetry (without conjugation).

Remark 1.5. Every inner product space is also a normed vector space, if we induce the norm from the inner product: $\|u\| = \sqrt{\langle u, u\rangle}$. However, the converse is not true, for example for the norms discussed in example 1.3, only the ℓ_2 norm can be induced by an inner product [25, Exercise 1.32].

Example 1.6. As mentioned in remark 1.1, we are particularly interested in vectors in the form of N-tuples. On the space $\mathbb{R}^N$ or $\mathbb{C}^N$, we usually define the inner product

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^\top \mathbf{x} = \sum_{n=1}^{N} x_n y_n, \text{ or } \langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^* \mathbf{x} = \sum_{n=1}^{N} x_n \overline{y_n}, \tag{1.3}$$

respectively. It can be easily shown that this inner product induces the ℓ_2 norm from example 1.3.

Next, we want to discuss the representation of vectors. The primary possibility is a basis, which allows to express any vector as a unique linear combination of the basis elements. For example in the case of the space $\mathbb{C}^N$, which has dimension N, any basis can be organized as columns of the regular matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_N]$. When we want to express a given vector $\mathbf{x} \in \mathbb{C}^N$ using the basis, we search for the coefficients $c_1, \ldots, c_N$ such that $\mathbf{x} = \sum_{n=1}^{N} c_n \mathbf{b}_n$, or, in matrix form, $\mathbf{x} = \mathbf{Bc}$. Since $\mathbf{B}$ is regular, we can directly compute $\mathbf{c} = \mathbf{B}^{-1}\mathbf{x}$. In the case of orthonormal basis, the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_N$ are of unit length and perpendicular to each other, i.e.,

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = \delta_{i,j} = \begin{cases} 1 & \text{for} \quad i = j, \\ 0 & \text{for} \quad i \neq j. \end{cases} \tag{1.4}$$

For such a basis, it holds $\mathbf{B}^{-1} = \mathbf{B}^*$, which results in $c_n = \mathbf{b}_n^* \mathbf{x} = \langle \mathbf{x}, \mathbf{b}_n \rangle$ and in the reconstruction property [24, Proposition 1.70]

$$\mathbf{x} = \sum_{n=1}^{N} \langle \mathbf{x}, \mathbf{b}_n \rangle \mathbf{b}_n. \tag{1.5}$$

An example of such a situation is the (normalized) discrete Fourier transform (DFT), which expresses a given vector using a basis consisting of complex exponentials – this basis is of the form [24, Example 5.4]

$$\mathbf{b}_n = \frac{1}{\sqrt{N}} \left[W_N^{(k-1)(n-1)} \right]_{k=1,\ldots,N}, \quad \text{where} \quad W_N = \mathrm{e}^{2\pi \mathrm{i}/N} \quad \text{and} \quad n = 1, \ldots, N. \tag{1.6}$$

In some cases, uniqueness of the representation is not needed, or it is even inconvenient, such as in situations where we wish to utilize the sparsity of the representation. For these cases, the so-called frames provide a suitable generalization: Frame is still a complete system (it generates the entire vector space) and allows a simple computation of coordinates for any vector in that space, but the representation obtained does not need to be unique.

Definition 1.7 (Frame [26, Definition 1.1.1]). Let $F = \{f_1, \dots, f_M\}$ be a subset of a vector space V over $\mathbb{F}$ with inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\| \cdot \|$. The set F is a frame for V if there exist constants $0 < A \leq B < \infty$ such that for every $u \in V$ it holds

$$A\|u\|^2 \leq \sum_{m=1}^{M} |\langle u, f_m \rangle|^2 \leq B\|u\|^2. \tag{1.7}$$

The constants A, B are called frame bounds. If there exist A, B such that $A = B$, then F is called tight frame. Furthermore, if $A = B = 1$, F is called Parseval (tight) frame.

There are numerous equivalent characterizations of frames. A crucial property is that thanks to the so-called Parseval inequality (1.7), any frame F for V spans the space V. In the case of tight frames, this is evident also from the following theorem.

Theorem 1.8 (Representation with tight frame [26, Proposition 1.1.4]). *Let $F = \{f_1, \dots, f_M\}$ be a tight frame for V with bounds $A = B$. Then for every $u \in V$ it holds*

$$u = \frac{1}{A} \sum_{m=1}^{M} \langle u, f_m \rangle f_m. \tag{1.8}$$

Comparing (1.8) with (1.5), we see that tight frames provide a very similar representation as bases. The difference is that frames can be redundant, i.e., it can hold $M > N$, where M denotes the number of elements of the frame and N denotes the space dimension.

The frame theory uses the notion of synthesis operator $T \colon \mathbb{F}^M \to V$ and its adjoint, the analysis operator $L \colon V \to \mathbb{F}^M$ [26, p. 4]:

$$T\mathbf{c} = \sum_{m=1}^{M} c_m f_m \quad \forall \mathbf{c} \in \mathbb{F}^M, \tag{1.9a}$$

$$Lu = [\langle u, f_m \rangle]_{m=1,\dots,M} \quad \forall u \in V. \tag{1.9b}$$

Equation (1.8) can be written using these operators as $u = \frac{1}{A} TLu$ for all $u \in V$, or $TL = AI$, where I denotes the identity on the space V.

Example 1.9 (Gabor frame, STFT, see e.g. [27]). This thesis will often make use of the so-called Gabor frames, which are also known by the analysis operator as short-time Fourier transform (STFT). The elements (atoms) of such a frame are formed as translations and modulations of a chosen window function. A careful choice of the window and the values of translation and modulation may lead to the resulting system being a (tight) frame.

Consider the space $\mathbb{C}^L$. A Gabor frame is defined by the window $\mathbf{g} \in \mathbb{C}^L$, assumed to be locally supported and with length $|\mathrm{supp}(\mathbf{g})| = W < L$, translation parameter a (we assume that L is divisible by a) and the number of frequency bins (modulations) F. The coefficients of the STFT of a vector $\mathbf{x} \in \mathbb{C}^L$ can be organized in a matrix $\mathbf{C} \in \mathbb{C}^{F\times(L/a)}$ such that[1]

$$\mathbf{C}(f+1, n+1) = \sum_{l=0}^{L-1} \mathbf{x}(l+1) \underbrace{\overline{\mathbf{g}(l-an+1)}\mathrm{e}^{-2\pi \mathrm{i} lf/F}}_{\text{complex conjugate of } \mathbf{g}_{f,n}(l+1)}, \tag{1.10}$$

where $f = 0, \dots, F-1$ is the modulation index, $n = 0, \dots, L/a-1$ is the translation index and $(l - an)$ is computed modulo L. In line with example 1.6, we can write $\mathbf{C}(f+1, n+1) = \langle \mathbf{x}, \mathbf{g}_{f,n} \rangle$. Atoms of a Gabor frame are illustrated in figure 1.2.

Note that in the matrix form of $\mathbf{C}$, the time-frequency structure of the coefficients is evident. However, it may be practical to consider a vectorized form. For $N = L/a$, we have $\mathbf{C} \in \mathbb{C}^{F\times N}$, which can be also represented in the form of a vector $\mathbf{c} \in \mathbb{C}^{FN}$.

Furthermore, note that equation (1.10) also reveals the connection of STFT with the Fourier transform, that is, a column of the coefficient matrix $\mathbf{C}$ contains the Fourier coefficients of a localized segment of the signal $\mathbf{x}$, weighted by the window $\mathbf{g}$. Denoting $\mathbf{F}$ the (possibly oversampled[2]) DFT and $\mathbf{x}_1, \dots, \mathbf{x}_N$ the (weighted) signal segments, it holds

$$\mathbf{C} = [\mathbf{F}\mathbf{x}_1, \dots, \mathbf{F}\mathbf{x}_N]. \tag{1.11}$$

Remark 1.10. Note that audio signals are commonly processed and/or visualized in the form of spectrogram. Usually, it visualizes the power temporal spectrum of the signal, i.e., the matrix $\mathbf{P} \in \mathbb{R}^{F\times N}$, $p_{fn} = |c_{fn}|^2$, where c_{fn} are individual time-frequency coefficients of the signal considered.

Remark 1.11 (Discrete-time signal formalism)**.** Approaching discrete signals as selections of points of continuous functions, or discrete functions, is not practical when implementing algorithms for signal processing. As a convenient simplification, we pose the following vector representation: A signal $x\colon \{t_1, t_2, \dots, t_L\} \to \mathbb{R}$ will be identified with a vector[3] $\mathbf{x} = [x(t_1), x(t_2), \dots, x(t_L)]^\top \in \mathbb{R}^L$. When working with a vector $\mathbf{x}$, one should keep in mind the underlying temporal information about the samples. For example, consider a signal $x\colon T \to \mathbb{R}$ and let us select two parts $x_1 = x|_{T_1}$ and $x_2 = x|_{T_2}$ of the same length, i.e., $T_1, T_2 \subset T$ and $|T_1| = |T_2|$. Then the vectors $\mathbf{x}_1$ a $\mathbf{x}_2$ are elements of the *same* vector space $\mathbb{R}^{|T_1|}$, even though from the perspective of discrete functions, x_1 and x_2 are functions of different time instances T_1 and T_2.

[1] http://ltfat.org/doc/gabor/dgt.html

[2] For example, the twice redundant DFT can be implemented as padding the signal to twice its length and then computing DFT. The backward transform is the inverse DFT, followed by cropping the result to the original length.

[3] Note that quantization is not considered in this thesis. If it was the case, signals should be elements of a discrete space, such as $\{-32\,768, \dots, 32\,767\}^L$ for 16-bit quantization.

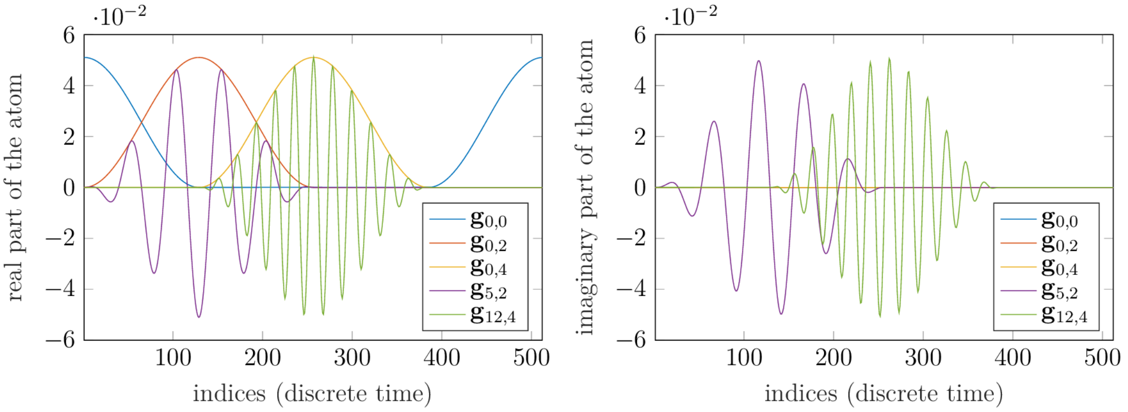


Fig. 1.2: Illustration of a Gabor frame. Atoms from equation (1.10) are depicted for a particular choice of vector length $L = 512$, Hann window of length $W = 256$, window shift (translation) $a = 256$ and $F = 256$ frequency bins. The imaginary part of any atom $\mathbf{g}_{0,n}$ is zero, which corresponds to equation (1.10) with the choice of a real window function $\mathbf{g}$ and $f = 0$.

Remark 1.12 (Real versus complex signals)**.** Since monophonic audio signals are considered in this thesis, it is natural to identify the *real* signal space with $\mathbb{R}^L$. However, as will be mentioned below, linear operators are usually defined between spaces over the same field, either $\mathbb{R}$ or $\mathbb{C}$. This is inconvenient, since we shall make use of the time-frequency representations, in particular using Gabor frames, which utilize *complex* coefficients, see example 1.9. This theoretical inconsistency can be formally treated in two ways:

1. Signals can be considered complex, possibly with the constraint of zero imaginary part. This will be the case in the Matlab implementation.
2. The coefficient space can be considered over real numbers (i.e., with $\mathbb{R}$ as the scalars). In particular, in case of DFT or STFT, we may further assume conjugate symmetry of the spectra (in the frequency direction).

In order to simplify the notation, signals will be considered real throughout the thesis, and the DFT or STFT coefficients complex without further elaboration.

1.1.2 Operators

In finite-dimensional vector spaces, where vectors correspond to N-tuples, any linear operation (transformation) is identifiable with multiplication by a suitable matrix. And vice versa: Any matrix $\mathbf{T} \in \mathbb{C}^{M\times N}$ defines a linear mapping $\mathbb{C}^N \to \mathbb{C}^M$ by setting $\mathbf{x} \mapsto \mathbf{Tx}$ for any $\mathbf{x} \in \mathbb{C}^N$ [24, Proposition 1.43]. More generally, a linear operator is defined as follows:

Definition 1.13 (Linear, continuous operator [24, Definitions 1.30, 2.69])**.** Let V, W be two vector spaces over $\mathbb{F}$. Operator $T\colon V \to \mathcal{R}(T) \subseteq W$, where $\mathcal{R}(T)$ denotes the range space of T, is called linear, or antilinear (conjugate linear), if for all $u, v \in V$ and for all $\alpha \in \mathbb{F}$, it holds

$$T(u+v) = T(u) + T(v) \quad \text{and} \quad T(\alpha \cdot u) = \alpha \cdot T(u), \text{ or } T(\alpha \cdot u) = \overline{\alpha} \cdot T(u). \tag{1.12}$$

Operator T is called continuous (or bounded), if there exists $C > 0$ such that $\|Tu\| \leq C\|u\|$ for all $u \in V$.[4]

It can be shown that the set of all bounded operators forms a vector space. In this thesis, we will only make use of the notion of operator norm, which itself is a norm (according to 1.2) on the space of operators.

Definition 1.14 (Operator norm [24, Definition 2.69, Lemma 2.70])**.** Let $T \colon V \to W$ be a bounded linear operator between two normed spaces. We define the norm of T as

$$\|T\| = \inf\{C \mid C > 0 : \|Tu\| \leq C\|u\| \; \forall u \in V\}. \tag{1.13}$$

Equivalently, the norm can be defined as $\|T\| = \sup_{\|u\|=1} \|Tu\|$, which can be seen as a measure of how much the transformation T deforms the unit ball $\{u \in V \mid \|u\| = 1\}$. Since in finite-dimensional spaces, linear operators have one-to-one correspondence to matrices (or, more precisely with multiplication of vectors by a matrix, see [24, Proposition 1.43], we can use definition 1.14 also to compute (operator) norms of matrices (besides e.g. the Frobenius norm mentioned in example 1.3).

In many cases, we want to express a *backwards* operation to a given transformation $T \colon V \to W$, i.e., some transformation $W \to V$ related to T. In the following chapters, we will make use of three different possibilities – the adjoint, inverse and pseudoinverse operator.

Definition 1.15 (Adjoint operator [24, Proposition 2.7])**.** Let $T \colon V \to W$ be a bounded linear operator between two inner product spaces. The linear operator $T^* \colon W \to V$ which for all $u, v \in V$ satisfies

$$\langle Tu, v \rangle = \langle u, T^* v \rangle \tag{1.14}$$

is called the operator adjoint to T.

The inverse of a operator $T \colon V \to W$ can be seen intuitively as such operator $T^{-1} \colon W \to V$, which, when composed with T, forms the identity (on V in the case

[4] Note that $\|u\|$ is a norm on the space V, whereas $\|Tu\|$ is a norm on the space W. It should be clear from the context which norm is used, which is why it will usually not be distinguished by notation.

$T^{-1}T$ and on W in the case of TT^{-1}). In other notation, $v = Tu$ is equivalent to $u = T^{-1}v$. However, the inverse does not always exist. If this is the case, the concept of pseudoinverse provides a convenient generalization.

Definition 1.16 (Pseudoinverse operator [28, Chapter 1.1])**.** Let $T\colon V \to W$ be a bounded linear operator between two inner product spaces. The operator $T^+\colon W \to V$ is called the pseudoinverse (Moore–Penrose inverse) of the operator T if it satisfies:

1. $TT^+T = T$,
2. $T^+TT^+ = T^+$,
3. $(TT^+)^* = TT^+$,
4. $(T^+T)^* = T^+T$.

Even though the formal definitions may be rather complicated, the situation is much simpler in the case of matrix operators: Consider an operator defined by a matrix $\mathbf{T}$. Then, the adjoint operator corresponds to the Hermitian transpose of the original matrix, i.e., $\mathbf{T}^*$ (only transpose in the case of real spaces, i.e., $\mathbf{T}^\top$). Similarly the inverse operator is defined as multiplication with the inverse matrix $\mathbf{T}^{-1}$. From this matrix case, one can also observe that the adjoint (and also pseudoinverse) always exist, but it is not the case of the inverse, which needs the matrix $\mathbf{T}$ to be regular. If the columns of $\mathbf{T}$ are linearly independent, it can be shown that the matrix $\mathbf{T}^*\mathbf{T}$ is regular, and we can express the pseudoinverse explicitly as $\mathbf{T}^+ = (\mathbf{T}^*\mathbf{T})^{-1}\mathbf{T}^*$.

The pseudoinverse is closely related to solutions of systems of linear equations in the form $\mathbf{A}\mathbf{x} = \mathbf{b}$, where the right-hand side $\mathbf{b} \in \mathbb{C}^M$ and the matrix $\mathbf{A} \in \mathbb{C}^{M\times N}$ are given and we search for $\mathbf{x} \in \mathbb{C}^N$. If the matrix $\mathbf{A}$ is regular, the system has a unique solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. If $\mathbf{A}$ is singular, the system either has infinitely many solutions (if $\mathbf{b}$ lies in the range space of $\mathbf{A}$, i.e., the set of linear combinations of the columns of $\mathbf{A}$), or no solution. Then, by putting $\hat{\mathbf{x}} = \mathbf{A}^+\mathbf{b}$ we minimize the divergence from the right-hand size:

$$\|\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\|_2 = \min_{\mathbf{x}\in\mathbb{C}^N} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2, \tag{1.15}$$

while from all $\mathbf{x}$ satisfiying (1.15), $\hat{\mathbf{x}}$ has itself minimal norm [28, Corollary 3, p. 109]. Note that this covers also the case when the system has infinitely many solutions and $\min_{\mathbf{x}\in\mathbb{C}^N} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 = 0$.

Remark 1.17 (Projection onto the range space)**.** Properties of the pseudoinverse allow to express the projection onto the range space of a linear operator $\mathbf{A}$ (see also definition 1.13 for a formal definition of the projection on a convex set) as

$$\operatorname{proj}_{\mathcal{R}(\mathbf{A})}(\mathbf{x}) = \mathbf{A}\mathbf{A}^+\mathbf{x} = \mathbf{A}(\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*\mathbf{x}. \tag{1.16}$$

The second equality holds if the columns of $\mathbf{A}$ are linearly independent, as discussed above. In particular, if $\mathbf{A}$ is the analysis operator of a Parseval tight frame, it holds $\mathbf{A}^*\mathbf{A} = \mathbf{I}$, which simplifies (1.16) to $\operatorname{proj}_{\mathcal{R}(\mathbf{A})}(\mathbf{x}) = \mathbf{A}\mathbf{A}^*\mathbf{x}$.

1.2 Optimization

Assume hereinafter that V is a vector space, equipped with a norm $\|\cdot\|$ and an inner product $\langle\cdot,\cdot\rangle$ if needed, and $f\colon V \to \mathbb{R}\cup\{\infty\}$ is a function with domain $\mathcal{D}(f) = \{u \in V \mid f(u) < \infty\} \subseteq V$.

An optimization problem is usually posed in one of the following forms:

$$m = \min_x f(x) \quad \text{subject to} \quad x \in C \tag{1.17a}$$

$$\hat{x} \in \operatorname*{arg\,min}_{x\in C} f(x), \tag{1.17b}$$

where $f(x)$ is referred to as the objective function. In the case (1.17a) we search for the minimum m of the function $f(x)$ over the set $C \subset V$. The constraint $x \in C$ might not be present, then we consider every $x \in \mathcal{D}(f)$. In the case (1.17b) we search for such $\hat{x}$ in which $f(x)$ attains its minimum over the set C. Using the notation of (1.17b), it holds $f(\hat{x}) = m$. Since this minimum can be attained at multiple points, the argument of the minima is generally a set. However, in most applications, the particular choice of an element of this set is not crucial, which is why we write simply $\hat{x} = \operatorname{arg\,min}_{x\in C} f(x)$.

Modern optimization-based signal processing applications usually rely on convex optimization, i.e., searching for a minimum of a convex function over a convex set. For the sake of completeness, we hereby recall definitions of these basic terms:

Definition 1.18 (Convex set [29, p. 2.1.4])**.** A set $C \subseteq V$ is convex if for all $u, v \in C$ and $0 \leq t \leq 1$, it holds $tu + (1-t)v \in C$.

Definition 1.19 (Convex function [29, p. 3.1.1])**.** A function $f\colon V \to \mathbb{R}$ is convex if its domain $\mathcal{D}(f)$ is a convex set and if for all $u, v \in \mathcal{D}(f)$ and $0 \leq t \leq 1$ it holds

$$f(tu + (1-t)v) \leq tf(u) + (1-t)f(v). \tag{1.18}$$

One of the key notions in convex optimization is also the (convex) conjugate of a given function f, sometimes also called the Fenchel–Legendre transform of f.

Definition 1.20 (Convex conjugate function [29, p. 3.3.1])**.** A (convex) conjugate of a function $f\colon V \to \mathbb{R}$ is the function $f^*\colon V \to \mathbb{R}$ defined point-wise as

$$f^*(y) = \sup_{x\in\mathcal{D}(f)} \left(\langle x, y\rangle - f(x)\right). \tag{1.19}$$

It can be shown that point-wise supremum of convex functions is a convex function. Since for arbitrary fixed $x \in V$ the function $f_x(y) = \langle x, y\rangle - f(x)$ is affine and thus convex (because $f(x)$ is just a number for fixed x), then also $f^*(y) = \sup_{x\in\mathcal{D}(f)} f_x(y)$ is convex, which justifies the term *convex* conjugate. Let us emphasize here that the convexity of f^* is independent on the (non)convexity of f.

Remark 1.21. Due to the presence of the supremum function, the expression (1.19) is well defined only for a real space V. In the complex case, we may define a real inner product for $u, v \in V$ as the real part of a given complex inner product on V, i.e., $\Re(\langle u, v\rangle)$. However, it will be a true inner product by definition 1.4 only if we limit ourselves to real scalars due to the homogeneity property. Alternative definitions of conjugate function for complex spaces exist [30]; such a generality is nonetheless out of scope of this thesis.

In signal processing, the so-called regularized inverse problems are of special importance. This refers to a situation where we observe a signal that has been degraded or transformed in some way and from this observation we want to estimate its original form. A special case is the observation given by a linear operator, which can be put in the context of systems of linear equations. For a given matrix $\mathbf{A}$, observed signal $\mathbf{b}$ and the unknown original $\mathbf{x}$, the following scheme applies:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \begin{cases} \text{linear observation model of } \mathbf{x} \\ \text{system of linear equations} \begin{cases} \exists\mathbf{A}^{-1} & \text{unique solution} \\ \nexists\mathbf{A}^{-1} & \begin{cases} \infty \text{ solutions, one must be chosen} \\ \text{no solution} \end{cases} \end{cases} \end{cases}$$

By the end of the section 1.1.2 we have mentioned the pseudoinverse operator as a practical approach to systems of linear equations with no unique solution, see equation (1.15) and the comments thereby.

However, the pseudoinverse always chooses the “smallest” solution, which may not be the most suitable one in practice. That is why we bring forward the concept of regularization, i.e., a function $R(\mathbf{x})$, which allows to mathematically distinguish between “good” and “bad” solutions, usually such that $R(\mathbf{x})$ attains small values for favorable choices of $\mathbf{x}$. A possible regularization option, related to the pseudoinverse, is the norm $R(\mathbf{x}) = \|\mathbf{x}\|$. If we similarly generalize the measure of divergence from the (not necessarily linear) observation, we arrive at the following generic regularized inverse problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} E(\mathbf{x}, \mathbf{b}) + R(\mathbf{x}). \tag{1.20}$$

The term $E(\mathbf{x}, \mathbf{b})$ ensures that the solution $\hat{\mathbf{x}}$ is in line with the observation $\mathbf{b}$. In the case of linear observation model, the function E can ensure satisfying the equality

$\mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$, if we can guarantee existence of at least one such solution. Otherwise, we can require the solution to be "close" to the right-hand side, e.g. by setting $E(\mathbf{x}, \mathbf{b}) = \|\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\|_2$.

1.2.1 Proximal operators

In this section, we introduce the concept of a proximal operator (corresponding to a given function), which turns out to be very practical for minimizing non-differentiable functions. Already in his original work, Jean-Jacques Moreau introduced the proximal operator as a generalization of the projection onto a closed subset or a subspace of a vector space [31] (see also the table 1.1 with selected examples). There are also connections between the proximal operator and the gradient of a differentiable function, see e.g. [32, Thm. 6.39], but these are less intuitive.

In the following, we will not discuss in detail all properties of functions needed for the definitions to be reasonable. We only denote $\mathcal{F}(V)$ the set of convex lower semicontinuous[5] functions on V with non-empty domain (from optimization point of view, functions that are not identically equal ∞). The symbol V denotes a real finite-dimensional inner product space, which can be without loss of generality identified with the space $\mathbb{R}^N$ of a desired dimension N, since the only difference is a particular choice of basis. Similarly, a linear operator A as a mapping from a space of dimension N to a space of dimension M can be identified with a matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, as previously mentioned in section 1.1.2. Let us remark that in some cases, the operators and algorithms from this and the subsequent section 1.2.2 are defined also in more general spaces, see e.g. [33, 34]. However, the applications considered in this thesis will only demand finite dimension.

Definition 1.22 (Proximal operator [32, Definition 6.1])**.** Given a function $f \in \mathcal{F}(V)$, the proximal mapping of f is the operator $\operatorname{prox}_f \colon V \to V$ given by

$$\operatorname{prox}_f(u) = \underset{v \in V}{\arg\min}\, f(v) + \tfrac{1}{2}\|v - u\|^2 \quad \forall u \in V. \tag{1.21}$$

Even though for a generic function f the minimization in (1.21) can have multiple solutions, the condition $f \in \mathcal{F}(V)$ ensures that the solution is unique and thus the proximal operator is well defined [32, thm. 6.3]. Particular examples of proximal operators relevant to this thesis are given in table 1.1.

Recall that the conjugate function was defined in definition 1.20, and the indi-

[5] see e.g. [32, Definition 2.5]

Tab. 1.1: Proximal operator of selected functions. In all cases it holds $f \in \mathcal{F}(\mathbb{R}^N)$, while assuming $g \in \mathcal{F}(\mathbb{R}^M)$, $\mathbf{A} \in \mathbb{R}^{M \times N}$, $\mathbf{y} \in \mathbb{R}^M$ and $\gamma > 0$.

function	$f(\mathbf{x})$	$\operatorname{prox}_f(\mathbf{x})$	reference
indicator	$I_C(\mathbf{x})$, $C \neq \emptyset$ closed	$\operatorname{proj}_C(\mathbf{x})$	[32, thm. 6.24]
quadratic	$\frac{\gamma}{2}\lVert\mathbf{A}\mathbf{x}-\mathbf{y}\rVert_2^2$	$(\mathbf{I}+\gamma\mathbf{A}^\top\mathbf{A})^{-1}(\mathbf{x}+\gamma\mathbf{A}^\top\mathbf{y})$	[35, tab. I]
ℓ_1 norm	$\gamma\lVert\mathbf{x}\rVert_1$	$\operatorname{soft}_\gamma(\mathbf{x})$	[32, ex. 6.8]
linear	$g(\mathbf{A}\mathbf{x})$, $\mathbf{A}\mathbf{A}^\top = \gamma\mathbf{I}$	$\mathbf{x}+\gamma^{-1}\mathbf{A}^\top\big(\operatorname{prox}_{\gamma g}(\mathbf{A}\mathbf{x})-\mathbf{A}\mathbf{x}\big)$	[32, thm. 6.15]
composition	$g(\mathbf{A}\mathbf{x})$, $\mathbf{A}^\top\mathbf{A} = \gamma\mathbf{I}$	$\gamma^{-1}\mathbf{A}^\top\operatorname{prox}_{\gamma g+I_{\mathcal{R}(\mathbf{A})}}(\mathbf{A}\mathbf{x})$	[1, lem. 2]
conjugate	$\gamma f^*(\mathbf{x})$	$\mathbf{x}-\gamma\operatorname{prox}_{\gamma^{-1}f}(\gamma^{-1}\mathbf{x})$	[32, thm. 6.45]

cator function of a set $C \subseteq V$ is defined as

$$I_C(u) = \begin{cases} 0 & \text{if } u \in C, \\ \infty & \text{if } u \notin C. \end{cases} \tag{1.22}$$

The proximal operator is thus indeed a generalization of the projection operator, since

$$\operatorname{proj}_C(u) = \operatorname*{arg\,min}_{v \in C} \tfrac{1}{2}\lVert v-u\rVert^2 = \operatorname*{arg\,min}_{v \in V} I_C(v) + \tfrac{1}{2}\lVert v-u\rVert^2 = \operatorname{prox}_{I_C}(u). \tag{1.23}$$

For a space $V = \mathbb{R}$ the soft thresholding as a proximal operator of the absolute value has the following form [32, lemma 6.5]:

$$\operatorname{prox}_{\gamma|\cdot|}(x) = \operatorname{soft}_\gamma(x) = \operatorname{sgn}(x)\max(|x|-\gamma, 0). \tag{1.24}$$

For the space $V = \mathbb{R}^N$, the soft thresholding operates entrywise and it is not distinguished in notation from the scalar version:

$$\operatorname{soft}_\gamma(\mathbf{x}) = [\operatorname{soft}_\gamma(x_1), \ldots, \operatorname{soft}_\gamma(x_N)]^\top. \tag{1.25}$$

1.2.2 Algorithms

In this section, we present several algorithms for solving optimization problems of a specific form, where we minimize the sum of several “simple” functions. In the case where these functions have a known proximal operator, we are able to minimize such a sum alternately with respect to the individual functions using the corresponding proximal operators and appropriate extrapolation steps.

Douglas–Rachford algorithm

Assume a generic optimization problem in the form

$$\hat{\mathbf{x}} = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^N} f(\mathbf{x}) + g(\mathbf{x}), \tag{1.26}$$

where $f, g \in \mathcal{F}(\mathbb{R}^N)$, $\mathrm{ri}(\mathcal{D}(f)) \cap \mathrm{ri}(\mathcal{D}(g)) \neq \emptyset$ a $f(\mathbf{x}) + g(\mathbf{x}) \to \infty$ if $\|\mathbf{x}\| \to \infty$. The Douglas–Rachford algorithm (DRA) 1.1 generates the sequence $\mathbf{x}^{(n)}$ which converges to the solution of the problem (1.26) [33, corollary 21], [35, proposition 9].

Algorithm 1.1: Douglas–Rachford algorithm [35, p. IV.]

```
input: $\varepsilon \in [0,1]$, $\gamma > 0$, $\mathbf{y}^{(0)} \in \mathbb{R}^N$
for $n = 1, 2, \dots$ do
    $\mathbf{x}^{(n)} = \mathrm{prox}_{\gamma g}(\mathbf{y}^{(n-1)})$
    choose $\lambda^{(n)} \in [\varepsilon, 2-\varepsilon]$
    $\mathbf{y}^{(n)} = \mathbf{y}^{(n-1)} + \lambda^{(n)}\left(\mathrm{prox}_{\gamma f}(2\mathbf{x}^{(n)} - \mathbf{y}^{(n-1)}) - \mathbf{x}^{(n)}\right)$
end
output: $\hat{\mathbf{x}} = \mathbf{x}^{(n)}$
```

Chambolle–Pock algorithm

In numerous applications, we need to optimize problems where a function is composed with a linear operator. In some cases we know the proximal operator of such a composition (see table 1.1). For arbitrary linear operator, the problem of the form

$$\hat{\mathbf{x}} = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^N} f(\mathbf{A}\mathbf{x}) + g(\mathbf{x}) \tag{1.27}$$

can be solved using Chambolle–Pock algorithm (CPA), also known as primal–dual algorithm. A generic form is in algorithm 1.2.

Algorithm 1.2: Chambolle–Pock algorithm [36]

```
input: $\tau$, $\sigma > 0$, $\theta \in [0,1]$, $\mathbf{x}^{(0)} \in \mathbb{R}^N$, $\mathbf{y}^{(0)} \in \mathbb{R}^M$, $\bar{\mathbf{x}}^{(0)} = \mathbf{x}^{(0)}$
for $n = 1, 2, \dots$ do
    $\mathbf{y}^{(n)} = \mathrm{prox}_{\sigma f^*}(\mathbf{y}^{(n-1)} + \sigma\mathbf{A}\bar{\mathbf{x}}^{(n-1)})$
    $\mathbf{x}^{(n)} = \mathrm{prox}_{\tau g}(\mathbf{x}^{(n-1)} - \tau\mathbf{A}^*\mathbf{y}^{(n)})$
    $\bar{\mathbf{x}}^{(n)} = \mathbf{x}^{(n)} + \theta(\mathbf{x}^{(n)} - \mathbf{x}^{(n-1)})$
end
output: $\hat{\mathbf{x}} = \mathbf{x}^{(n)}$
```

Assume that $g \in \mathcal{F}(\mathbb{R}^N)$, $f, f^* \in \mathcal{F}(\mathbb{R}^M)$, while f^* is the conjugate of f (see definition 1.20), and $\mathbf{A} \in \mathbb{R}^{M \times N}$ represents a linear operator $\mathbb{R}^N \to \mathbb{R}^M$. If[6] $\theta = 1$ and $\tau\sigma\|\mathbf{A}\|^2 \leq 1$ (see definition 1.14), then the sequence $\mathbf{x}^{(n)}$ generated by the algorithm 1.2 converges to the solution of the problem (1.27) [36].

ADMM

Problem (1.27) can be treated alternatively, if we introduce a new variable $\mathbf{z}$, tied by the constraint $\mathbf{z} = \mathbf{A}\mathbf{x}$, and solve for both $\mathbf{x}$ and $\mathbf{z}$. The equivalent reformulation of (1.27) is

$$(\hat{\mathbf{x}}, \hat{\mathbf{z}}) = \operatorname*{arg\,min}_{\mathbf{x} \in \mathbb{R}^N, \mathbf{z} \in \mathbb{R}^M} f(\mathbf{z}) + g(\mathbf{x}) \quad \text{subject to } \mathbf{A}\mathbf{x} - \mathbf{z} = \mathbf{0}. \tag{1.28}$$

Problem in the form of (1.28) can be solved using a variant of the Lagrange multiplier method known as Alternating Direction Method of Multipliers (ADMM). The method is derived using the augmented Lagrangian [38, p. 2.3]

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{z}) + g(\mathbf{x}) + \mathbf{y}^\top(\mathbf{A}\mathbf{x} - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{z}\|^2 \tag{1.29}$$

with multipliers $\mathbf{y} \in \mathbb{R}^M$, or equivalently with $\mathbf{u} = \mathbf{y}/\rho$ in the (scaled) form

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{u}) = f(\mathbf{z}) + g(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{z} + \mathbf{u}\|^2 - \frac{\rho}{2}\|\mathbf{u}\|^2. \tag{1.30}$$

ADMM then consists of the minimization of L_ρ with respect to $\mathbf{x}$ and $\mathbf{z}$ and of a suitable update of the multipliers $\mathbf{u}$. This is summarized by the algorithm 1.3.

Algorithm 1.3: ADMM [38, p. 3.1.1]

```
input: $\mathbf{z}^{(0)} \in \mathbb{R}^M$, $\mathbf{u}^{(0)} \in \mathbb{R}^M$, $\rho > 0$
for $n = 1, 2, \dots$ do
    $\mathbf{x}^{(n)} = \arg\min_{\mathbf{x} \in \mathbb{R}^N} g(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{z}^{(n-1)} + \mathbf{u}^{(n-1)}\|^2$
    $\mathbf{z}^{(n)} = \arg\min_{\mathbf{z} \in \mathbb{R}^M} f(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x}^{(n)} - \mathbf{z} + \mathbf{u}^{(n-1)}\|^2$
    $\mathbf{u}^{(n)} = \mathbf{u}^{(n-1)} + \mathbf{A}\mathbf{x}^{(n)} - \mathbf{z}^{(n)}$
end
output: $\hat{\mathbf{x}} = \mathbf{x}^{(n)}$, $\hat{\mathbf{z}} = \mathbf{z}^{(n)}$
```

Denote $(\mathbf{x}^*, \mathbf{z}^*)$ the solution of the problem (1.28) and $p^* = f(\mathbf{z}^*) + g(\mathbf{x}^*)$ the optimal value of the objective function. Then, under rather mild conditions[7], one can show the convergence of the residuals $\mathbf{r}^{(n)} = \mathbf{A}\mathbf{x}^{(n)} - \mathbf{z}^{(n)} \to \mathbf{0}$, the objective

[6] Even though the original paper proves convergence for $\tau\sigma\|\mathbf{A}\|^2 < 1$, the finite-dimensional case allows also equality in this constraint [37, theorem 3.3].

[7] Remarkably when $f \in \mathcal{F}(\mathbb{R}^M)$ and $g \in \mathcal{F}(\mathbb{R}^N)$ and when the (unaugmented) Lagrangian L_0 has a saddle point.

function $f(\mathbf{z}^{(n)}) + g(\mathbf{x}^{(n)}) \to p^*$, and the dual variable $\mathbf{z}^{(n)} \to \mathbf{z}^*$ [38, p. 3.2.1]. For the convergence of the primal variable $\mathbf{x}^{(n)} \to \mathbf{x}^*$, additional conditions need to be satisfied. An example of a sufficient condition is that $\mathbf{A}^\top\mathbf{A}$ is a regular matrix and $\mathrm{ri}(\mathcal{D}(f)) \cap \mathbf{A}\,\mathrm{ri}(\mathcal{D}(g)) \neq \emptyset$ [35, p. VI.D].

Remark 1.23 (Proximal form of ADMM)**.** Step 2 can be reformulated using the proximal operator as $\mathbf{x}^{(n)} = \mathrm{prox}_{g/\rho}(\mathbf{A}^\top(\mathbf{z}^{(n-1)} - \mathbf{u}^{(n-1)}))$, if $\mathbf{A}^\top\mathbf{A} = \mathbf{I}$, in analogy with [2, remark 3]. Directly from definition 1.22, step 3 has a proximal form $\mathbf{z}^{(n)} = \mathrm{prox}_{f/\rho}(\mathbf{A}\mathbf{x}^{(n)} + \mathbf{u}^{(n-1)})$.

Remark 1.24 (Complex form of ADMM)**.** ADMM can be used also if f and g in equation (1.28) are real functions of complex variables. Formally, however, in such a case the augmented Lagrangian must be reformulated so that it remains a real function. Instead of equation (1.29) we use a novel definition

$$L_\rho(\mathbf{x},\mathbf{z},\mathbf{y}) = f(\mathbf{z}) + g(\mathbf{x}) + \mathbf{y}^\top \begin{bmatrix} \Re(\mathbf{A}\mathbf{x}-\mathbf{z}) \\ \Im(\mathbf{A}\mathbf{x}-\mathbf{z}) \end{bmatrix} + \frac{\rho}{2} \left\| \begin{bmatrix} \Re(\mathbf{A}\mathbf{x}-\mathbf{z}) \\ \Im(\mathbf{A}\mathbf{x}-\mathbf{z}) \end{bmatrix} \right\|_2^2, \tag{1.31}$$

where for $\mathbf{x} \in \mathbb{C}^N$, $\mathbf{z} \in \mathbb{C}^M$ and $\mathbf{A} \in \mathbb{C}^{M\times N}$ the dual variable is $\mathbf{y} \in \mathbb{R}^{2M}$. For this formulation, we have the same augmented Lagrangian (1.30), only with the difference that $\mathbf{u} \in \mathbb{R}^{2M}$. In practice, this modification is not necessary and we can work with complex vectors, because for any complex vector $\mathbf{c}$, it is easy to show that

$$\|\mathbf{c}\|_2^2 = \left\| \begin{bmatrix} \Re(\mathbf{c}) \\ \Im(\mathbf{c}) \end{bmatrix} \right\|_2^2. \tag{1.32}$$

The subproblems on lines 2 and 3 in algorithm 1.3 can remain the same also for complex variables and step 4 consists in summation of complex vectors, which is equivalent to summing the real and imaginary parts separately.

Remark 1.25. For solving problems of type (1.27), which may also contain further summands and different linear operators, there exist more complex algorithms. An example is the proximal algorithm introduced independently by Condat and Vũ [39, 40]. In other cases, it is possible to convert complex problems by appropriate combination functions to a problem solvable by one of the algorithms mentioned above, see e.g. [3]. It can also be advantageous to use the properties of proximal operators for separable functions [41, 42, 43].

Non-negative matrix factorization

A different, but for this thesis relevant problem, is NMF. The goal is to factorize a non-negative matrix $\mathbf{V} \in \mathbb{R}^{M\times N}$ as a multiplication of two non-negative matrices

$\mathbf{W} \in \mathbb{R}^{M\times K}$ and $\mathbf{H} \in \mathbb{R}^{K\times N}$. If we allow some deviation of $\mathbf{V}$ from the product $\mathbf{WH}$, then NMF solves the problem

$$(\hat{\mathbf{W}}, \hat{\mathbf{H}}) = \operatorname*{arg\,min}_{\substack{\mathbf{W}\in\mathbb{R}^{M\times K}\\ \mathbf{H}\in\mathbb{R}^{K\times N}}} D(\mathbf{V} \mid \mathbf{WH}) \quad \text{subject to } w_{mk} \geq 0, h_{kn} \geq 0 \; \forall m, n, k, \tag{1.33}$$

where $D(\mathbf{A} \mid \mathbf{B})$ measures the divergence between two matrices $\mathbf{A}$ and $\mathbf{B}$. An example of such a measure is the square of the Euclidean distance $\|\mathbf{A} - \mathbf{B}\|_{\mathrm{F}}^2 = \sum_{i,j} |a_{ij} - b_{ij}|^2$ or the so-called Itakura–Saito divergence

$$D_{\mathrm{IS}}(\mathbf{A} \mid \mathbf{B}) = \sum_{i,j} d_{\mathrm{IS}}(a_{ij} \mid b_{ij}) = \sum_{i,j} \Big(\frac{a_{ij}}{b_{ij}} - \log \frac{a_{ij}}{b_{ij}} - 1 \Big), \tag{1.34}$$

both being special cases of β-divergence [44, p. 2.1]. The problem (1.33) is not convex (only for fixed $\mathbf{W}$ or fixed $\mathbf{H}$), but algorithms exist which allow to find at least a local optimum [45, 44]. For the case featuring Itakura–Saito divergence a possible solver is in algorithm 1.4, where $\odot$ denotes entrywise operations (multiplication, power) and also the fraction is understood as entry-wise operation ($\frac{\mathbf{A}}{\mathbf{B}} = \mathbf{A} \odot \mathbf{B}^{\odot[-1]}$).

Algorithm 1.4: NMF [44, Alg. 1]

```
input: non-negative matrices $\mathbf{W}^{(0)} \in \mathbb{R}^{M\times K}$ and $\mathbf{H}^{(0)} \in \mathbb{R}^{K\times N}$
for $n = 1, 2, \dots$ do
    $\mathbf{W}^{(n)} = \mathbf{W}^{(n-1)} \odot \dfrac{\big((\mathbf{W}^{(n-1)}\mathbf{H}^{(n-1)})^{\odot[-2]} \odot \mathbf{V}\big)\,(\mathbf{H}^{(n-1)})^\top}{(\mathbf{W}^{(n-1)}\mathbf{H}^{(n-1)})^{\odot[-1]}(\mathbf{H}^{(n-1)})^\top}$
    $\mathbf{H}^{(n)} = \mathbf{H}^{(n-1)} \odot \dfrac{(\mathbf{W}^{(n)})^\top\,\big((\mathbf{W}^{(n)}\mathbf{H}^{(n-1)})^{\odot[-2]} \odot \mathbf{V}\big)}{(\mathbf{W}^{(n)})^\top(\mathbf{W}^{(n)}\mathbf{H}^{(n-1)})^{\odot[-1]}}$
end
output: $\hat{\mathbf{W}} = \mathbf{W}^{(n)}$, $\hat{\mathbf{H}} = \mathbf{H}^{(n)}$
```

For higher numerical stability, it is possible to normalize the columns of the matrix $\mathbf{W}^{(n)}$ after each iteration and multiply the rows of the matrix $\mathbf{H}^{(n)}$ proportionally, such that the product is preserved.

Remark 1.26. The multiplicative updates of algorithm 1.4 may be problematic due to possible division by zero. To overcome this issue, a relaxed problem can be solved instead of (1.33):

$$(\hat{\mathbf{W}}, \hat{\mathbf{H}}) = \operatorname*{arg\,min}_{\substack{\mathbf{W}\in\mathbb{R}^{M\times K}\\ \mathbf{H}\in\mathbb{R}^{K\times N}}} D(\mathbf{V} \mid \mathbf{WH} + \varepsilon) \quad \text{subject to } w_{mk} \geq 0, h_{kn} \geq 0 \; \forall m, n, k, \tag{1.35}$$

where $\varepsilon > 0$ is a small relaxation parameter. This leads to the relaxed updates

(iteration counter is omitted for brevity)

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{\left((\mathbf{WH} + \varepsilon)^{\odot[-2]} \odot \mathbf{V}\right) \mathbf{H}^\top}{(\mathbf{WH} + \varepsilon)^{\odot[-1]} \mathbf{H}^\top}, \tag{1.36a}$$

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^\top \left((\mathbf{WH} + \varepsilon)^{\odot[-2]} \odot \mathbf{V}\right)}{\mathbf{W}^\top (\mathbf{WH} + \varepsilon)^{\odot[-1]}}, \tag{1.36b}$$

as can be derived from equations (8)–(12) in [44].

2 State of the art

For easier orientation in the described methods, we first introduce the interpolation problem in section 2.1 using the formal concepts introduced in the previous chapter. We then discuss the main classes of methods from current literature which achieve state-of-the-art results in audio signal reconstruction. These are mainly the methods based on autoregressive modeling (section 2.2), sparsity of the time-frequency representation (section 2.3), self-similarity (section 2.4), and methods based on deep learning (section 2.5).

2.1 Generic task of interpolating missing segments in an audio signal

Consider an unknown undamaged signal $\mathbf{x}^{\text{true}} \in \mathbb{R}^L$. Assume that there is a sample dropout in this signal, where some samples are considered known (reliable) and the remaining samples are to be estimated. We formalize this split using the set of indices of known samples M and its complement $\overline{M} = \{1, \dots, L\} \setminus M$. The sample dropout can be formalized as a restriction via the operator $D\colon \mathbb{R}^L \to \mathbb{R}^{|M|}$, $D(\mathbf{x}^{\text{true}}) = \mathbf{x}^{\text{true}}(M)$. Recall that, in accordance with remark 1.11, the restriction respects the underlying temporal localization of the reliable samples.

Denote $\mathbf{x}^{\text{obs}} = D(\mathbf{y}) = \mathbf{x}^{\text{true}}(M) \in \mathbb{R}^{|M|}$ the observed signal and let us define the set of feasible interpolated signals as[1]

$$\mathit{\Gamma} = D^{-1}(\mathbf{x}^{\text{obs}}) = \{\mathbf{x} \in \mathbb{R}^L \mid \mathbf{x}(M) = \mathbf{x}^{\text{obs}}\}. \tag{2.1}$$

The most generic interpolation problem can be posed as the task of finding a suitable $\hat{\mathbf{x}} \in \mathit{\Gamma}$. Although we will encounter various approaches to determine the suitability of the solution in the following, we will most often proceed by formulating a suitable regularized inverse problem of the form given by equation (1.20). It suffices to define $E(\mathbf{x}, \mathbf{x}^{\text{obs}}) = I_{\mathit{\Gamma}}(\mathbf{x})$, where $I_{\mathit{\Gamma}}$ is the indicator function of the set $\mathit{\Gamma}$ according to (1.22), which results in the problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} I_{\mathit{\Gamma}}(\mathbf{x}) + R(\mathbf{x}). \tag{2.2}$$

Remark 2.1 (Of indices and matrices)**.** For the derivation of some of the methods, it will be convenient to rewrite the condition for the membership of the signal $\mathbf{x}$ in the set $\mathit{\Gamma}$ as a set of linear constraints. Specifically, for the selection of $\mathbf{x}(M)$,

[1] Equation (2.1) uses the notion of preimage of the element $\mathbf{x}^{\text{obs}}$ under the operator D, i.e., the set of all $\mathbf{x}$ such that $D(\mathbf{x}) = \mathbf{x}^{\text{obs}}$.

we define a matrix $\mathbf{M}$, which is constructed from the identity matrix of order L by selecting the rows given by the indices in M, i.e., $\mathbf{M} \in \mathbb{R}^{|M| \times L}$. Similarly, using the complement $\overline{M}$, we define the matrix $\overline{\mathbf{M}} \in \mathbb{R}^{(L-|M|) \times L}$. With this notation, we can define the constraint

$$\mathbf{x} \in \mathit{\Gamma} \quad \Longleftrightarrow \quad \mathbf{M}\mathbf{x} = \mathbf{x}^{\mathrm{obs}} = \mathbf{M}\mathbf{x}^{\mathrm{true}}. \tag{2.3}$$

A second equivalent form of the condition for $\mathbf{x}$ can be obtained by imputing zeros at the missing sample indices in both $\mathbf{x}^{\mathrm{true}}$ and the considered signal, which can be expressed as

$$\mathbf{x} \in \mathit{\Gamma} \quad \Longleftrightarrow \quad \mathbf{M}^\top\mathbf{M}\mathbf{x} = \mathbf{M}^\top\mathbf{x}^{\mathrm{obs}} = \mathbf{M}^\top\mathbf{M}\mathbf{x}^{\mathrm{true}}. \tag{2.4}$$

This variant may sometimes be more practical, as the manipulation of uncorrupted samples does not require knowledge of their count because the signal is not truncated by the multiplication with $\mathbf{M}^\top\mathbf{M}$.

Remark 2.2. For the matrices defined in remark 2.1, it always holds $\mathbf{M}^\top\mathbf{M} + \overline{\mathbf{M}}^\top\overline{\mathbf{M}} = \mathbf{I}$, i.e., the identity matrix (of size $L = 5$ in the previous example). This is clear because both $\mathbf{M}^\top\mathbf{M}$ and $\overline{\mathbf{M}}^\top\overline{\mathbf{M}}$ are diagonal matrices such that

$$\left(\mathbf{M}^\top\mathbf{M}\right)(i,i) = \begin{cases} 1 & i \in M, \\ 0 & \text{otherwise,} \end{cases} \qquad \left(\overline{\mathbf{M}}^\top\overline{\mathbf{M}}\right)(i,i) = \begin{cases} 0 & i \in M, \\ 1 & \text{otherwise.} \end{cases} \tag{2.5}$$

In other words, it holds $\begin{bmatrix} \mathbf{M}^\top & \overline{\mathbf{M}}^\top \end{bmatrix} \begin{bmatrix} \mathbf{M} \\ \overline{\mathbf{M}} \end{bmatrix} = \mathbf{I}$, i.e., the matrix $\begin{bmatrix} \mathbf{M} \\ \overline{\mathbf{M}} \end{bmatrix}$ is unitary (its transpose coincides with its inverse).

Remark 2.3 (Considering noise)**.** In the formulations above, we have assumed that the observations are not contaminated by noise, so the samples that are not lost are truly reliable and we want to preserve them in the reconstruction. We call such a problem formulation (and its solution) *consistent*. In practice, it may be reasonable to consider noise, although then the problem is not a true interpolation in the mathematical sense. As a result, even signals that are sufficiently close to the set $\mathit{\Gamma}$ in a suitable metric are feasible (we refer to this case as the *inconsistent* formulation).

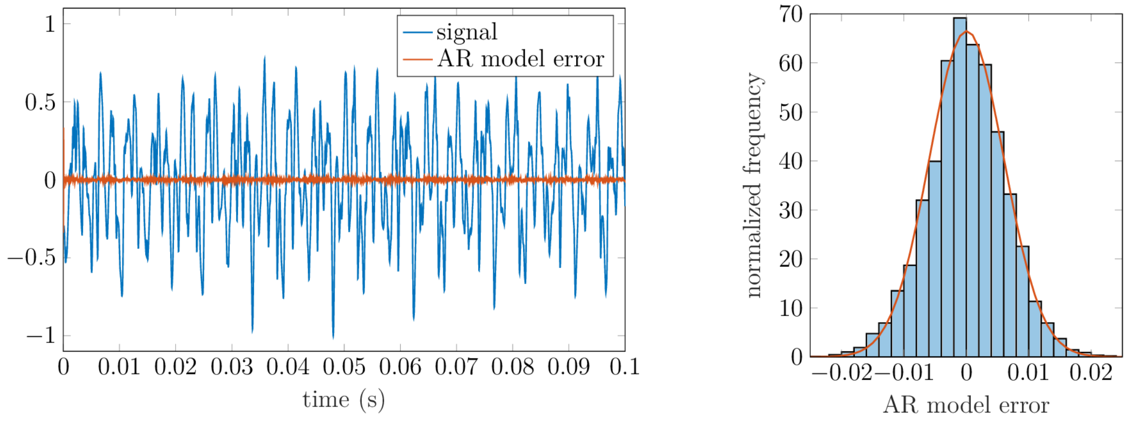
Note that inconsistent formulation may be reasonable even in the noiseless case. For example, a signal can be assumed to only approximately follow a designed model. As a consequence, regularization can be applied while not forcing entrywise equality with observations.

2.2 Methods based on autoregressive modeling

Autoregressive (AR) processes have long been a popular way to model audio signals. These are a type of stochastic processes where the signal (process) $\mathbf{x}$ is governed by the equation

$$x_n = -\sum_{i=1}^{p} a_i x_{n-i} + e_n, \tag{2.6}$$

where $\mathbf{e}$ is the vector of AR model errors with respect to the signal $\mathbf{x}$ and p is the model order. Thus, the equation (2.6) prescribes the element x_n as a linear combination of the p previous elements, except for the error term e_n, which ideally corresponds to white noise. Alternatively, we can say that the signal $\mathbf{x}$ corresponds to the output of an IIR filter with coefficients $\mathbf{a}$, whose input is the noise signal $\mathbf{e}$. This model is used, for example, by speech codecs such as CELP (code-excited linear prediction), where speech is modeled as noise or a sequence of pulses passing through an AR filter [46, chapter 4]. However, the model is also suitable for modeling musical signals, as shown in figure 2.1, which shows a signal – a short section of a string quartet recording – and its deviation from the AR model of order $p = 32$.



(a) the signal $\mathbf{x}$ and the errors $\mathbf{e}$, as defined in (2.6)

(b) histogram of the samples of the error vector $\mathbf{e}$

Fig. 2.1: Illustration of the AR nature of audio signals. In addition, the histogram shows the probability density of the fitted normal distribution (when ignoring 1 % most outlying values). For a true AR process, the error terms should exactly follow the normal distribution. For the analysis to be comprehensive, we would still have to show that the individual components of the vector $\mathbf{e}$ are uncorrelated. In this illustrative example, it would turn out that the order of the model p must be very high (of the order of 1000) for the considered signal to actually correspond to the realization of an AR process.

In the context of audio reconstruction, one of the state-of-the-art methods using the AR model is Janssen's algorithm, presented in 1986 [47]. It is a solution to the problem

$$(\hat{\mathbf{x}}, \hat{\mathbf{a}}) = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^L,\, \mathbf{a}\in\mathbb{R}^p} Q(\mathbf{x}, \mathbf{a}) + I_{\Gamma}(\mathbf{x}), \tag{2.7}$$

where $Q(\mathbf{x}, \mathbf{a}) = \|\mathbf{e}\|^2 = \sum_{n=p+1}^{L} |x_n + \sum_{i=1}^{p} a_i x_{n-i}|$ is the norm of the error term[2] from equation (2.6) (for given $\mathbf{a}$ and $\mathbf{x}$), or a measure of how well the signal matches its autoregressive approximation. This is a non-convex problem, but in practice, it can be solved in a satisfiable way by alternating optimization with respect to the vector of $\mathbf{a}$ coefficients and with respect to the signal $\mathbf{x}$. Despite its age and conceptual simplicity, it is still the leading method in the field of filling in missing sections of audio signals [4].

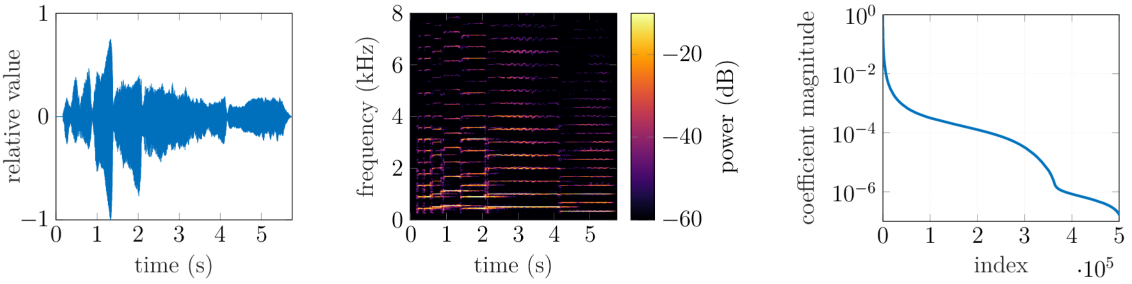
An alternative approach based on AR modeling is the approach of Etter [49]. The problem of interpolating a missing audio segment is split into two extrapolation problems, forward and backward, and the extrapolation is performed according to the AR model of the signal in the left and right neighborhood of the missing segment, respectively. The advantage is the straightforwardness of the method and a lower computational complexity compared to Janssen's iterative algorithm; the disadvantage is the need for an intact neighborhood on both sides of the segment to be completed. Similar methods were also presented by Kauppinen [50, 51] or Esquef [52]. The algorithm is also implemented in Matlab as the function fillgaps.

In addition, variations on the original task based on AR modeling appear in the literature, such as requiring sparsity of the AR coefficients [53, 54, 55]. This theoretical concept has been applied to filling the missing segments in speech signals [56, 57]. Methods for related signal restoration tasks, such as declipping, can be based on a similar principle, where we impose conditions on the reconstructed signal instead of the AR model.

2.3 Methods based on spectral sparsity

A wide range of methods is based on the assumption of spectral sparsity of speech and music signals. Although the signal itself may not contain many zero values (i.e., it is not sparse by itself), with a suitable transformation we can obtain a representation that is significantly sparser. In audio signal processing, such a transformation is, for example, STFT (see 1.9) or discrete cosine transform (DCT), see figure 2.2.

[2] This defines the so-called forward prediction error. Alternatively, the Burg's method [48, Sec. 5.1.2] estimates the AR coefficients are estimated such that forward and backward prediction errors are minimized simultaneously.



(a) signal in the time domain

(b) signal in the time-frequency domain

(c) time-frequency coefficients ordered by magnitude

Fig. 2.2: Illustration of sparsity, which is not a property of the signal itself (a), but of its time-frequency representation (b). Sparsity in the sense of a high number of non-significant entries is emphasized by the figure (c), where the coefficients from the spectrogram are ordered by absolute value, showing a steep decrease.

The sparsity assumption can then be used to pose interpolation as a regularized problem.

This idea appears for the first time, together with the term *audio inpainting* in the work od Adler et al. [20], who attempt to solve the following optimization task:

$$\hat{\mathbf{c}} = \operatorname*{arg\,min}_{\mathbf{c}\in\mathbb{C}^{FN}} \|\mathbf{c}\|_0 \quad \text{subject to} \quad \|\mathbf{x}^{\text{obs}} - \mathbf{MTc}\|_2^2 \leq \varepsilon, \tag{2.8}$$

where $\mathbf{c}$ are the coefficients of the signal in a suitable transform (DCT or STFT) and $\mathbf{T}$ synthesizes the signal from these coefficients; the solution is then the signal $\hat{\mathbf{x}} = \mathbf{T}\hat{\mathbf{c}}$. The parameter $\varepsilon > 0$ regulates the tolerated deviation of the solution from the observed samples, which is different from (2.2), where the indicator function enforces this deviation to be zero (in the context of the note 2.3, this is therefore an inconsistent formulation). Also note that for surjective $\mathbf{T}$ (which is the case of frames), problem (2.8) is solvable even with $\varepsilon = 0$. However, allowing a non-zero tolerance naturally leads to sparser solutions.

The task (2.8) is problematic due to the presence of sparsity as a non-convex function. Even though promising strategies based on mixed-integer programming exist for such tasks [58, 59], the audio processing application is further complicated by the problem scale. The solution must therefore be sought heuristically, with no guarantee of convergence to the global optimum. The authors of [20] choose the Orthogonal Matching Pursuit (OMP) [60] and show that their approach can compete with Janssen's algorithm for holes of length of up to a few milliseconds. The advantage of the OMP algorithm is its simple concept, its disadvantage is the computational complexity.

Because the sparsity assumption may be too simplistic, more complex models can be found in the literature. One of them is structured sparsity, where we do not directly minimize the number of non-zero coefficients in the spectrogram, but define certain groups of coefficients and require a low number of these groups [61]. From the spectrogram in figure 2.2b we can presume that suitable groups are, for example, a horizontal or vertical series of several neighboring coefficients (in the time-frequency plane) – the former representing the harmonic components of the tones, the latter the transient components. It turns out that in the case of audio signal interpolation, such an approach can provide a significant improvement in the reconstruction quality [62], similarly to the problem of audio declipping [63].

In addition, these works consider the so-called convex relaxation, where the non-convex sparsity is replaced by the norm $\|\cdot\|_1$, which sums the absolute values of the argument entries (see also 1.3), and propose the following two formulations:

$$\hat{\mathbf{c}} = \operatorname*{arg\,min}_{\mathbf{c}\in\mathbb{C}^{FN}} \|\mathbf{c}\|_1 + I_{\Gamma}(\mathbf{Tc}), \tag{2.9a}$$

$$\hat{\mathbf{x}} = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^{L}} \|\mathbf{Lx}\|_1 + I_{\Gamma}(\mathbf{x}). \tag{2.9b}$$

The problems (2.9) are already convex, thus they are efficiently solvable using proximal algorithms [35].

Remark 2.4 (Synthesis and analysis formulation)**.** In the literature concerning audio signal reconstruction, the *synthesis* and *analysis* kinds of optimization formulations are sometimes distinguished. Examples of a synthesis formulation are (2.9a) and (2.8). The variables of such a problem are the (sparse) signal coefficients from which the signal is synthesized using the operator $\mathbf{T}$. On the other hand, the analysis formulation (2.9b) has as variable directly the sought signal and the sparsity is assumed after analyzing this signal with the operator $\mathbf{L}$. If the transformations $\mathbf{T}$ and $\mathbf{L}$ were invertible, then the two formulations would be equivalent. In the case of redundant frame representation (where the number of coefficients FN exceeds the length of the signal L), this is generally not the case [64].

Regularization using non-local sparsity [65] has also appeared in the literature. This is a variant of structured sparsity, but the sparse groups of coefficients are not necessarily concentrated in time, instead they can be spread over the whole time range of the signal. For example, this allows to utilize the repetitive nature of musical signals.

For finding sparse solutions in various domains, we also use a procedure which is referred to as dictionary learning, where in addition to the sought variable (signal or coefficients) we also look for the transformation such that the solution can be truly sparse, see e.g. [66, 67]. For signal interpolation, generic dictionary-learning approaches may not be appropriate, as we often require some additional properties of the transformation we are looking for, e.g., that it is a (tight) frame. A wider area of application has been described in the field of declipping of audio signals [68, 69, 70]. In combination with the OMP, dictionary learning has been used to interpolate short gaps (up to length of 15 ms) [71]. Recent literature considers also longer missing segments [72, 73], when more advanced algorithms developed in this thesis are supplemented by the learning phase (see chapter 4).

2.4 Methods based on self-similarity

A separate class consists of methods based on self-similarity. These methods do not work with regularized formulations of type (2.2), nor do they try to extrapolate missing sections from the intact context. Rather, the goal is to disguise the missing segment by filling in the signal with a section from the intact part of the signal such that the filled-in segment is imperceptible in the resulting signal. An example is the *self-content-based audio inpainting* method, which works by splitting the signal into blocks of a given length and searching for a similar block in place of the corrupted [74]. A second example is the method based on a similarity graph, which focuses on the smoothness of the transition from the uncorrupted context to the section added at the missing location [75].

2.5 Methods based on deep learning

The last class of methods widely represented in recent literature is based on machine learning, specifically on deep neural networks. In a broader sense, these methods can be put in the context of the regularized task (2.2), where, however, the regularization $R(\mathbf{x})$ is not hand-crafted – as in case of the (structured) sparsity or autoregressive modeling – but is learned using a sufficient number of training signals.

Although neural networks are now dominant in many domains and generative models are gaining much attention, generating (musical) audio signals is rather marginal [76, 77, 78]. While more applications deal with speech processing (including interpolation of missing segments caused by packet loss during transmission [79]), music signal processing is more complicated due to its multi-scale nature [80].

Currently, most advanced deep learning algorithms for audio signal interpolation are based on the so-called context encoders, which are networks that learn a minimalist (low-dimensional) representation of complex inputs (here spectrograms) [81]. In the case of interpolation, the network learns a representation of the uncorrupted context and, in the decoding part, produces a longer signal from this representation that contains also the concealed segment. A newer variant then additionally includes the so-called *adversarial loss*, where the learning process involves a second network (discriminator) in addition to the generator. The discriminator learns alongside the generator to distinguish real and artificially generated signals, which further forces the generator to produce more realistic signals during the learning process [82]. The most recent developments focus on so-called diffusion models [83, 84].

A drawback of methods that rely only on deep neural networks is the problematic interpretability and controllability of the results. A compromise between these methods and classical methods is either the so-called *deep unfolding* [85], or *plug-and-play* methods [86]. These methods find applications mainly in image processing, and recently there have also emerged applications to audio signal reconstruction [87, 88], but studies dealing with the interpolation of (especially longer) sections of audio signals using these methods are still lacking.

3 Thesis aims and objectives

The overall goal of the thesis is the development of new algorithms for interpolation of missing or corrupted sections of musical audio signals, aiming at increasing the quality of the restored audio signals. The ambition is to challenge the state-of-the-art methods especially in the context of middle-length gaps, i.e., drop-outs of length of tens of milliseconds. To achieve the main goal, the aims of the thesis can be structured in the following, consecutive objectives.

Formalization of the interpolation task

In order to propose algorithms for audio interpolation, the problem needs to be formalized. Most of the state-of-the-art methods are based on mathematical optimization, more precisely inverse problems. Such problems consist in minimization of the sum of the data-fidelity and regularization terms. The first aim of the thesis is to formulate the problem in a way suitable for the employment of modern optimization methods.

Design of the priors on the audio signal and suitable algorithms

To benefit from algorithms used to solve inverse problems, a convenient prior must be designed. This thesis aims to focus on two particular possibilities – sparsity and low-rank structure of the spectrogram of the original audio signal.

Sparsity has been among the most popular priors in past decades, both from the synthesis and analysis perspectives. Since modern optimization algorithms are commonly built upon proximal splitting, a part of the work will focus on convex relaxations of the sparsity objective. Regarding the low-rank prior, one of the state-of-the-art methods in audio declipping is based on a probabilistic formulation using non-negative matrix factorization (NMF). The aim of part of the thesis is to study profoundly this approach and design its variant for audio interpolation.

The design of the priors must be followed by the proposition of suitable algorithms for solving the resulting optimization problems. In the context of convex optimization, we may rely on proximal splitting methods. For probabilistic modeling, maximum likelihood estimation (MLE) is a commonly used strategy, realized through the expectation–maximization (EM) algorithm.

Proposals for modification of the state-of-the-art models and methods

The aim is to build novel algorithms for the sparsity-motivated problem, and to propose modifications that allow to overcome the main drawbacks, such as the loss of signal energy inside the interpolated segment. This could be achieved in several ways:

1. modifications of the optimization problem, such as by proposing weighting strategies in the regularizing norm,
2. post-processing of the interpolated signal,
3. decline from the requirement of convexity and proposition of a heuristic algorithm for a non-convex problem.

In the part of the thesis concerning probabilistic modeling, we will follow up on the research in the related field of audio declipping [89]. The aim is to draw new perspectives on the existing work, allowing design of alternative algorithms.

Implementation and evaluation of the proposed methods in a reproducible numerical experiment

A natural goal is the objective evaluation of the performance of the proposed methods, which must be preceded by an efficient implementation of these methods. Since high objective reconstruction quality is what we aim to achieve, the evaluation will be performed using objective metrics. Popular tools rely on the comparison of the restored signal with a reference, which means that the experiment will be designed based on clean audio signals with simulated degradation, such that the reference is available. To express the quality numerically, objective evaluation will be carried out by an entrywise metric, such as the signal-to-distortion ratio (SDR), as well as by metrics predicting the results of subjective tests (PEMO-Q [90], Perceptual Evaluation of Audio Quality (PEAQ) [91, 92]).

While subjective evaluation via standardized listening test would be a preferable way to draw conclusions, it is excluded from the aims of the thesis due to expected volume of signals to be analyzed.

4 Methods based on sparse representations

In this chapter, we focus on the development of methods based on spectral sparsity, the basis of which was already laid in section 2.3. In particular, we will build upon the historically first, non-convex formulation (2.8) and relaxed convex tasks (2.9).

The primary motivation for development here is the attractiveness of modern optimization methods based on proximal operators, which allow numerically efficient solution of inverse problems in signal processing. Especially in the case of solving convex problems (2.9) and in the situation of dropout of larger number of consecutive samples, however, the current great disadvantage is the decrease of the energy of the interpolated signal inside the filled gap. This phenomenon is illustrated in figure 4.1, where we see that the reconstruction[1] near the boundaries of the filled segment is satisfactory, but towards the middle of the segment the signal energy decreases substantially (while the estimation appears correct in terms of frequency and phase). Such a decrease in an interpolated segment of length in the order of tens of milliseconds is then also noticeable from a subjective listening point of view. The cause of the drop in energy may be precisely the convex relaxation of the task, which, by the principle of the applied ℓ_1 norm, not only requires a low number of significant time-frequency coefficients, but also penalizes those significant coefficients.

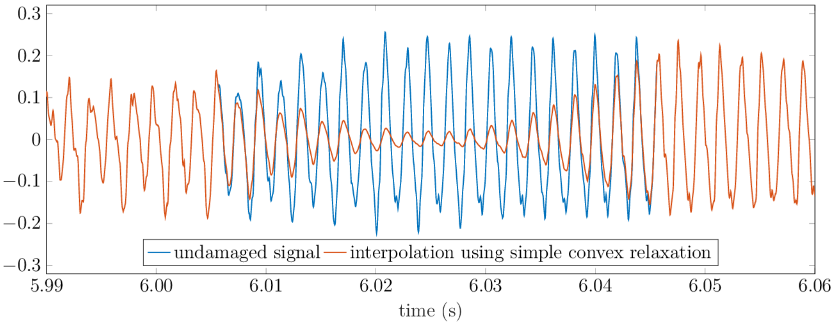


Fig. 4.1: Energy drop in the interpolated segment of the signal. We can see the original undamaged signal (violin tone recording) in which a sample dropout of length 40 ms was simulated and was interpolated as a solution to the analysis problem (2.9b).

Section 4.1 provides a closer look at the causes of the energy drop in the case of convex formulations and proposes alternative formulations and algorithms for their optimization. In section 4.2, we then return to the original idea of (unrelaxed) sparsity and propose algorithms that are competitive in terms of objective reconstruction quality and computational time, in contrast to the historical OMP.

[1] The reconstruction was performed by the demonstration script available at `https://github.com/ondrejmokry/InpaintingRevisited`.

4.1 Methods based on convex relaxation

In this section we will build upon the relaxed synthesis and analysis formulations (2.9a) and (2.9b), which we will complement with the possibility to weight the individual (time-frequency) coefficients by a vector of weights $\mathbf{w} \in \mathbb{R}^{FN}$:

$$\hat{\mathbf{c}} = \operatorname*{arg\,min}_{\mathbf{c}\in\mathbb{C}^{FN}} \|\mathbf{w} \odot \mathbf{c}\|_1 + I_\Gamma(\mathbf{Tc}), \tag{4.1a}$$

$$\hat{\mathbf{x}} = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^{L}} \|\mathbf{w} \odot \mathbf{Lx}\|_1 + I_\Gamma(\mathbf{x}), \tag{4.1b}$$

where the symbol $\odot$ denotes the entrywise product. Note that the dimension of $\mathbf{c}$ and $\mathbf{w}$ refers to the assumed time-frequency nature of the coefficients (see examples 1.9). For the sake of clarity, the dimension of the time-frequency coefficients will be denoted K instead of FN at some places in this section.

Remark 4.1. We leave the interpretation and specific form of $\mathbf{w}$ open at this point and will return to it in section 4.1.3 after we have derived algorithms for solving the problems (4.1). However, let us stress out that if $w_n > 0$ for all n, then the weighed norm $\|\mathbf{x}\|_{\mathbf{w},p} = \|\mathbf{w} \odot \mathbf{x}\|_p$ is a true norm according to definition 1.2 for any ℓ_p norm (see example 1.3).

In section 1.2.2, several algorithms were presented for solving optimization problems that involve the sum of two functions whose proximal operators we can evaluate. In the following sections, we will derive the specific forms of the algorithms for solving the two problems (4.1a) and (4.1b). Further modifications will be proposed in the subsequent section 4.1.3.

4.1.1 Solving the synthesis formulation (4.1a)

Since we aim at using proximal algorithms, we first derive the proximal operator of the weighted ℓ_1 norm, $\operatorname{prox}_{\|\cdot\|_{\mathbf{w},1}}$. We can conveniently use a property of the proximal operator of separable functions, which can be simply put as follows [32, theorem 6.6]: If $f(\mathbf{c}) = f_1(c_1) + f_2(c_2) + \cdots + f_K(c_K)$, then

$$\operatorname{prox}_f(\mathbf{c}) = [\operatorname{prox}_{f_1}(c_1), \operatorname{prox}_{f_2}(c_2), \ldots, \operatorname{prox}_{f_K}(c_K)]^\top. \tag{4.2}$$

Since on the space $\mathbb{R}^1$ the absolute value coincides with the ℓ_1 norm, it holds $\operatorname{prox}_{\gamma|\cdot|}(c) = \operatorname{soft}_\gamma(c)$ (see table 1.1). Considering that $\|\mathbf{c}\|_{\mathbf{w},1} = |w_1c_1| + \cdots + |w_Kc_K|$, the aforementioned property (4.2) can be used to express $\operatorname{prox}_{\|\cdot\|_{\mathbf{w},1}}(\mathbf{c}) = [\operatorname{soft}_{w_1}(c_1), \ldots, \operatorname{soft}_{w_K}(c_K)]^\top$, which we denote as $\operatorname{soft}_{\mathbf{w}}(\mathbf{c})$. The eventual multiplication of the entire function $\|\cdot\|_{\mathbf{w},1}$ by a scalar is then trivial – the scalar becomes part of the weights $\mathbf{w}$, so we multiply all the thresholds by it:

$$\operatorname{prox}_{\gamma\|\cdot\|_{\mathbf{w},1}}(\mathbf{c}) = \operatorname{prox}_{\|\cdot\|_{\gamma\mathbf{w},1}}(\mathbf{c}) = [\operatorname{soft}_{\gamma w_1}(c_1), \ldots, \operatorname{soft}_{\gamma w_K}(c_K)]^\top. \tag{4.3}$$

The second summand in the objective function (4.1a) is the indicator function of the set $\mathit{\Gamma}$, whose proximal operator is the projection onto this set, $\operatorname{proj}_{\mathit{\Gamma}}$, according to the table 1.1. It can be shown (see remark 4.2) that, considering (2.4), the following holds:

$$\begin{aligned}\operatorname{proj}_{\mathit{\Gamma}}(\mathbf{x}) &= (\mathbf{I} - \mathbf{M}^\top\mathbf{M})\mathbf{x} + \mathbf{M}^\top\mathbf{M}\mathbf{x}^{\text{true}} \\ &= (\mathbf{I} - \mathbf{M}^\top\mathbf{M})\mathbf{x} + \mathbf{M}^\top\mathbf{x}^{\text{obs}} \\ &= \overline{\mathbf{M}}^\top\overline{\mathbf{M}}\mathbf{x} + \mathbf{M}^\top\mathbf{x}^{\text{obs}}.\end{aligned} \tag{4.4}$$

With the knowledge of these proximal operators, the usage of CPA (algorithm 1.2) might be suggested, because the indicator function in the problem (4.1a) is in composition with the (linear) synthesis operator $\mathbf{T}$. Nonetheless, if this operator corresponds to a tight frame, i.e., $\mathbf{T}\mathbf{T}^* = \mathbf{T}\mathbf{L} = A\mathbf{I}$ for a constant $A > 0$ (see definition 1.7, theorem 1.8 and equation (1.9)), we can use the composition property of table 1.1 and use the proximal operator of $I_{\mathit{\Gamma}} \circ \mathbf{T}$ (the symbol $\circ$ denotes the composition):

$$\operatorname{prox}_{I_{\mathit{\Gamma}}\circ\mathbf{T}}(\mathbf{c}) = \mathbf{c} + \tfrac{1}{A}\mathbf{L}(\operatorname{proj}_{\mathit{\Gamma}}(\mathbf{T}\mathbf{c}) - \mathbf{T}\mathbf{c}), \quad \mathbf{L} = \mathbf{T}^*, \quad \mathbf{T}\mathbf{L} = A\mathbf{I}. \tag{4.5}$$

In this case, scalar multiplication by $\gamma > 0$ does not influence the indicator function, thus with the knowledge of the proximal operators according to (4.3) and (4.5) we can solve the problem (4.1a) using DRA (algorithm 1.1). The specific form of DRA is in algorithm 4.1. For a more detailed derivation of step 4 see remark 4.3.

Algorithm 4.1: Douglas–Rachford algorithm for solving (4.1a)

```
  input: $\varepsilon \in [0,1], \gamma > 0, \mathbf{d}^{(0)} \in \mathbb{C}^{FN}$
1 for $n = 1, 2, \ldots$ do
2     $\mathbf{c}^{(n)} = \operatorname{soft}_{\gamma\mathbf{w}}(\mathbf{d}^{(n-1)})$
3     choose $\lambda^{(n)} \in [\varepsilon, 2-\varepsilon]$
4     $\mathbf{d}^{(n)} = (1-\lambda^{(n)})\mathbf{d}^{(n-1)} + \lambda^{(n)}\mathbf{c}^{(n)} + \lambda^{(n)}\frac{1}{A}\mathbf{L}\left(\mathbf{M}^\top\mathbf{x}^{\text{obs}} - \mathbf{M}^\top\mathbf{M}\mathbf{T}(2\mathbf{c}^{(n)} - \mathbf{d}^{(n-1)})\right)$
5 end
6 $\mathbf{c}^{(n+1)} = \mathbf{c}^{(n)} + \frac{1}{A}\mathbf{L}\left(\mathbf{M}^\top\mathbf{x}^{\text{obs}} - \mathbf{M}^\top\mathbf{M}\mathbf{T}\mathbf{c}^{(n)}\right)$
  output: $\hat{\mathbf{c}} = \mathbf{c}^{(n+1)}$
```

Note that from a theoretical point of view, the desired membership $\mathbf{T}\mathbf{c}^{(n)} \in \mathit{\Gamma}$ would be guaranteed as a limit in an infinite number of iterations. In practice, we therefore add an extra projection (line 6) to ensure a consistent solution for arbitrary termination of the algorithm. At the same time, it is clear that the solution of the algorithm 4.1 is a vector of time-frequency coefficients. To obtain the interpolated signal itself, it suffices to synthesize the solution of the algorithm; step 6 then ensures that this synthesized signal also lies in the set $\mathit{\Gamma}$.

Remark 4.2 (Derivation of the projection)**.** From the definition of projection (see also equation 1.23), it holds for $\mathbf{x} \in \mathbb{R}^L$

$$\operatorname{proj}_{\mathit{\Gamma}}(\mathbf{x}) = \operatorname*{arg\,min}_{\mathbf{z} \in \mathit{\Gamma}} \|\mathbf{z} - \mathbf{x}\|_2, \tag{4.6}$$

which is equivalent to minimizing $\|\mathbf{z} - \mathbf{x}\|_2^2 = \sum_{n=1}^{L} |z_n - x_n|^2$. At the same time, the set of feasible signals $\mathit{\Gamma}$ is defined entrywise according to equation (2.1): The constraint $\mathbf{x}(M) = \mathbf{x}^{\text{true}}(M)$ is equivalent to $x_n = x_n^{\text{true}}$ for $n \in M$. This allows us to decompose the problem into individual entries[2]:

$$\operatorname{proj}_{\mathit{\Gamma}}(\mathbf{x})(n) = \operatorname*{arg\,min}_{z} |z - x_n|^2 \text{ subject to } \begin{cases} z = x_n^{\text{true}} \text{ if } n \in M, \\ z \text{ arbitrary if } n \notin M. \end{cases} \tag{4.7}$$

Together, we have

$$\operatorname{proj}_{\mathit{\Gamma}}(\mathbf{x})(n) = \begin{cases} x_n^{\text{true}} \text{ for } n \in M, \\ x_n \text{ for } n \notin M, \end{cases} \tag{4.8}$$

or, the reliable samples (defined by the index set M) are preserved from the observed signal $\mathbf{x}^{\text{true}}$ and the remaining samples are taken from the argument $\mathbf{x}$. This can be written compactly precisely as $\operatorname{proj}_{\mathit{\Gamma}}(\mathbf{x}) = (\mathbf{I} - \mathbf{M}^\top\mathbf{M})\mathbf{x} + \mathbf{M}^\top\mathbf{M}\mathbf{x}^{\text{true}}$ (see also remark 2.1).

Remark 4.3 (Derivation of the projection step of algorithm 4.1)**.** For a tight frame, it holds according to (4.5) and (4.4) that

$$\begin{aligned} \operatorname{prox}_{I_{\mathit{\Gamma}} \circ \mathbf{T}}(\mathbf{c}) &= \mathbf{c} + \tfrac{1}{A}\mathbf{L}(\operatorname{proj}_{\mathit{\Gamma}}(\mathbf{T}\mathbf{c}) - \mathbf{T}\mathbf{c}) \\ &= \mathbf{c} + \tfrac{1}{A}\mathbf{L}\big((\mathbf{I} - \mathbf{M}^\top\mathbf{M})\mathbf{T}\mathbf{c} + \mathbf{M}^\top\mathbf{x}^{\text{obs}} - \mathbf{T}\mathbf{c}\big) \\ &= \mathbf{c} + \tfrac{1}{A}\mathbf{L}\big(\mathbf{M}^\top\mathbf{x}^{\text{obs}} - \mathbf{M}^\top\mathbf{M}\mathbf{T}\mathbf{c}\big). \end{aligned}$$

Then

$$\begin{aligned} \mathbf{d}^{(n)} &= \mathbf{d}^{(n-1)} + \lambda^{(n)}\left(\operatorname{prox}_{I_{\mathit{\Gamma}} \circ \mathbf{T}}(2\mathbf{c}^{(n)} - \mathbf{d}^{(n-1)}) - \mathbf{c}^{(n)}\right) \\ &= \mathbf{d}^{(n-1)} + \lambda^{(n)}\left(2\mathbf{c}^{(n)} - \mathbf{d}^{(n-1)} + \tfrac{1}{A}\mathbf{L}\left(\mathbf{M}^\top\mathbf{x}^{\text{obs}} - \mathbf{M}^\top\mathbf{M}\mathbf{T}(2\mathbf{c}^{(n)} - \mathbf{d}^{(n-1)})\right) - \mathbf{c}^{(n)}\right) \\ &= (1 - \lambda^{(n)})\mathbf{d}^{(n-1)} + \lambda^{(n)}\mathbf{c}^{(n)} + \lambda^{(n)}\tfrac{1}{A}\mathbf{L}\left(\mathbf{M}^\top\mathbf{x}^{\text{obs}} - \mathbf{M}^\top\mathbf{M}\mathbf{T}(2\mathbf{c}^{(n)} - \mathbf{d}^{(n-1)})\right). \end{aligned}$$

[2] We use the notation from remark 1.1.

In particular, we obtain for $\lambda^{(n)} = 1$ that

$$\mathbf{d}^{(n)} = \mathbf{c}^{(n)} + \tfrac{1}{A}\mathbf{L}\left(\mathbf{M}^\top\mathbf{x}^{\text{obs}} - \mathbf{M}^\top\mathbf{M}\mathbf{T}(2\mathbf{c}^{(n)} - \mathbf{d}^{(n-1)})\right).$$

Remark 4.4 (Alternative algorithms)**.** To solve problem (4.1a), DRA could be used with conversely assigned functions, or there could be several ways to use a more complex algorithm such as CPA. However, our goal is to find a solution to the optimization problem and study how suitable this solution is in terms of signal quality. The way of finding the solution is not crucial for us and hence alternative algorithms are not being considered.

4.1.2 Solving the analysis formulation (4.1b)

The analysis formulation (4.1b), like the synthesis variant, involves the sum of two functions with known (and easily enumerable) proximal operators, one of which is in composition with a linear operator. However, unlike the synthesis formulation, in practical situations the condition $\mathbf{L}\mathbf{L}^* = A\mathbf{I}$, which would guarantee easy enumerability of the proximal operator of the composition of the ℓ_1 norm and the analysis operator $\mathbf{L}$, is not satisfied. In the case of a tight frame, it holds $\mathbf{L}^*\mathbf{L} = \mathbf{T}\mathbf{L} = A\mathbf{I}$ and according to table 1.1

$$\operatorname{prox}_{\|\cdot\|_{\mathbf{w},1}\circ\mathbf{L}}(\mathbf{x}) = \tfrac{1}{A}\mathbf{T}\operatorname{prox}_{\|\cdot\|_{\mathbf{w},1}+I_{\mathcal{R}(\mathbf{L})}}(\mathbf{L}\mathbf{x}). \tag{4.9}$$

However, we cannot effectively evaluate a proximal operator containing an indicator function in practice [1].

To solve problem (4.1b) we therefore use CPA following algorithm 1.2. Since in this case the presence of the linear operator clearly distinguishes the functions used, according to equation (1.27), $f = \|\cdot\|_{\mathbf{w},1}$ and $g = I_\Gamma$. For a particular form of CPA we need to know the proximal operator of the (convex) conjugate function of the function f (see definition 1.20). Using table 1.1, we have (see remark 4.5)

$$\operatorname{prox}_{\sigma(\|\cdot\|_{\mathbf{w},1})^*}(\mathbf{c}) = \mathbf{c} - \sigma\operatorname{prox}_{\sigma^{-1}\|\cdot\|_{\mathbf{w},1}}(\sigma^{-1}\mathbf{c}) = \mathbf{c} - \sigma\operatorname{soft}_{\sigma^{-1}\mathbf{w}}(\sigma^{-1}\mathbf{c}) = \mathbf{c} - \operatorname{soft}_{\mathbf{w}}(\mathbf{c}). \tag{4.10}$$

The resulting specific form of CPA is in algorithm 4.2.

As mentioned in section 1.2.2, the convergence of the algorithm is guaranteed for the choice of the parameters $\theta = 1$ and $\sigma\tau\|\mathbf{L}\|^2 \leq 1$. Specifically in the case of tight frame, it holds $\|\mathbf{L}\|^2 = \|\mathbf{L}^*\mathbf{L}\| = \|\mathbf{T}\mathbf{L}\| = \|A\mathbf{I}\| = A$ [24, Proposition 2.71].

Remark 4.5 (Derivation of the proximal step)**.** From equation (1.24), it holds for

Algorithm 4.2: Chambolle–Pock algorithm for solving (4.1b)

```
  input: $\tau, \sigma > 0, \theta \in [0,1], \mathbf{x}^{(0)} \in \mathbb{R}^L, \mathbf{c}^{(0)} \in \mathbb{C}^{FN}, \bar{\mathbf{x}}^{(0)} = \mathbf{x}^{(0)}$
1 for $n = 1, 2, \ldots$ do
2     $\mathbf{c}^{(n)} = \mathbf{c}^{(n-1)} + \sigma\mathbf{L}\bar{\mathbf{x}}^{(n-1)} - \mathrm{soft}_{\mathbf{w}}(\mathbf{c}^{(n-1)} + \sigma\mathbf{L}\bar{\mathbf{x}}^{(n-1)})$
3     $\mathbf{x}^{(n)} = \mathrm{proj}_{\Gamma}(\mathbf{x}^{(n-1)} - \tau\mathbf{T}\mathbf{c}^{(n)})$
4     $\bar{\mathbf{x}}^{(n)} = \mathbf{x}^{(n)} + \theta(\mathbf{x}^{(n)} - \mathbf{x}^{(n-1)})$
5 end
  output: $\hat{\mathbf{x}} = \mathbf{x}^{(n)}$
```

any scalar argument c and parameters $\sigma > 0$ and $w > 0$

$$\begin{aligned}
\sigma\,\mathrm{soft}_{\sigma^{-1}w}(\sigma^{-1}c) &= \sigma\,\mathrm{sgn}(\sigma^{-1}c)\max\left(\left|\sigma^{-1}c\right| - \sigma^{-1}w, 0\right) \\
&= \sigma\,\mathrm{sgn}(c)\max\left(\sigma^{-1}(|c| - w), 0\right) \\
&= \sigma\,\mathrm{sgn}(c)\sigma^{-1}\max\left(|c| - w, 0\right) \\
&= \mathrm{soft}_w(c).
\end{aligned}$$

Remark 4.6 (Clip function)**.** Using the properties $c = \mathrm{sgn}(c)\,|c|$ and $\max(a,b) = -\min(-a,-b)$, it holds

$$\begin{aligned}
c - \mathrm{soft}_w(c) &= c - \mathrm{sgn}(c)\max(|c| - w, 0) \\
&= \mathrm{sgn}(c)\,|c| + \mathrm{sgn}(c)\min(w - |c|, 0) \\
&= \mathrm{sgn}(c)\min(w, |c|).
\end{aligned}$$

This function, denoted also clip_w, performs hard clipping of its argument on the level w. Values of c less than w (in absolute value) remain unchanged, other values are replaced by the value of w with sign preserved according to the argument c. In algorithm 4.2, it means that line 2 can be rewritten as $\mathbf{c}^{(n)} = \mathrm{clip}_{\mathbf{w}}(\mathbf{c}^{(n-1)} + \sigma\mathbf{L}\bar{\mathbf{x}}^{(n-1)})$.

4.1.3 Choosing the weights and other modifications

In this section, we focus on the problem illustrated by figure 4.1, i.e., the interpolation of a longer dropout (compact gap) in an otherwise undamaged signal. In this case, the described approaches based on minimizing the ℓ_1 norm provide a solution that does not achieve sufficient energy in the interpolated segment. We first aim to design the weights for the problem (4.1) to support the energy of the interpolated segment and thus improve the quality of the reconstruction. Subsequently, we also propose several possible extensions to the optimization algorithm itself.

Offset and overlap-based weights

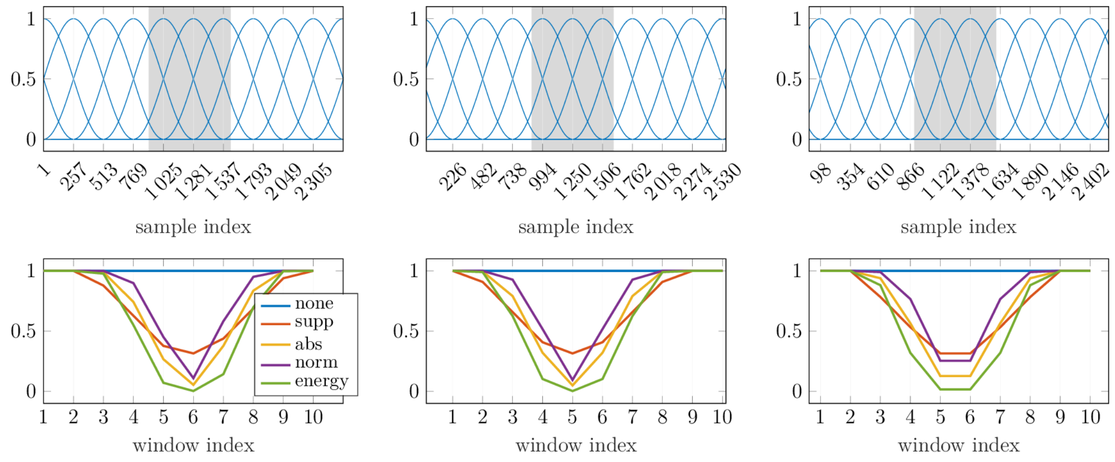
Assume further the use of the time-frequency representation in the role of the operators $\mathbf{L}$ and $\mathbf{T}$, namely the Gabor frame. According to example 1.9, the analysis (and in the case of a tight frame also the synthesis) is performed over windows of a given shape, length W and shift a. Each shift corresponds to F modulations of this window, giving a total of $FL/a = FN$ atoms for a signal of length L, denoted[3], $\mathbf{g}_1, \ldots, \mathbf{g}_{FN} \in \mathbb{C}^L$. Following on [4], we hypothesize that the cause of the energy drop in the interpolated segment is that the windows contributing largely to this segment are not sufficiently represented in the sparse signal representation. The motivation for this hypothesis is the interpretation of the problem (4.1) that we are looking for a sparse representation of the uncorrupted part of the signal. Atoms having large overlap with the missing segment (or small overlap with the undamaged part of the signal) may be irrelevant to this representation, hence they do not play a significant role in the solution. However, in the synthesis of the whole (reconstructed) signal, the energy corresponding to such atoms is crucial.[4]

We therefore choose weights $\mathbf{w} \in \mathbb{R}^{FN}$ that measure the overlap of the gap and the corresponding windows of our time-frequency representation in various ways. However, since this overlap apparently depends on the position of the gap with respect to the „grid“ of the individual windows (shifts), we additionally propose the so-called *offset*, which allows the window shifts to be aligned with the position of the gap. In the *full* variant, the center of the gap corresponds to the peak of a window for some shift, while in the *half* variant the center of the gap is in the centerline of two adjacent windows, see figure 4.2. We refer to the variant without any offset as *none*.

As a measure of the overlap of atoms, we then choose the following options (for completeness, including the variant without weighting). Recall that according to remark 2.1, the matrix $\mathbf{M}$ implements the selection of samples corresponding to the

[3] Unlike example 1.9, here we index the atoms linearly for simplicity.

[4] An extreme example is a gap of such a length that it fully covers several shifts of the window. The solution to such a problem then necessarily contains a sequence of zero samples, since windows completely falling into the gap in the sparse representation necessarily have zero coefficients; see [93] for details. In this thesis, such a pathological situation is not further addressed.



(a) variant *none* (b) variant *full* (c) variant *half*

Fig. 4.2: Offset variants and the corresponding weights $\mathbf{w}$ for atoms with no modulation. The top row shows the shifted windows for calculating the time-frequency representation. The gray area represents the sample dropout. The vertical grid shows the samples corresponding to the centers of each window (with the spacing given by the parameter $a = 256$). The bottom row then shows the vectors of weights, horizontally aligned with the top series of plots; we choose a continuous display for better readability, but in reality these are vectors of 10 values. The legend for the weight variants is common to all graphs.

uncorrupted signal:

$$w_n = 1 \qquad \text{no weights,} \tag{4.11a}$$

$$w_n = \frac{|\mathrm{supp}(\mathbf{Mg}_n)|}{|\mathrm{supp}(\mathbf{g}_n)|} \qquad \text{support-based,} \tag{4.11b}$$

$$w_n = \frac{\|\mathbf{Mg}_n\|_1}{\|\mathbf{g}_n\|_1} \qquad \ell_1\text{-norm-based,} \tag{4.11c}$$

$$w_n = \frac{\|\mathbf{Mg}_n\|_2}{\|\mathbf{g}_n\|_2} \qquad \ell_2\text{-norm-based,} \tag{4.11d}$$

$$w_n = \frac{\|\mathbf{Mg}_n\|_2^2}{\|\mathbf{g}_n\|_2^2} \qquad \text{energy-based.} \tag{4.11e}$$

Possible values of weights for individual (non-modulated) windows are also shown in figure 4.2. Note that when the *full* or *half* offset is chosen, the obtained weight vectors are symmetric along the centerline of the gap.

Remark 4.7. In the case where we are working with a Gabor frame, the denominators in the preceding formulas are unnecessary, since the values are the same for all atoms; they can therefore be viewed as a positive scalar multiple of the function $\|\cdot\|_{\mathbf{w},1}$, which has no effect on the argument of the minimum we are looking for. This fact may be seen from the particular shape of the individual atoms of the Gabor frame – it is always a modulation (and translation, which we need not consider now) of the window $\mathbf{g}$, and this modulation does not affect the absolute values of the individual samples[5]. However, we keep the denominators in the formulas to illustrate the idea that we are measuring the fraction of the atom in the uncorrupted part of the signal. In addition, the formulas (4.11) are also applicable to representations other than Gabor frames.

Iterative re-weighting

The iterative approach does not aim to support specific time-frequency coefficients based on their influence on the interpolated segment, but to approximate, per individual case, the ℓ_0 norm by the weighted ℓ_1 norm [94, 5, 4]. The principle is to compute weights based on a solution estimate, where small time-frequency coefficients are given large weights (hence are penalized more by minimizing the weighted ℓ_1 norm) and vice versa. Specifically, in the iterative procedure, we determine an initial estimate, for example using the variant with no weights according to (4.11a), recalculate new weights based on this solution, update the estimate, and repeat the procedure. In the synthesis model, the solution are the time-frequency coefficients, which we can directly use to compute the weights for the next iteration, see algorithm 4.3. As a follow-on to [94], we suggest an analysis variant (algorithm 4.4), which additionally includes a step 3 in which we analyze the current solution for the subsequent calculation of the weights [5, 4]. Note that in both algorithms, the parameter ε represents a small offset introduced to avoid division by zero.

Algorithm 4.3: Synthesis model with iterative re-weighting

```
input: initial weights $\mathbf{w}^{(0)} \in \mathbb{R}^{FN}$, $\varepsilon > 0$
for $i = 0, 1, \dots$ do
    $\mathbf{c}^{(i)} = \arg\min_{\mathbf{c} \in \mathbb{C}^{FN}} \|\mathbf{w}^{(i)} \odot \mathbf{c}\|_1 + I_\Gamma(\mathbf{Tc})$
    $w_k^{(i+1)} = 1/(|c_k^{(i)}| + \varepsilon)$, $k = 1, \dots, FN$
end
output: $\hat{\mathbf{x}} = \mathbf{Tc}^{(i)}$
```

[5] In the continuous case, the modulation of the window $g(x)$ with a parameter f is the function $\mathrm{e}^{2\pi\mathrm{i}fx} \cdot g(x)$. However, in the formulas (4.11), the modules of the vector elements come out when breaking down the individual norms, see example 1.3. If we calculate the modulus for the value of the modulated window, we get $|\mathrm{e}^{2\pi\mathrm{i}fx} \cdot g(x)| = |\mathrm{e}^{2\pi\mathrm{i}fx}| \cdot |g(x)| = |g(x)|$. The same property naturally holds in the discrete case, which is why modulation does not really affect the absolute value of the elements.

```
Algorithm 4.4: Analysis model with iterative re-weighting
    input: initial weights $\mathbf{w}^{(0)} \in \mathbb{R}^{FN}$, $\varepsilon > 0$
for $i = 0, 1, \ldots$ do
    $\mathbf{x}^{(i)} = \arg\min_{\mathbf{x}\in\mathbb{R}^L} \|\mathbf{w}^{(i)} \odot \mathbf{Lx}\|_1 + I_\Gamma(\mathbf{x})$
    $\mathbf{c}^{(i)} = \mathbf{Lx}^{(i)}$
    $w_k^{(i+1)} = 1/(|c_k^{(i)}| + \varepsilon)$, $k = 1, \ldots, FN$
end
    output: $\hat{\mathbf{x}} = \mathbf{x}^{(i)}$
```

Iterative shortening of the interpolated segment

As in the previous case, we propose a procedure that solves the interpolation as a sequence of several tasks. The motivation is that the reconstruction by the basic sparsity-based method is more satisfactory at the edges of the filled segment, as shown, for example, in figure 4.1. We base our approach on this observation, and propose to repeatedly fill in the edges of the gap, thus progressively shortening the gap, until the whole gap is filled.

Formally, we consider that the missing segment is defined by indices s (first missing sample) and f (last missing sample) and its shortening always proceeds by r samples from both sides, see algorithm 4.5. The number of iterations of this algorithm obviously depends on the length of the hole $f-s+1$ and the value of r, see line 1 of the algorithm. Step 3 can be implemented by any interpolation algorithm, however, we focus on the above-derived algorithms 4.1 or 4.2.

```
Algorithm 4.5: Iterative shortening of the interpolated segment
    input: $\mathbf{x}^{(0)} \in \mathbb{R}^L$, gap borders $s^{(0)} = s$ a $f^{(0)} = f$, parameter $r$
1 $I = \lceil (f^{(0)} - s^{(0)} + 1)/(2r) \rceil$                    // number of iterations
2 for $i = 0, 1, \ldots, I-1$ do
3     $\mathbf{x}^{(i+1)}$ = interpolation of $\mathbf{x}^{(i)}$ with a gap between samples $s^{(i)}$ and $f^{(i)}$
4     $s^{(i+1)} = s^{(i)} + r$                    // shrink the gap from the left
5     $f^{(i+1)} = f^{(i)} - r$                    // shrink the gap from the right
6 end
    output: $\hat{\mathbf{x}} = \mathbf{x}^{(i+1)}$
```

According to the notation in problems (4.1a) and (4.1b) we use algorithm 4.5 to obtain a sequence of sets of feasible solutions $\Gamma^{(i+1)} \subset \Gamma^{(i)}$ and a sequence of weight vectors $\mathbf{w}^{(i)}$. For a solution in each iteration, it holds

$$\mathbf{x}^{(i)} \in \Gamma^{(i+1)}. \tag{4.12}$$

Note that if we do not consider any atom weighting mechanism in the solution in step

3, the same objective function will be optimized in each iteration of algorithm 4.5. Thus, according to equation (4.12), $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)}$ and the resulting solution will be no different from simply using the default algorithm 4.1 or 4.2.

Time domain compensation for energy loss

In this section, we no longer manipulate the flow of the interpolation algorithm or the problem as such, but propose a heuristic approach to compensate for the energy loss by modifying the resulting signal. The motivation here is that we observe a similar energy drop in nearby gaps of the same length interpolated by the same algorithm.

The compensation procedure for a given (already interpolated) gap has the following basic steps:

- we simulate several other gaps in its vicinity, from which we obtain the (temporal) energy profile of the original and the interpolated signal,
- based on the ratios of these energies, we determine the optimal so-called compensation function,
- we multiply the interpolated signal entrywise by this function to obtain the desired energy increase in the interpolated segment.

To formalize this procedure, we first pose some basic assumptions.

Assumption 4.8 (Assumptions on the compensation function)**.** For simplicity, let us consider a continuous case where a continuous function $x(t)\colon [0,T] \to \mathbb{R}^L$ is considered in place of the signal $\mathbf{x} \in \mathbb{R}^L$. Furthermore, assume that a previously discussed offset strategy applies. The goal is to interpolate the segment for[6] $t \in [s,f] \subset [0,T]$ and then to multiply the resulting signal with a compensation function $q(t)\colon [0,T] \to [1,\infty)$, which meets the following assumptions:

1. $q(t)$ is a smooth function,
2. $q(t) = 1$ for $t \in [0,T] \setminus [s,f]$,
3. $\frac{\mathrm{d}q}{\mathrm{d}t} = 0$ for $t = s$ a $t = f$,
4. $\frac{\mathrm{d}q}{\mathrm{d}t} \geq 0$ for $0 < t < \frac{s+f}{2}$,
5. $\frac{\mathrm{d}q}{\mathrm{d}t} \leq 0$ for $\frac{s+f}{2} < t < f$,
6. $q(s+t) = q(f-t)$ for $t \in [0, f-s]$.

The assumptions of smoothness (item 1) and smooth connection to a constant value of 1 outside the interpolated gap (items 2 and 3) are natural, since we do not want to cause any jumps in the reconstructed signal. Symmetry (item 6) together with a non-decreasing trend in the first half of the segment and a non-increasing one in the second half (items 4 and 5) is based on the offset assumption. Due to it (by

[6] Unlike in the rest of the chapter, here we consider s, f as values of (continuous) time, not discrete indices.

choosing either *half* or *full* offset) we can assume a symmetric decrease of energy, which is indeed observed for example in figure 4.3.

The design of a suitable compensation function $q(t)$ and the corresponding discrete vector $\mathbf{q}$ is summarized by algorithm 4.6. The key steps are illustrated in figures 4.3 and 4.4.

Algorithm 4.6: Computation of the compensation vector $\mathbf{q}$

```
   input: the signal with interpolated time interval $[s, f]$, distribution of $J$
          additional gaps, time positions $t_1, \dots, t_I$ of the $I$ segments inside
          the initial gap
 1 simulation and interpolation of $J$ in the vicinity of the initial gap
 2 $\mathbf{X}$ = energy matrix where $x_{ij}$ = the energy of the $i$-th segment of the $j$-th
     interpolated gap
 3 $\mathbf{Y}$ = energy matrix where $y_{ij}$ = the energy of the $i$-th segment of the intact
     signal corresponding to the $j$-th gap
 4 compute multipliers $\mathbf{m}$ as the least squares fit of the energy profiles $\mathbf{X}$ to
     the targets $\mathbf{Y}$:

     $\mathbf{m} = \operatorname*{arg\,min}_{\mathbf{m}' \in \mathbb{R}^m} \|\mathbf{Y} - \operatorname{diag}(\mathbf{m}') \cdot \mathbf{X}\|_2^2 = \left[ \frac{\sum_{j=1}^{J} y_{ij} x_{ij}}{\sum_{j=1}^{J} x_{ij}^2} \right]_{i=1,\dots,I}$     (4.13)

 5 $n_i = \sqrt{m_i}$ for $i = 1, \dots, I$
 6 for $i = 1, \dots, \lfloor I/2 \rfloor$ do
 7     $p = (n_i + n_{I+1-i})/2$
 8     $n_i = p$
 9     $n_{I+1-i} = p$
10 end
11 $q(t)$ = spline interpolation of the points $[s, 1], [t_1, n_1], \dots, [t_I, n_I], [f, 1]$ with
     zero derivative in $s$ and $f$
12 sample $q(t) \mapsto \mathbf{q} \in \mathbb{R}^h$, where $h$ is the number of initially interpolated
     samples (gap length)
   output: compensation vector $\mathbf{q}$
```

Step 5 is justified by the fact that the vector $\mathbf{m}$ represents energy ratios, while the output of our calculation must correspond to the ratio of the values of the signal itself. The loop from line 6 (symmetrization of the vector $\mathbf{n}$) then follows the reasoning of using an offset, where we assume a symmetric evolution of the energy decrease with respect to the gap centerline.

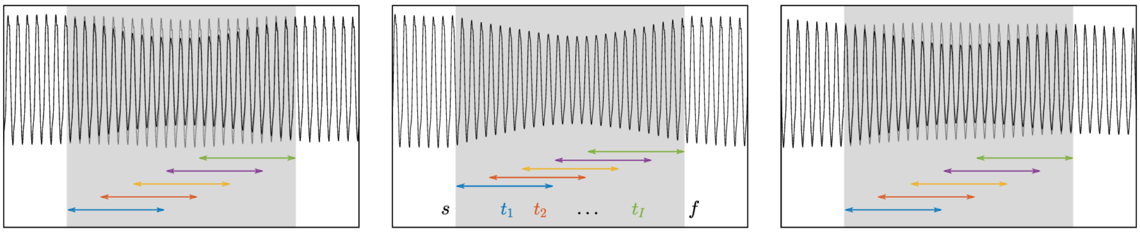
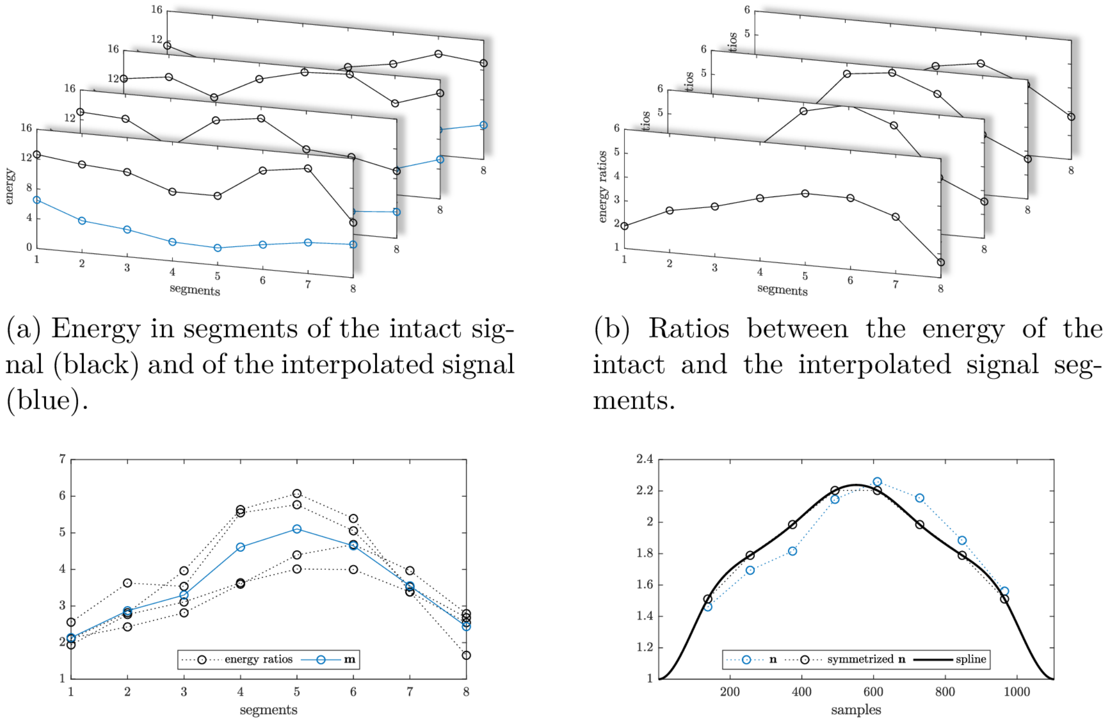


Fig. 4.3: Segments for calculating the compensation of the energy drop in the time domain. The reconstruction of the initial dropout (middle graph, dropout location indicated by the gray area between times s and f) is accompanied by two additional dropouts in its vicinity (left and right graphs), where the intact signal is available (plotted in gray). The position of the $I = 5$ segments, symbolized by the colored arrows, is the same relative to the gap in all three graphs. Adapted from [4].



(a) Energy in segments of the intact signal (black) and of the interpolated signal (blue).

(b) Ratios between the energy of the intact and the interpolated signal segments.

(c) Ratios from the plot 4.4b and the vector **m** according to equation (4.13).

(d) Symmetrization of the vector $\mathbf{n} = \sqrt{\mathbf{m}}$ and the resulting $\mathbf{q}$.

Fig. 4.4: Visualization of the energy drop compensation procedure in the time domain. We use $J = 4$ simulated dropouts and the compensation function uses the energy from $I = 8$ overlapping segments for the calculation. Adapted from [4].

4.2 Heuristic approaches to the non-convex problem

The ultimate goal of the sparsity-based methods is finding a signal $\hat{\mathbf{x}} \in \mathbb{R}^L$ and/or the corresponding time-frequency coefficients $\hat{\mathbf{c}} \in \mathbb{C}^{FN}$, such that the signal is consistent with the observations ($\hat{\mathbf{x}} \in \mathit{\Gamma}$) and the coefficients are sparse ($\|\hat{\mathbf{c}}\|_0 \ll FN$). As already mentioned in section 2.3, the non-convexity of the ℓ_0 norm causes that the task is not feasible. To make the problem feasible, we might either relax the ℓ_0 norm to the ℓ_1 norm, or relax the condition $\hat{\mathbf{x}} \in \mathit{\Gamma}$; the latter option has been explored in the pioneer work [20] solving the problem (2.8) with the OMP algorithm. However, relaxing the problem using ℓ_1 norm is likely to result in energy drop in the interpolated segments, because this norm also penalizes the significant coefficients, as presented in section 4.1. On the other hand, the approach based on the OMP is not very efficient and the restoration quality is mediocre.

As an alternative, and in line with [6], this section derives the Sparse Audio Inpainter (SPAIN) – an algorithm inspired by its precursor for audio declipping, Sparse Audio Declipper (SPADE) [95, 7]. Some technical parts of the derivation are omitted and can be found in the detailed technical report [2].

4.2.1 From ADMM to sparse audio inpainting

As in the cases of the convex relaxation in problems (2.9) and (4.1), we start from two closely related formulations, distinguished by the presence of the synthesis or the analysis operator:

$$(\hat{\mathbf{x}}, \hat{\mathbf{c}}) = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^L,\, \mathbf{c}\in\mathbb{C}^{FN}} \|\mathbf{c}\|_0 \quad \text{subject to} \quad \mathbf{x} \in \mathit{\Gamma},\ \|\mathbf{x} - \mathbf{T}\mathbf{c}\|_2 \leq \varepsilon, \tag{4.14a}$$

$$(\hat{\mathbf{x}}, \hat{\mathbf{c}}) = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^L,\, \mathbf{c}\in\mathbb{C}^{FN}} \|\mathbf{c}\|_0 \quad \text{subject to} \quad \mathbf{x} \in \mathit{\Gamma},\ \|\mathbf{L}\mathbf{x} - \mathbf{c}\|_2 \leq \varepsilon. \tag{4.14b}$$

Note that in contrast to (4.1), the optimization in either the synthesis or analysis variant of (4.14) is carried out over both $\mathbf{x}$ and $\mathbf{c}$.

Even though (4.14) relaxes the relationship of the consistent signal $\mathbf{x}$ and its sparse coefficients $\mathbf{c}$, it is still NP-hard due to the presence of the ℓ_0 norm. The idea of the SPAIN algorithm is to approach (4.14) with the ADMM (algorithm 1.3) where each iteration searches for κ-sparse coefficients $\mathbf{c}$. The value of κ is governed by a preset schedule, such that it increases during iterations of ADMM, and the algorithm stops when the condition $\|\mathbf{x} - \mathbf{T}\mathbf{c}\|_2 \leq \varepsilon$ or $\|\mathbf{L}\mathbf{x} - \mathbf{c}\|_2 \leq \varepsilon$ is satisfied for a chosen value of ε.

4.2.2 S-SPAIN

Synthesis-based Sparse Audio Inpainter (S-SPAIN) uses the ADMM based on the following reformulation of (4.14a) for fixed sparsity κ (compare with (1.28)):

$$(\hat{\mathbf{x}}, \hat{\mathbf{c}}) = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^L,\, \mathbf{c}\in\mathbb{C}^{FN}} I_\Gamma(\mathbf{x}) + I_{\ell_0\leq\kappa}(\mathbf{c}) \quad \text{subject to} \quad \mathbf{Tc} - \mathbf{x} = \mathbf{0}, \tag{4.15}$$

where the function $I_{\ell_0\leq\kappa}$ denotes the indicator function of the set of κ-sparse vectors $\{\mathbf{c}\in\mathbb{C}^{FN} \mid \|\mathbf{c}\|_0 \leq \kappa\}$.

Note that (4.15) is not meant to be necessarily solvable, since meeting all three conditions ($\mathbf{x}\in\Gamma$, $\|\mathbf{c}\|_0\leq\kappa$, $\mathbf{Tc}-\mathbf{x}=\mathbf{0}$) for a small value of κ is improbable in practice. This is, however, not a problem, since the formulation (4.15) serves only as a starting point to derive the ADMM steps, which will approximate the solution such that the condition $\mathbf{Tc}-\mathbf{x}=\mathbf{0}$ is effectively relaxed.

To employ algorithm 1.3, we need to find solutions to the subproblems[7]

$$\mathbf{c}^{(n)} = \operatorname*{arg\,min}_{\mathbf{c}\in\mathbb{C}^{FN}} I_{\ell_0\leq\kappa}(\mathbf{c}) + \tfrac{\varrho}{2}\|\mathbf{Tc} - \mathbf{x}^{(n-1)} + \mathbf{u}^{(n-1)}\|_2^2, \tag{4.16a}$$

$$\mathbf{x}^{(n)} = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^L} I_\Gamma(\mathbf{x}) + \tfrac{\varrho}{2}\|\mathbf{Tc}^{(n)} - \mathbf{x} + \mathbf{u}^{(n-1)}\|_2^2. \tag{4.16b}$$

Equation (4.16a) corresponds to the update on line 2 of algorithm 1.3 with fixed $\mathbf{x}^{(n-1)}\in\mathbb{R}^L$ and $\mathbf{u}^{(n-1)}\in\mathbb{R}^L$. Similarly, (4.16b) corresponds to line 3 of the same algorithm with fixed $\mathbf{c}^{(n)}\in\mathbb{C}^{FN}$ and $\mathbf{u}^{(n-1)}\in\mathbb{R}^L$.

The update (4.16a) is a sparse synthesis problem, where the vector $\mathbf{x}^{(n-1)}-\mathbf{u}^{(n-1)}$ is to be optimally approximated[8] by a κ-sparse combination of the atoms (columns of $\mathbf{T}$). In case of non-orthogonal $\mathbf{T}$, such as when $\mathbf{T}$ corresponds to the synthesis operator of an overcomplete frame, this problem is NP-hard. In line with [95], we can use the approximation

$$\mathbf{c}^{(n)} = \mathbf{c}_{\text{hard}}^{(n)} = \operatorname{hard}_\kappa\left(\mathbf{L}(\mathbf{x}^{(n-1)} - \mathbf{u}^{(n-1)})\right), \tag{4.17}$$

where the operator $\operatorname{hard}_\kappa$ denotes hard thresholding, keeping κ largest values of its argument and setting the rest to zero.

Remark 4.9. The choice (4.17) can be easily justified in the case of Parseval tight

[7] The factor $\frac{\varrho}{2}$ in both (4.16a) and (4.16b) could be omitted due to the nature of the indicator function and the fact that the actual value of the minima is not used.

[8] Note that the approximation can be exact for a suitable choice of $\mathbf{T}$ and large enough κ, meaning that the objective function of the attains zero value. Then, the task (4.16a) in fact corresponds to the so-called exact sparse problem, see e.g. [96].

frame, i.e., in the case $\mathbf{TL} = \mathbf{I}$ (see definition 1.7). Then

$$\begin{aligned}\|\mathbf{Tc} - \mathbf{x}^{(n-1)} + \mathbf{u}^{(n-1)}\|_2^2 &= \|\mathbf{Tc} - \mathbf{TL}(\mathbf{x}^{(n-1)} - \mathbf{u}^{(n-1)})\|_2^2 \\ &= \|\mathbf{T}(\mathbf{c} - \mathbf{L}(\mathbf{x}^{(n-1)} - \mathbf{u}^{(n-1)}))\|_2^2 \\ &\leq \|\mathbf{c} - \mathbf{L}(\mathbf{x}^{(n-1)} - \mathbf{u}^{(n-1)})\|_2^2,\end{aligned}$$

since for a Parseval tight frame with analysis operator, it can be shown that $\|\mathbf{Tc}\| \leq \|\mathbf{c}\|$ for any $\mathbf{c}$ [97, Lemma 5.4.5].[9] It can also be shown (see [2, Remark 2]) that the κ-sparse $\mathbf{c}$ minimizing the norm $\|\mathbf{c} - \mathbf{L}(\mathbf{x}^{(n-1)} - \mathbf{u}^{(n-1)})\|_2^2$ can be expressed as $\mathbf{c}_{\text{hard}}^{(n)} = \text{hard}_\kappa(\mathbf{L}(\mathbf{x}^{(n-1)} - \mathbf{u}^{(n-1)}))$. Returning to the above inequality, we have $\|\mathbf{Tc}_{\text{hard}}^{(n)} - \mathbf{x}^{(n-1)} + \mathbf{u}^{(n-1)}\|_2^2 \leq \|\mathbf{c}_{\text{hard}}^{(n)} - \mathbf{L}(\mathbf{x}^{(n-1)} - \mathbf{u}^{(n-1)})\|_2^2$, which gives us an upper bound on the error of the sparse approximation. This can be interpreted as a guarantee of the goodness of $\mathbf{c}_{\text{hard}}^{(n)}$ as an approximate solution to the problem (4.16a).

As an alternative, we have proposed in [6] to use the OMP [60] to find the sparse approximation of $\mathbf{v} = \mathbf{x}^{(n-1)} - \mathbf{u}^{(n-1)}$. In our case, this greedy iterative algorithm works with a sequence of candidate solutions $\mathbf{z}^{(i)}$ such that $\|\mathbf{z}^{(i)}\|_0 = i$, and residuals $\mathbf{r}^{(i)} = \mathbf{Tz}^{(i)} - \mathbf{v}$. The algorithm starts from $\mathbf{z}^{(0)} = \mathbf{0}$. Then, in i-th iteration, a column from $\mathbf{T}$ which is maximally correlated with the current residual $\mathbf{r}^{(i)}$ is added to the support of $\mathbf{z}^{(i)}$, increasing its ℓ_0 norm by one, and $\mathbf{z}^{(i+1)}$ is obtained as an optimal approximation of $\mathbf{v}$ with the columns of $\mathbf{T}$ defined by the currently chosen support[10]. After κ iterations, we obtain $\mathbf{z}^{(\kappa)}$ as an approximate solution to (4.16a), which we denote $\mathbf{c}_{\text{OMP}}^{(n)}$.

Since the OMP is an iterative algorithm which uses a computationally demanding projection in each step, using $\mathbf{c}^{(n)} = \mathbf{c}_{\text{OMP}}^{(n)}$ is much less efficient than the choice $\mathbf{c}^{(n)} = \mathbf{c}_{\text{hard}}^{(n)}$. On the other hand, OMP can provide a better candidate in terms of the objective function of (4.16a).

Contrary to the subproblem (4.16a), which needed to be approximated, the update (4.16b) can be computed precisely as a projection $\mathbf{x}^{(n)} = \text{proj}_\Gamma(\mathbf{Tc}^{(n)} + \mathbf{u}^{(n-1)})$ (see table 1.1 and the definition of the proximal operator 1.22).

Altogether, S-SPAIN is summarized by algorithm 4.7, where the scheduling of κ is taken from [95], i.e., it increases by the value of s every r iterations.

As mentioned before, the algorithm stops when the norm $\|\mathbf{Tc}^{(n)} - \mathbf{x}^{(n)}\|_2$ drops below the desired tolerance ε.

[9] Note that [97, Lemma 5.4.5] only discusses coefficient vectors which are orthogonal to the null space of $\mathbf{T}$. Nevertheless, since $\mathbf{Tc} = \mathbf{0}$ for $\mathbf{c}$ in the null space, the property can be easily extended for any $\mathbf{c}$.

[10] Since the support is fixed now, this approximation problem is not NP-hard and it has a closed form solution.

Algorithm 4.7: S-SPAIN

```
input: $\mathbf{x}(0) \in \mathbb{R}^L$, $\mathbf{u}^{(0)} \in \mathbb{R}^L$, sparsity scheduling parameters $r$, $s$
$\kappa^{(0)} = 0$
for $n = 1, 2, \ldots$ do
    if $(n-1) \mod r = 0$ then
        $\kappa^{(n)} = \kappa^{(n-1)} + s$
    else
        $\kappa^{(n)} = \kappa^{(n-1)}$
    end
    if hard thresholding chosen then
        $\mathbf{c}^{(n)} = \mathrm{hard}_{\kappa^{(n)}}\left(\mathbf{L}(\mathbf{x}^{(n-1)} - \mathbf{u}^{(n-1)})\right)$
    else
        $\mathbf{c}^{(n)}$ = solution of the OMP algorithm with dictionary $\mathbf{T}$ and $\kappa^{(n)}$
          iterations to approximate the vector $\mathbf{x}^{(n-1)} - \mathbf{u}^{(n-1)}$
    end
    $\mathbf{x}^{(n)} = \mathrm{proj}_\Gamma\left(\mathbf{T}\mathbf{c}^{(n)} + \mathbf{u}^{(n-1)}\right)$
    $\mathbf{u}^{(n)} = \mathbf{u}^{(n-1)} + \mathbf{T}\mathbf{c}^{(n)} - \mathbf{x}^{(n)}$
end
output: $\hat{\mathbf{x}} = \mathbf{x}^{(n)}$, $\hat{\mathbf{c}} = \mathbf{c}^{(n)}$
```

4.2.3 A-SPAIN

As in the synthesis case, the Analysis-based Sparse Audio Inpainter (A-SPAIN) is based on the following reformulation of (4.14b) to fit problem (1.28) suitable for the ADMM:

$$(\hat{\mathbf{x}}, \hat{\mathbf{c}}) = \mathop{\arg\min}_{\mathbf{x}\in\mathbb{R}^L,\, \mathbf{c}\in\mathbb{C}^{FN}} \; I_\Gamma(\mathbf{x}) + I_{\ell_0 \le \kappa}(\mathbf{c}) \quad \text{subject to} \quad \mathbf{L}\mathbf{x} - \mathbf{c} = \mathbf{0}. \tag{4.18}$$

To employ ADMM, the augmented Lagrangian is minimized in two alternating steps (compare to (4.16) and lines 2 and 3 of algorithm 1.3):

$$\mathbf{c}^{(n)} = \mathop{\arg\min}_{\mathbf{c}\in\mathbb{C}^{FN}} I_{\ell_0 \le \kappa}(\mathbf{c}) + \tfrac{\varrho}{2}\|\mathbf{L}\mathbf{x}^{(n-1)} - \mathbf{c} + \mathbf{u}^{(n-1)}\|_2^2, \tag{4.19a}$$

$$\mathbf{x}^{(n)} = \mathop{\arg\min}_{\mathbf{x}\in\mathbb{R}^L} I_\Gamma(\mathbf{x}) + \tfrac{\varrho}{2}\|\mathbf{L}\mathbf{x} - \mathbf{c}^{(n)} + \mathbf{u}^{(n-1)}\|_2^2. \tag{4.19b}$$

Remark 4.10 (Update ordering)**.** We have chosen a different order of the updates in equations (4.19) compared with the general form of ADMM in algorithm 1.3. This will create an almost equivalent variation on the ADMM, see e.g. [38, pp. 14, 22]. The reason for this alteration is to provide a more straightforward comparison of A-SPAIN and S-SPAIN. We will follow this convention in the rest of the derivation, including the final algorithm 4.8 with the coefficient update on line 8 and the signal update on line 9.

In the case of (Parseval) tight frames, it can be shown that the subproblem (4.19b) is equivalent to the explicitly solvable minimization [2, Remark 3]

$$\mathbf{x}^{(n)} = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^L} I_{\Gamma}(\mathbf{x}) + \tfrac{\rho}{2}\|\mathbf{x} - \mathbf{T}(\mathbf{c}^{(n)} - \mathbf{u}^{(n-1)})\|_2^2 = \operatorname{proj}_{\Gamma}\left(\mathbf{T}(\mathbf{c}^{(n)} - \mathbf{u}^{(n-1)})\right). \quad (4.20)$$

According to [2, Remark 2], the solution to problem (4.19a) can be found using the hard thresholding operator:

$$\mathbf{c}^{(n)} = \operatorname{hard}_{\kappa^{(n)}}\left(\mathbf{L}(\mathbf{x}^{(n-1)} + \mathbf{u}^{(n-1)})\right). \quad (4.21)$$

In contrast to equation (4.17) in the synthesis case, it is the exact solution in this case, not only a possible approximation.

A-SPAIN is summarized in algorithm 4.8. It includes the same sparsity scheduling as in the case of S-SPAIN. The stopping criterion follows the original formulation (4.14a), i.e., we stop when $\|\mathbf{L}\mathbf{x}^{(n)} - \mathbf{c}^{(n)}\|_2 < \varepsilon$ for chosen value of ε.

Algorithm 4.8: A-SPAIN

```
input: $\mathbf{x}(0) \in \mathbb{R}^L$, $\mathbf{u}^{(0)} \in \mathbb{C}^{FN}$, sparsity scheduling parameters $r$, $s$
$k^{(0)} = 0$
for $n = 1, 2, \dots$ do
    if $(n-1) \mod r = 0$ then
        $k^{(n)} = k^{(n-1)} + s$
    else
        $k^{(n)} = k^{(n-1)}$
    end
    $\mathbf{c}^{(n)} = \operatorname{hard}_{k^{(n)}}\left(\mathbf{L}\mathbf{x}^{(n-1)} + \mathbf{u}^{(n-1)}\right)$
    $\mathbf{x}^{(n)} = \operatorname{proj}_{\Gamma}\left(\mathbf{T}(\mathbf{c}^{(n)} - \mathbf{u}^{(n-1)})\right)$
    $\mathbf{u}^{(n)} = \mathbf{u}^{(n-1)} + \mathbf{L}\mathbf{x}^{(n)} - \mathbf{c}^{(n)}$
end
output: $\hat{\mathbf{x}} = \mathbf{x}^{(n)}$, $\hat{\mathbf{c}} = \mathbf{c}^{(n)}$
```

5 Methods based on low-rank representations

All the methods previously discussed were based upon sparsity of the time-frequency representation of plausible audio signals. In section 2.3, we have also briefly mentioned the social sparsity approach which allows to exploit structures in the spectrogram such as its tonal and transient components. However, there exists a more *semantic* model of audio, which is connected to the non-negative matrix factorization (NMF).

Consider a (power) spectrogram $\mathbf{P} \in \mathbb{R}^{F \times N}$ of a non-distorted audio signal. This can be seen as a matrix whose dimensions correspond to the number of frequency bins F and number of temporal frames N of the STFT used to analyze the signal, see example 1.9 and remark 1.10. The goal of NMF is to (approximately) factorize $\mathbf{P}$ as $\mathbf{P} \approx \mathbf{W}\mathbf{H}$ with non-negative matrices $\mathbf{W} \in \mathbb{R}^{F \times K}$ and $\mathbf{H} \in \mathbb{R}^{K \times N}$, and with small value of K. The interpretation, and the possible relation to semantics of audio, is that the matrix $\mathbf{W}$ contains spectral patterns the audio is composed of (such as individual tones), while the matrix $\mathbf{H}$ models the activation of these patterns in time. This interpretation also justifies the non-negativity assumption, since the non-negative power spectrogram $\mathbf{P}$ is modeled as a superposition of power spectral patterns. The intuition is illustrated by figure 5.1 which shows the NMF of a spectrogram of a violin recording.

This chapter is devoted to derivation of algorithms which use the NMF model[1] to interpolate missing audio segments, as previously published in [8]. The task is posed as an estimation problem, which assumes that the spectrogram of the interpolated audio signal is a collection of Gaussian random variables, and if organized in a matrix, the variances of those variables exhibit the NMF structure. This approach is partly based on a successful application to audio declipping [89, 98].

The core of the interpolation method lies in estimating this set of variables given the observed parts of the signal. Section 5.2 presents an estimation approach based on the expectation–maximization (EM) algorithm [99], which treats the missing signal samples as latent variables of the problem. Two algorithms are described, depending on the domain of the so-called *complete data*, which is a part of the design of the EM algorithm. Section 5.3 then presents an innovative approach to solving the estimation problem using an alternating-minimization strategy instead of the EM algorithm. For implementation-related practical remarks, see appendix A.

[1] We will use the acronym NMF not only for the factorization itself, but also in related contexts, such as to describe the low-rank structure of a matrix imposed by the factorization, the “modelability” of a matrix as a low-rank product etc.

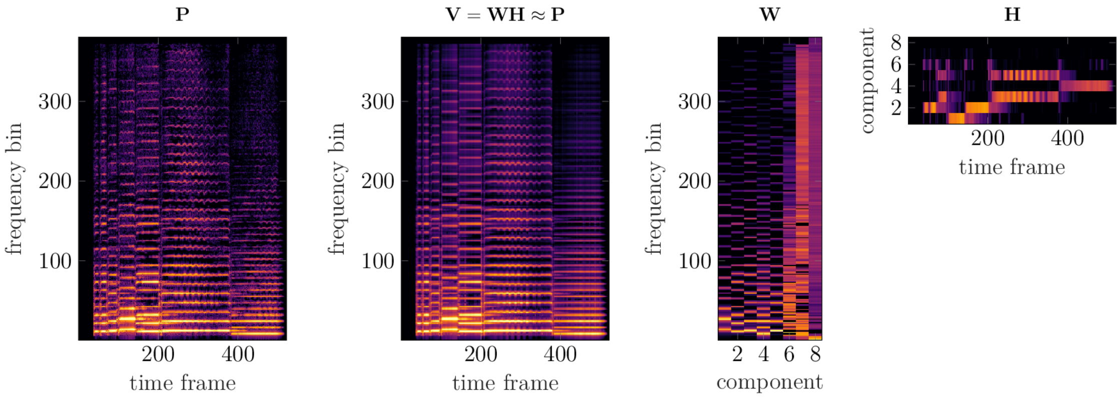


Fig. 5.1: Spectrogram decomposition with NMF using $K = 8$ components. In the matrix $\mathbf{W}$, we may identify 5 tones (first 5 components), distinguished by different fundamental frequency and different mixture of the harmonics. Components 6–8 are rather atonal; these cover for example the screeching sound of the violin bow (component 6) and some residual noises. On the other hand, the matrix $\mathbf{H}$ dictates how to "compose" the signal using these components. For example, we see that the tone represented by component 2 appears twice at the beginning and then it is repeated after two other notes (component 3 and 1). Another remarkable example is the vibrato after frame 200, which is represented as repeated switching between components 3 and 5 with fundamental frequencies close to each other.

5.1 Assumptions and notation for the NMF modeling

Recall that $\mathbf{x}^{\text{true}} \in \mathbb{R}^L$ denotes the original, undistorted time domain signal. The goal is to interpolate the missing samples given a partial observation $\mathbf{x}^{\text{obs}} = \mathbf{M}\mathbf{x}^{\text{true}}$ and a low-rank NMF model of the power STFT spectrum of the original signal. The restored signal, i.e., an estimation of $\mathbf{x}^{\text{true}}$, shall be denoted $\hat{\mathbf{x}} \in \mathbb{R}^L$.

Denote $\mathbf{X}^{\text{true}} = \{\mathbf{x}_1^{\text{true}}, \ldots, \mathbf{x}_N^{\text{true}}\}$ the windowed time domain signal[2], $\mathbf{x}_n^{\text{true}} \in \mathbb{R}^W$ for $n = 1, \ldots, N$. Although independence of the individual frames is not in fact true in the case of overlapping frames, we will assume it throughout this chapter in order to simplify the derivations. Due to the correspondence of the whole signal and its windowed version[3], estimating $\hat{\mathbf{x}} \in \mathbb{R}^L$ corresponds to estimating the individual frames as $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_N\}$.

[2] Due to the constant window length W, we could also consider the windowed signal to be a matrix $\mathbf{X}^{\text{true}} = [\mathbf{x}_1^{\text{true}}, \ldots, \mathbf{x}_N^{\text{true}}] \in \mathbb{R}^{W \times N}$. However, since the length of the observed and missing parts of the frames may differ for each n, we use the set notation for the sake of consistency.

[3] We are aware of the fact that in the case of overlapping (and window-weighted) time frames, the correspondence of the signal and its framed version is not one-to-one. This is because using the overlap-add procedure, several different framed signals may produce a single time domain signal. However, this ambiguity is not considered significant.

In each time frame n, the reliable samples can be selected by the multiplication with the selection matrix $\mathbf{M}_n$, as in remark 2.1, which gives the observed part of the windowed signal $\mathbf{X}^{\text{obs}} = \{\mathbf{x}_1^{\text{obs}}, \dots, \mathbf{x}_N^{\text{obs}}\}$. Selection with the complementary masks $\overline{\mathbf{M}}_n$ for $n = 1, \dots, N$ results in the parts of the signal to be interpolated, denoted $\mathbf{X}^{\text{miss}} = \{\mathbf{x}_1^{\text{miss}}, \dots, \mathbf{x}_N^{\text{miss}}\}$. To establish the low-rank, NMF-related representation of the spectrum of the original signal, we will resort to the STFT (see example 1.9) in practice. However, the concept will be treated in a slightly more general manner, for which we postulate the following assumptions.

Assumption 5.1 (Reconstruction operator)**.** There exists a linear reconstruction operator, represented by the matrix $\mathbf{T} \in \mathbb{C}^{W \times F}$, that reconstructs a signal window $\mathbf{x}_n^{\text{true}} \in \mathbb{R}^W$ from the representation $\mathbf{c}_n^{\text{true}} \in \mathbb{C}^F$ as

$$\mathbf{x}_n^{\text{true}} = \mathbf{T}\mathbf{c}_n^{\text{true}}, \quad n = 1, \dots, N. \tag{5.1}$$

There also exists the analysis operator, represented by the matrix $\mathbf{L} \in \mathbb{C}^{F \times W}$.

Remark 5.2. Recall that for the sake of simplicity, we slightly abuse the notation of the synthesis and analysis operators as defined in equation (1.9) and used in the previous chapter. Here, the operators $\mathbf{T}$ and $\mathbf{L}$ only operate on the signal window or coefficients related to this window, not the whole signal. Also, the relationship of $\mathbf{T}$ and $\mathbf{L}$ is clear in the context of frames (where they are mutually adjoint), but we keep it general for now and discuss the connection later in section 5.2.3.

Based on the established notation, the (synthesis) time-frequency coefficients of the framed signal $\mathbf{X}^{\text{true}}$ can be organized in a matrix $\mathbf{C}^{\text{true}} = [\mathbf{c}_1^{\text{true}}, \dots, \mathbf{c}_N^{\text{true}}]$, see also example 1.9. Similarly, an estimation of these coefficients (which allows reconstruction of $\hat{\mathbf{X}}$) shall be denoted $\hat{\mathbf{C}} = [\hat{\mathbf{c}}_1, \dots, \hat{\mathbf{c}}_N]$.

Assumption 5.3 (Gaussian coefficients)**.** Coefficients of the original audio in the time-frequency domain are all mutually independent, and each coefficient follows a complex circular zero-mean Gaussian distribution (see remark 5.4)

$$c_{fn}^{\text{true}} \sim \mathcal{N}(0, v_{fn}). \tag{5.2}$$

Note that due to the assumed independence of the individual time-frequency coefficients, we can rewrite the distribution for the spectrum of a frame as

$$\mathbf{c}_n^{\text{true}} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{D}_n\right), \quad \mathbf{D}_n = \operatorname{diag}\left([v_{fn}]_{f=1,\dots,F}\right), \tag{5.3}$$

where the symbol $\mathcal{N}$ now represents a multivariate complex Gaussian distribution.

Remark 5.4 (Gaussian distribution)**.** Similarly to the real case, the (P-variate) complex zero-mean Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ can be defined using a $P \times P$ Hermitian positive definite complex covariance matrix $\boldsymbol{\Sigma}$ via the probability density function (PDF) [100, Sec. 7]

$$p(\mathbf{u}) = \det\left(\pi\boldsymbol{\Sigma}\right)^{-1} \exp\Big(-\mathbf{u}^{*}\boldsymbol{\Sigma}^{-1}\mathbf{u}\Big). \tag{5.4}$$

It turns out that this PDF exhibits the circular symmetry, which means that it holds $p(\mathrm{e}^{\mathrm{i}\theta}\mathbf{u}) = p(\mathbf{u})$ for any phase shift θ [101, Appendix A.1.3]. In this work, we allow also a shift by a vector $\boldsymbol{\mu} \in \mathbb{C}^{P}$, which leads to the distribution denoted $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and defined naturally by the PDF [44, Appendix A]

$$p(\mathbf{u}) = \det\left(\pi\boldsymbol{\Sigma}\right)^{-1} \exp\Big(-(\mathbf{u}-\boldsymbol{\mu})^{*}\boldsymbol{\Sigma}^{-1}(\mathbf{u}-\boldsymbol{\mu})\Big). \tag{5.5}$$

If $\mathbf{u} \in \mathbb{C}^{P}$ follows the circular symmetric (thus inherently zero-mean) Gaussian distribution with a covariance matrix $\boldsymbol{\Sigma}$, and $\mathbf{A} \in \mathbb{C}^{Q\times P}$ is a full-rank constant matrix, then $\mathbf{v} = \mathbf{A}\mathbf{u}$ follows the circular symmetric Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^{*})$ [101, Appendix A.1.3]. Considering also the shift from zero, as in (5.5), and a constant vector $\mathbf{b} \in \mathbb{C}^{Q}$, we can generalize the formula for affine transformation of real Gaussian vectors (see e.g. [102, Thm. 4.4]) into the complex case:

$$\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \Longrightarrow \quad \mathbf{v} = \mathbf{A}\mathbf{u} + \mathbf{b} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^{*}). \tag{5.6}$$

Assumption 5.5 (NMF structure of the coefficients)**.** The variance matrix $\mathbf{V} = [v_{fn}]$ has the low-rank NMF structure

$$v_{fn} = \sum_{k=1}^{K} w_{fk}h_{kn}, \tag{5.7}$$

where K is small and the variables are non-negative reals. This model amounts to $\mathbf{V} = \mathbf{W}\mathbf{H}$ with $\mathbf{W}$ and $\mathbf{H}$ being, respectively, $F \times K$ and $K \times N$ non-negative matrices [89, Sec. 2.2].

Based on assumption 5.1 (Reconstruction operator), the observation is given by the *linear* model

$$\mathbf{x}_n^{\mathrm{obs}} = \mathbf{M}_n\mathbf{x}_n^{\mathrm{true}} = \mathbf{M}_n(\mathbf{T}\mathbf{c}_n^{\mathrm{true}}) = (\mathbf{M}_n\mathbf{T})\,\mathbf{c}_n^{\mathrm{true}}. \tag{5.8}$$

5.2 Maximum likelihood estimation by treating the missing samples as latent variables

The audio interpolation task can be posed as a maximum likelihood estimation (MLE) problem of the following form:

$$\hat{\mathbf{W}}, \hat{\mathbf{H}} = \operatorname*{arg\,max}_{\mathbf{W},\mathbf{H}} p(\, \mathbf{X}^{\text{obs}} \mid \mathbf{W}, \mathbf{H}\,). \tag{5.9}$$

In words, we search for the parameters $\mathbf{W}, \mathbf{H}$ of the statistical model described in the previous section 5.1, such that under this statistical model, the observed data is most probable. The windowed signal block estimate $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_n\}_{n=1,\ldots,N}$ is then recovered from $\hat{\mathbf{W}}$, $\hat{\mathbf{H}}$ and the reliable data $\mathbf{X}^{\text{obs}} = \{\mathbf{x}_n^{\text{obs}}\}_{n=1,\ldots,N}$ via Wiener filtering (see (5.14a)) and synthesis using $\mathbf{T}$ [98, Sec. IV.J].

From the linear observation model (5.8) and the assumed distribution of $\mathbf{c}_n^{\text{true}}$ in (5.3), we may derive that $(\mathbf{x}_n^{\text{obs}} \mid \mathbf{W}, \mathbf{H}) \sim \mathcal{N}(\mathbf{0}, \mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top)$, see also equation (5.12) below. This allows to express the objective of (5.9) as

$$\begin{aligned} p(\mathbf{X}^{\text{obs}} \mid \mathbf{W}, \mathbf{H}) &= \prod_{n=1}^{N} p\left(\mathbf{x}_n^{\text{obs}} \mid \mathbf{W}, \mathbf{H}\right) \\ &= \prod_{n=1}^{N} \det\left(\pi\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1} \exp\left(-(\mathbf{x}_n^{\text{obs}})^\top(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top)^{-1}\mathbf{x}_n^{\text{obs}}\right), \end{aligned} \tag{5.10}$$

where the product is a corollary of the assumption of independence of the individual signal frames. Note that even though we have a closed expression for the objective of the problem (5.9), it is non-convex with respect to the sought parameters $\mathbf{W}, \mathbf{H}$. Thus, we propose to employ the EM algorithm to aim at finding at least a local optimum.

5.2.1 The EM-*tf* algorithm

To design a particular form of the EM algorithm, we set the *incomplete data* to correspond to the set of all the windows of the observed reliable signal, i.e., $\mathbf{X}^{\text{obs}}$. The *complete data* correspond to the STFT spectrum $\mathbf{C}^{\text{true}} \in \mathbb{C}^{F\times N}$ of the original signal. The parameters to be estimated are $\theta = \{\mathbf{W}, \mathbf{H}\}$, while their current estimate is denoted $\tilde{\theta}$.

The EM algorithm aims at solving the MLE problem (5.9) by minimizing the functional

$$Q(\theta, \tilde{\theta}) = -\int \log p(\mathbf{C}^{\text{true}} \mid \theta) p(\mathbf{C}^{\text{true}} \mid \mathbf{X}^{\text{obs}}, \tilde{\theta})\, \mathrm{d}\mathbf{C}^{\text{true}} \tag{5.11}$$

by repeating two principal steps:

1. **E-step**: Compute $Q(\theta, \tilde{\theta})$, i.e., the expected value $\hat{\mathbf{C}}$ of $\mathbf{C}^{\text{true}} \in \mathbb{C}^{F \times N}$ (the STFT spectrum of the original clean signal $\mathbf{x}^{\text{true}}$) using the current estimate $\tilde{\theta}$ of the parameters $\mathbf{W}$, $\mathbf{H}$ and the reliable data $\mathbf{X}^{\text{obs}}$.
2. **M-step**: Use the data from the expectation step as if it were actually measured to determine a maximum-likelihood estimated of the parameters $\tilde{\theta}$ by minimizing $Q(\theta, \tilde{\theta})$ with respect to θ. In our case, this step corresponds to updating $\mathbf{W}$ and $\mathbf{H}$ as a decomposition of the posterior power spectrogram $\mathbf{P}$.

After meeting a chosen stopping criterion (or after a fixed number of iterations), the algorithm ends with $\hat{\mathbf{C}}$ from the E-step as its output. The whole signal $\hat{\mathbf{x}} \in \mathbb{R}^L$ can be formed by an overlap-add scheme from the corresponding time-domain frames $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_n = \mathbf{T}\hat{\mathbf{c}}_n\}_{n=1,\dots,N}$.

Remark 5.6 (Signal constraints)**.** In the original paper [89], the NMF-based algorithm described above is applied to the problem of audio declipping. In that case, signal constraints (such as exceeding the clipping levels) can be enforced heuristically between the E-step and the M-step. This is done by modifying the distribution of the time-frequency coefficients (by updating $\hat{\mathbf{c}}_n$ and $\hat{\mathbf{\Sigma}}_n$ of (5.14) for each n), such that the corresponding restored time-domain frames meet the given signal constraints. Then, the posterior power spectrogram $\mathbf{P}$ is re-estimated given the distribution of $\hat{\mathbf{c}}_n$, the parameters $\mathbf{W}$, $\mathbf{H}$ and the reliable data $\mathbf{X}^{\text{obs}}$. However, no constraints are needed in our interpolation setting where only the missing samples are being estimated without modifying the known samples, thus the only constraint $\hat{\mathbf{x}} \in \mathit{\Gamma}$ (see section 2.1) is satisfied.

The whole algorithm is detailed in algorithm 5.1. The individual steps are justified in the following subsections.

E-step

This step estimates the STFT spectrum of $\mathbf{x}^{\text{true}}$ by estimating the frequency spectrum $\mathbf{c}_n^{\text{true}}$ of $\mathbf{x}_n^{\text{true}}$ for each n. It follows from linearity of the transform in (5.8) and from the distribution (5.3) that remark 5.4 applies and it holds

$$\begin{bmatrix} \mathbf{c}_n^{\text{true}} \\ \mathbf{x}_n^{\text{obs}} \end{bmatrix} = \begin{bmatrix} \mathbf{c}_n^{\text{true}} \\ (\mathbf{M}_n\mathbf{T})\,\mathbf{c}_n^{\text{true}} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{D}_n & \mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top \\ \mathbf{M}_n\mathbf{T}\mathbf{D}_n & \mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top \end{bmatrix}\right). \tag{5.12}$$

Since $\mathbf{W}$ and $\mathbf{H}$ are fixed in the E-step, the distribution of the measurement $\mathbf{x}_n^{\text{obs}}$ is known. The aim is to estimate the complete data $\mathbf{c}_n^{\text{true}}$ in the sense of minimal mean squared error (MSE), i.e., to find its expected value $\hat{\mathbf{c}}_n$. This is clear from the conditional probability of the STFT coefficients given the observed time domain samples:

$$p(\mathbf{C}^{\text{true}} \mid \mathbf{X}^{\text{obs}}, \tilde{\theta}) = p(\mathbf{C}^{\text{true}} \mid \mathbf{X}^{\text{obs}}, \mathbf{W}, \mathbf{H}) \\ = \prod_{n=1}^{N} p(\mathbf{c}_n^{\text{true}} \mid \mathbf{x}_n^{\text{obs}}, \mathbf{W}, \mathbf{H}) = \prod_{n=1}^{N} \mathcal{N}(\hat{\mathbf{c}}_n, \hat{\boldsymbol{\Sigma}}_n), \quad (5.13)$$

where, using the Gaussian assumptions and (5.12), it holds [103, Theorem 10.3]

$$\hat{\mathbf{c}}_n = \mathbf{D}_n \mathbf{T}^* \mathbf{M}_n^\top \left(\mathbf{M}_n \mathbf{T} \mathbf{D}_n \mathbf{T}^* \mathbf{M}_n^\top \right)^{-1} \mathbf{x}_n^{\text{obs}}, \quad (5.14a)$$

$$\hat{\boldsymbol{\Sigma}}_n = \mathbf{D}_n - \mathbf{D}_n \mathbf{T}^* \mathbf{M}_n^\top \left(\mathbf{M}_n \mathbf{T} \mathbf{D}_n \mathbf{T}^* \mathbf{M}_n^\top \right)^{-1} \mathbf{M}_n \mathbf{T} \mathbf{D}_n. \quad (5.14b)$$

Note that by the notation $\mathcal{N}(\hat{\mathbf{c}}_n, \hat{\boldsymbol{\Sigma}}_n)$ in (5.13) and also later in the same context, we actually mean the probability density p associated with the distribution $\mathcal{N}(\hat{\mathbf{c}}_n, \hat{\boldsymbol{\Sigma}}_n)$, see equation (5.5).

Remark 5.7 (Connection to the Moore–Penrose pseudoinverse)**.** If we set $\mathbf{B} = \mathbf{M}_n \mathbf{T} \sqrt{\mathbf{D}}_n$ (the square root is applied on the diagonal elements), we see from equation (5.14a) that $\hat{\mathbf{c}}_n = \sqrt{\mathbf{D}}_n \mathbf{B}^* (\mathbf{B}\mathbf{B}^*)^{-1} \mathbf{x}_n^{\text{obs}} = \sqrt{\mathbf{D}}_n \mathbf{B}^+ \mathbf{x}_n^{\text{obs}}$, where $\mathbf{B}^+$ is the Moore–Penrose pseudoinverse of $\mathbf{B}$, see also definition 1.16. In words, $\hat{\mathbf{c}}_n$ is a least-squares solution to the linear system $\mathbf{B}\mathbf{z} = \mathbf{x}_n^{\text{obs}}$, scaled row-wise by the square roots of the diagonal entries of $\mathbf{D}_n$.

The expression $p(\mathbf{C}^{\text{true}} \mid \theta)$ from (5.11) can be broken down using (5.3) as

$$p(\mathbf{C}^{\text{true}} \mid \theta) = \prod_{n=1}^{N} p(\mathbf{c}_n^{\text{true}} \mid \theta) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{0}, \mathbf{D}_n), \quad (5.15)$$

where the parameters $\mathbf{W}, \mathbf{H}$ appear in the matrices $\mathbf{D}_n = \operatorname{diag}([v_{fn}]_{f=1,\dots,F})$ due to assumption 5.5 (NMF structure of the coefficients).

M-step

It can be shown that the minimization of (5.11) with respect to the parameters $\mathbf{W}, \mathbf{H}$ is equivalent to the minimization of the Itakura–Saito divergence between $\mathbf{P}$ and the product $\mathbf{W}\mathbf{H}$, $D_{\text{IS}}(\mathbf{P} \mid \mathbf{W}\mathbf{H})$ (see equation (1.34)) [89, 44]. Here, $\mathbf{P} = [p_{fn}]$ denotes the posterior power spectrum, which is given from (5.14) as

$$p_{fn} = \mathbb{E}\left(\left| c_{fn}^{\text{true}} \right|^2 \mid \mathbf{x}_n^{\text{obs}}, \mathbf{W}, \mathbf{H} \right) = \left| \hat{\mathbf{c}}_n(f) \right|^2 + \hat{\boldsymbol{\Sigma}}_n(f, f). \quad (5.16)$$

The minimization of $D_{\text{IS}}(\mathbf{P} \mid \mathbf{W}\mathbf{H})$ can be performed by the multiplicative updates of algorithm 1.4.

Algorithm 5.1: Audio interpolation via EM-*tf*.

```
input: reliable samples $\mathbf{X}^{\text{obs}} = \{\mathbf{x}_n^{\text{obs}}\}_{n=1,\dots,N}$, respective selection matrices
       $\{\mathbf{M}_n\}_{n=1,\dots,N}$, linear transform $\mathbf{T} \in \mathbb{C}^{W \times F}$
initialize $\mathbf{W} \in \mathbb{R}^{F \times K}$, $\mathbf{H} \in \mathbb{R}^{K \times N}$ non-negative
repeat
    // E-step:
    for $n = 1, \dots, N$ do
        $\mathbf{D}_n \leftarrow \operatorname{diag}\left([v_{fn}]_{f=1,\dots,F}\right)$ with $[v_{fn}]_{f=1,\dots,F}$ being the $n$-th column of
         the matrix $\mathbf{V} = \mathbf{WH}$
        $\hat{\mathbf{c}}_n \leftarrow \mathbf{D}_n \mathbf{T}^* \mathbf{M}_n^\top \left(\mathbf{M}_n \mathbf{T} \mathbf{D}_n \mathbf{T}^* \mathbf{M}_n^\top\right)^{-1} \mathbf{x}_n^{\text{obs}}$
        $\hat{\boldsymbol{\Sigma}}_n \leftarrow \mathbf{D}_n - \mathbf{D}_n \mathbf{T}^* \mathbf{M}_n^\top \left(\mathbf{M}_n \mathbf{T} \mathbf{D}_n \mathbf{T}^* \mathbf{M}_n^\top\right)^{-1} \mathbf{M}_n \mathbf{T} \mathbf{D}_n$
        $p_{fn} \leftarrow |\hat{\mathbf{c}}_n(f)|^2 + \hat{\boldsymbol{\Sigma}}_n(f,f),\ f = 1, \dots, F$
    end
    // M-step:
    repeat
        $\mathbf{W} \leftarrow \mathbf{W} \odot \dfrac{\left((\mathbf{WH})^{\odot[-2]} \odot \mathbf{P}\right)\mathbf{H}^\top}{(\mathbf{WH})^{\odot[-1]}\mathbf{H}^\top}$ with $\mathbf{P} = [p_{fn}]$
        $\mathbf{H} \leftarrow \mathbf{H} \odot \dfrac{\mathbf{W}^\top\left((\mathbf{WH})^{\odot[-2]} \odot \mathbf{P}\right)}{\mathbf{W}^\top(\mathbf{WH})^{\odot[-1]}}$ with $\mathbf{P} = [p_{fn}]$
        normalize columns of $\mathbf{W}$, scale rows of $\mathbf{H}$
    until satisfied with the factorization
until convergence criteria met
output: $\hat{\mathbf{C}} = [\hat{\mathbf{c}}_1, \dots, \hat{\mathbf{c}}_N]$, $\hat{\mathbf{W}} = \mathbf{W}, \hat{\mathbf{H}} = \mathbf{H}$
```

5.2.2 The EM-t algorithm

An alternative formulation of the problem

As an alternative to the previously described EM-*tf* algorithm, the aim of the derivation in this section is to explicitly include the missing samples (which are to be interpolated) into the problem formulation. We build the formulation upon the same statistical model, given by assumptions 5.1 (Reconstruction operator), 5.3 (Gaussian coefficients) and 5.5 (NMF structure of the coefficients). The MLE formulation thus stays exactly the same as before:

$$\hat{\mathbf{W}}, \hat{\mathbf{H}} = \operatorname*{arg\,max}_{\mathbf{W},\mathbf{H}} p(\,\mathbf{X}^{\text{obs}} \mid \mathbf{W}, \mathbf{H}\,). \tag{5.9 revisited}$$

The difference is in the setup of the EM algorithm: As in the previous section 5.2.1, the *incomplete data* correspond to the observed reliable signal $\mathbf{X}^{\text{obs}}$. The difference is that we set the *complete data* to correspond to the time domain samples, including both the observed and the missing ones, i.e., $\{\mathbf{X}^{\text{obs}}, \mathbf{X}^{\text{miss}}\}$.

Using this setting, we may rewrite the EM functional in (5.11) as

$$Q(\theta, \tilde{\theta}) = -\int \log p(\mathbf{X}^{\text{true}} \mid \theta) p(\mathbf{X}^{\text{miss}} \mid \mathbf{X}^{\text{obs}}, \tilde{\theta})\, \mathrm{d}\mathbf{X}^{\text{miss}}. \tag{5.17}$$

E-step

Since the same statistical model as in section 5.2.1 is assumed, we recall from (5.3) that $\mathbf{c}_n^{\text{true}} \sim \mathcal{N}(\mathbf{0}, \mathbf{D}_n)$ with $\mathbf{D}_n = \operatorname{diag}\left([v_{fn}]_{f=1,\dots,F}\right)$ and $\mathbf{V} = [v_{fn}] = \mathbf{WH}$. Since it holds $\mathbf{x}_n^{\text{true}} = \mathbf{T}\mathbf{c}_n^{\text{true}}$, the distribution of the time domain in frame n is $\mathbf{x}_n^{\text{true}} \sim \mathcal{N}(\mathbf{0}, \mathbf{T}\mathbf{D}_n\mathbf{T}^*)$ (see remark 5.4), i.e.,

$$p(\mathbf{X}^{\text{true}} \mid \theta) = \prod_{n=1}^{N} p(\mathbf{x}_n^{\text{true}} \mid \theta) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{0}, \mathbf{T}\mathbf{D}_n\mathbf{T}^*). \tag{5.18}$$

Now to express the term $p(\mathbf{X}^{\text{miss}} \mid \mathbf{X}^{\text{obs}}, \tilde{\theta})$, we need to find, for each n, the distribution of $\mathbf{x}_n^{\text{miss}}$ given the observed samples $\mathbf{x}_n^{\text{obs}}$ and the model parameters $\mathbf{W}$ and $\mathbf{H}$. For this purpose, we use the following reordering of the vector $\mathbf{x}_n^{\text{true}}$:

$$\begin{bmatrix} \mathbf{M}_n\mathbf{x}_n^{\text{true}} \\ \overline{\mathbf{M}}_n\mathbf{x}_n^{\text{true}} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_n^{\text{obs}} \\ \mathbf{x}_n^{\text{miss}} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top & \mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\overline{\mathbf{M}}_n^\top \\ \overline{\mathbf{M}}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top & \overline{\mathbf{M}}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\overline{\mathbf{M}}_n^\top \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{bmatrix}\right). \tag{5.19}$$

Due to the Gaussian assumptions, the posterior distribution of the complete data can be expressed according to [103, Theorem 10.2] as $(\mathbf{x}_n^{\text{miss}} \mid \mathbf{x}_n^{\text{obs}}, \mathbf{W}, \mathbf{H}) \sim \mathcal{N}(\hat{\mathbf{x}}_n^{\text{miss}}, \hat{\mathbf{S}}_n)$

with mean

$$\hat{\mathbf{x}}_n^{\text{miss}} = \mathbf{S}_{21}\mathbf{S}_{11}^{-1}\mathbf{x}_n^{\text{obs}} = \overline{\mathbf{M}}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}\mathbf{x}_n^{\text{obs}} \tag{5.20}$$

and covariance matrix

$$\begin{aligned}\hat{\mathbf{S}}_n &= \mathbf{S}_{22} - \mathbf{S}_{21}\mathbf{S}_{11}^{-1}\mathbf{S}_{12}\\ &= \overline{\mathbf{M}}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\overline{\mathbf{M}}_n^\top - \overline{\mathbf{M}}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\overline{\mathbf{M}}_n^\top.\end{aligned} \tag{5.21}$$

In terms of the EM functional (5.17), this yields

$$\begin{aligned}p(\mathbf{X}^{\text{miss}} \mid \mathbf{X}^{\text{obs}}, \tilde{\theta}) &= p(\mathbf{X}^{\text{miss}} \mid \mathbf{X}^{\text{obs}}, \mathbf{W}, \mathbf{H})\\ &= \prod_{n=1}^{N} p(\mathbf{x}_n^{\text{miss}} \mid \mathbf{x}_n^{\text{obs}}, \mathbf{W}, \mathbf{H}) = \prod_{n=1}^{N} \mathcal{N}(\hat{\mathbf{x}}_n^{\text{miss}}, \hat{\mathbf{S}}_n).\end{aligned} \tag{5.22}$$

Remark 5.8. The relation to the estimation of $\hat{\mathbf{c}}_n$ and $\hat{\mathbf{\Sigma}}_n$ defined by equation (5.14) in section 5.2.1 is the following:

$$\hat{\mathbf{x}}_n^{\text{miss}} = \overline{\mathbf{M}}_n\mathbf{T}\hat{\mathbf{c}}_n, \tag{5.23a}$$

$$\hat{\mathbf{S}}_n = \overline{\mathbf{M}}_n\mathbf{T}\hat{\mathbf{\Sigma}}_n\mathbf{T}^*\overline{\mathbf{M}}_n^\top. \tag{5.23b}$$

This precisely corresponds to $\mathbf{x}_n^{\text{miss}} = \overline{\mathbf{M}}_n\mathbf{x}_n^{\text{true}} = \overline{\mathbf{M}}_n(\mathbf{T}\mathbf{c}_n^{\text{true}}) = (\overline{\mathbf{M}}_n\mathbf{T})\mathbf{c}_n^{\text{true}}$, in line with the linear observation model in equation (5.8). In other words, given the estimates $\hat{\mathbf{c}}_n$ and $\hat{\mathbf{\Sigma}}_n$ of section 5.2.1 and the relation $\mathbf{x}_n^{\text{miss}} = (\overline{\mathbf{M}}_n\mathbf{T})\mathbf{c}_n^{\text{true}}$, we arrive at the same estimate of the missing samples $\mathbf{x}_n^{\text{miss}}$ as derived in equations (5.20) and (5.21).

M-step

The crucial part of the EM algorithm is the estimation of the parameters $\mathbf{W}, \mathbf{H}$ in the M-step, given the posterior distribution of the missing temporal samples $\mathbf{X}^{\text{miss}}$. Even though equations (5.18), (5.22) and (5.23) allow to express the functional (5.17) in the closed form, it is expensive to compute and optimize directly. Thus, we proceed to re-estimate the frequency coefficients corresponding to the signal estimated by equation (5.23) and update $\mathbf{W}$ and $\mathbf{H}$ as the factorization of this spectrum.

For this, recall the analysis operator $\mathbf{L}$ from assumption 5.1. Using this operator, we may define the posterior spectrum as $\mathbf{C}^{\text{alt}} = [\mathbf{c}_1^{\text{alt}}, \ldots, \mathbf{c}_N^{\text{alt}}]$, where $\mathbf{c}_n^{\text{alt}} = \mathbf{L}\mathbf{x}_n^{\text{true}}$. With the reordering used in equation (5.19), and in line with remark 2.2, we may rewrite this relation as

$$\mathbf{c}_n^{\text{true}} = \mathbf{L}(\mathbf{M}_n^\top \mathbf{M}_n + \mathbf{L}\overline{\mathbf{M}}_n^\top \overline{\mathbf{M}}_n)\mathbf{x}_n^{\text{true}} = \mathbf{L}\mathbf{M}_n^\top \mathbf{x}_n^{\text{obs}} + \mathbf{L}\overline{\mathbf{M}}_n^\top \mathbf{x}_n^{\text{miss}}. \tag{5.24}$$

Here, the term $\mathbf{L}\mathbf{M}_n^\top \mathbf{x}_n^{\text{obs}}$ represents the (fixed, known) contribution of the reliable samples to the frequency spectrum $\mathbf{c}_n^{\text{true}}$ of the frame n. The term $\mathbf{L}\overline{\mathbf{M}}_n^\top \mathbf{x}_n^{\text{miss}}$, on the other hand, represents the (random, unknown) contribution of the missing samples. As such, we get $\mathbf{c}_n^{\text{true}}$ as an affine transformation of a Gaussian random vector $\mathbf{x}^{\text{miss}} \sim \mathcal{N}(\hat{\mathbf{x}}_n^{\text{miss}}, \hat{\mathbf{S}}_n)$. This affine transformation is defined by the constant vector is $\mathbf{L}\mathbf{M}_n^\top \mathbf{x}_n^{\text{obs}}$ and the linear operator $\mathbf{L}\overline{\mathbf{M}}_n^\top$. Using the results (5.20) and (5.21), we may thus write that given $\mathbf{x}_n^{\text{obs}}, \mathbf{W}, \mathbf{H}$, it holds $\mathbf{c}_n^{\text{true}} \sim \mathcal{N}\left(\hat{\mathbf{c}}_n^{\text{alt}}, \hat{\mathbf{\Sigma}}_n^{\text{alt}}\right)$ with

$$\hat{\mathbf{c}}_n^{\text{alt}} = \mathbf{L}\mathbf{M}_n^\top \mathbf{x}_n^{\text{obs}} + \mathbf{L}\overline{\mathbf{M}}_n^\top \hat{\mathbf{x}}_n^{\text{miss}}, \quad \hat{\mathbf{\Sigma}}_n^{\text{alt}} = \mathbf{L}\overline{\mathbf{M}}_n^\top \hat{\mathbf{S}}_n \overline{\mathbf{M}}_n \mathbf{L}^*. \tag{5.25}$$

Using the results (5.14) for the EM-*tf* algorithm, the above expressions could be simplified to

$$\hat{\mathbf{c}}_n^{\text{alt}} = \mathbf{L}\mathbf{T}\hat{\mathbf{c}}_n, \quad \hat{\mathbf{\Sigma}}_n^{\text{alt}} = \mathbf{L}\mathbf{T}\hat{\mathbf{\Sigma}}_n \mathbf{T}^* \mathbf{L}^*. \tag{5.26}$$

In line with equation (5.16) in the EM-*tf* algorithm, the posterior power spectrogram is then defined entrywise as

$$p_{fn} = \mathbb{E}\left(\left|s_{fn}\right|^2 \mid \mathbf{x}_n^{\text{obs}}, \mathbf{W}, \mathbf{H}\right) = \left|\hat{\mathbf{c}}_n^{\text{alt}}(f)\right|^2 + \hat{\mathbf{\Sigma}}_n^{\text{alt}}(f, f), \tag{5.27}$$

and factorized using the multiplicative updates of algorithm 1.4. The whole procedure is summarized in algorithm 5.2.

Algorithm 5.2: Audio interpolation via EM-t.

```
input: reliable samples $\mathbf{X}^{\text{obs}} = \{\mathbf{x}_n^{\text{obs}}\}_{n=1,\dots,N}$, respective selection matrices
       $\{\mathbf{M}_n\}_{n=1,\dots,N}$, linear transforms $\mathbf{T} \in \mathbb{C}^{W \times F}$, $\mathbf{L} \in \mathbb{C}^{F \times W}$
initialize $\mathbf{W} \in \mathbb{R}^{F \times K}$, $\mathbf{H} \in \mathbb{R}^{K \times N}$ non-negative
repeat
    // E-step:
    for $n = 1, \dots, N$ do
        $\mathbf{D}_n \leftarrow \operatorname{diag}\left([v_{fn}]_{f=1,\dots,F}\right)$ with $[v_{fn}]_{f=1,\dots,F}$ being the $n$-th column of
         the matrix $\mathbf{V} = \mathbf{WH}$
        $\hat{\mathbf{c}}_n^{\text{alt}} \leftarrow \mathbf{LTD}_n\mathbf{T}^*\mathbf{M}_n^\top \left(\mathbf{M}_n\mathbf{TD}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1} \mathbf{x}_n^{\text{obs}}$
        $\hat{\mathbf{\Sigma}}_n^{\text{alt}} \leftarrow \mathbf{LT}\left(\mathbf{D}_n - \mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\left(\mathbf{M}_n\mathbf{TD}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}\mathbf{M}_n\mathbf{TD}_n\right)\mathbf{T}^*\mathbf{L}^*$
        $p_{fn} \leftarrow \left|\hat{\mathbf{c}}_n^{\text{alt}}(f)\right|^2 + \hat{\mathbf{\Sigma}}_n^{\text{alt}}(f,f),\ f = 1, \dots, F$
    end
    // M-step:
    repeat
        $\mathbf{W} \leftarrow \mathbf{W} \odot \dfrac{\left((\mathbf{WH})^{\odot[-2]} \odot \mathbf{P}\right)\mathbf{H}^\top}{(\mathbf{WH})^{\odot[-1]}\mathbf{H}^\top}$ with $\mathbf{P} = [p_{fn}]$
        $\mathbf{H} \leftarrow \mathbf{H} \odot \dfrac{\mathbf{W}^\top\left((\mathbf{WH})^{\odot[-2]} \odot \mathbf{P}\right)}{\mathbf{W}^\top(\mathbf{WH})^{\odot[-1]}}$ with $\mathbf{P} = [p_{fn}]$
        normalize columns of $\mathbf{W}$, scale rows of $\mathbf{H}$
    until satisfied with the factorization
until convergence criteria met
output: $\hat{\mathbf{C}}^{\text{alt}} = \left[\hat{\mathbf{c}}_1^{\text{alt}}, \dots, \hat{\mathbf{c}}_N^{\text{alt}}\right], \hat{\mathbf{W}} = \mathbf{W}, \hat{\mathbf{H}} = \mathbf{H}$
```

5.2.3 Conditions for the equivalence of EM-*tf* and EM-*t*

Assumption 5.1 postulated the operators $\mathbf{T}$ and $\mathbf{L}$ only as being linear with no other relation. However, there are several natural setups, some of which make the approaches of EM-*tf* and EM-*t* equivalent. The following list briefly discusses several possible choices of the operators $\mathbf{T}$ and $\mathbf{L}$:

1. $\mathbf{T}$ is unitary, $\mathbf{L} = \mathbf{T}^* = \mathbf{T}^{-1}$:

   It holds $\mathbf{LT} = \mathbf{I}$, thus the relations (5.26) yield $\hat{\mathbf{c}}_n = \hat{\mathbf{c}}_n^{\text{alt}}$ and $\hat{\mathbf{\Sigma}}_n = \hat{\mathbf{\Sigma}}_n^{\text{alt}}$, which means that EM-*tf* and EM-*t* coincide. As an example, this is the case when $\mathbf{L}$ is a unitary DFT represented by the matrix $\mathbf{F} \in \mathbb{C}^{F \times W}$ with $F = W$ (i.e., the number of frequency bins equals the number of time-domain samples in a signal window, see also (1.6)). Then, $\mathbf{T} = \mathbf{L}^{-1}$ is the inverse DFT.

2. $\mathbf{T}$ is invertible, $\mathbf{L} = \mathbf{T}^{-1}$:

   Also in this case, we have from (5.26) that $\hat{\mathbf{c}}_n = \hat{\mathbf{c}}_n^{\text{alt}}$ and $\hat{\mathbf{\Sigma}}_n = \hat{\mathbf{\Sigma}}_n^{\text{alt}}$.

3. $\mathbf{T}$ is a Parseval tight frame synthesis operator, $F > W$, $\mathbf{L} = \mathbf{T}^*$, $\mathbf{TL} = \mathbf{I}$:

   The two models are no longer equivalent, since $\mathbf{LT}$ represents the projection operator on the range space of $\mathbf{L}$. This can be seen directly from equation (1.16) and the assumption $\mathbf{TL} = \mathbf{L}^*\mathbf{L} = \mathbf{I}$. In the redundant case $F > W$, this projection differs from identity. As an example, this is the case of redundant DFT, such as with $F = 2W$, i.e., twice as much frequency bins as is the number of time-domain samples in a signal window.

4. $\mathbf{T}$ is a Parseval tight frame analysis operator, $F < W$, $\mathbf{L} = \mathbf{T}^*$, i.e., $\mathbf{LT} = \mathbf{I}$:

   As in the first case, the relation $\mathbf{LT} = \mathbf{I}$ results in the equivalence of EM-*tf* and EM-*t*. However, the assumption $F < W$ means that we do not have enough frequency coefficients to reconstruct *an arbitrary* signal in the framed time domain. To be specific, the time-domain solution (in each frame) is restricted to the range space of $\mathbf{T}$.

5. $\mathbf{T}$ is arbitrary, $\mathbf{L} = \mathbf{T}^+$:

   In this case, the matrix $\mathbf{LT} = \mathbf{T}^+\mathbf{T}$ represents the orthogonal projection onto the range space of $\mathbf{T}^*$. This is in general different from the identity, unless the range space of $\mathbf{T}^*$ is the whole coefficient space $\mathbb{C}^F$.

6. $\mathbf{L}$ is arbitrary, $\mathbf{T} = \mathbf{L}^+$:

   Similarly to the previous option, $\mathbf{LT} = \mathbf{LL}^+$ represents the orthogonal projection onto the range space of $\mathbf{L}$. This is in general different from the identity, unless the range space of $\mathbf{L}$ is the whole coefficient space $\mathbb{C}^F$.

5.3 Maximum likelihood estimation by treating the missing samples as parameters: the AM algorithm

Coming back to the MLE formulation (5.9), we may notice that the problem itself was posed without explicitly considering the samples to be interpolated. However, an alternative problem can be formulated including the missing samples as parameters:

$$\hat{\mathbf{W}}, \hat{\mathbf{H}}, \hat{\mathbf{X}}^{\text{miss}} = \underset{\mathbf{W}, \mathbf{H}, \mathbf{X}^{\text{miss}}}{\arg\max}\, p\Big(\{\mathbf{X}^{\text{obs}}, \underbrace{\mathbf{X}^{\text{miss}}}_{\text{parameters}}\} \mid \mathbf{W}, \mathbf{H}\Big), \tag{5.28}$$

which is equivalent to

$$\hat{\mathbf{W}}, \hat{\mathbf{H}}, \hat{\mathbf{X}}^{\text{miss}} = \underset{\mathbf{W}, \mathbf{H}, \mathbf{X}^{\text{miss}}}{\arg\min}\, -\log p\Big(\{\mathbf{X}^{\text{obs}}, \mathbf{X}^{\text{miss}}\} \mid \mathbf{W}, \mathbf{H}\Big), \tag{5.29}$$

under the assumptions 5.1 (Reconstruction operator), 5.3 (Gaussian coefficients) and 5.5 (NMF structure of the coefficients). Since $\mathbf{X}^{\text{obs}}$ is fixed in the problem, we may write the distribution of $p(\{\mathbf{X}^{\text{obs}}, \mathbf{X}^{\text{miss}}\} \mid \mathbf{W}, \mathbf{H})$ as $p(\{\mathbf{X}^{\text{obs}}, \mathbf{X}^{\text{miss}}\} \mid \mathbf{W}, \mathbf{H}, \mathbf{X}^{\text{obs}})$.

Now recall that it stems directly from the assumptions that the distribution of the reordered windows of the signal is

$$\begin{bmatrix} \mathbf{x}_n^{\text{obs}} \\ \mathbf{x}_n^{\text{miss}} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{M}_n \\ \overline{\mathbf{M}}_n \end{bmatrix} \mathbf{T}\mathbf{D}_n\mathbf{T}^* \begin{bmatrix} \mathbf{M}_n \\ \overline{\mathbf{M}}_n \end{bmatrix}^\top\right). \tag{5.19 with simplified notation}$$

To write down the probability density function, recall that the concatenation of the selection matrices is unitary (see remark 2.2), thus it can be omitted in the determinant appearing in equation (5.5). Furthermore, it simplifies the inverse of the covariance matrix, because

$$\left(\begin{bmatrix} \mathbf{M}_n \\ \overline{\mathbf{M}}_n \end{bmatrix} \mathbf{T}\mathbf{D}_n\mathbf{T}^* \begin{bmatrix} \mathbf{M}_n \\ \overline{\mathbf{M}}_n \end{bmatrix}^\top\right)^{-1} = \begin{bmatrix} \mathbf{M}_n \\ \overline{\mathbf{M}}_n \end{bmatrix} (\mathbf{T}\mathbf{D}_n\mathbf{T}^*)^{-1} \begin{bmatrix} \mathbf{M}_n \\ \overline{\mathbf{M}}_n \end{bmatrix}^\top. \tag{5.30}$$

Thus, using formula (5.5) for the Gaussian distribution, it holds

$$\begin{aligned} &p\left(\begin{bmatrix} \mathbf{x}_n^{\text{obs}} \\ \mathbf{x}_n^{\text{miss}} \end{bmatrix} \mid \mathbf{W}, \mathbf{H}\right) \\ &= \det\left(\pi\mathbf{T}\mathbf{D}_n\mathbf{T}^*\right)^{-1} \exp\left(-\begin{bmatrix} \mathbf{x}_n^{\text{obs}} \\ \mathbf{x}_n^{\text{miss}} \end{bmatrix}^\top \begin{bmatrix} \mathbf{M}_n \\ \overline{\mathbf{M}}_n \end{bmatrix} (\mathbf{T}\mathbf{D}_n\mathbf{T}^*)^{-1} \begin{bmatrix} \mathbf{M}_n \\ \overline{\mathbf{M}}_n \end{bmatrix}^\top \begin{bmatrix} \mathbf{x}_n^{\text{obs}} \\ \mathbf{x}_n^{\text{miss}} \end{bmatrix}\right). \end{aligned} \tag{5.31}$$

We propose to minimize $-\log p$ from formulation (5.29) via alternating optimization. Due to the assumed independence of the time frames, we can perform the optimization frame-wise by repeating two steps:

1. minimization with respect to the variables $\mathbf{x}_n^{\mathrm{miss}}$,
2. minimization with respect to the model parameters $\mathbf{W}, \mathbf{H}$.

In the following sections, both problems are treated in detail, and connection with the EM algorithm is briefly discussed.

Updating the signal

Minimizing $-\log p$, or maximizing p, with respect to $\mathbf{x}_n^{\mathrm{miss}}$ is equivalent to finding the mode of the conditional distribution of $\mathbf{x}_n^{\mathrm{miss}}$ given $\mathbf{x}_n^{\mathrm{obs}}$, $\mathbf{W}$ and $\mathbf{H}$. Due to the Gaussian assumptions, the mode is equal to the expected value

$$\hat{\mathbf{x}}_n^{\mathrm{miss}} = \mathbb{E}\left(\mathbf{x}_n^{\mathrm{miss}} \mid \mathbf{x}_n^{\mathrm{obs}}, \mathbf{W}, \mathbf{H}\right) = \overline{\mathbf{M}}_n \mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top \left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1} \mathbf{x}_n^{\mathrm{obs}}. \tag{5.20 revisited}$$

The whole signal frame, including the estimated missing samples $\hat{\mathbf{x}}_n^{\mathrm{miss}}$ and the fixed observed samples $\mathbf{x}_n^{\mathrm{obs}}$, can be composed together as

$$\hat{\mathbf{x}}_n = \mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top \left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1} \mathbf{x}_n^{\mathrm{obs}}. \tag{5.32}$$

Updating the model

We want to derive the updates directly from the optimization problem (5.29). Since the missing samples are fixed in this step, we do not need to reorder the samples of the signal frame $\mathbf{x}_n$, which leads to

$$p\left(\mathbf{x}_n \mid \mathbf{W}, \mathbf{H}\right) = \det\left(\pi\mathbf{T}\mathbf{D}_n\mathbf{T}^*\right)^{-1} \exp\left(-(\mathbf{x}_n)^\top \left(\mathbf{T}\mathbf{D}_n\mathbf{T}^*\right)^{-1} \mathbf{x}_n\right), \tag{5.33}$$

and therefore

$$-\log p\left(\mathbf{x}_n \mid \mathbf{W}, \mathbf{H}\right) = \log\det\left(\pi\mathbf{T}\mathbf{D}_n\mathbf{T}^*\right) + (\mathbf{x}_n)^\top \left(\mathbf{T}\mathbf{D}_n\mathbf{T}^*\right)^{-1} \mathbf{x}_n. \tag{5.34}$$

To simplify the derivation of the method, let us pose the following assumption.

Assumption 5.9 (Invertibility of the synthesis)**.** The operator $\mathbf{T}$ is invertible and $\mathbf{L} = \mathbf{T}^{-1}$. In particular, this means that $\mathbf{T}$ is square, i.e., $F = W$.

Under assumption 5.9, we see that $\det\left(\pi\mathbf{T}\mathbf{D}_n\mathbf{T}^*\right) = \pi^W \det\left(\mathbf{T}\right)^2 \det\left(\mathbf{D}_n\right)$ and $(\mathbf{T}\mathbf{D}_n\mathbf{T}^*)^{-1} = (\mathbf{T}^{-1})^*\mathbf{D}_n^{-1}\mathbf{T}^{-1}$. Since $\mathbf{T}$ does not depend on the parameters $\mathbf{W}$ and

$\mathbf{H}$ which are being optimized in this step, the problem boils down to

$$\underset{\mathbf{W},\mathbf{H}}{\arg\min}\ \log\det\left(\mathbf{D}_n\right)+\left(\mathbf{T}^{-1}\mathbf{x}_n\right)^*\mathbf{D}_n^{-1}\left(\mathbf{T}^{-1}\mathbf{x}_n\right). \tag{5.35}$$

To break down the objective (5.35), recall that we have defined in assumption 5.3 that $\mathbf{D}_n = \operatorname{diag}\left([v_{fn}]_{f=1,\dots,F}\right)$ (which is straightforward to invert) and denote $\mathbf{T}^{-1}\mathbf{x}_n = \mathbf{c}_n$. Then, the objective can be rewritten as

$$\log\prod_{f=1}^{F} v_{fn}+\sum_{f=1}^{F}\mathbf{c}_n^*(f)\frac{1}{v_{fn}}\mathbf{c}_n(f)=\sum_{f=1}^{F}\log v_{fn}+\sum_{f=1}^{F}\frac{\left|\mathbf{c}_n(f)\right|^2}{v_{fn}}. \tag{5.36}$$

Now recall from (1.34) that the Itakura–Saito divergence is defined as

$$d_{\mathrm{IS}}(x\mid y)=\frac{x}{y}-\log\frac{x}{y}-1=\frac{x}{y}-\log x+\log y-1. \tag{5.37}$$

Optimizing $d_{\mathrm{IS}}(x\mid y)$ with respect to y allows to omit terms independent on y, thus

$$\underset{y}{\arg\min}\, d_{\mathrm{IS}}(x\mid y) \quad = \quad \underset{y}{\arg\min}\,\frac{x}{y}+\log y. \tag{5.38}$$

This clearly corresponds to the expression in (5.36), therefore the minimization (5.35) is equivalent to

$$\underset{v_{fn}}{\arg\min}\sum_{f=1}^{F} d_{\mathrm{IS}}\left(\left|\mathbf{c}_n(f)\right|^2\mid v_{fn}\right),\quad \mathbf{c}_n=\mathbf{T}^{-1}\mathbf{x}_n,\quad v_{fn}=\sum_k w_{fk}h_{kn}. \tag{5.39}$$

Taking into account all the frames finally leads to the desired result that $\mathbf{W}$ and $\mathbf{H}$ are obtained by minimizing $D_{\mathrm{IS}}(\mathbf{P}\mid\mathbf{WH})$ where $p_{fn}=\left|(\mathbf{T}^{-1}\hat{\mathbf{x}}_n)(f)\right|^2$ and $\hat{\mathbf{x}}_n$ is the signal estimate from the previous step. The whole procedure is summarized in algorithm 5.3.

Remark 5.10. A simple heuristic possibility is to compute the spectrum of $\hat{\mathbf{x}}_n$ as

$$\hat{\mathbf{c}}_n=\mathbf{L}\hat{\mathbf{x}}_n=\mathbf{LTD}_n\mathbf{T}^*\mathbf{M}_n^\top\left(\mathbf{M}_n\mathbf{TD}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}\mathbf{x}_n^{\mathrm{obs}} \tag{5.40}$$

and the power spectrogram $p_{fn}=\left|\hat{\mathbf{c}}_n(f)\right|^2$, which would be applicable especially for $\mathbf{T}$ which is not invertible. Then, the multiplicative updates can be applied to minimize $D_{\mathrm{IS}}(\mathbf{P}\mid\mathbf{WH})$. However, this approach is not justified by the optimization (5.33) with respect to $\mathbf{W},\mathbf{H}$. The problem is that if we cannot compute the inverse of $(\mathbf{TD}_n\mathbf{T}^*)^{-1}$ as $\mathbf{AD}_n^{-1}\mathbf{B}$ for some matrices $\mathbf{A},\mathbf{B}$, we cannot separate the individual diagonal entries of $\mathbf{D}_n$ to fit it to the Itakura–Saito NMF problem.

Remark 5.11. Note that under assumption 5.9 (Invertibility of the synthesis), EM-*tf* is equivalent to EM-*t* (this is the first case discussed in section 5.2.3). However, the alternating minimization produces a slightly different algorithm, because the covariance matrix is not considered in computing the power spectrogram – compare line 7 of algorithm 5.3 with line 7 of algorithm 5.1.

Also note that the signal update in the AM algorithm closely resembles the E-step in EM-*tf* and the model update is equivalent to the M-step of EM-*tf*. However, the updates in AM are named differently on purpose to emphasize that AM is not derived from the EM algorithm.

Algorithm 5.3: Audio interpolation via AM.

```
input: reliable samples $\mathbf{X}^{\mathrm{obs}} = \{\mathbf{x}_n^{\mathrm{obs}}\}_{n=1,\dots,N}$, respective selection matrices
       $\{\mathbf{M}_n\}_{n=1,\dots,N}$, invertible linear transform $\mathbf{T} \in \mathbb{C}^{W \times F}$
initialize $\mathbf{W} \in \mathbb{R}^{F \times K}$, $\mathbf{H} \in \mathbb{R}^{K \times N}$ non-negative
repeat
    // Signal update:
    for $n = 1, \dots, N$ do
        $\mathbf{D}_n \leftarrow \operatorname{diag}\left([v_{fn}]_{f=1,\dots,F}\right)$ with $[v_{fn}]_{f=1,\dots,F}$ being the $n$-th column of
         the matrix $\mathbf{V} = \mathbf{WH}$
        $\hat{\mathbf{c}}_n \leftarrow \mathbf{D}_n \mathbf{T}^* \mathbf{M}_n^\top \left(\mathbf{M}_n \mathbf{T} \mathbf{D}_n \mathbf{T}^* \mathbf{M}_n^\top\right)^{-1} \mathbf{x}_n^{\mathrm{obs}}$
        $\hat{\mathbf{x}}_n \leftarrow \mathbf{T}^{-1} \hat{\mathbf{c}}_n$
        $p_{fn} \leftarrow |\hat{\mathbf{c}}_n(f)|^2, \ f = 1, \dots, F$
    end
    // Model update:
    repeat
        $\mathbf{W} \leftarrow \mathbf{W} \odot \dfrac{\left((\mathbf{WH})^{\odot[-2]} \odot \mathbf{P}\right)\mathbf{H}^\top}{(\mathbf{WH})^{\odot[-1]}\mathbf{H}^\top}$ with $\mathbf{P} = [p_{fn}]$
        $\mathbf{H} \leftarrow \mathbf{H} \odot \dfrac{\mathbf{W}^\top\left((\mathbf{WH})^{\odot[-2]} \odot \mathbf{P}\right)}{\mathbf{W}^\top(\mathbf{WH})^{\odot[-1]}}$ with $\mathbf{P} = [p_{fn}]$
        normalize columns of $\mathbf{W}$, scale rows of $\mathbf{H}$
    until satisfied with the factorization
until convergence criteria met
output: $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_N]$, $\hat{\mathbf{W}} = \mathbf{W}$, $\hat{\mathbf{H}} = \mathbf{H}$
```

6 Experiments

This chapter is dedicated to experimental evaluation of the methods proposed in chapters 4 and 5. Possible ways to numerically assess the quality of the interpolated signals are discussed in section 6.1. Section 6.2 then introduces heuristic indicators related to the convergence of the algorithms. While global convergence is theoretically provable for the methods based on convex relaxation, heuristic criteria need to be employed in the other cases. This also concerns SPAIN and the NMF-based methods, which is why later in this chapter, the heuristic convergence indicators are employed to illustrate progression towards a stable solution. The experiment design is outlined in section 6.3, followed by evaluation of the individual methods (section 6.4 and 6.5). The final section 6.6 presents comparison of selected methods with the state of the art using objective criteria.

6.1 Metrics of reconstruction quality

The signal-to-distortion ratio (SDR) for the reference signal $\mathbf{x}^{\mathrm{true}}$ and the interpolated signal $\hat{\mathbf{x}}$ is defined (in decibels) as

$$\mathrm{SDR}(\hat{\mathbf{x}}, \mathbf{x}^{\mathrm{true}}) = 10 \log_{10} \frac{\|\mathbf{x}^{\mathrm{true}}\|^2}{\|\hat{\mathbf{x}} - \mathbf{x}^{\mathrm{true}}\|^2}. \tag{6.1}$$

In words, it corresponds to the signal-to-noise ratio (SNR) value where $\mathbf{x}^{\mathrm{true}}$ serves as the (clean) signal and $\hat{\mathbf{x}} - \mathbf{x}^{\mathrm{true}}$, i.e., the entrywise difference of the true signal and its estimate, represents the noise. However, SDR is the preferred notation, since it highlights that the difference between the two signals is perceived as distortion.

Note that in light of remark 2.3, all the methods developed throughout this thesis can be labeled as *consistent*, since none of the algorithms alters the reliable/known samples. Using the masking operators $\mathbf{M}$ and $\overline{\mathbf{M}}$ (see also remark 2.1), the formula (6.1) for SDR is equivalent to

$$\mathrm{SDR}(\hat{\mathbf{x}}, \mathbf{x}^{\mathrm{true}}) = 10 \log_{10} \frac{\|\mathbf{M}\mathbf{x}^{\mathrm{true}}\|^2 + \|\overline{\mathbf{M}}\mathbf{x}^{\mathrm{true}}\|^2}{\underbrace{\|\mathbf{M}(\hat{\mathbf{x}} - \mathbf{x}^{\mathrm{true}})\|^2}_{=0} + \|\overline{\mathbf{M}}(\hat{\mathbf{x}} - \mathbf{x}^{\mathrm{true}})\|^2}, \tag{6.2}$$

where $\|\mathbf{M}(\hat{\mathbf{x}} - \mathbf{x}^{\mathrm{true}})\|^2 = 0$ due to consistency of the solution $\hat{\mathbf{x}}$. This means that the SDR value does not only consider the quality of the interpolation in the missing parts of the signal, but depends also on the energy of the reliable signal $\|\mathbf{M}\mathbf{x}^{\mathrm{true}}\|^2$. However, this term does not reflect in any way the quality of the interpolation. For this reason, the SDR is usually re-defined, such that it only considers the interpolated samples instead of the whole signal, i.e., as

$$\mathrm{SDR}_{\mathrm{miss}}(\hat{\mathbf{x}}, \mathbf{x}^{\mathrm{true}}) = \mathrm{SDR}(\overline{\mathbf{M}}\hat{\mathbf{x}}, \overline{\mathbf{M}}\mathbf{x}^{\mathrm{true}}) = 10 \log_{10} \frac{\|\overline{\mathbf{M}}\mathbf{x}^{\mathrm{true}}\|^2}{\|\overline{\mathbf{M}}(\hat{\mathbf{x}} - \mathbf{x}^{\mathrm{true}})\|^2}. \tag{6.3}$$

Remark 6.1 (Relation to MSE)**.** A common entrywise error metric is the MSE, which can be expressed using the norm as $\mathrm{MSE}(\hat{\mathbf{x}}, \mathbf{x}^{\mathrm{true}}) = \frac{1}{L}\|\hat{\mathbf{x}} - \mathbf{x}^{\mathrm{true}}\|^2$, where L is the signal length. We can derive that

$$\begin{aligned}\mathrm{SDR}(\hat{\mathbf{x}}, \mathbf{x}^{\mathrm{true}}) &= 10 \log_{10} \frac{\|\mathbf{x}^{\mathrm{true}}\|^2}{L \cdot \mathrm{MSE}(\hat{\mathbf{x}}, \mathbf{x}^{\mathrm{true}})} \\ &= 10 \log_{10} \|\mathbf{x}^{\mathrm{true}}\|^2 - 10 \log_{10} L - 10 \log_{10} \mathrm{MSE}(\hat{\mathbf{x}}, \mathbf{x}^{\mathrm{true}}). \end{aligned} \tag{6.4}$$

In words, SDR is inversely proportional to the logarithm of MSE, biased by the logarithm of the energy of the reference signal and of the signal length. Even though minimizing MSE and maximizing SDR with respect to $\hat{\mathbf{x}}$ are equivalent problems, SDR is the preferred indicator, since it relates the error to the energy of the reference signal.

Even though sample-wise metrics such as SDR and MSE objectively measure the interpolation quality, they do not necessarily reflect the *perceived* quality of the interpolated signal. In practice, the listener's perception might be the actual objective. However, as mentioned previously in chapter 3, the only way to conduct such an evaluation is through standardized listening tests with a proper number of listeners and in controlled environment. While such test can be designed for a comparison of a few test signals, it is hardly a feasible method for evaluation of large volumes of data.

For this reason, objective methods predicting the subjective quality of sound are a common compromise. Unfortunately, no approach customized to evaluation of interpolated audio signals has been established so far. A common, universal choice is the Perceptual Evaluation of Audio Quality (PEAQ) [91], which predicts the human rating of the difference between the degraded (or interpolated) and the reference signals, and outputs a single value denoted as objective difference grade (ODG). These values correspond to particular descriptions of the audible impairment, see table 6.1. We use the Matlab code[1] implemented according to the version of PEAQ based on the ITU-R recommendation BS.1387-1 [92].

An alternative to PEAQ is the PEMO-Q [90], also implemented in Matlab[2].

[1] http://www.mmsp.ece.mcgill.ca/Documents/Software/

[2] The software has been previously available for non-commercial use from https://www.hoertech.de/de/produkte/pemo-q.html (see e.g. the archived version of the web at http://web.archive.org/web/20200925200149/https://www.hoertech.de/de/produkte/pemo-q.html).

PEMO-Q computes a scalar called the perceptual similarity measure (PSM), which can be mapped onto the ODG score as used in the PEAQ.

Tab. 6.1: Interpretation of the ODG

ODG	Impairment description
0	Imperceptible
−1	Perceptible, but not annoying
−2	Slightly annoying
−3	Annoying
−4	Very annoying

6.2 Convergence indicators

In the case of iterative algorithms, it is crucial to assess convergence of the iterates. In the case of algorithms for convex optimization, theoretical convergence guarantees are available, see the individual examples in section 1.2.2. For the heuristic algorithms (this is the case of SPAIN or NMF-based methods), we must rely on empirical indicators. These may include tracking the objective function of the respective optimization problem or its relative change: Denote $f(u)$ the objective function, which depends on the variable(s) denoted compactly as u, and $\{u^{(0)}, u^{(1)}, \dots\}$ the iterates generated by the chosen algorithm. It is natural to define the relative objective change (ROC) in n-th iteration as the quantity

$$\mathrm{ROC}(n) = \frac{\left| f(u^{(n)}) - f(u^{(n-1)}) \right|}{|f(u^{(n-1)})|}. \tag{6.5}$$

Similarly, for vector iterates $\{\mathbf{u}^{(0)}, \mathbf{u}^{(1)}, \dots\}$, the relative solution change (RSC) is instinctively defined as

$$\mathrm{RSC}(n) = \frac{\|\mathbf{u}^{(n)} - \mathbf{u}^{(n-1)}\|}{\|\mathbf{u}^{(n-1)}\|}. \tag{6.6}$$

From the practical point of view, RSC is the preferred indicator, since progression of RSC towards zero suggests that further iterations are not likely change significantly the output of the algorithm, hence the reconstruction quality. This is expected to be in line with the evolution of SDR in iterations, computed according to either formula (6.1) or (6.3) with $\hat{\mathbf{x}}$ being the current iterate (or its time-domain counterpart, e.g., if the iterates belong to the time-frequency space).

6.3 Dataset and experiment protocol

The overall focus of the thesis is on the interpolation of real-world musical signals. However, to allow the quality assessment described before, the impairment is simulated using initially clean signals, such that the ground truth (reference) is always available.

In the demonstrative examples, we use individual recordings with a sampling rate of 16 or 44.1 kHz. For larger comparative experiments, we rely on the set of 10 musical recordings from the EBU SQAM dataset [104, 105], sampled at 44.1 kHz and shortened to 7 seconds, as used in recent related publications [4, 72, 98].

Regarding the setting of the extrapolation problem, two scenarios are usually considered in the literature. In the first scenario, samples at random positions are discarded. While real-world degradation is not likely to appear in this way, the scenario allows to interpolate significant portions of the input signal (at reasonable quality), such as 80 % of lost samples considered in [62, 1].

The second, and herein preferred scenario consists in the interpolation of compact gaps. In particular, the methods considered in this thesis aim at filling middle-length gaps, i.e., missing segments of length in the range of tens of milliseconds. Drop-outs of a given length are simulated at pseudo-random positions across the clean signals in a particular way which has several convenient effects:

1. A single test signal always includes gaps of equal length, which allows to study performance of the algorithms depending on the gap length.
2. Adjacent gaps are guaranteed to be separated by long enough reliable context, such that context-based methods (e.g. the extrapolation technique described in section 2.2) can treat every gap independently.
3. Existence of multiple gaps in a single audio signals increases the reliability of the objective, perception-based metrics.

The gap distribution is illustrated by figure 6.1, showing a selection of degraded signals as used in [8] and later in sections 6.5 and 6.6. The protocol used in [4], which projects also to section 6.4.1, was slightly different; in particular, the gap positions were allowed to differ depending on the source signal. However, and most importantly, both strategies preserve fairness of the experiment in the sense that all methods are being compared on the same gaps.

For any details not mentioned in the text, as well as the raw data, see the links to the source codes in appendix B.

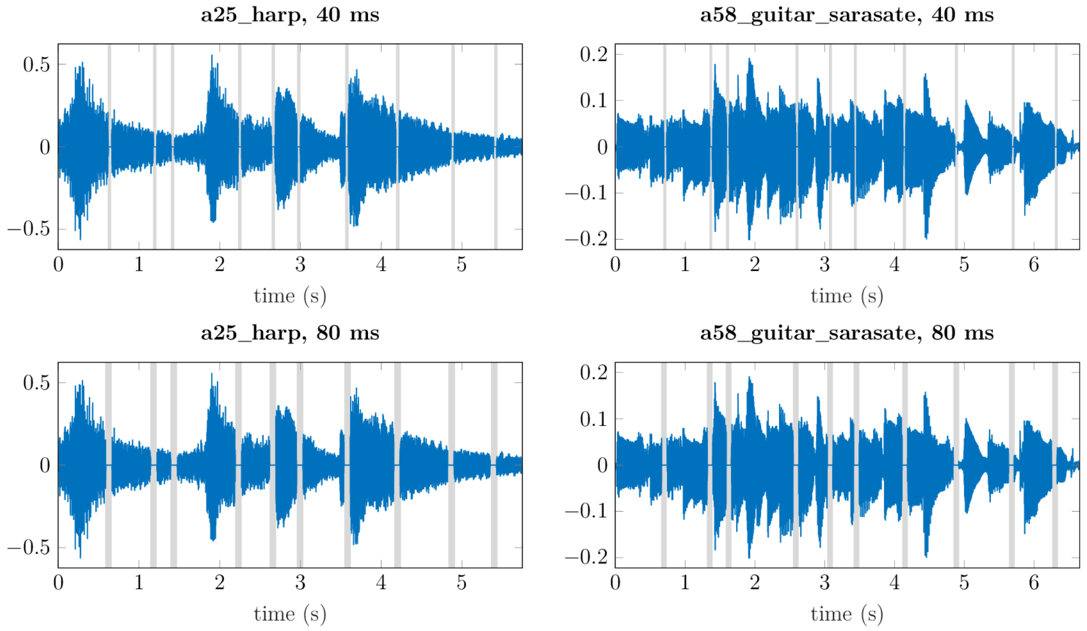


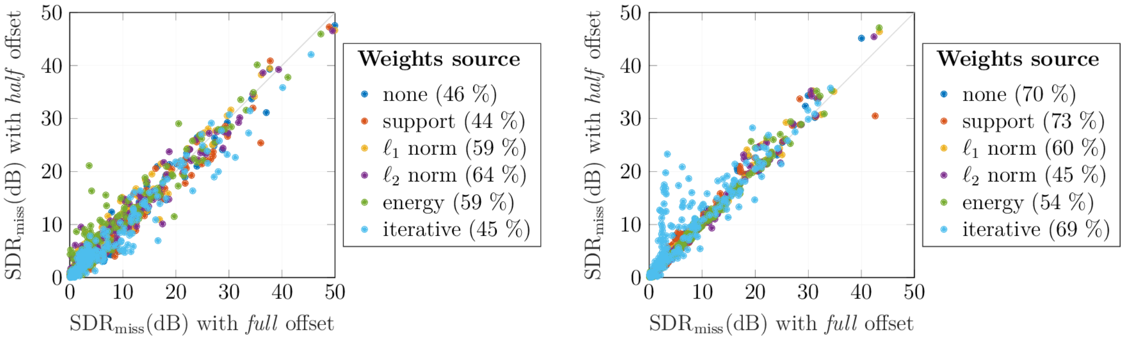
Fig. 6.1: Example of the gap distribution for two signals and two gap lengths. The gaps are indicated by the gray areas, as well as zero values of the signal. Observe that due to the pseudo-random scheme of the gap generation, the individual dropouts appear at virtually the same positions independent on gap length, as well as in same places in the signal, relative to its length.

6.4 Evaluation of sparsity-based methods

The experiments in this section aim at demonstrating the effectiveness of the methods described in chapter 4. Since a rather large number of particular algorithms and variants have been proposed, the evaluation is divided into smaller experiments. The algorithms based on convex relaxation (section 4.1) and on heuristic use of ADMM (section 4.2) are discussed in subsections 6.4.1 and 6.4.2, respectively.

6.4.1 Convex relaxation

The first experiment concerns the offset choice. Since symmetry of the energy profile inside the interpolated gap is a prerequisite of some of the methods, the focus is on choosing among the *full* and *half* variants (see section 4.1.3). Figure 6.2 presents a comparison using numerous choices of the signal, gap length and position and atom weighting. While the choice of the offset variant appears to have only a minor effect in case of the synthesis model, the *half* variant is favored in case of the analysis model. The most remarkable difference is in the analysis model using iterative weights, where *half* offset largely improves the results with low SDR, compared to the *full* variant (see figure 6.2b). For this reason, the subsequent experiments are performed using the *half* offset only.



(a) synthesis model (53 %)

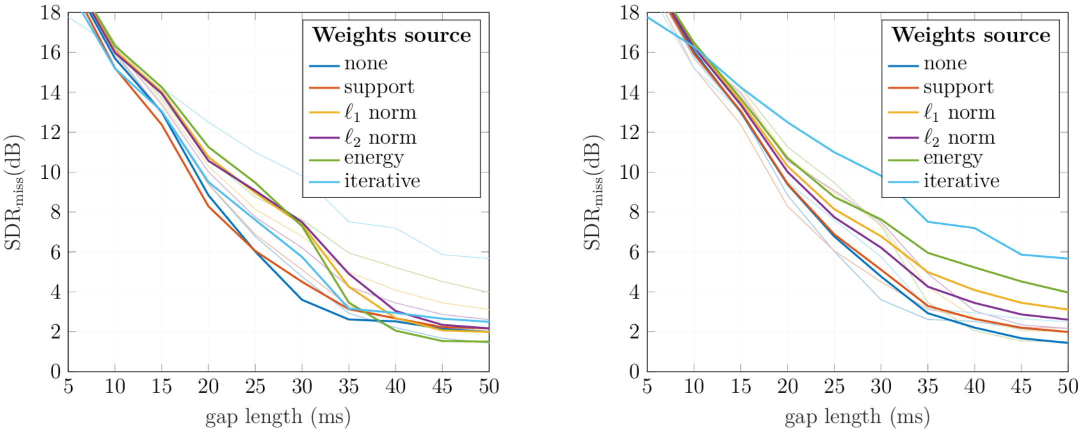
(b) analysis model (61 %)

Fig. 6.2: Experiment on choosing the offset. Each point of the scatter plot represents one of 200 interpolated gaps (10 signals, 5 gap lengths from 10 to 50 ms, 4 gaps per signal). Per each weighting type and each model, the percentage represents the fraction of results above the diagonal line, i.e., in favor of the *half* offset.

Figure 6.2 includes an overview of possible choices of the atom weights – either the explicit variants from equation (4.11), or the iterative choices defined via algorithms 4.3 and 4.4. However, figure 6.3 presents a clearer and broader comparison depending on the model (synthesis or analysis) and gap length. The first observation is that the proposed atom weighting globally leads to higher quality (in terms of SDR) compared to the non-weighted reference. The best choice of weights depends on the signal model – while iterative weights surpass all other options in the analysis model (except for the shortest gaps), the best results in the synthesis case are obtained with weights computed using ℓ_2 norm or energy (squared ℓ_2 norm) of the atoms. Finally, this experiment reveals that in most cases, the analysis model is a better choice than the synthesis one, especially with proper atom weighting.

To evaluate the approaches of iterative gap shortening (algorithm 4.5) and time domain compensation (algorithm 4.6), an illustrative example with fixed weighting approach is presented in figure 6.4. Looking at the dependence of reconstruction quality with respect to the gap length, the iterative gap shortening leads to opposite effects for the synthesis and analysis models (see figure 6.4a). Despite the observed improvement in the synthesis model, the SDR increase is only minor (compare with figure 6.3), and it does not outweigh the increased computational cost caused by the iterative nature of the shortening.

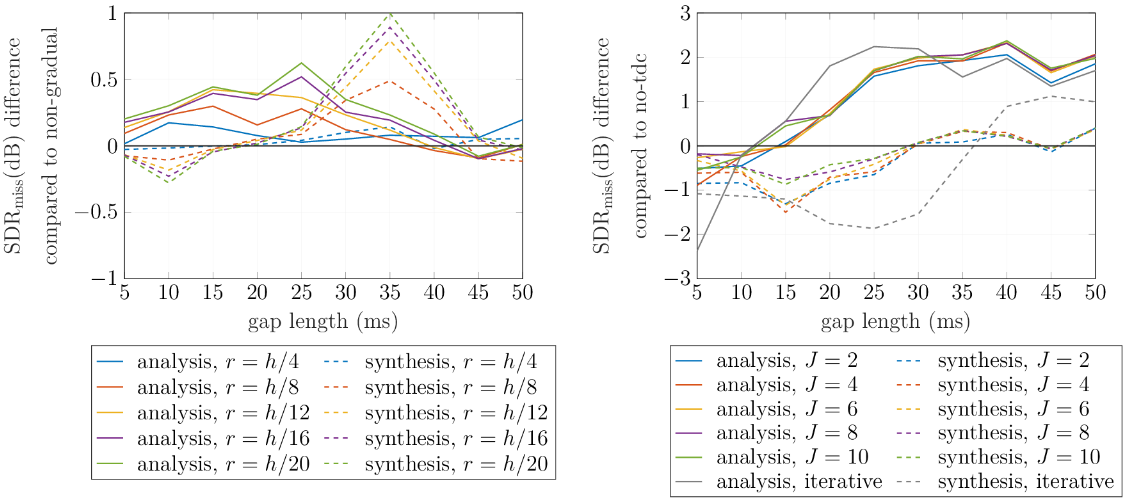
The time domain compensation is appealing in the analysis case, where this strategy leads to comparable results with the algorithm using iterative weights (see figure 6.4b). Note that both approaches (iterative re-weighting of algorithm 4.4 and the time domain compensation of algorithm 4.6) include multiple runs of the simple interpolation algorithm 4.2. In re-weighting, it is one run per each iteration of algorithm 4.4, in time domain compensation, it is one run per each “training” gap (denoted J in figure 6.4b). The notable observation here is that in the latter case, even values of the parameter J as small as 2 lead to a measurable improvement.



(a) synthesis model

(b) analysis model

Fig. 6.3: Experiment on weighting the atoms, for *half* offset. In each graph, the lines in lighter shade represent results from the other model to simplify visual comparison (i.e., both graphs plot the data for both the synthesis and the analysis model, but differ in what is highlighted).



(a) Iterative shortening of the gap with shrinkage parameter r, and h denoting the length of the gap in number of samples.

(b) Time domain compensation with J denoting the number of “training” gaps.

Fig. 6.4: Experiment on time-domain modifications based on algorithms 4.5 and 4.6. Both illustrations use *half* offset and energy-based atom weights (4.11e). For the time domain compensation in 6.4b, results using the iterative weights are also plotted for reference (this is mainly motivated by the results shown in figure 6.3b).

6.4.2 Heuristic approaches

To illustrate the action of SPAIN, we take a close look at the convergence properties and resulting performance of the proposed variants on two illustrative interpolation problems. Based on section 6.3, the first scenario is a random loss of 60 % samples, the second scenario features gaps of length 30 ms. Table 6.2 shows results of both interpolation problems using the variants of SPAIN proposed in section 4.2: S-SPAIN (H) and S-SPAIN (OMP) represent the synthesis variants with the coefficient update realized using hard thresholding and OMP, respectively (see algorithm 4.7). A-SPAIN (algorithm 4.8) uses exact evaluation of the coefficient update, thus it only appears as a single variant. Additionally, S-SPAIN and A-SPAIN coincide in case of an invertible transform [2, Section 7]. This is the case of choosing DFT with no redundancy, which is displayed in the last row of table 6.2.

Regarding the reconstruction quality, table 6.2 shows that redundancy in the transform is beneficial. On the other hand, using OMP in S-SPAIN promised more precise coefficient update, but did not lead to better quality in this example.

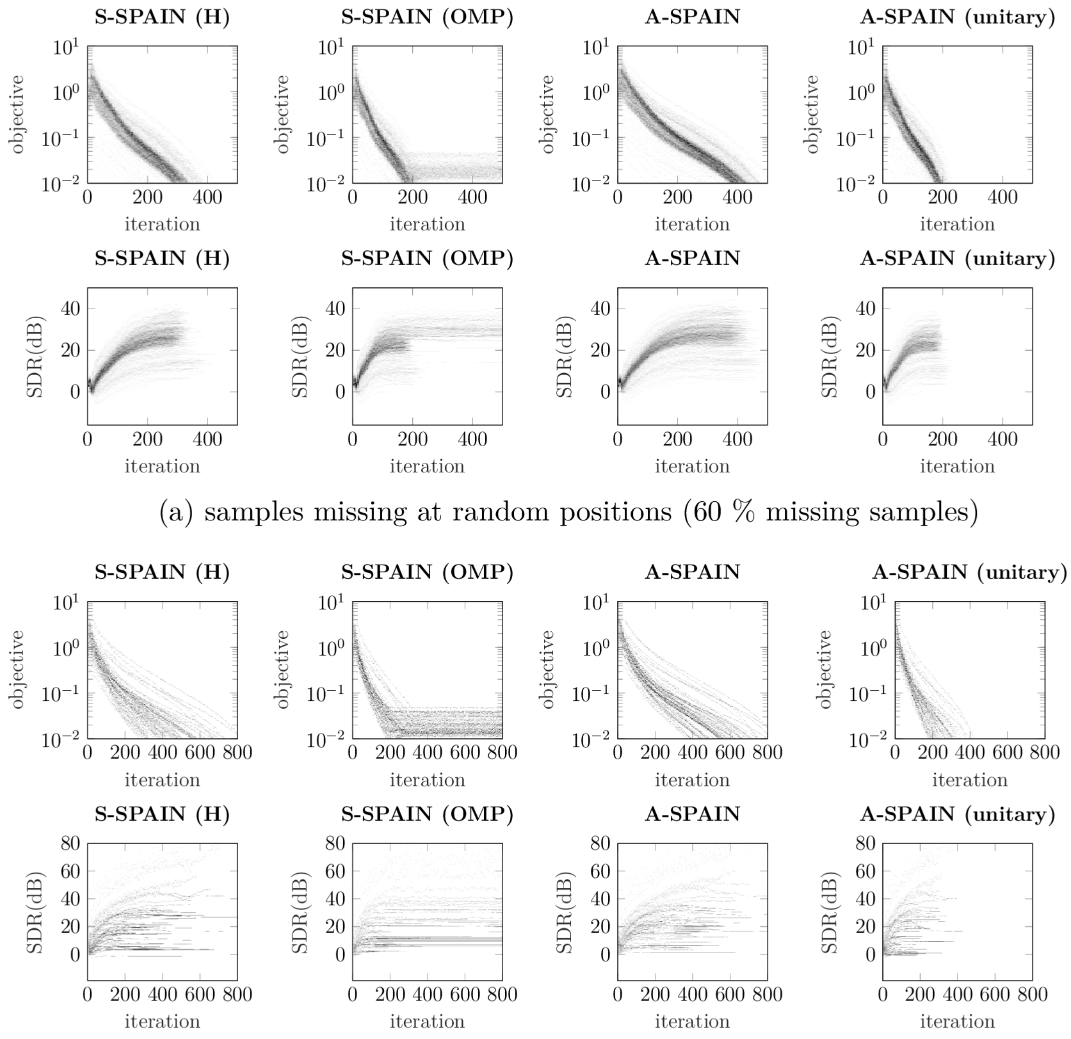
The convergence properties are studied more in figure 6.5. Since SPAIN is applied frame-wise, the stopping criterion may lead to different number of iterations in each frame. For this reason, figure 6.5 shows overlay of results for all the frames at the same time.[3] Concerning convergence of the objective, S-SPAIN (OMP) appears to be faster than S-SPAIN (H). However, it struggles to decrease the value in the range below 10^{-1} in case of several time frames. Convergence of the objective of A-SPAIN using the same redundancy of DFT was the slowest. On the other hand, the unitary variant was the fastest (but reaching lower SDR values).

Note that the observations regarding all variants of SPAIN, i.e., the preferability of hard thresholding in S-SPAIN and overall lead of A-SPAIN, are in line with the original publication [6], where the experiment was conducted using larger set of signals.

[3] Note that in the case of compact gaps (figure 6.5b), only a minority of time frames is effectively being interpolated. For this reason, plots in figure 6.5b feature less lines compared to figure 6.5a.

Tab. 6.2: Illustrative comparison of SPAIN variants. Note that SDR is computed both on the whole signal and on the interpolated samples only. In contrast to the previous experiments, a single signal (violin recording) sampled at 16 kHz is used.

method	DFT red.	60 % missing samples				10 × 30 ms gaps			
		SDR (dB)	SDR$_{\text{miss}}$ (dB)	PEMO-Q ODG	PEAQ ODG	SDR (dB)	SDR$_{\text{miss}}$ (dB)	PEMO-Q ODG	PEAQ ODG
S-SPAIN (H)	2	29.35	27.10	−2.59	−2.33	23.60	11.80	−1.34	−0.47
S-SPAIN (OMP)	2	27.56	25.31	−2.90	−3.13	21.60	9.81	−2.94	−0.58
A-SPAIN	2	29.55	27.31	−2.62	−2.18	22.32	10.52	−1.61	−0.47
A-SPAIN	1	27.80	25.55	−2.87	−3.05	17.09	5.29	−2.99	−0.59



(a) samples missing at random positions (60 % missing samples)

(b) samples missing in compact gaps (10 × 30 ms gaps)

Fig. 6.5: Illustrative comparison of SPAIN variants, with focus on convergence properties. The unitary variant denotes A-SPAIN using DFT with no redundancy (see also table 6.2). The algorithm is applied frame-wise, and it is stopped when the tolerance $\varepsilon = 10^{-2}$ is reached. For this reason, number of iterations in each frame may differ, which is also why all metrics are shown per each frame.

6.5 Evaluation of NMF-based methods

The first goal of this section is to demonstrate the convergence properties of the estimators described in chapter 5. The illustrative signal is an excerpt of the song *Mamavatu*, containing the acoustic guitar and the drums, sampled at 16 kHz. The sample loss was in this preliminary test was simulated according to the random scenario, loosing 60 % of the signal samples. The corrupted signal was interpolated using EM-*tf*, EM-t and AM (algorithms 5.1, 5.2, and 5.3, respectively). All algorithms shared the settings #1 of table 6.3. To allow comparison of all three algorithms, two settings of the transform $\mathbf{T}$ were considered – either the inverse DFT (i.e. with the number of frequency channels $F = W$), or its redundant variant with $F = 2W$. These choices represent cases 1 and 3 of section 5.2.3, respectively.

The results are presented in figure 6.6, where possible convergence was tracked using several metrics (see section 6.2). The AM algorithm features remarkable convergence properties, while the SDR indicates better performance for the EM-based algorithms. This motivates inclusion of the compromise algorithm denoted AM-to-EM-*tf*, which starts by a chosen number of AM iterations (5 in this example) and then switches to EM-*tf*.

The second experiment, which concludes this section, focuses on validation of the observations from figure 6.6 in the compact gap scenario. For the sake of this experiment, the set of 10 signals described in section 6.3 was used, with gaps of length ranging from 20 to 80 ms. Only the case of $F = W$ and invertible transform $\mathbf{T}$ representing the inverse DFT was considered, see the settings #2 from table 6.3 for details.

The results, presented in figure 6.7, support the previous observations, especially that the difference in reconstruction quality (in terms of SDR) between AM and EM-*tf* is not significant. Nonetheless, AM reached its peak faster, as seen also in the RSC. Note that due to computational demands of the problem, objective was not tracked in this case. However, the previous experiment suggests that RSC mostly corresponds to the convergence of the algorithm with respect to its objective value (see figure 6.6). A new observation is that this phenomenon depends on the gap length – the longer the gap, the slower the convergence of EM-*tf* is, compared to AM.

Tab. 6.3: Overview of the experiment settings for NMF based methods. All cases share sine window with 50 % overlap [106, Sec. V.C]. The value F corresponds to the length of the (possibly oversampled) DFT. The inner loop refers to the multiplicative updates of $\mathbf{W}$ and $\mathbf{H}$ in each of the algorithms.

	data and degradation			W	F	K	iterations		figure
	signals	sample rate	sample loss				outer	inner	
#1	1	16 kHz	60 %	1024	1024 or 2048	20	70	10	6.6
#2	10	44.1 kHz	10 gaps	4096	4096	20	50	10	6.7
#3	10	44.1 kHz	10 gaps	4096	4096	20	100	10	6.8

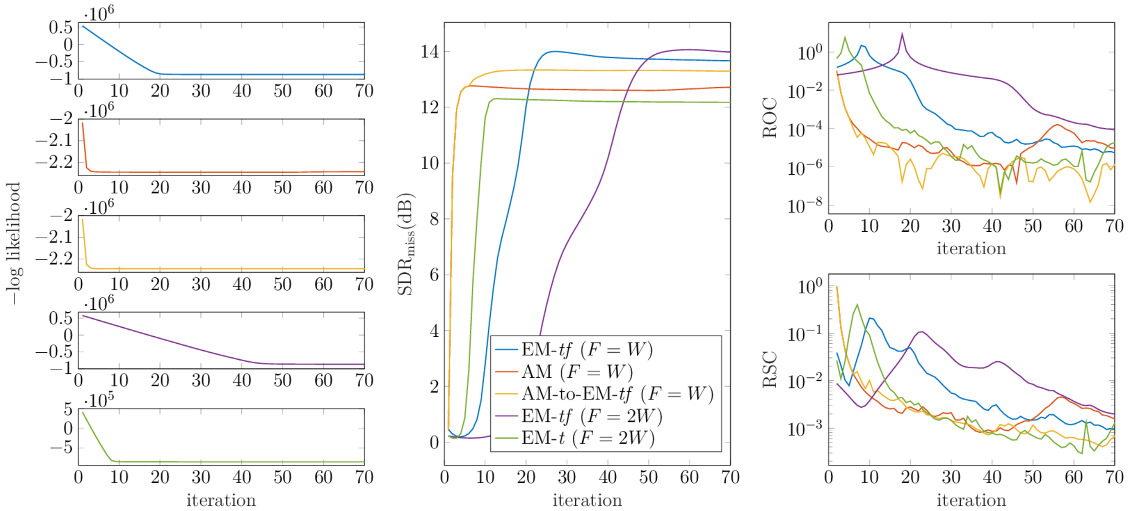


Fig. 6.6: Comparison of the convergence properties of EM-*tf*, EM-*t* and AM, including the switching variant AM-to-EM-*tf*. The legend in the middle plot applies for the whole figure. Note that the first column does not show comparable quantities, since the objective depends on the choice of F and also on the algorithm. In particular, the formula for log likelihood switches after the initializing iterations of AM-to-EM-*tf*, which is disregarded on purpose in the plot.

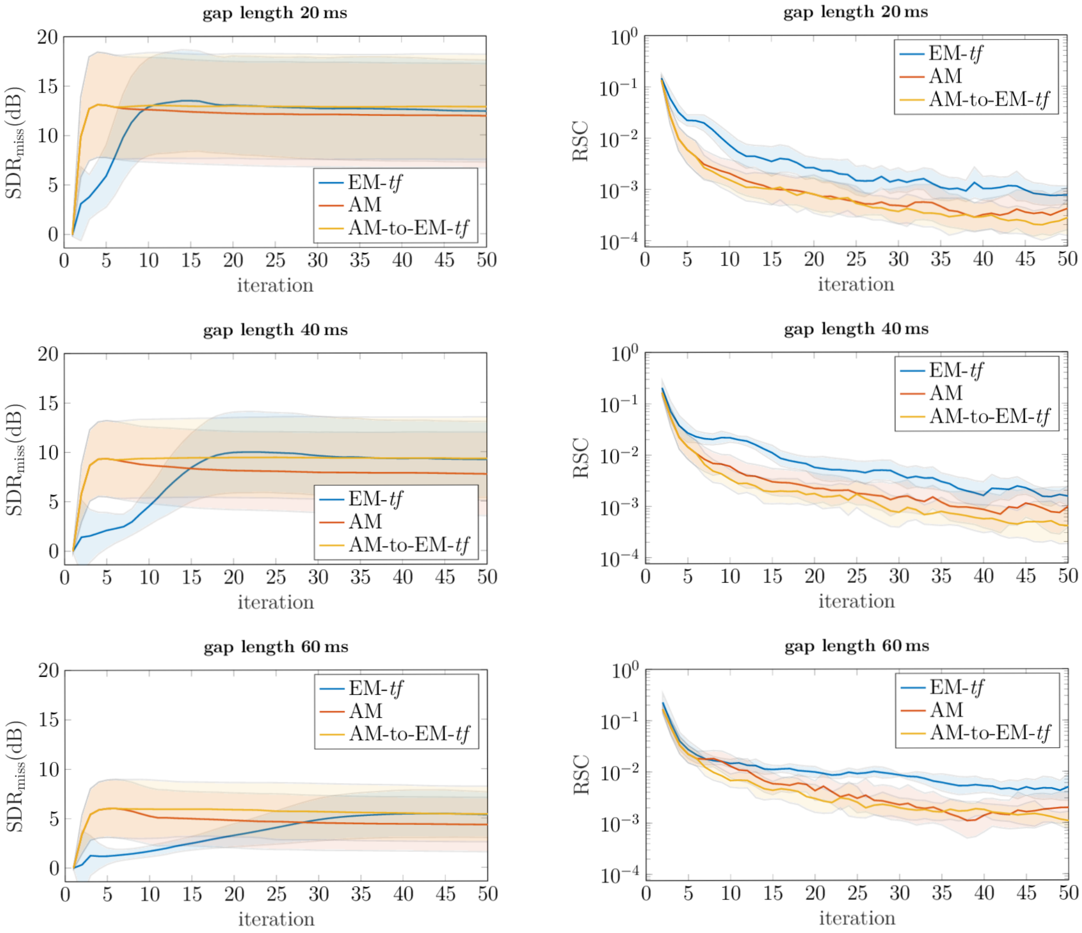


Fig. 6.7: Comparison of the performance of EM-*tf*, AM, and the switching variant AM-to-EM-*tf* (switching after 5 iterations). The left column shows the evolution of the SDR over iterations, the right column shows the RSC. Both the metrics are averaged over the dataset and plotted together with 95% confidence interval represented by the light colored areas.

6.6 Comparative study featuring the state-of-the-art interpolators

The derivation and evaluation of the interpolation methods is concluded by comparing selected algorithms with the state of the art. The focus is on the interpolation of middle-length gaps, which allows the use of context-based approaches. From the variety of available methods, several representative competitors are chosen:

- The AR-based methods are usually represented by Janssen's iterative approach [47] or by Etter's forward and backward extrapolation [49], see section 2.2 for details. An extension in the present experiment is the modified Janssen's method, which differs from the original one in signal segmentation – the AR model is presumably estimated per overlapping time frames in the original method, but the modified version considers each gap (including short context around it) separately. Furthermore, all these methods are presented in two versions, depending on the chosen estimator of the AR coefficients and denoted by the respective Matlab function: lpc (minimizing forward prediction error) and arburg (Burg's method [48, Sec. 5.1.2]).
- The convex approach (section 4.1) is represented by the analysis based algorithm 4.2 with either energy-based or iterative weights (see equation (4.11e) and algorithm 4.4, respectively).
- A-SPAIN (algorithm 4.8) serves as a representative of the ADMM-based family of methods. However, successors of SPAIN have already been developed, namely SPAIN-MOD and SPAIN-LEARNED [72], outperforming the original algorithm. SPAIN-MOD treats the gaps together with their context, instead of overlapping signal frames. SPAIN-LEARNED further improves the quality of the interpolation by a dictionary learning step, namely by deforming the STFT to allow a sparser representation than using the STFT.[4]
- The NMF-based methods are represented by EM-*tf* and AM, i.e., algorithms 5.1 and 5.3, respectively.

Remark 6.2 (Particular settings of the signal models)**.** With the intention of fair comparability, all relevant methods used a sine window of length $W = 4096$ samples (approx. 92 ms) with 50% overlap. This concerns all methods using STFT (i.e., methods based on ℓ_1 minimization, A-SPAIN-LEARNED and NMF), A-SPAIN and Janssen. Methods based on NMF and ℓ_1 norm were applied with equal window length and number of frequency channels ($F = W$). On the other hand, A-SPAIN-LEARNED used $F = 2W$, and similarly A-SPAIN used DFT with redundancy 2.

[4] Similar modification has been proposed also for the convex approach; however, it has been shown to be inferior to A-SPAIN-LEARNED [73].

The AR-based methods used a model of order $p = 512$. The context of modified Janssen's and Etter's methods was set to 4096 samples, while A-SPAIN-LEARNED used a longer context (8192 samples) for the sake of the dictionary learning. These values were chosen based on the corresponding studies, where they have shown good performance. For particular choices of all the parameters of the individual methods, please refer to the published source code accompanying the article [8] (see the link in appendix B).

Remark 6.3 (AR-based methods)**.** In current literature, the AR-based methods are usually represented by Etter's method using either linear prediction [4] or Burg's method [81, 107]. Janssen algorithm appeared in the seminal article *Audio Inpainting* [20] using linear prediction error and temporal windows. The modified version was used only recently in [8], inspired by the approach of SPAIN-MOD [72]. Regarding the AR model estimators, Kauppinen and Roth prefer the Burg's method for audio signal extrapolation [51, Sec. 4.2]. However, comparison of the estimators in the context of audio interpolation, especially when being part of the iterative Janssen's method, is missing in the literature.

Remark 6.4 (Deep-learning-based methods)**.** Study including methods based upon deep learning is left for future research. Comparability with optimization-based methods is difficult because successful methods usually aim at spectrogram inpainting [81, 82, 108]. This approach is hardly adaptable to the scenarios discussed in section 6.3, where temporal samples are being discarded, since such a degradation does not correspond to a simple drop-out of spectrogram columns. Furthermore, neural networks are usually trained for a fixed input size, which makes testing with variable gap length difficult.

For the compact-gap scenario and 10 test signals (see section 6.3), the averaged performance results are shown in figure 6.8 (for complete results, see appendix C). From the proposed methods (displayed in bold), the NMF-based approaches (both EM-*tf* and AM) scored the best in terms of the SDR. For gaps above 40 ms, performance of the NMF-based methods drops rather fast, similarly to Janssen. A possible cause is the estimation of the missing samples via Wiener filtering (as mentioned in section 5.2) from a limited number of samples. This effect appears especially when long gaps are considered, since those gaps induce presence of signal frames with large number of missing samples. A similar step appears within the Janssen algorithm in the signal update step [109, Sec. 3.3].

The sparsity based methods appeared to be inferior in this experiment. However, note that previously surpassed methods such as plain ℓ_1 minimization or the OMP (see e.g. the experiment in [4]) have been omitted.

The perceptually-motivated comparison was realized using both PEMO-Q and PEAQ (see section 6.1). The first observation is that PEAQ was mostly indecisive in this study (except for disfavoring the iteratively weighted ℓ_1 relaxation). This might be due to the fact that local degradation is not largely reflected in the PEAQ algorithm.

More credible scores were obtained using PEMO-Q. The largest discrepancy between the ODG and SDR results appears in the case of A-SPAIN (reaches scores comparable with EM-*tf* and AM) and the convex approach with iterative weights (scores the lowest globally). Also the (frame-wise) Janssen's method was slightly more favored in terms of ODG, compared to the SDR.

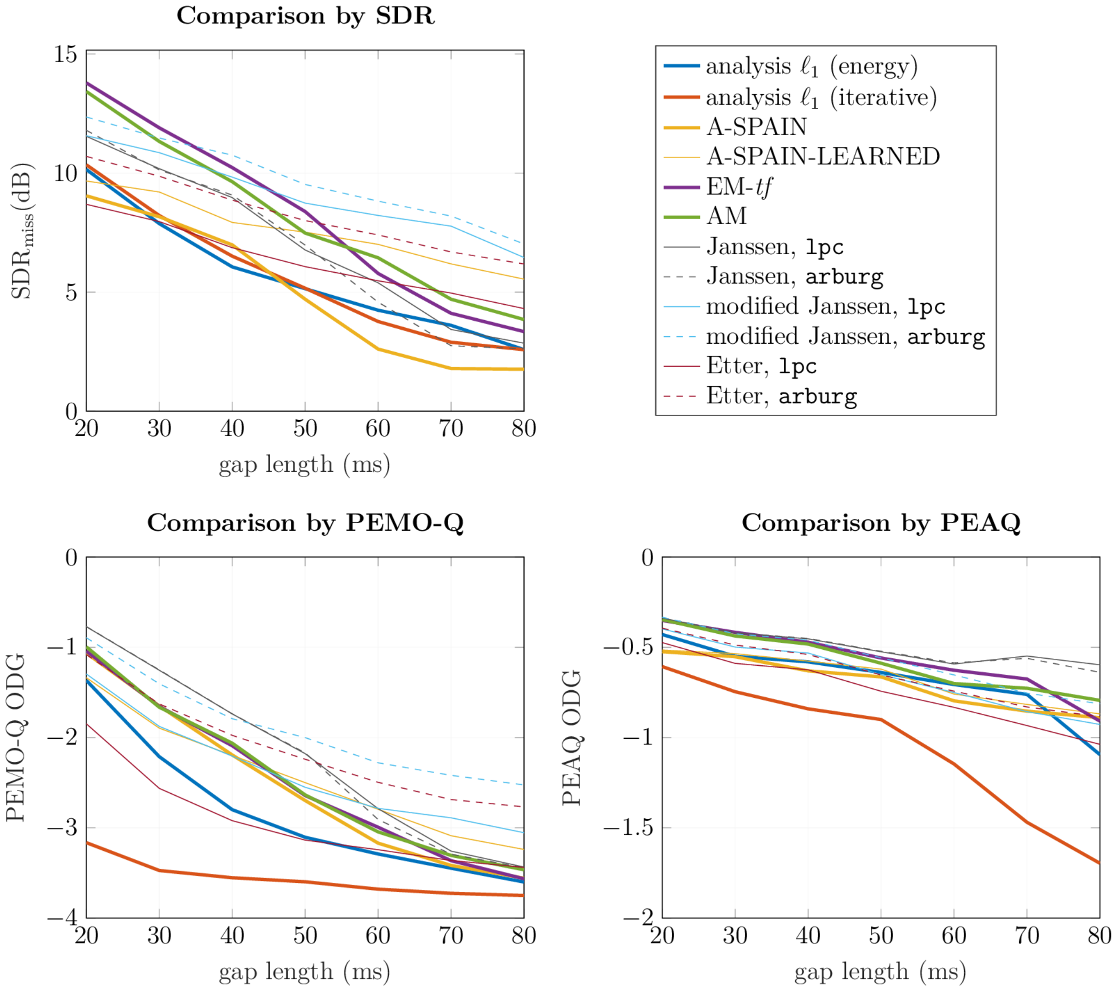


Fig. 6.8: Comparison of the developed algorithms with the state-of-the-art methods for interpolation of short to middle-length gaps. The legend is common for all the plots. Where applicable, the quality metrics were tracked during iterations and the figure shows the peak values. Note that this strategy is applied separately per each test instance, as well as per each quality criterion. Also note that for the sake of readability, the ODG axis for PEAQ is cropped, even though the ODG ranges from −4 to 0.

Tab. 6.4: Elapsed times of the methods used in the comparative study. Both iterations and time to peak are counted/measured to the point where the peak SDR is reached, where applicable. In other words, the iterations and durations come from the same data used to generate the SDR subplot of figure 6.8. The maximum number of (outer) iterations of EM-*tf*, AM and all variants of the Janssen algorithm was set to 100. The mean and median values are computed across all the signals and gap lengths considered in the experiment.

method	mean values			median values		
	iterations to peak	time (s) to peak	time (s) per iteration	iterations to peak	time (s) to peak	time (s) per iteration
analysis ℓ_1 (energy)	—	53.30	—	—	48.34	—
analysis ℓ_1 (iterative)	—	675.72	—	—	747.51	—
A-SPAIN	—	43.55	—	—	42.87	—
A-SPAIN-LEARNED	—	6599.70	—	—	6325.10	—
EM-*tf*	42.74	2602.37	62.66	23	1432.15	61.68
AM	17.29	740.55	41.82	4	161.20	41.12
Janssen, `lpc`	15.13	23.58	1.38	3	4.92	1.26
Janssen, `arburg`	9.66	15.44	1.92	3	5.36	1.44
modified Janssen, `lpc`	5.19	10.26	1.67	1	2.59	1.45
modified Janssen, `arburg`	6.16	15.45	2.29	2	3.92	2.13
Etter, `lpc`	—	0.03	—	—	0.03	—
Etter, `arburg`	—	0.34	—	—	0.34	—

Even though the overall focus of the experiment is on reconstruction quality rather than speed of the algorithms, we present the elapsed times for the sake of completeness in table 6.4.

First, note that the number of iterations have been only tracked in the case of the Janssen algorithm and the NMF-based methods. For the convex approach, the number of iterations was fixed rather high (2000), since the aim was to converge to global minimum of the respective convex problem. In the case of the iterative weights, the outer cycle was run for 10 iterations (see algorithm 4.4), leading to 10 times higher computational load compared to using the energy-based weights. As previously mentioned in section 6.4.2, tracking times and performance per iteration in SPAIN is not feasible, since meeting the stopping criterion largely changes from frame to frame (see figure 6.5). While this is not the case of A-SPAIN-LEARNED, which is based on the STFT, the computational load is hugely increased by the dictionary learning step. Finally, Etter's method is non-iterative, and is clearly the fastest according to table 6.4.

Regarding the NMF-based methods, we observe that the number of iterations needed to reach the peak SDR is much lower for AM, compared to EM-*tf*. In particular, observe from the median values that in half of the cases, AM reached its peak SDR in less then 4 iterations. This corresponds also to the experiments presented in section 6.5. Furthermore, the time per iteration of AM is slightly lower than that of EM-*tf*, which is in line with [8, Rem. 4]. However, in the context of the rest of the methods, both EM-*tf* and AM rank among the slower ones.

Concerning the variants of the Janssen algorithm, the results indicate slightly higher computational cost of the Burg's method. Overall, the elapsed times to peak rank among the faster methods, especially with regard to a very fast progression towards the peak.

The experimental part of the thesis is concluded by an illustration of the performance of the methods on a particular signal. To this end, figures 6.9 and 6.10 present an example taken from the comparative experiment. The signal under test is the guitar recording with gap length of 40 ms. Only an excerpt of the reconstruction is shown for readability reasons, as well as only a subset of methods, with the iteration being plotted chosen based on SDR. This particular test instance was selected due to the observation that the results for the guitar signal correlate the most with the average values for all signals (compare figures 6.8 and C.1).

A detailed inspection of the solution of the convex relaxation (denoted as the analysis ℓ_1) in both figure 6.9 and 6.10 reveals the remainder of a minor energy drop in both interpolated gaps. On the other hand, it appears that no "parasite" components are introduced in the signal.

Notably, A-SPAIN-LEARNED exhibits an unusual behavior in the first of the displayed gaps and represents reliably only the lowest frequency content (see the corresponding plot in figure 6.10). While this individual case did not largely affect the average performance shown in figure 6.8, it has a visible effect in context of the particular signal (see figure C.1).

Since all the methods discussed are context-based, interpolation of the first displayed gap is expected to be contaminated by the close presence of the onset of a new chord. This onset is evident from the waveform envelope in figure 6.9, while its overflow into the interpolated gap is more visible in the spectrograms in figure 6.10. The most high-frequency content (above 4 kHz) is introduced in the case of A-SPAIN; on the other hand, this effect is not too apparent in the case of EM-*tf* and Janssen algorithm.

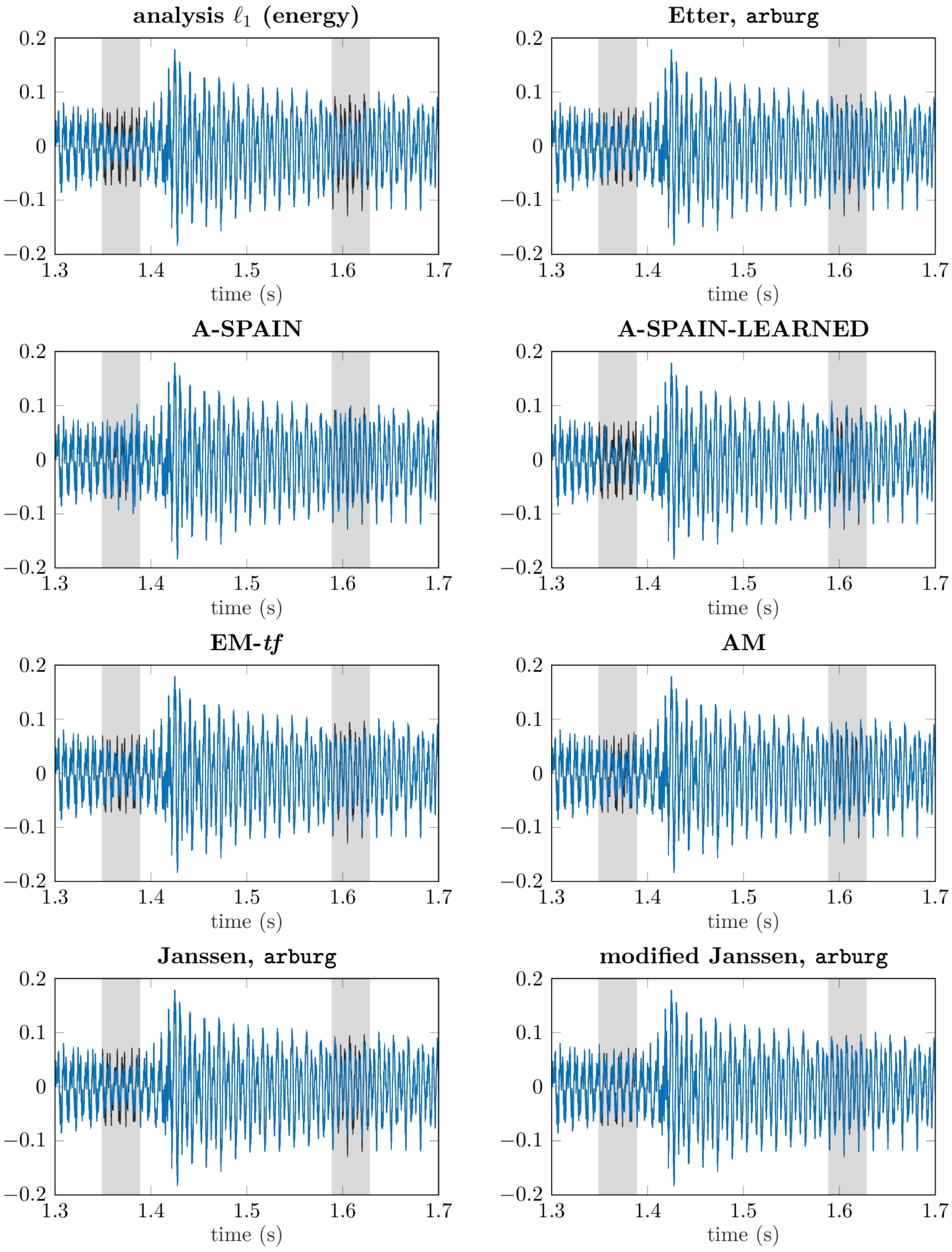


Fig. 6.9: Illustrative reconstruction (waveforms) of the guitar recording (signal a58_guitar_sarasate in appendix C) with gaps of length 40 ms (highlighted by the gray areas). The blue waveform represents the interpolated signal, the black parts (visible only in the gaps) come from the reference signal.

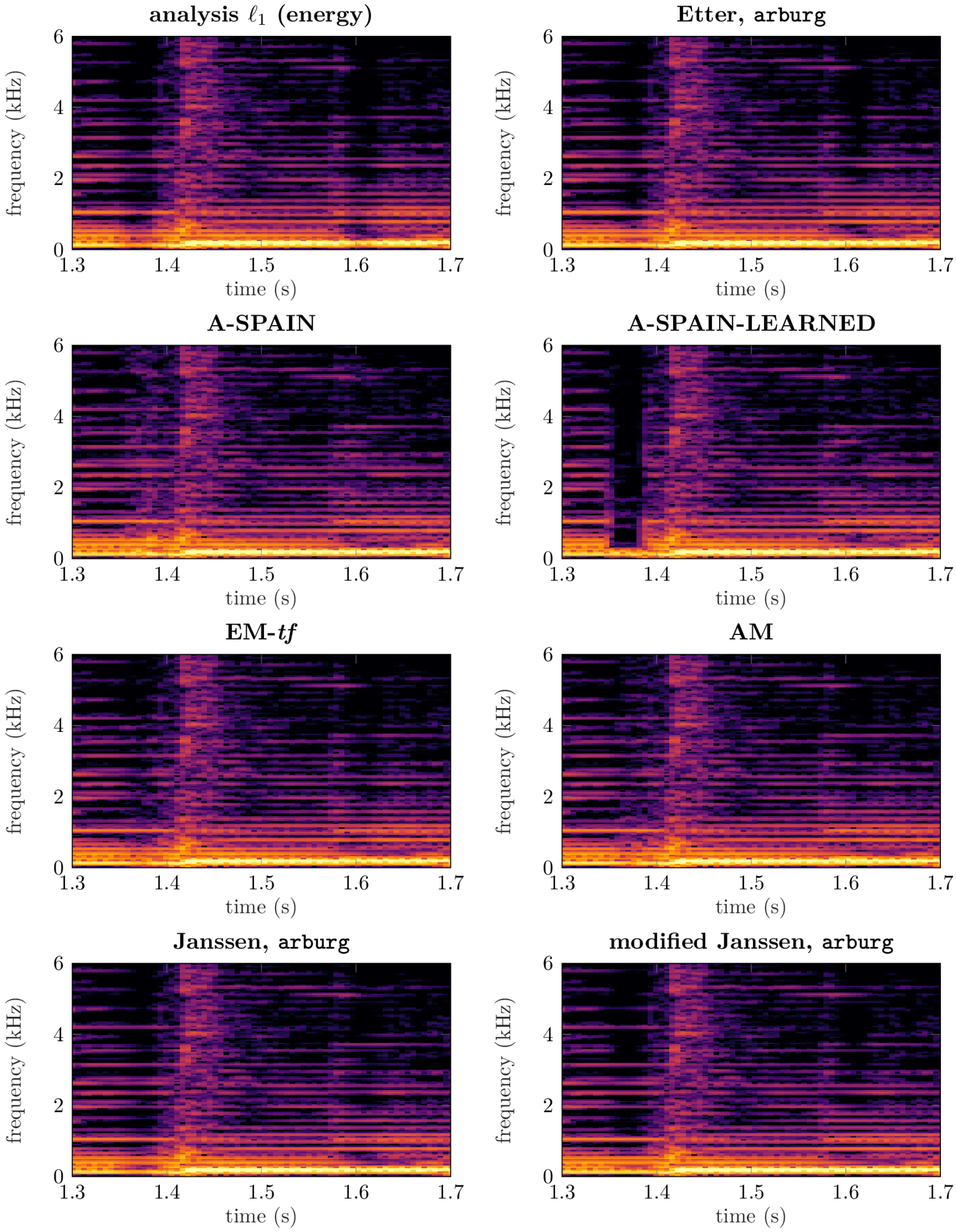


Fig. 6.10: Illustrative reconstruction (spectrograms) of the guitar recording (signal a58_guitar_sarasate in appendix C) with gaps of length 40 ms. The gap positions are not highlighted, however, they exactly match those of figure 6.9. Note that all the plots share the colormap, even though the color scale is not displayed.

Conclusions and perspectives

The thesis was devoted to the study and development of new algorithms for the interpolation of missing parts in audio signals. The main ambition was twofold: The first goal was to improve the performance of existing methods based on sparsity of audio signals in a transformed domain. The second goal was to challenge the state of the art using novel methods based either on sparse or on structured signal representations.

To fulfill the goals of the thesis, chapter 1 compiled necessary theoretical background regarding inverse problems and audio signal processing, with focus on audio signal interpolation. The overview of the state of the art was presented in chapter 2, where the goals of the thesis emerged and have been stated later in chapter 3. In the core part of the thesis, two main classes of methods were studied, and both classes have been enriched by novel algorithms and modifications of the existing ones.

Chapter 4 presented the use of spectrogram sparsity to solve the audio interpolation problem. The synthesis and analysis perspectives have been formulated in a relaxed, convex form, and suitable proximal algorithms have been adopted for solving the respective problems. This approach has been supplemented with novel atom weighting and two time-domain-based heuristic procedures to tackle the energy loss in the interpolated signal segment. Alternatively, the declipping algorithm SPADE has been adapted for audio interpolation and coined SPAIN. In addition to the synthesis- and analysis-based flavors (S-SPAIN and A-SPAIN, respectively), the novel variant of S-SPAIN using sub-iterations of OMP has been proposed.

Chapter 5 introduced three interpolation algorithms based on probabilistic modeling and on the assumption that the spectrograms of convenient audio signals have a particular structure related to the NMF. As in the case of SPAIN, EM-*tf*, is based on a successful precursor for audio declipping. The other two, EM-*t* and AM, are newly derived estimators from the same assumptions, serving as alternatives to EM-*tf*.

The evaluation of the proposed methods in chapter 6, including the comparison with the state of the art, fulfilled the goals of the thesis. The experiments concerned all aforementioned classes of methods, namely algorithms based on the convex relaxation of the sparsity assumption, on heuristic use of the ADMM for the non-convex problem (i.e., SPAIN), and on signal model using probabilistic NMF. All previously derived algorithms have been empirically proven to be viable in terms of both entrywise (SDR) and psychoacoustics-based quality metrics (PEMO-Q, PEAQ). While PEAQ did not give decisive results, both SDR and PEMO-Q indicated dominance of the NMF-based methods. In terms of SDR, the state of the art has been improved in the case of short gaps (10–30 ms) and equal results have been achieved up to gap length of 50 ms. On the other hand, PEMO-Q favored Janssen's iterative AR-based algorithm. Even though AR-based methods dominated in the case of gap lengths above 60 ms, a notable competitor was A-SPAIN-LEARNED, a successor of the A-SPAIN algorithm.

While most of the dissertation ideas have been previously published by the author (see the list of author's bibliography), the thesis included a more thorough treatment, as well as new perspectives on the topic. Regarding the experiments, the illustration of SPAIN in section 6.4.2 is novel. Furthermore, the state of the art comparison in section 6.6 not only gathered the results from the original experiments, but included more reference methods and a more complex analysis.

Even though chapter 6 presented broad range of experiments to evaluate the methods and compare the results with the state of the art, two more aspects could be suggested: first, comparison of optimization- and learning-based methods, and second, standardized listening testing. The first aspect stems from unavailability of a fair test protocol allowing such comparison, as discussed in remark 6.4. Similarly, proper listening tests are considered infeasible for the amount of data evaluated objectively throughout chapter 6. However, a possible future goal is to design a listening test aiming at validation of the PEMO-Q outputs in the presence of audible artifacts specific for interpolated signals.

Further perspectives for future research regarding optimization-based methods include, but are not limited to, the following directions:

The first promising idea is to employ psychoacoustics in the interpolation process, aiming at subjective listening quality of the resulting signal. Atom weighting based on hearing thresholds and masking have been proposed in the context of audio declipping [110], which can be readily applied in place of the weights designed in chapter 4 of the thesis. However, preliminary tests in this direction have not proved the approach viable in case of gap interpolation. Another idea is the use of psychoacoustically motivated representations of audio signals instead of STFT, such as ERBlets [111], which has been studied only moderately in [62].

The second possible direction is further research in previously established combinations with deep learning, such as using the deep prior [112] or unfolding [113]. While deep prior has been recently applied for interpolation of gaps in audio signals [108], unrolling has been studied only in the context of spectrogram phase reconstruction [87]. A novel possible approach consists in using parts of successful neural networks as regularizing terms in an optimization-based method. For example, suitability of a candidate signal could be measured using a previously trained discriminator of a generative adversarial network [114].

Author's Bibliography

1. MOKRÝ, Ondřej; RAJMIC, Pavel. Approximal operator with application to audio inpainting. *Signal Processing*. 2021, vol. 179, p. 107807. ISSN 0165-1684. Available from DOI: 10.1016/j.sigpro.2020.107807.
2. ZÁVIŠKA, Pavel; MOKRÝ, Ondřej; RAJMIC, Pavel. *S-SPADE Done Right: Detailed Study of the Sparse Audio Declipper Algorithms*. 2018-09. Tech. rep. Brno University of Technology. Available from arXiv: 1809.09847 [math.OC].
3. RAJMIC, Pavel; ZÁVIŠKA, Pavel; VESELÝ, Vítězslav; MOKRÝ, Ondřej. A New Generalized Projection and Its Application to Acceleration of Audio Declipping. *Axioms*. 2019, vol. 8, no. 3. ISSN 2075-1680. Available from DOI: 10.3390/axioms8030105.
4. MOKRÝ, Ondřej; RAJMIC, Pavel. Audio Inpainting: Revisited and Reweighted. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2020, vol. 28, pp. 2906–2918. Available from DOI: 10.1109/taslp.2020.3030486.
5. MOKRÝ, Ondřej; RAJMIC, Pavel. Reweighted l1 minimization for audio inpainting. In: *Proceedings of the 2019 SPARS workshop*. Toulouse, 2019. Available also from: https://www.utko.fekt.vut.cz/~rajmic/papers/Mokry_Rajmic-Reweighted_l1_inpainting-SPARS2019.pdf.
6. MOKRÝ, Ondřej; ZÁVIŠKA, Pavel; RAJMIC, Pavel; VESELÝ, Vítězslav. Introducing SPAIN (SParse Audio INpainter). In: *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019.
7. ZÁVIŠKA, Pavel; RAJMIC, Pavel; MOKRÝ, Ondřej; PRŮŠA, Zdeněk. A Proper Version of Synthesis-based Sparse Audio Declipper. In: *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom, 2019, pp. 591–595. ISSN 2379-190X. Available from DOI: 10.1109/ICASSP.2019.8682348.
8. MOKRÝ, Ondřej; MAGRON, Paul; OBERLIN, Thomas; FÉVOTTE, Cédric. Algorithms for audio inpainting based on probabilistic nonnegative matrix factorization. *Signal Processing*. 2022, p. 10. ISSN 1872-7557. Available from DOI: 10.1016/j.sigpro.2022.108905.
9. MOKRÝ, Ondřej; RAJMIC, Pavel; ZÁVIŠKA, Pavel. Flexible framework for audio reconstruction. In: *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx2020)*. Vienna, Austria, 2020-21, vol. 1. ISSN 2413-6689. Available also from: https://dafx2020.mdw.ac.at/proceedings/index.html.

10. ZÁVIŠKA, Pavel; RAJMIC, Pavel; MOKRÝ, Ondřej. Audio Dequantization Using (Co)Sparse (Non)Convex Methods. In: *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Toronto, Canada, 2021, pp. 701–705. ISSN 2379-190X. Available from DOI: 10.1109/ICASSP39728.2021.9414637.

11. MOKRÝ, Ondřej; ZÁVIŠKA, Pavel. Inconsistent Audio Declipping Performance Enhancement Based on Audio Inpainting. In: *Proceedings of the 27th Conference STUDENT EEICT 2021*. Brno University of Technology, Faculty of Electrical Engineering and Communication, 2021, pp. 596–600. ISBN 978-80-214-5614-3.

12. ZÁVIŠKA, Pavel; RAJMIC, Pavel; MOKRÝ, Ondřej. Audio declipping performance enhancement via crossfading. *Signal Processing*. 2022, vol. 192, p. 108365. ISSN 0165-1684. Available from DOI: 10.1016/j.sigpro.2021.108365.

13. ZÁVIŠKA, Pavel; RAJMIC, Pavel; MOKRÝ, Ondřej. Multiple Hankel matrix rank minimization for audio inpainting. In: *2023 46th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2023. Available from DOI: 10.1109/tsp59544.2023.10197741.

Bibliography

14. CAMBRIDGE UNIVERSITY PRESS & ASSESSMENT. *INTERPOLATION | English Meaning – Cambridge Dictionary* [online]. 2023. [visited on 2023-12-30]. Available from: https://dictionary.cambridge.org/dictionary/english/interpolation.

15. MACH, Vaclav. Denoising Phonogram Cylinders Recordings Using Structured Sparsity. In: *7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. 2015, pp. 314–319.

16. JAYANT, Nuggehally S.; CHRISTENSEN, Susan W. Effects of Packet Losses in Waveform Coded Speech and Improvements Due to an Odd-Even Sample-Interpolation Procedure. *IEEE Transactions on Communications*. 1981, vol. 29, no. 2, pp. 101–109. Available from DOI: 10.1109/tcom.1981.1094975.

17. DEJDAR, Petr; MOKRÝ, Ondřej; ČÍŽEK, Martin; RAJMIC, Pavel; MÜNSTER, Petr; SCHIMMEL, Jiří; PRAVDOVÁ, Lenka; HORVÁTH, Tomáš; ČÍP, Ondřej. Characterization of sensitivity of optical fiber cables to acoustic vibrations. *Scientific Reports*. 2023, vol. 13, no. 1. Available from DOI: 10.1038/s41598-023-34097-9.

18. DEJDAR, Petr; MOKRY, Ondrej; MUNSTER, Petr; SPURNY, Vladimir; HORVATH, Tomas; TOMASOV, Adrian; JASIM, Ali A.; PODRAZKY, Ondrej; BORODKIN, Andrei; HONZATKO, Pavel. Negative Curvature Hollow-Core Photonic Crystal Fibers Sensitivity to Acoustic Vibration in the Audible Spectrum. In: *2023 International Workshop on Fiber Optics on Access Networks (FOAN)*. IEEE, 2023. Available from DOI: 10.1109/foan59927.2023.10328118.

19. SHANNON, Claude Elwood. Communication in the Presence of Noise. *Proceedings of the IRE*. 1949, vol. 37, no. 1, pp. 10–21. ISSN 0096-8390. Available from DOI: 10.1109/jrproc.1949.232969.

20. ADLER, Amir; EMIYA, Valentin; JAFARI, Maria G.; ELAD, Michael; GRIBONVAL, Rémi; PLUMBLEY, Mark D. Audio Inpainting. *IEEE Transactions on Audio, Speech, and Language Processing*. 2012, vol. 20, no. 3, pp. 922–932. ISSN 1558-7916. Available from DOI: 10.1109/TASL.2011.2168211.

21. ANDREWS, C.A.; DAVIES, J.M.; SCHWARZ, G.R. Adaptive data compression. *Proceedings of the IEEE*. 1967, vol. 55, no. 3, pp. 267–277. Available from DOI: 10.1109/proc.1967.5481.

22. MATHEWS, M. Extremal coding for speech transmission. *IEEE Transactions on Information Theory*. 1959, vol. 5, no. 3, pp. 129–136. Available from DOI: 10.1109/tit.1959.1057504.

23. STRANG, Gilbert. *Introduction to linear algebra*. Wellesley-Cambridge Press, 2009. ISBN 9780980232721.

24. HAN, D.; KORNELSON, K.; LARSON, D.; WEBER, E. *Frames for Undergraduates*. American Mathematical Society, 2008. ISBN 9780821842126.

25. HEIL, Christopher. *A Basis Theory Primer: Expanded Edition (Applied and Numerical Harmonic Analysis)*. Birkhäuser, 2010. ISBN 978-0-8176-4687-5.

26. CHRISTENSEN, Ole; CHRISTENSEN, Khadija Laghrida. *Approximation Theory, From Taylor Polynomials to Wavelets*. Boston: Birkhäuser, 2004. ISBN 0-8176-3600-5.

27. BALAZS, P.; DÖRFLER, M.; KOWALSKI, M.; TORRÉSANI, B. Adapted and adaptive linear time-frequency representations: a synthesis point of view. *IEEE Signal Processing Magazine*. 2013, vol. 30, no. 6, pp. 20–31. Available from DOI: 10.1109/MSP.2013.2266075.

28. BEN-ISRAEL, Adi; GREVILLE, Thomas N.E. *Generalized Inverses: Theory and Applications*. Springer-Verlag, 2003. ISBN 0-387-00293-6.

29. BOYD, Stephen Poythress; VANDENBERGHE, Lieven. *Convex Optimization*. Cambridge University Press, 2004. ISBN 0521833787.

30. BERNDTSSON, Bo; CORDERO-ERAUSQUIN, Dario; KLARTAG, Bo'az; RUBINSTEIN, Yanir A. Complex Legendre duality. *American Journal of Mathematics*. 2020, vol. 142, no. 1, pp. 323–339. Available from DOI: 10.1353/ajm.2020.0008.

31. MOREAU, Jean Jacques. Proximité et dualité dans un espace hilbertien. *Bulletin de la société mathématique de France*. 1965, vol. 93, pp. 273–299. ISSN 0037-9484.

32. BECK, Amir. *First-Order Methods in Optimization*. SIAM-Society for Industrial and Applied Mathematics, 2017. ISBN 9781611974980.

33. COMBETTES, P.L.; PESQUET, J.C. A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing*. 2007, vol. 1, no. 4, pp. 564–574. Available from DOI: 10.1109/JSTSP.2007.910264.

34. CONDAT, L.; KITAHARA, D.; CONTRERAS, A.; HIRABAYASHI, A. Proximal Splitting Algorithms for Convex Optimization: A Tour of Recent Advances, with New Twists. *SIAM Review*. 2023, vol. 65, no. 2, pp. 375–435. ISSN 1095-7200. Available from DOI: 10.1137/20m1379344.

35. COMBETTES, P. L.; PESQUET, J. C. Proximal splitting methods in signal processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. 2011, vol. 49, pp. 185–212. Available from DOI: 10.1007/978-1-4419-9569-8_10.

36. CHAMBOLLE, Antonin; POCK, Thomas. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*. 2011, vol. 40, no. 1, pp. 120–145. ISSN 1573-7683. Available from DOI: 10.1007/s10851-010-0251-1.

37. CONDAT, Laurent. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*. 2013, vol. 158, no. 2, pp. 460–479. Available from DOI: 10.1007/s10957-012-0245-9.

38. BOYD, Stephen P.; PARIKH, Neal; CHU, Eric; PELEATO, Borja; ECKSTEIN, Jonathan. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*. 2011, vol. 3, no. 1, pp. 1–122. Available also from: http://dblp.uni-trier.de/db/journals/ftml/ftml3.html#BoydPCPE11.

39. CONDAT, L. A Generic Proximal Algorithm for Convex Optimization—Application to Total Variation Minimization. *Signal Processing Letters, IEEE*. 2014, vol. 21, no. 8, pp. 985–989. ISSN 1070-9908. Available from DOI: 10.1109/LSP.2014.2322123.

40. VŨ, Bang Công. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*. 2011, vol. 38, no. 3, pp. 667–681. Available from DOI: 10.1007/s10444-011-9254-8.

41. SCHLOEGL, Matthias; HOLLER, Martin; SCHWARZL, Andreas; BREDIES, Kristian; STOLLBERGER, Rudolf. Infimal convolution of total generalized variation functionals for dynamic MRI. *Magnetic Resonance in Medicine*. 2017, vol. 78, no. 1, pp. 142–155. Available from DOI: 10.1002/mrm.26352.

42. BREDIES, Kristian; HOLLER, Martin. A TGV-Based Framework for Variational Image Decompression, Zooming, and Reconstruction. Part II: Analytics. *Siam Journal On Imaging Sciences*. 2015, vol. 8, no. 4, pp. 2851–2886. ISSN 1936-4954. Available from DOI: 10.1137/15M1023865.

43. MOKRÝ, Ondřej; VITOUŠ, Jiří. Unrolled L+S decomposition for compressed sensing in magnetic resonance imaging. *Elektrorevue*. 2022, vol. 24, no. 3, pp. 86–93. ISSN 1213-1539. Available also from: http://hdl.handle.net/11012/214078.

44. FÉVOTTE, Cédric; BERTIN, Nancy; DURRIEU, Jean-Louis. Nonnegative matrix factorization with the Itakura–Saito divergence: With application to music analysis. *Neural computation*. 2009, vol. 21, no. 3, pp. 793–830.

45. LEE, Daniel; SEUNG, H. Sebastian. Algorithms for Non-negative Matrix Factorization. In: LEEN, T.; DIETTERICH, T.; TRESP, V. (eds.). *Advances in Neural Information Processing Systems*. MIT Press, 2000, vol. 13.

46. SPANIAS, Andreas; PAINTER, Ted; ATTI, Venkatraman. *Audio Signal Processing and Coding* [online]. John Wiley & Sons, Inc., 2005 [visited on 2024-02-15]. ISBN 978-0-471-79147-8. Available from DOI: 10.1002/0470041978.

47. JANSSEN, A. J. E. M.; VELDHUIS, R. N. J.; VRIES, L. B. Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes. *IEEE Trans. Acoustics, Speech and Signal Processing*. 1986, vol. 34, no. 2, pp. 317–330. Available from DOI: 10.1109/TASSP.1986.1164824.

48. BROCKWELL, Peter J.; DAVIS, Richard A. *Introduction to Time Series and Forecasting*. Springer International Publishing, 2016. ISBN 9783319298542. ISSN 2197-4136. Available from DOI: 10.1007/978-3-319-29854-2.

49. ETTER, W. Restoration of a discrete-time signal segment by interpolation based on the left-sided and right-sided autoregressive parameters. *IEEE Transactions on Signal Processing*. 1996, vol. 44, no. 5, pp. 1124–1135. ISSN 1053-587X. Available from DOI: 10.1109/78.502326.

50. KAUPPINEN, Ismo; KAUPPINEN, Jyrki. Reconstruction Method for Missing or Damaged Long Portions in Audio Signal. *Journal of the Audio Engineering Society*. 2002, vol. 50, no. 7/8, pp. 594–602. Available also from: http://www.aes.org/e-lib/browse.cfm?elib=11068.

51. KAUPPINEN, Ismo; ROTH, Kari. Audio Signal Extrapolation – Theory and Applications. In: *Proceedings of the 5th International Conference on Digital Audio Effects (DAFx-02)*. Hamburg, Germany, 2002, pp. 105–110. Available also from: http://www2.hsu-hh.de/ant/dafx2002/papers/DAFX02_Kauppinen_Roth_signal_extrapolation.pdf.

52. ESQUEF, Paulo Antonio Andrade; VÄLIMÄKI, Vesa; ROTH, Kari; KAUPPINEN, Ismo. Interpolation of long gaps in audio signals using the warped Burg's method. In: *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*. London, UK, 2003, pp. 18–23.

53. VESELÝ, V.; TONNER, J. Sparse Parameter Estimation in Overcomplete Time Series Models. *Austrian Journal of Statistics*. 2006, vol. 35, pp. 371–378. Available also from: http://www.stat.tugraz.at/AJS/ausg062+3/062Vesely.pdf.

54. GIACOBELLO, Daniele; CHRISTENSEN, Mads Graesboll; DAHL, Joachim; JENSEN, Soren Holdt; MOONEN, Marc. Joint estimation of short-term and long-term predictors in speech coders. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009. Available from DOI: 10.1109/icassp.2009.4960532.

55. GIACOBELLO, D.; CHRISTENSEN, M. G.; MURTHI, M. N.; JENSEN, S. H.; MOONEN, M. Sparse Linear Prediction and Its Applications to Speech Processing. *IEEE Transactions on Audio, Speech, and Language Processing*. 2012, vol. 20, no. 5, pp. 1644–1657. Available from DOI: 10.1109/tasl.2012.2186807.

56. DUFERA, Bisrat Derebssa; ENEMAN, Koen; WATERSCHOOT, Toon van. Missing Sample Estimation Based on High-Order Sparse Linear Prediction for Audio Signals. In: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018. Available from DOI: 10.23919/eusipco.2018.8553567.

57. DUFERA, Bisrat Derebssa; ADUGNA, Eneyew; ENEMAN, Koen; WATERSCHOOT, Toon van. Restoration of Click Degraded Speech and Music Based on High Order Sparse Linear Prediction. In: *2019 IEEE AFRICON*. IEEE, 2019. Available from DOI: 10.1109/africon46755.2019.9133792.

58. BOURGUIGNON, Sebastien; NININ, Jordan; CARFANTAN, Herve; MONGEAU, Marcel. Exact Sparse Approximation Problems via Mixed-Integer Programming: Formulations and Computational Performance. *IEEE Transactions on Signal Processing*. 2016, vol. 64, no. 6, pp. 1405–1419. ISSN 1941-0476. Available from DOI: 10.1109/tsp.2015.2496367.

59. DONNE, Diego Delle; KOWALSKI, Matthieu; LIBERTI, Leo. A Novel Integer Linear Programming Approach for Global L0 Minimization. *Journal of Machine Learning Research*. 2023, vol. 24, no. 382, pp. 1–28. Available also from: http://jmlr.org/papers/v24/21-0741.html.

60. PATI, Y.C.; REZAIIFAR, R.; KRISHNAPRASAD, P.S. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*. 1993, pp. 40–44. ISSN 1058-6393.

61. KOWALSKI, M.; SIEDENBURG, K.; DÖRFLER, M. Social Sparsity! Neighborhood Systems Enrich Structured Shrinkage Operators. *Signal Processing, IEEE Transactions on*. 2013, vol. 61, no. 10, pp. 2498–2511. ISSN 1053-587X. Available from DOI: 10.1109/TSP.2013.2250967.

62. LIEB, Florian; STARK, Hans-Georg. Audio inpainting: Evaluation of time-frequency representations and structured sparsity approaches. *Signal Processing*. 2018, vol. 153, pp. 291–299. Available from DOI: 10.1016/j.sigpro.2018.07.012.

63. SIEDENBURG, Kai; KOWALSKI, Matthieu; DÖRFLER, Monika. Audio declipping with social sparsity. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1577–1581.

64. ELAD, Michael; MILANFAR, Peyman; RUBINSTEIN, Ron. Analysis versus synthesis in signal priors. In: *Inverse Problems*. 2007, vol. 23, pp. 947–968.

65. TOUMI, Ichrak; EMIYA, Valentin. Sparse Non-Local Similarity Modeling for Audio Inpainting. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018. Available from DOI: 10.1109/icassp.2018.8462187.

66. ENGAN, K.; AASE, S.O.; HAKON HUSOY, J. Method of optimal directions for frame design. In: *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*. 1999, vol. 5, 2443–2446 vol.5. ISSN 1520-6149. Available from DOI: 10.1109/ICASSP.1999.760624.

67. AHARON, M.; ELAD, M.; BRUCKSTEIN, A. M. K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations. *IEEE Transactions on Signal Processing*. 2006, vol. 54, pp. 4311–4322.

68. RENCKER, Lucas; BACH, Francis; WANG, Wenwu; PLUMBLEY, Mark D. Consistent Dictionary Learning for Signal Declipping. In: *Latent Variable Analysis and Signal Separation*. Springer International Publishing, 2018, pp. 446–455. Available from DOI: 10.1007/978-3-319-93764-9_41.

69. RENCKER, Lucas; BACH, Francis; WANG, Wenwu; PLUMBLEY, Mark D. Sparse Recovery and Dictionary Learning From Nonlinear Compressive Measurements. *IEEE Transactions on Signal Processing*. 2019, vol. 67, no. 21, pp. 5659–5670. Available from DOI: 10.1109/tsp.2019.2941070.

70. LI, Bin; RENCKER, Lucas; DONG, Jing; LUO, Yuhui; PLUMBLEY, Mark D.; WANG, Wenwu. Sparse Analysis Model Based Dictionary Learning for Signal Declipping. *IEEE Journal of Selected Topics in Signal Processing*. 2021, vol. 15, no. 1, pp. 25–36. Available from DOI: 10.1109/JSTSP.2021.3051746.

71. MACH, Václav; OZDOBINSKI, Roman. Optimizing dictionary learning parameters for solving Audio Inpainting problem. *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems*. 2013, vol. 2, no. 1, pp. 40–45. ISSN 1805-5443. Available from DOI: 10.11601/ijates.v2i1.34.

72. TAUBÖCK, Georg; RAJBAMSHI, Shristi; BALAZS, Peter. Dictionary Learning for Sparse Audio Inpainting. *IEEE Journal of Selected Topics in Signal Processing*. 2021, vol. 15, no. 1, pp. 104–119. Available from DOI: 10.1109/jstsp.2020.3046422.

73. RAJBAMSHI, Shristi; TAUBÖCK, Georg; HOLIGHAUS, Nicki; BALAZS, Peter. Audio Inpainting via ℓ_1-Minimization and Dictionary Learning. In: *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021. Available from DOI: 10.23919/eusipco54536.2021.9616132.

74. BAHAT, Yuval; SCHECHNER, Yoav Y.; ELAD, Michael. Self-content-based audio inpainting. *Signal Processing*. 2015, vol. 111, no. 0, pp. 61–72. ISSN 0165-1684. Available from DOI: 10.1016/j.sigpro.2014.11.023.

75. PERRAUDIN, Nathanael; HOLIGHAUS, Nicki; MAJDAK, Piotr; BALAZS, Peter. Inpainting of long audio segments with similarity graphs. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2018, pp. 1–1. Available from DOI: 10.1109/taslp.2018.2809864.

76. DONAHUE, Chris; MCAULEY, Julian; PUCKETTE, Miller. Adversarial Audio Synthesis. In: *International Conference on Learning Representations*. 2019. Available also from: https://arxiv.org/abs/1802.04208.

77. ENGEL, Jesse; AGRAWAL, Kumar Krishna; CHEN, Shuo; GULRAJANI, Ishaan; DONAHUE, Chris; ROBERTS, Adam. GANSynth: Adversarial Neural Audio Synthesis. In: *International Conference on Learning Representations*. 2019. Available also from: https://arxiv.org/abs/1902.08710.

78. MARAFIOTI, Andrés; PERRAUDIN, Nathanaël; HOLIGHAUS, Nicki; MAJDAK, Piotr. Adversarial Generation of Time-Frequency Features with application in audio synthesis. In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, vol. 97, pp. 4352–4362. Proceedings of Machine Learning Research. Available also from: https://proceedings.mlr.press/v97/marafioti19a.html.

79. LEE, Bong-Ki; CHANG, Joon-Hyuk. Packet Loss Concealment Based on Deep Neural Networks for Digital Speech Transmission. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2016, vol. 24, no. 2, pp. 378–387. Available from DOI: 10.1109/taslp.2015.2509780.

80. DIELEMAN, Sander; OORD, Aäron van den; SIMONYAN, Karen. The challenge of realistic music generation: modelling raw audio at scale. In: *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*. arXiv, 2018. Available from DOI: 10.48550/ARXIV.1806.10474.

81. MARAFIOTI, Andrés; PERRAUDIN, Nathanaël; HOLIGHAUS, Nicki; MAJDAK, Piotr. A Context Encoder For Audio Inpainting. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2019, vol. 27, no. 12, pp. 2362–2372. Available from DOI: 10.1109/taslp.2019.2947232.

82. MARAFIOTI, Andres; MAJDAK, Piotr; HOLIGHAUS, Nicki; PERRAUDIN, Nathanaël. GACELA: A Generative Adversarial Context Encoder for Long Audio Inpainting of Music. *IEEE Journal of Selected Topics in Signal Processing*. 2021, vol. 15, no. 1, pp. 120–131. Available from DOI: 10.1109/JSTSP.2020.3037506.

83. LIU, Kaiyang; GAN, Wendong; YUAN, Chenchen. MAID: A Conditional Diffusion Model for Long Music Audio Inpainting. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023. Available from DOI: 10.1109/icassp49357.2023.10095769.

84. MOLINER, Eloi; LEHTINEN, Jaakko; VÄLIMÄKI, Vesa. Solving Audio Inverse Problems with a Diffusion Model. In: *ICASSP 2023 – 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023. Available from DOI: 10.1109/icassp49357.2023.10095637.

85. CHEN, Xiaohan; LIU, Jialin; WANG, Zhangyang; YIN, Wotao. Theoretical Linear Convergence of Unfolded ISTA and Its Practical Weights and Thresholds. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Montréal, Canada: Curran Associates Inc., 2018, pp. 9079–9089. NIPS'18.

86. VENKATAKRISHNAN, S. V.; BOUMAN, C. A.; WOHLBERG, B. Plug-and-Play priors for model based reconstruction. In: *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013. Available from DOI: 10.1109/globalsip.2013.6737048.

87. VIAL, Pierre-Hugo; MAGRON, Paul; OBERLIN, Thomas; FEVOTTE, Cedric. Learning the Proximity Operator in Unfolded ADMM for Phase Retrieval. *IEEE Signal Processing Letters*. 2022, vol. 29, pp. 1619–1623. Available from DOI: 10.1109/lsp.2022.3189275.

88. TANAKA, Tomoro; YATABE, Kohei; YASUDA, Masahiro; OIKAWA, Yasuhiro. APPLADE: Adjustable Plug-and-Play Audio Declipper Combining DNN with Sparse Optimization. In: *ICASSP 2022 – 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022. Available from DOI: 10.1109/icassp43922.2022.9747089.

89. BILEN, Çağdaş; OZEROV, Alexey; PÉREZ, Patrick. Audio declipping via nonnegative matrix factorization. In: *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2015, pp. 1–5. Available from DOI: 10.1109/WASPAA.2015.7336948.

90. HUBER, R.; KOLLMEIER, B. PEMO-Q—A New Method for Objective Audio Quality Assessment Using A Model of Auditory Perception. *IEEE Trans. Audio Speech Language Proc.* 2006, vol. 14, no. 6, pp. 1902–1911. Available from DOI: 10.1109/TASL.2006.883259.

91. THIEDE, Thildo; TREURNIET, William C.; BITTO, Roland; SCHMIDMER, Christian; SPORER, Thomas; BEERENDS, John G.; COLOMES, Catherine; KEYHL, Michael; STOLL, Gerhard; BRANDENBURG, Karlheinz; FEITEN, Bernhard. PEAQ – The ITU standard for objective measurement of perceived audio quality. *Journal of the Audio Engineering Society*. 2000, vol. 48, no. 1/2, pp. 3–29. Available also from: http://www.aes.org/e-lib/browse.cfm?elib=12078.

92. KABAL, P. *An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality*. 2002-05. Tech. rep. MMSP Lab Technical Report, Dept. Electrical & Computer Engineering, McGill University.

93. RAJMIC, Pavel; BARTLOVÁ, Hana; PRŮŠA, Zdeněk; HOLIGHAUS, Nicki. Acceleration of audio inpainting by support restriction. In: *2015 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, 2015, pp. 325–329. Available from DOI: 10.1109/icumt.2015.7382451.

94. CANDES, Emmanuel J.; WAKIN, Michael B.; BOYD, Stephen P. Enhancing Sparsity by Reweighted ℓ_1 Minimization. *Journal of Fourier Analysis and Applications*. 2008, vol. 14, pp. 877–905.

95. KITIĆ, Srđan; BERTIN, Nancy; GRIBONVAL, Rémi. Sparsity and cosparsity for audio declipping: a flexible non-convex approach. In: *LVA/ICA 2015 – The 12th International Conference on Latent Variable Analysis and Signal Separation*. Liberec, Czech Republic, 2015, pp. 243–250.

96. CHERFAOUI, Farah; EMIYA, Valentin; RALAIVOLA, Liva; ANTHOINE, Sandrine. Recovery and Convergence Rate of the Frank–Wolfe Algorithm for the m-Exact-Sparse Problem. *IEEE Transactions on Information Theory*. 2019, vol. 65, no. 11, pp. 7407–7414. ISSN 1557-9654. Available from DOI: 10.1109/tit.2019.2919640.

97. CHRISTENSEN, Ole. *Frames and Bases, An Introductory Course*. Boston: Birkhäuser, 2008. ISBN 9780817646776.

98. ZÁVIŠKA, Pavel; RAJMIC, Pavel; OZEROV, Alexey; RENCKER, Lucas. A Survey and an Extensive Evaluation of Popular Audio Declipping Methods. *IEEE Journal of Selected Topics in Signal Processing*. 2021, vol. 15, no. 1, pp. 5–24. Available from DOI: 10.1109/JSTSP.2020.3042071.

99. DEMPSTER, Arthur P; LAIRD, Nan M; RUBIN, Donald B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*. 1977, vol. 39, no. 1, pp. 1–22.

100. GOODMAN, N. R. Statistical Analysis Based on a Certain Multivariate Complex Gaussian Distribution (An Introduction). *The Annals of Mathematical Statistics*. 1963, vol. 34, no. 1, pp. 152–177. ISSN 0003-4851. Available from DOI: 10.1214/aoms/1177704250.

101. TSE, David. *Fundamentals of wireless communication*. Ed. by VISWANATH, Pramod. Cambridge: Cambridge University Press, 2012. ISBN 9780511807213. Title from publisher's bibliographic system (viewed on 05 Oct 2015).

102. GRAY, Robert M.; DAVISSON, Lee D. *An Introduction to Statistical Signal Processing*. Cambridge University Press, 2004. ISBN 9780521131827. Available from DOI: 10.1017/cbo9780511801372.

103. KAY, Steven M. *Fundamentals of Statistical Processing, Volume I: Estimation Theory*. Prentice Hall, 1993. ISBN 0133457117.

104. *EBU SQAM CD: Sound Quality Assessment Material recordings for subjective tests* [online]. 2008. Available also from: https://tech.ebu.ch/publications/sqamcd.

105. *Sound Quality Assessment Material recordings for subjective tests*. Geneva: European Broadcasting Union, 2008. Available also from: https://tech.ebu.ch/docs/tech/tech3253.pdf. EBU – TECH 3253.

106. HARRIS, F.J. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*. 1978, vol. 66, no. 1, pp. 51–83. Available from DOI: 10.1109/proc.1978.10837.

107. MOLINER, E.; VÄLIMÄKI, V. *Diffusion-Based Audio Inpainting*. 2023-09. Tech. rep. Available from DOI: 10.48550/arXiv.2305.15266.

108. MIOTELLO, F.; PEZZOLI, M.; COMANDUCCI, L.; ANTONACCI, F.; SARTI, A. Deep Prior-Based Audio Inpainting Using Multi-Resolution Harmonic Convolutional Neural Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2023, pp. 1–11. ISSN 2329-9304. Available from DOI: 10.1109/TASLP.2023.3324556.

109. OUDRE, Laurent. Interpolation of Missing Samples in Sound Signals Based on Autoregressive Modeling. *Image Processing On Line*. 2018, vol. 8, pp. 329–344. Available from DOI: 10.5201/ipol.2018.23.

110. ZÁVIŠKA, P.; RAJMIC, P.; SCHIMMEL, J. Psychoacoustically Motivated Audio Declipping Based on Weighted l1 Minimization. In: *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*. Budapest, Hungary, 2019, pp. 338–342. ISSN null. Available from DOI: 10.1109/TSP.2019.8769109.

111. NECCIARI, T.; BALAZS, P.; HOLIGHAUS, N.; SØNDERGAARD, P. L. The ERBlet transform: An auditory-based time-frequency representation with perfect reconstruction. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013, pp. 498–502. ISSN 1520-6149. Available from DOI: 10.1109/ICASSP.2013.6637697.

112. ULYANOV, Dmitry; VEDALDI, Andrea; LEMPITSKY, Victor. Deep Image Prior. *International Journal of Computer Vision*. 2020, vol. 128, no. 12. ISSN 1573-1405. Available from DOI: 10.1007/s11263-020-01303-4.

113. MONGA, Vishal; LI, Yuelong; ELDAR, Yonina C. Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing. *IEEE Signal Processing Magazine*. 2021, vol. 38, no. 2, pp. 18–44. Available from DOI: 10.1109/MSP.2020.3016905.

114. GOODFELLOW, Ian; POUGET-ABADIE, Jean; MIRZA, Mehdi; XU, Bing; WARDE-FARLEY, David; OZAIR, Sherjil; COURVILLE, Aaron; BENGIO, Yoshua. Generative Adversarial Nets. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014, vol. 27.

115. BILEN, Çağdaş; OZEROV, Alexey; PÉREZ, Patrick. Solving Time-Domain Audio Inverse Problems Using Nonnegative Tensor Factorization. *IEEE Transactions on Signal Processing*. 2018, vol. 66, no. 21, pp. 5604–5617. ISSN 1053-587X. Available from DOI: 10.1109/TSP.2018.2869113.

116. VIAL, P.-H.; MAGRON, P.; OBERLIN, T.; FÉVOTTE, C. Phase retrieval with Bregman divergences and application to audio signal recovery. *IEEE Journal of Selected Topics in Signal Processing*. 2021, vol. 15, no. 1, pp. 51–64. Available from DOI: 10.1109/JSTSP.2021.3051870.

117. FÉVOTTE, C.; KOWALSKI, M. Estimation With Low-Rank Time–Frequency Synthesis Models. *IEEE Transactions on Signal Processing*. 2018, vol. 66, no. 15, pp. 4121–4132.

List of abbreviations

ADMM	Alternating Direction Method of Multipliers
AR	autoregressive
CPA	Chambolle–Pock algorithm
DCT	discrete cosine transform
DFT	discrete Fourier transform
DRA	Douglas–Rachford algorithm
DSP	digital signal processing
EM	expectation–maximization
MLE	maximum likelihood estimation
MSE	mean squared error
NMF	non-negative matrix factorization
ODG	objective difference grade
OMP	Orthogonal Matching Pursuit
PDF	probability density function
PEAQ	Perceptual Evaluation of Audio Quality
PSM	perceptual similarity measure
ROC	relative objective change
RSC	relative solution change
SDR	signal-to-distortion ratio
SNR	signal-to-noise ratio
SPADE	Sparse Audio Declipper
SPAIN	Sparse Audio Inpainter
A-SPAIN	Analysis-based Sparse Audio Inpainter
S-SPAIN	Synthesis-based Sparse Audio Inpainter
STFT	short-time Fourier transform

List of symbols and notation

Symbol	Description	Link
	General notation	
$\mathbb{R}$, $\mathbb{C}$	real, complex numbers	
$\mathbb{F}$	any number field, usually $\mathbb{R}$ or $\mathbb{C}$	
$\Re(c)$, $\Im(c)$, $\bar{c}$	real and imaginary part of a number c, complex conjugate	
$\operatorname{sgn}(c)$	sign function, $\operatorname{sgn}(c) = \frac{c}{\lvert c\rvert}$ for $c \in \mathbb{R}$ or $c \in \mathbb{C}$, except for $\operatorname{sgn}(0) = 0$	
$\exp(c)$	exponential function, alias to e^c for $c \in \mathbb{R}$ or $c \in \mathbb{C}$	
$\lfloor r \rfloor$, $\lceil r \rceil$	floor, ceiling of a real number r	
(a, b), $[a, b]$	open, closed interval from a to b, $a, b \in \mathbb{R}$ (half-open intervals analogously)	
$\lvert C\rvert$	cardinality of the set C (not to be confused with absolute value of a number)	
$\operatorname{ri}(C)$	relative interior of the set C	
$\lVert \cdot \rVert$	norm on any vector space, including the operator norm	1.2, 1.14
$\lVert \cdot \rVert_p$	p-norm of a vector	1.3
$\lVert \cdot \rVert_\mathrm{F}$	Frobenius norm of a matrix	1.3
$\langle \cdot, \cdot \rangle$	inner product of two vectors	1.4
$\delta_{i,j}$	Kronecker delta, $\delta_{i,j} = 1$ for $i = j$, otherwise 0	
$\mathcal{R}(T)$	range space of an operator T	1.13
T^*, T^{-1}, T^+	adjoint, inverse, pseudoinverse of an operator T	1.15, 1.16
$f \circ g$	composition of functions or operators, $(f \circ g)(x) = f(g(x))$	
	Vectors and matrices	
$\mathbf{x}$, $\mathbf{x}(n) = x_n$	vector $\mathbf{x} = [x_1, \dots, x_N]^\top \in \mathbb{F}^N$, its n-th entry (scalar component)	1.1
$\mathbf{x}(M)$	selection by an index set, e.g., $\mathbf{x}(\{m, n\}) = [x_m, x_n]^\top$	1.1
$\operatorname{supp}(\mathbf{x})$	support of a vector, i.e., the set of indices of its non-zero entries, $\operatorname{supp}\mathbf{x} = \{n \mid x_n \neq 0\}$	
$\mathbf{A}^\top$, $\mathbf{A}^*$	transpose, Hermitian transpose of the matrix $\mathbf{A}$	
$\odot$	Hadamard (entrywise) product of vectors or matrices	
$\mathbf{A}^{\odot[n]}$	entrywise n-th power of the matrix $\mathbf{A}$	
$\det(\mathbf{A})$	determinant of the matrix $\mathbf{A}$	
$\operatorname{diag}(\mathbf{x})$	diagponal matrix from the vector $\mathbf{x} \in \mathbb{F}^N$, $\mathbf{A} = \operatorname{diag}(\mathbf{x}) \in \mathbb{F}^{N \times N}$ with $a_{nn} = x_n$ and $a_{mn} = 0$ for $m \neq n$	
$d_\mathrm{IS}(a \mid b)$	Itakura–Saito divergence between numbers a and b	(1.34)
$D_\mathrm{IS}(\mathbf{A} \mid \mathbf{B})$	Itakura–Saito divergence between matrices $\mathbf{A}$ and $\mathbf{B}$	(1.34)

Symbol	Description	Link
	Optimization	
inf, sup	infimum, supremum of a set or a function	
$\mathcal{D}(f)$	domain of a function $f\colon V \to \mathbb{R} \cup \{\infty\}$, $\mathcal{D}(f) = \{u \in V \mid f(u) < \infty\} \subseteq V$	
f^*	convex conjugate function	1.20
$\mathcal{F}(V)$	set of lower semicontinuous convex functions on V with non-empty domain	
I_C	indicator function of a set C, $I_C(x) = 0$ for $x \in C$, otherwise ∞	
prox_f	proximal operator of a function f	1.22
proj_C	projection onto a set C	(1.23)
soft_τ	soft thresholding with threshold τ	(1.24)
hard_k	hard thresholding, keeping k largest entries	
	Probability	
$p(\mathbf{x})$	probability density function	
$p(\mathbf{x} \mid \mathbf{y})$	conditional probability density function in $\mathbf{x}$, given $\mathbf{y}$	
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	(multivariate) Gaussian (normal) distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$	5.4
$\mathbb{E}$	expected value	
	Notation used in the interpolation tasks	
$\mathbf{x}^{\text{true}} \in \mathbb{R}^L$	original (non-degraded) signal	
M	set of indices of reliable (non-degraded) samples (entries)	
$\overline{M}$	set of indices of missing (degraded) samples (entries), $\overline{M} = \{1, \dots, L\} \setminus M$	
$\mathbf{x}^{\text{obs}}$	observed (non-degraded) part of the signal $\mathbf{x}^{\text{true}} = \mathbf{x}^{\text{true}}(M)$	
$\mathit{\Gamma}$	set of feasible interpolated signals, $\mathit{\Gamma} = \{\mathbf{x} \in \mathbb{R}^L \mid \mathbf{x}(M) = \mathbf{x}^{\text{obs}}\}$	(2.1)

List of appendices

A Practical remarks on methods based on low-rank representations

This appendix comments on a few practical details encountered in the implementation of the algorithms derived in chapter 5. The discussed technicalities did not appear in the published article [8].

A.1 Ensuring conjugate symmetry of the spectra

If the transforms $\mathbf{L}$ and $\mathbf{T} = \mathbf{L}^*$ correspond to the DFT and its inverse, respectively, and the audio signal is real, we should ensure conjugate symmetry of the spectrum $\mathbf{c}_n$ in each temporal frame n. Otherwise, we cannot guarantee that the solution of the interpolation problem is a real signal. This means we need to relax the independence in assumption 5.3, since the values of the (real, non-negative) matrix $\mathbf{V} = \mathbf{WH}$ need to be symmetric in the frequency direction.

Note that the conjugate symmetry of the spectra in the context of NMF is mentioned in the work by Bilen et al. [115, Sec. III.E]. There, it is proposed to enforce symmetry by updating the matrix $\mathbf{W}$, similarly to enforcing signal constraints as mentioned in remark 5.6. A less detailed discussion is also present in [116, Sec. III.D], whereas [117, Sec. III] proposes both complex and real formulations of a problem related to NMF.

Without modifying the model and the whole derivation, we can rely on the fact that *a frequency-symmetric initialization of the matrix* $\mathbf{W}$ *results in real temporal solution*, which can be demonstrated as follows.

1. First, consider $\hat{\mathbf{c}}_n$ defined by (5.14a) and $\hat{\mathbf{c}}_n^{\mathrm{alt}}$ from equation (5.25). If $\mathbf{D}_n = \operatorname{diag}\left([v_{fn}]_{f=1,\dots,F}\right)$ with $\mathbf{V} = [v_{fn}] = \mathbf{WH}$ is a diagonal matrix with its (real, non-negative) diagonal displaying the desired symmetry, which can be ensured by the initialization, we can show that:
   (a) $\mathbf{TD}_n\mathbf{T}^*$ is a real matrix. Denote $\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_F]$, where $\mathbf{t}_1, \dots, \mathbf{t}_F$ are column vectors which are either real or $\mathbf{t}_{f_1}$ is complex conjugate to $\mathbf{t}_{f_2}$ for some $f_1, f_2 \in \{1, \dots, F\}$. We shall denote this complex conjugacy as $\mathbf{t}_{f_1} = \overline{\mathbf{t}_{f_2}}$, meaning that the vectors are entrywise complex conjugate. With this notation, we have

$$\mathbf{TD}_n\mathbf{T}^* = \sum_{f=1}^{F} v_{fn}\mathbf{t}_f\mathbf{t}_f^*. \tag{A.1}$$

   Since for each matrix $\mathbf{t}_{f_1}\mathbf{t}_{f_1}^*$, there is a complex conjugate matrix $\mathbf{t}_{f_2}\mathbf{t}_{f_2}^*$ present in the sum, the whole sum is real.

(b) It directly follows that $\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top$ is real matrix, thus the inverse $\left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}$ is also real and $\left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}\mathbf{x}_n^{\mathrm{obs}}$ is a real vector.

(c) Now multiplication with $\mathbf{M}_n^\top$ realizes the imputation of zeros in places of the missing samples, and $\mathbf{T}^* = \mathbf{L}$ is the DFT, thus the frequency coefficients $\mathbf{T}^*\mathbf{M}_n^\top\left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}\mathbf{x}_n^{\mathrm{obs}}$ are conjugate symmetric, as desired.

(d) Finally, the multiplication with $\mathbf{D}_n$ does not break the symmetry, since the diagonal itself is symmetric.

(e) In case of $\hat{\mathbf{c}}_n^{\mathrm{alt}}$ in EM-*t*, the multiplication with $\mathbf{LT}$ only realizes inverse DFT (creating a real signal), followed by DFT, which again results in a conjugate symmetric coefficients.

2. Now focus on $\hat{\mathbf{\Sigma}}_n$. Clearly this is a Hermitian matrix, thus its diagonal is real, and we only need the diagonal in subsequent computations of both EM-*tf* and EM-*t*. We need to ensure that it is also symmetric:

(a) It holds

$$\hat{\mathbf{\Sigma}}_n = \mathbf{D}_n - \mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}\mathbf{M}_n\mathbf{T}\mathbf{D}_n \tag{A.2}$$

$$= \mathbf{D}_n - \begin{bmatrix} v_{1n}\mathbf{t}_1^* \\ \vdots \\ v_{Fn}\mathbf{t}_F^* \end{bmatrix}\mathbf{M}_n^\top\left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}\mathbf{M}_n\begin{bmatrix} v_{1n}\mathbf{t}_1 & \dots & v_{Fn}\mathbf{t}_F \end{bmatrix}, \tag{A.3}$$

which leads to the expression for the diagonal entries

$$\hat{\mathbf{\Sigma}}_n(f,f) = v_{fn} - (v_{fn}\mathbf{t}_f^*)\mathbf{M}_n^\top\left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}\mathbf{M}_n(v_{fn}\mathbf{t}_f). \tag{A.4}$$

(b) For a pair f_1, f_2 such that $\mathbf{t}_{f_1} = \overline{\mathbf{t}_{f_2}}$ and $v_{f_2n} = v_{f_1n}$, it holds

$$\hat{\mathbf{\Sigma}}_n(f_2,f_2) = v_{f_2n} - (v_{f_2n}\mathbf{t}_{f_2}^*)\mathbf{M}_n^\top\left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}\mathbf{M}_n(v_{f_2n}\mathbf{t}_{f_2}) \tag{A.5}$$

$$= v_{f_1n} - (v_{f_1n}\mathbf{t}_{f_1}^\top)\mathbf{M}_n^\top\left(\mathbf{M}_n\mathbf{T}\mathbf{D}_n\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1}\mathbf{M}_n(v_{f_1n}\overline{\mathbf{t}_{f_1}}) \tag{A.6}$$

$$= \overline{\hat{\mathbf{\Sigma}}_n(f_1,f_1)} = \hat{\mathbf{\Sigma}}_n(f_1,f_1), \tag{A.7}$$

since the diagonal is real.

3. It remains to show that the multiplicative update of $\mathbf{W}$ does not break its symmetry in frequency direction. This is straightforward since this property is kept by entrywise operations and also by multiplication with a matrix from the right-hand side.

A.2 Implementing the multiplicative updates

As mentioned in remark 1.26, division by zero can be avoided by introducing a minor offset in the Itakura–Saito divergence.

In the alternating minimization approach, this can be back-tracked to a relaxed problem where it is assumed (compare to (5.3))

$$\mathbf{c}_n^{\text{true}} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{D}_n + \varepsilon\mathbf{I}\right), \quad \mathbf{D}_n = \operatorname{diag}\left(\left[v_{fn}\right]_{f=1,\dots,F}\right). \tag{A.8}$$

Consider now the approach of section 5.3 and the AM algorithm. Taking into account the equation for the objective function (5.31), this results in maximizing

$$p\left(\mathbf{x}_n \mid \mathbf{W}, \mathbf{H}\right) = \det\left(\pi\mathbf{T}(\mathbf{D}_n + \varepsilon\mathbf{I})\mathbf{T}^*\right)^{-1} \exp\left(-(\mathbf{x}_n)^\top \left(\mathbf{T}(\mathbf{D}_n + \varepsilon\mathbf{I})\mathbf{T}^*\right)^{-1} \mathbf{x}_n\right), \tag{A.9}$$

which is further equivalent to minimizing the objective

$$-\log p\left(\mathbf{x}_n \mid \mathbf{W}, \mathbf{H}\right) = \log\det\left(\pi\mathbf{T}(\mathbf{D}_n + \varepsilon\mathbf{I})\mathbf{T}^*\right) + (\mathbf{x}_n)^\top \left(\mathbf{T}(\mathbf{D}_n + \varepsilon\mathbf{I})\mathbf{T}^*\right)^{-1} \mathbf{x}_n. \tag{A.10}$$

It means that the term $(\mathbf{D}_n + \varepsilon\mathbf{I})$ should replace $\mathbf{D}_n$ also in the signal update in the AM algorithm.

We arrive at a similar result while developing the EM-*tf* and EM-*t* algorithms from the assumption in equation (A.8). The term $(\mathbf{D}_n + \varepsilon\mathbf{I})$ appears in the whole derivation instead of $\mathbf{D}_n$. The change projects itself also in the multiplicative updates of $\mathbf{W}$ and $\mathbf{H}$, as desired. The resulting objective function for this case reads

$$\begin{aligned} p\left(\mathbf{x}_n^{\text{obs}} \mid \mathbf{W}, \mathbf{H}\right) &= \det\left(\pi\mathbf{M}_n\mathbf{T}(\mathbf{D}_n + \varepsilon\mathbf{I})\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1} \\ &\quad \cdot \exp\left(-(\mathbf{x}_n^{\text{obs}})^\top(\mathbf{M}_n\mathbf{T}(\mathbf{D}_n + \varepsilon\mathbf{I})\mathbf{T}^*\mathbf{M}_n^\top)^{-1}\mathbf{x}_n^{\text{obs}}\right), \end{aligned} \tag{A.11}$$

or, in the negative logarithmic form,

$$\begin{aligned} -\log p\left(\mathbf{x}_n^{\text{obs}} \mid \mathbf{W}, \mathbf{H}\right) &= \log\det\left(\pi\mathbf{M}_n\mathbf{T}(\mathbf{D}_n + \varepsilon\mathbf{I})\mathbf{T}^*\mathbf{M}_n^\top\right) \\ &\quad + (\mathbf{x}_n^{\text{obs}})^\top \left(\mathbf{M}_n\mathbf{T}(\mathbf{D}_n + \varepsilon\mathbf{I})\mathbf{T}^*\mathbf{M}_n^\top\right)^{-1} \mathbf{x}_n^{\text{obs}}. \end{aligned} \tag{A.12}$$

B Implementation

The research described in this thesis has been realized using the Matlab software. The implementation of the algorithms developed in chapters 4 and 5 is available through the following links, related to the respective publications:

- Introducing SPAIN (SParse Audio INpainter) [6]:
  `https://www.utko.fekt.vut.cz/~rajmic/software/SPAIN.zip`
- Audio Inpainting: Revisited and Reweighted [4]:
  `https://github.com/ondrejmokry/InpaintingRevisited`
- Algorithms for audio inpainting based on probabilistic nonnegative matrix factorization [8]:
  `https://github.com/ondrejmokry/InpaintingNMF`

Regarding the gap generation procedure, the aforementioned source codes may contain minor inconsistencies (gap lengths, randomized positions etc.). For this reason, a repository with pre-generated degraded signals is available at `https://github.com/ondrejmokry/TestSignals` for future use.

C Complete results from the state-of-the-art study

In this appendix, figures C.1, C.3 and C.4 show the evaluation of the interpolation algorithms per each signal of the dataset individually, i.e., the data that generated the mean values plotted in figure 6.8.

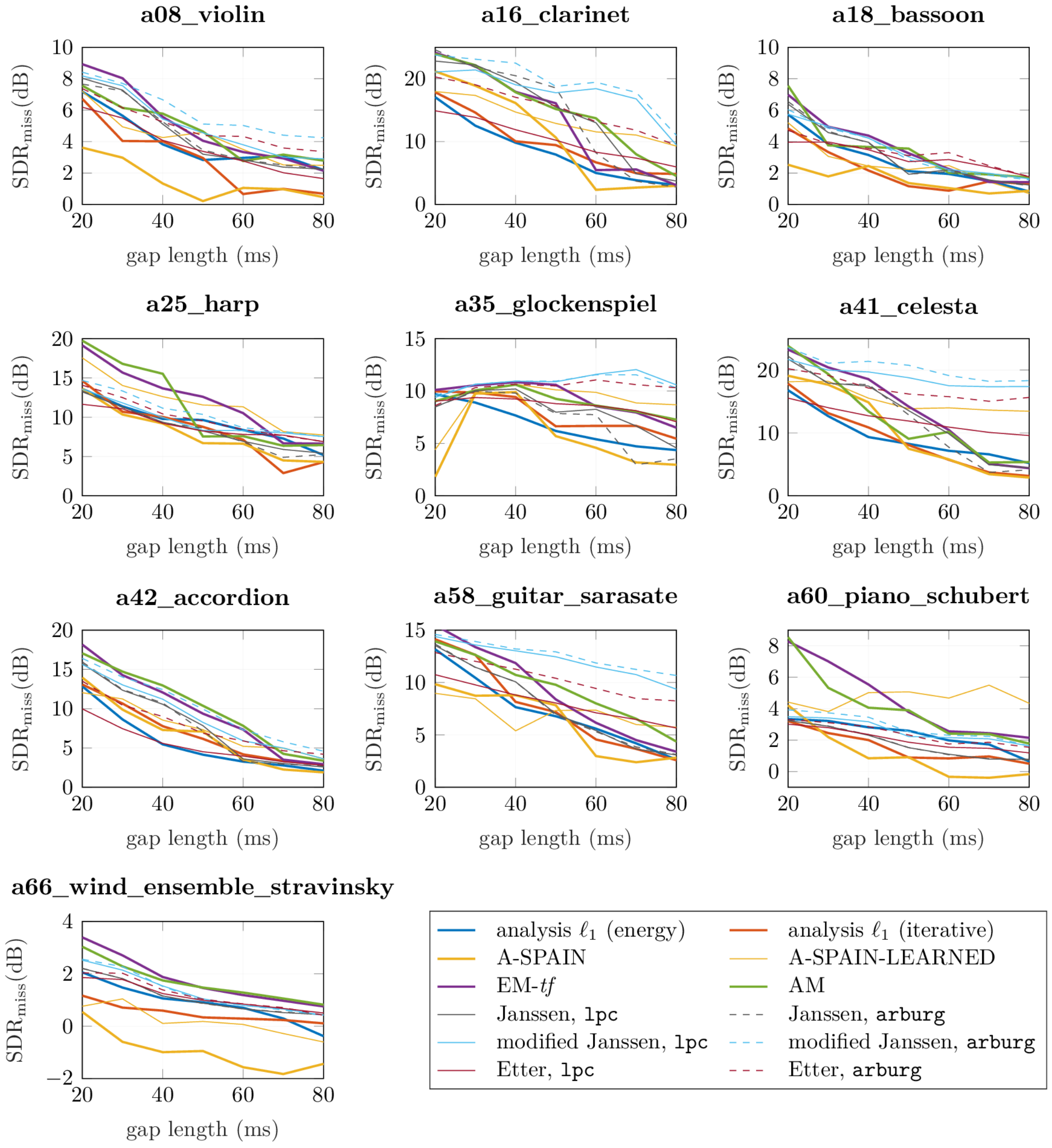


Fig. C.1: State-of-the-art comparison – complete SDR values. Note that the vertical axis is adjusted to the data in the individual plots.

The SDR values in figure C.1 reveal several appealing facts. For example, the lead of the AR-based methods for gaps above 50 ms (especially modified Janssen, Etter) appears to come not only from the sparse signals (clarinet, glockenspiel, celesta), but also from the guitar sample. On the other hand, the NMF-based methods dominate in case of some of the polyphonic instruments (harp, piano) and the complex recording (wind ensemble).

A noticeable outlier is the results of A-SPAIN and A-SPAIN-LEARNED in the case of glockenspiel and gap length 20 ms. This particular case appears to be affected by the fact that one of the gaps ends in the middle of an onset, as seen in figure C.2. The offset is than excessively represented in the interpolated gap. For a slightly longer gap, this onset is lost completely, thus it does not have any negative effect on the interpolation.

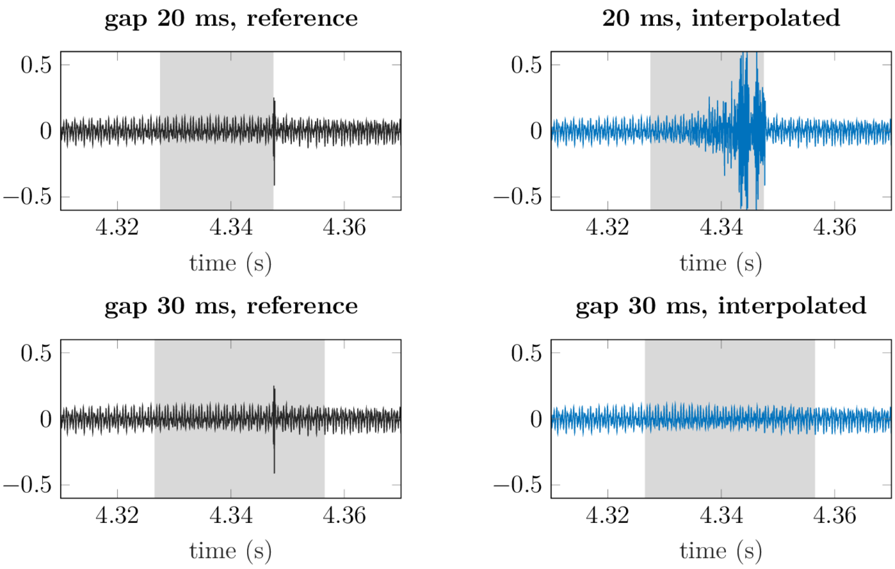


Fig. C.2: Example of a problematic interpolation using A-SPAIN. The signal under test is a35_glockenspiel, and the focus is on a single gap position which caused counter-intuitive results. As in figure 6.9, the gray area represents the sample dropout..

Regarding the psychoacoustically motivated metrics, PEMO-Q mostly aligns with the results in terms of SDR, see figure C.3. The mean values in figure 6.8 indicate the largest difference between SDR and ODG in the case of iteratively weighted ℓ_1 relaxation and A-SPAIN. Figure C.3 confirms that these differences are observed consistently in the whole dataset.

The ODG values computed using PEAQ in figure C.4 indicate that this algorithm is not very sensitive to the differences in the interpolated audio. A notable exception is the piano recording, for which the interpolation algorithms scored across the whole range of ODG. For the rest of the signals, PEAQ did not indicate large differences between the algorithms for gap lengths below 60 ms, except for the interatively weighted ℓ_1 relaxation. However, with respect to the whole ODG scale, the observed differences in terms of PEAQ are only minor.

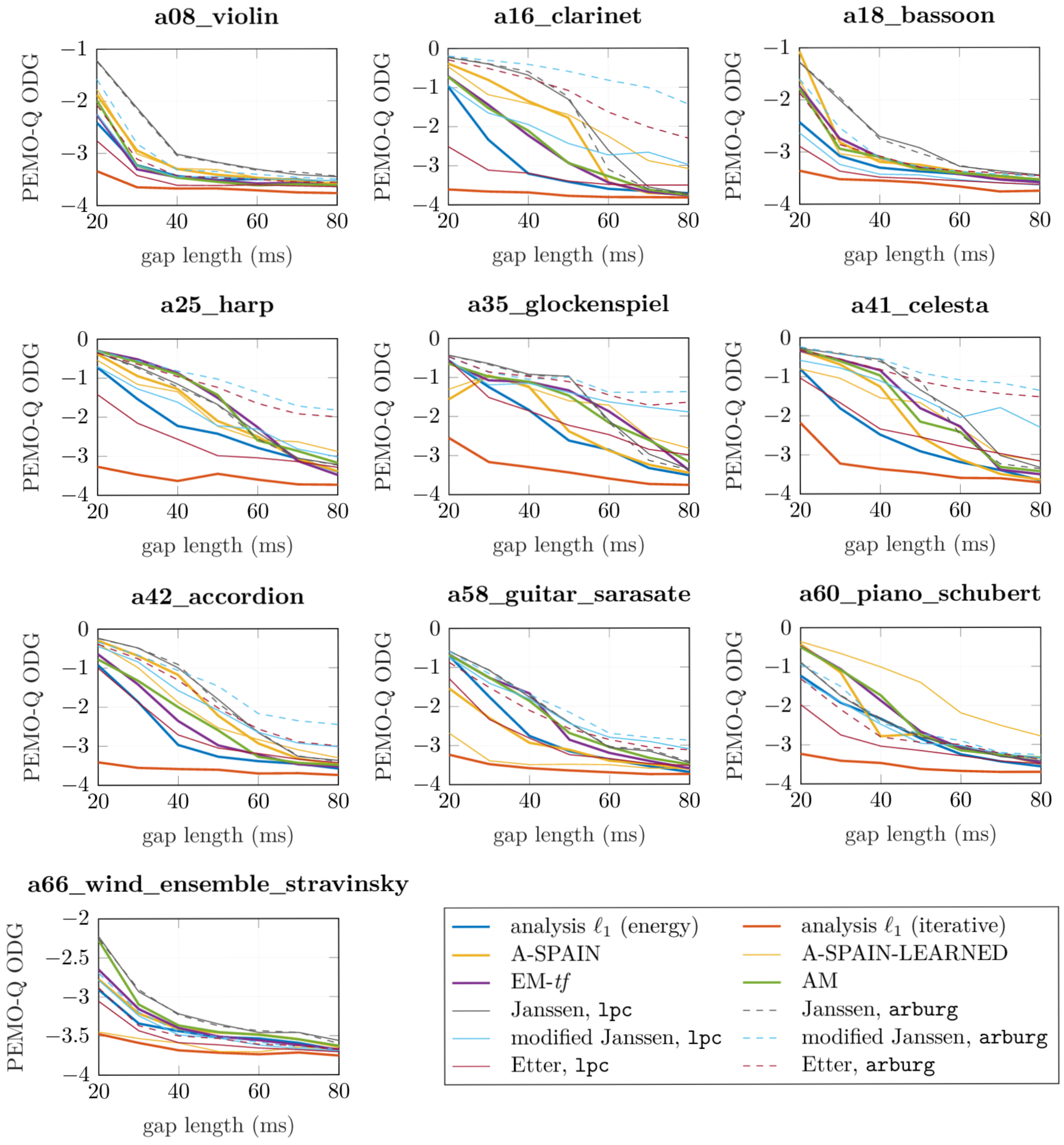


Fig. C.3: State-of-the-art comparison – complete ODG values via PEMO-Q. Note that the vertical axis is cropped in case of a08_violin, a18_bassoon and a66_wind_ensemble_stravinsky

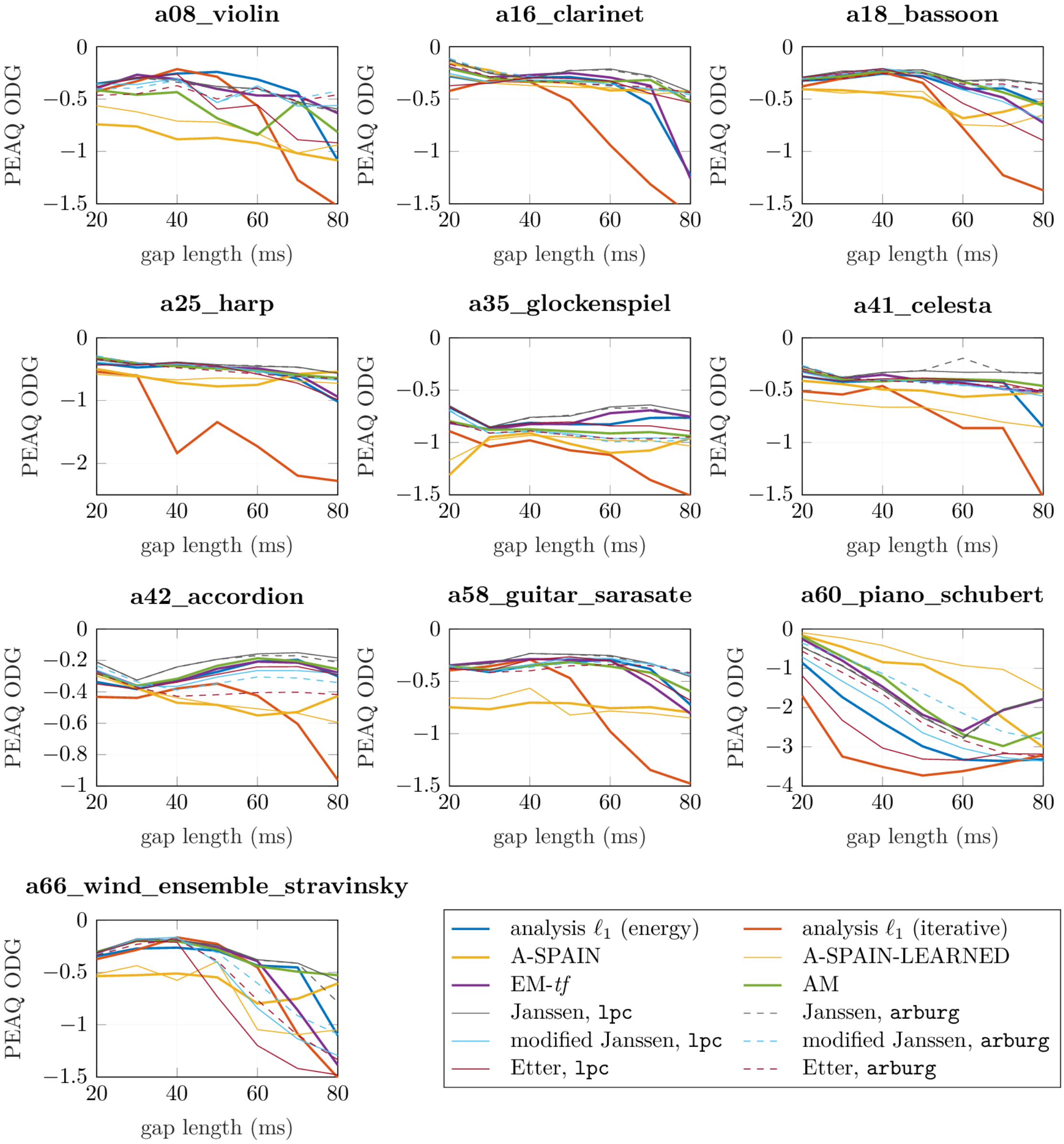


Fig. C.4: State-of-the-art comparison – complete ODG values via PEAQ. Note that the vertical axis covers the whole ODG scale only in the case of a60_piano_schubert.