Audio Inpainting with Sparse Representation and Multilayered Expansion

Eto Sun

Department of Music Research
Schulich School of Music
McGill University
Montréal, Canada

December 2023

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Arts.

Abstract

This thesis proposes a novel hybrid audio inpainting approach that takes into account the diversity of signals and increases the quality of the reconstruction. Audio inpainting aims at recovering missing or degraded parts of an audio signal based on its reliable segments. The degradations arise in various real-world scenarios, such as impulsive noise, clipping, transmission errors, and physical damage of storage medium. Existing inpainting approaches suffer from several limitations, such as energy drop in the reconstructed signal, especially for longer gaps, and poor reconstruction quality for non-stationary signals, such as signals with modulations, or transients, and for noisy signals.

In the proposed approach, left and right reliable neighborhoods around the gap are used to reconstruct the signal of the gap. Based on the fact that an audio signal can be considered as a mixture of three components: sinusoids (or tonal), transients, and noise, each pair of reliable neighborhoods is divided into these components using a structural sparse decomposition technique for subsequent analysis. The gap is reconstructed by extrapolating the parameters estimated from the reliable neighborhoods of each component. The estimated parameters of each component are extracted separately based on their own acoustic characteristics. Many methods used in this thesis, such as the structural sparse decomposition, partial tracking, and extrapolation algorithms, are refined for more robust inpainting results.

A series of experiments are conducted to evaluate the performance of the various stages of the hybrid approach and compare it with other state-of-the-art inpainting approaches on both synthesized and real audio signals, considering various evaluation metrics such as signal-to-noise ratio and objective difference grade. The results demonstrate the hybrid approach to achieve high-quality reconstruction and low computational complexity across a wide range of gap lengths and signal types, especially for longer gaps (longer than 50 ms) and stationary signals.

Résumé

Cette thèse propose une nouvelle approche hybride de restoration audio (*Audio Inpainting*) qui prend en compte la diversité des signaux sonore et améliore la qualité de la reconstruction. La restoration audio vise à reconstituer les parties manquantes ou dégradées d'un signal audio à partir de segments fiables. Dans les situations réelles les dégradations sont du type bruit impulsif, écrêtage, erreurs de transmission ou dûes à la détérioration physique de supports de stockage. Les approches existantes de la restoration audio souffrent de plusieurs limitations, telles que la chute d'énergie dans les signaux reconstruits, surtout pour les plus longues brèches, et la mauvaise qualité de reconstruction de signaux non stationnaires, tels que des signaux modulés ou transitoires, ainsi que de signaux bruités.

Dans l'approche proposée, les voisinages fiables gauche et droit autour de la brèche sont utilisés pour reconstruire le signal dans la brèche. En se fondant sur le fait qu'un signal audio peut être considéré comme un mélange de trois composantes : sinusoïdes (ou tons), transitoires et bruits, chaque paire de voisinage fiable est décomposé en ces composantes à l'aide d'une technique de décomposition parcimonieuse structurée pour une analyse ultérieure. La brèche est reconstruite en extrapolant les paramètres estimés de chaque composante sur ces voisinages fiables. Les paramètres estimés de chaque composante sont extraits séparément en fonction de leurs propres caractéristiques acoustiques. De nombreuses méthodes utilisées dans cette thèse, telles que la décomposition parcimonieuse structurée, le suivi partiel et les algorithmes d'extrapolation, sont affinées afin d'obtenir des résultats de restoration audio plus robustes.

Une série d'expériences est menée pour évaluer la performance des différentes étapes de l'approche hybride ainsi que la comparer avec d'autres approches tirées de l'état de l'art de la restoration audio sur des signaux tant synthétisés que réels, et en considérant diverses mesures d'évaluation telles que le rapport signal sur bruit et la note différentielle objective. Les résultats montrent que l'approche hybride permet d'obtenir une reconstruction de haute qualité pour une large gamme de longueurs et de types de brèches avec une bonne préservation de l'énergie et une extension de partiels.

Acknowledgements

First of all, I would like to express my great appreciation to my supervisor, Professor Philippe Depalle. Thank you for your invaluable mentorship, from my ignorance of the audio signal processing field to my determination to pursue a Ph.D. in this field. I am neither a good analyzer of intricate knowledge nor a good synthesizer of remarkable eurekas, but you always have a way to decompose the knowledge into comprehensible components and help me to reconnect the fragmented ideas into purposeful trajectories. Thank you for your immense patience and support, especially during the last months of thesis writing. Sometimes I feel that I am physically and mentally trapped on the Penrose stairs in the processes of model design and thesis writing. You always help me out of this endless fine-tuning loop with new perspectives, thorough assistance, thoughtfulness, and humor. I would also appreciate your financial support by offering me a research assistantship to compensate for the costly international student tuition fees.

I am also grateful to all the professors in the Music Technology area. I thank Professor Ichiro Fujinaga for the opportunity to work on a Music Information Retrieval project and for improving my presentation skills during the seminar. I also thank Professor Gary Scavone, Stephen McAdams, and Marcelo Wanderley for providing the enlightening seminars that helped me broaden my horizons in various fields.

Thanks to all members of the Signal Processing and Control Laboratory (SPCL). I gained a lot of insight and state-of-the-art at each group meeting and enjoyed the academic atmosphere, even across the computer screens (and the Canada/US border). I am also grateful to my roommate and friend, Yifan Huang, for providing academic and living rescue while I was preparing to combat the deadlines.

Last but not least, I would like to thank my family and my friends. Thank you for all your support, encouragement, inspiration, and love. I wouldn't be who I am now without you all.

Contents

List of Figures

List of Tables

List of Acronyms

Notation	Description	Page
TV-ISD	time-varying Itakura-Saito distance	ix
TF	time-frequency	2
SPAIN	SParse Audio INpainter	2
SNR	signal-to-noise ratio	2
MP	Matching Pursuit	8
LASSO	least absolute shrinkage and selection operator	9
ISTA	Iterative Shrinkage-Thresholding Algorithm	9
FISTA	Fast Iterative Shrinkage-Thresholding Algorithm	10
WGL	windowed-group-lasso	12
STFT	short-time Fourier transform	21
DDM	Distribution Derivative Method	22
PT	partial tracking	24
MQ-PT	McAulay-Quatieri partial tracking	24
AR	autoregressive	34
IIR	infinite impulse response	34
FIR	finite impulse response	35
PSD	power spectral density	36
LV	Loris-Verhoeven	44
RMSE	root-mean-square energy	49
LAR	Log Area Ratio	61
ISD	Itakura-Saito distance	63
ODG	objective difference grade	63
PEAQ	Perceptual Evaluation of Audio Quality	69
IQR	interquartile range	86

List of Symbols

Notation	Description	Page
$\mathbf{y}$	input signal	1
Γ	set of all feasible signals	1
$\mathbf{M}$	masking matrix	1
$y[n]$	discrete time signal	6
$\mathbf{x}$	vector of coefficients of atoms	6
$\mathbf{\Phi}$	dictionary	6
ϕ	atom	6
ψ	window	6
H	hop size	6
$\langle \mathbf{y}, \phi \rangle$	inner product between vectors $\mathbf{y}$ and ϕ	6
$\mathbf{\Phi}^{\mathrm{H}}$	conjugate transpose (Hermitian transpose) of $\mathbf{\Phi}$	7
$\mathbf{S}$	frame operator	7
$\lVert \cdot \rVert_p$	ℓ_p norm	7
$\mathbf{x}^{(\mathrm{k})}$	variable $\mathbf{x}$ at the k-th iteration	8
$\mathbf{r}$	residual	8
λ	shrinkage parameter	9
$\mathcal{S}$	soft-thresholding operator	9
$(\cdot)^+$	ReLU function, $(x)^+ = \max(x, 0)$	9
$\lVert \cdot \rVert_{p,q}$	$\ell_{p,q}$ mixed norm	11
$\mathcal{H}$	hard-thresholding operator	15
$\mathbf{w}$	weight vector	17
$\odot$	element-wise multiplication	17
φ	phase	20
$\Re\{x\}$	real part of x	20
$\Im\{x\}$	imaginary part of x	22

1

Introduction

1.1 Audio Degradation

Degradation of audio signals is ubiquitous in the real world and can happen at many stages of audio content production. The impulsive noise and clipping introduce unwanted components while recording. Transmission errors, like internet packet losses, will miss samples of an audio stream. Moreover, the storage mediums can be physically damaged so that the information cannot be retrieved accurately. The degradation can drastically reduce the perceived audio quality. Consequently, there is a growing need to restore the missing or distorted parts of the audio signal, which leads to the emergence of a research field known as *audio inpainting* (Adler et al. 2012). Audio inpainting involves the recovery of missing or degraded parts of an audio signal based on its reliable segments. This problem was previously referred to as *audio restoration* in earlier decades, prior to the adoption of the term audio inpainting (Godsill and Rayner 1998).

1.2 Audio Inpainting

Adler et al. (2012) proposed a general framework for reconstructing the missing or severely damaged parts of an audio signal named *audio inpainting*. Let $\mathbf{y} \in \mathbb{R}^N$ be an audio signal with N samples. Assume that the indices of its missing or highly degraded samples are known. These samples are referred to as *unreliable* samples. The other samples will be considered *reliable*. The recovered signal should be the same as the original signal in the reliable part, in other words, it should belong to the following set $\Gamma_{\mathbf{y}}$:

$$\Gamma_{\mathbf{y}} = \{\mathbf{s} \in \mathbb{R}^N : \mathbf{M}_{\mathrm{R}}\mathbf{s} = \mathbf{M}_{\mathrm{R}}\mathbf{y}\} \tag{1.1}$$

where $\mathbf{M}_{\mathrm{R}} \in \mathbb{R}^{N\times N}$ is a square diagonal matrix whose k-th diagonal value is 1 if the k-th sample of the original signal is reliable, otherwise it is 0, which means $\mathbf{M}_{\mathrm{R}}\mathbf{y}$ will contain all reliable samples.

In addition, the signal $\mathbf{y}$ can not only be represented in the time domain but also in a transformed domain (such as the Gabor transform). The goal of audio inpainting is to find a signal $\mathbf{s} \in \Gamma_{\mathbf{y}}$ that is perceptually similar to the undegraded version of signal $\mathbf{y}$.

For some kinds of degradations (such as clipping), some prior knowledge about the missing data could be incorporated to narrow down the solution space, so that the reconstruction signals will be in a subset of $\Gamma_{\mathbf{y}}$. Furthermore, the degradation could be analyzed in the time domain or in the time-frequency (TF) domain, which leads to different approaches for reconstruction. For example, the clipping can be seen as an amplitude cutoff in the time domain or as a harmonic distortion in the TF domain.

In the rest of the thesis, we mostly consider the *gap* scenario. A gap refers to a temporal segment with consecutive samples missing. Gaps can usually be divided into two categories based on their duration: *short* gaps and *long* gaps.

Short gaps are usually less than 100 milliseconds in duration, and they are sometimes further classified into short gaps (< 10 ms) and medium gaps (10–100 ms) (Taubock et al. 2021). In this scenario, the signal is more likely to be approximately stationary for the duration of the gap. One classic but still state-of-the-art approach is Janssen's method, which interpolates the missing samples iteratively according to their neighborhood based on the autoregressive model (Janssen et al. 1986). Recently, techniques in the time-frequency domain have been introduced to solve the inpainting problem, such as sparse decomposition (Adler et al. 2012; Mokrý et al. 2019; Mokrý and Rajmic 2020) and non-negative matrix factorization (Mokrý et al. 2023).

Long gaps, on the other hand, are more than 100 milliseconds in duration. This is a more challenging situation because the signal is usually non-stationary. Different approaches could be used, such as sinusoidal modelling (Esquef et al. 2003; Lukin and Todd 2008), waveform substitution based on self-similarity (Perraudin et al. 2018), and deep learning methods (Marafioti et al. 2019; Marafioti et al. 2021; Moliner and Välimäki 2023). These approaches typically require a longer context (or even the entire range of the signal) to reconstruct the gap.

In this thesis, we mainly focus on gaps of length 10–500 milliseconds.

1.3 Motivation

We started our exploration with the sparsity-based models, especially the SParse Audio INpainter (SPAIN), which has better signal-to-noise ratio (SNR) and speed compared to Janssen's method for very short gaps (Mokrý et al. 2019). While experimenting with the method, we found that this method (as well as other sparsity-based methods) has an assumption that does not always hold for

the signals: the undegraded signal should be sparse, which means the signal can be represented as a linear combination of a few simple waveforms. This assumption leads to poor reconstruction quality on certain signals, such as those that are fast-varying or noisy.

To further investigate the problem, we need to consider the characteristics of the input signal itself. An audio signal can be considered a mixture of three components: *sinusoids*, *transients*, and *noise* (Verma and Meng 2000). The sinusoids (also known as the *tonal* part) can be generalized as the slow-varying deterministic part, which is mostly stationary (or cyclo-stationary) in the longer term. The transients represent the fast-varying deterministic part, which consists of components that have a short duration, a wide spectral bandwidth, and are usually located at the beginning or end of a sustained sound. The noise refers to the stochastic part of the signal, which is often referred to as the residual of the signal. Different methods are generally used for analyzing and re-synthesizing these three components, which may explain the inconsistency in the reconstruction quality of sparsity-based methods on different signals. This model provides a structured view of the various audio signals and is widely used in the fields of additive synthesis (Verma and Meng 2000; Tantibundhit et al. 2006) and audio encoding (Daudet and Torrésani 2002). However, it is rarely discussed in the context of audio inpainting.

In order to separate these three components from the mixture, one technique that can be used is called *sparse decomposition*. This method is able to provide a signal representation that is not only more interpretable, but also contains structured information extracted from the signal (Kereliuk and Depalle 2011; Siedenburg and Dörfler 2011). Moreover, sparse decomposition can be combined with other methods that are tailored for each component, such as sinusoidal modeling for the tonal part, to achieve a better reconstruction quality than inpainting methods using only the sparsity-based techniques.

Therefore, we will build a hybrid approach to improve the perceived quality of the reconstruction. The hybrid approach will utilize structural sparse decomposition to obtain these three components and reconstruct them in different ways.

1.4 Contributions

The main contribution of the thesis is the new hybrid approach for audio inpainting. The model links approaches from different fields and exhibits high-quality reconstruction across a wide range of gap lengths and signal types. We provide a comparison and discussion of typical inpainting approaches while considering signal diversity. Moreover, a novel approach to determining the weight of neighborhood in social sparsity is proposed in the decomposition process. A heuristic method

is introduced to automatically tune the hyperparameters of sparse decomposition algorithms. We also refined the partial tracking and extrapolation algorithms to obtain more robust results.

1.5 Organization of the Thesis

Chapter 2 provides an overview of typical approaches for audio inpainting, including sparse decomposition, additive synthesis, and noise reconstruction. The weaknesses of these methods are discussed. Chapter 3 introduces a new hybrid approach for audio inpainting and elaborates on each component in detail. Chapter 4 analyzes our hybrid approach and compares it with other state-of-the-art techniques through various experiments. Chapter 5 summarizes this thesis, outlines the strengths and limitations of our approach, and addresses some possibilities for future research.

2

Review of Audio Inpainting Methods

This chapter reviews the typical approaches for audio inpainting. We focus on three main categories of methods: sparse decomposition, sinusoidal modeling, and autoregressive modeling. Each category is discussed in detail, encompassing the underlying principles, related inpainting methods, and their weaknesses. The rest of this chapter is organized as follows. Section 2.1 provides an in-depth explanation of sparse decomposition methods, covering fundamental concepts such as atoms and frames, and explores their applications in audio inpainting. Section 2.2 discusses both analysis and synthesis techniques based on the sinusoidal modeling, along with an exploration to related audio inpainting methods. Section 2.3 introduces autoregressive modeling, specifically examining linear prediction methods, spectral envelope estimation techniques, and their applications to audio inpainting.

2.1 Sparse Decomposition

Sparse decomposition, relying on the concept of *atomic modeling*, is a technique that represents or approximates an audio signal as a linear combination of a few elementary waveforms selected from a large waveform bank. The sparse decomposition can be applied to audio inpainting, based on finding appropriate constraints associated to the problem. In this section, we first review the basic concepts of sparse decomposition, and then present and discuss some existing sparsity-based audio inpainting methods.

2.1.1 Atoms and dictionaries

In the atomic modeling, a time-domain audio signal $y[n]$ can be written as a linear combination of simple waveforms,

$$y[n] = \sum_i x_i \phi_i[n] \tag{2.1}$$

where $\phi_i[n]$ are the simple waveforms (*atoms*), and x_i are the weights of them. The set of atoms forms a *dictionary*. We can rewrite the model into matrix form,

$$\mathbf{y} = \mathbf{\Phi x} \tag{2.2}$$

where $\mathbf{y} \in \mathbb{R}^N$ is an audio signal, $\mathbf{\Phi} \in \mathbb{R}^{N \times M}$ is a dictionary whose columns $\boldsymbol{\phi}_m$ are atoms.

The choices of the set of atoms $\phi_i[n]$ are infinite. In order to represent a wide variety of signals and to have good time-frequency (TF) domain properties, a Gabor frame could be used to build a set of atoms as will be described below.

2.1.2 Gabor frames

A *Gabor atom* is defined as

$$\phi_{m,k}[n] = \psi[n - Hm] e^{2\pi jk/K} \tag{2.3}$$

where ψ is a window, H is a hop size, K is the number of frequency shifts or frequency channels. The signal length N should be divisible by the hop size H. Therefore, for a signal with length N, we can build $P = KN/H$ different Gabor atoms. The set of these Gabor atoms is referred to as the *Gabor dictionary*.

By choosing the appropriate ψ, H, and K, a *Gabor frame* can be formed by satisfying the condition that for all signals $\mathbf{y} \in \mathbb{C}^N$, there exist constants $A, B > 0$ so that:

$$A\|\mathbf{y}\|_2^2 \leq \sum_{m,k} \left| \langle \mathbf{y}, \boldsymbol{\phi}_{m,k} \rangle \right|^2 \leq B\|\mathbf{y}\|_2^2 \tag{2.4}$$

where A and B are called the *frame bounds*.

Frames need fewer restrictions and are easier to construct than bases, as they introduce redundancy to the dictionaries for more flexibility. Redundancy can provide interpretable representations in the TF domain.

The signal $\mathbf{y}$ can be synthesized from the Gabor atoms $\boldsymbol{\phi}_{m,k}$ and their coefficients $x_{m,k}$:

$$y[n] = \sum_{m,k} x_{m,k}\, \phi_{m,k}[n] \quad \iff \quad \mathbf{y} = \mathbf{\Phi x} \tag{2.5}$$

where $\mathbf{\Phi} : \mathbb{C}^P \rightarrow \mathbb{C}^N$ is the *synthesis operator*, $\mathbf{x} \in \mathbb{C}^P$ is the vector of coefficients of Gabor atoms. The coefficients of atoms can be calculated as:

$$x_{m,k} = \langle \mathbf{y}, \phi_{m,k} \rangle \quad \Longleftrightarrow \quad \mathbf{x} = \mathbf{\Phi}\mathbf{y} \tag{2.6}$$

where $\mathbf{\Phi}^{\mathrm{H}}$ is the conjugate transpose (Hermitian transpose) of $\mathbf{\Phi}$ and is referred to as the *analysis operator*. By composing the synthesis and analysis operators, we can define the *frame operator* $\mathbf{S}$ so that:

$$\mathbf{S}\mathbf{y} = \mathbf{\Phi}\mathbf{\Phi}^{\mathrm{H}}\mathbf{y}. \tag{2.7}$$

If the frame operator $\mathbf{S}$ is a diagonal positive-definite matrix, i.e., all values from its main diagonal are greater than 0 and all other values are 0, then we say that the frame operator satisfies the *painless case* (Balazs et al. 2013). In addition, if $\mathbf{S}$ is a positive multiple of the identity matrix ($\mathbf{S} = c\mathbf{I}_N$, where c is a positive number and $\mathbf{I}_N$ is the $N \times N$ identity matrix), the frame is called a *tight* frame (Dörfler 2001). In this case, we have $A = B$ for Eq. (2.4). The tight frames are used for various applications and will be mentioned in later sections of the thesis.

2.1.3 Sparse approximation

Many audio signals can be represented (or approximated) by a small amount of atoms (Kereliuk 2013). That means most value in $\mathbf{x}$ will be zero if $\mathbf{\Phi}$ is a redundant dictionary (a frame). A sparse representation of an audio signal describes the signal in a more interpretable way and could be used for high-level analysis and processing. However, there is a trade-off between sparsity and the quality of approximation. That leads to the sparse approximation problem, which can be formalized as an optimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 \leq \varepsilon \tag{2.8}$$

where $\|\mathbf{x}\|_0$ is the ℓ_0 "norm" of $\mathbf{x}$ which counts the non-zero coefficients in $\mathbf{x}$, $\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2$ measures the residual energy, and ε is the residual energy threshold. Since we try to minimize the error between the original signal $\mathbf{y}$ and the synthesized signal $\mathbf{\Phi}\mathbf{x}$, this model is referred to as the *synthesis* model, and we say that $\mathbf{x}$ is a *sparse representation* of $\mathbf{y}$.

In contrast to the synthesis model, an alternative approach called the *analysis* model has gained growing interest within the last decade. Instead of controlling the number of non-zero coefficients of atoms, the analysis model tries to sparsify the transformed signal by applying the analysis operator to the signal:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{\Phi}^{\mathrm{H}}\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{x}\|_2^2 \leq \varepsilon \tag{2.9}$$

where $\mathbf{\Phi}^{\mathrm{H}}$ is the analysis operator and satisfies $\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{H}}$ to be a positive multiple of the identity matrix, we say that $\mathbf{x}$ is a *cosparse representation* of $\mathbf{y}$ (Nam et al. 2013). Although these two models look similar, they tend to generate very different decomposition results.

Although finding the optimal solution to this problem is very challenging, a suboptimal solution is usually built as an approximation based on available algorithms. There are two main approaches to solving this problem: the greedy approach and the relaxation approach.

2.1.3.1 Greedy approach

Greedy algorithms iteratively select atoms from a given dictionary. The coefficients of selected atoms are updated to minimize the residual error at each iteration.

Suppose that we start with an empty selection and all coefficients are zero ($\mathbf{x}^{(0)} = \mathbf{0}$). Then we need to choose one atom from the dictionary that has the most influence on the signal. That could be achieved by finding the maximum absolute value of the correlation between atoms and the signal:

$$\arg\max_k |\langle \mathbf{y}, \boldsymbol{\phi}_k \rangle| \tag{2.10}$$

where k is the index of the atom, representing the k-th atom (k-th column of the dictionary). Then we can update the selection $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha \boldsymbol{\delta}_k$, where $\alpha = \langle \mathbf{y}, \boldsymbol{\phi}_k \rangle$ is the coefficient, $\boldsymbol{\delta}_k$ is a vector that is 1 for the k-th value and 0 for the others. The residual can then be calculated by:

$$\mathbf{r}^{(1)} = \mathbf{y} - \mathbf{\Phi}\mathbf{x}^{(1)} = \mathbf{r}^{(0)} - \alpha \boldsymbol{\phi}_k \tag{2.11}$$

where $\mathbf{r}^{(0)} = \mathbf{y}$ is the residual before selecting any atoms. That forms one iteration of this algorithm, and the next atom to be selected will be the one with the highest correlation with the residual $\mathbf{r}^{(1)}$. This method is known as *matching pursuit* (MP) algorithm, and it is one of the earliest and most well-known greedy algorithms for sparse decomposition (Mallat and Zhang 1993). The procedure is summarized in Algorithm 1.

2.1.3.2 Relaxation approach

Instead of making locally optimal decisions at each iteration, the relaxation approach aims to modify the optimization problem itself and obtain a global approximate solution through iterations.

We start by rewriting the problem (Eq. (2.8)) to an unconstrained form:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 + \lambda \mathcal{R}(\mathbf{x}) \tag{2.12}$$

Algorithm 1. Matching Pursuit (MP)

Input: input signal $\mathbf{y}$, dictionary $\mathbf{\Phi}$

Output: coefficients of atoms $\mathbf{x}$

Initialization: $\mathsf{n} = 1, \mathbf{x}^{(0)} = \mathbf{0}, \mathbf{r}^{(0)} = \mathbf{y}$

1: **repeat**

2: $k^{(\mathsf{n})} = \arg\max_k \left|\langle \mathbf{r}^{(\mathsf{n}-1)}, \boldsymbol{\phi}_k \rangle\right|$

3: $\alpha^{(\mathsf{n})} = \langle \mathbf{r}^{(\mathsf{n}-1)}, \boldsymbol{\phi}_{k^{(\mathsf{n})}} \rangle$

4: $\mathbf{x}^{(\mathsf{n})} = \mathbf{x}^{(\mathsf{n}-1)} + \alpha^{(\mathsf{n})} \boldsymbol{\delta}_{k^{(\mathsf{n})}}$

5: $\mathbf{r}^{(\mathsf{n})} = \mathbf{r}^{(\mathsf{n}-1)} - \alpha^{(\mathsf{n})} \boldsymbol{\phi}_{k^{(\mathsf{n})}}$

6: $\mathsf{n} = \mathsf{n} + 1$

7: **until** stopping condition

where $\lambda > 0$ represents the *Lagrange multiplier*, controlling the strength of the constraint, and $\mathcal{R}$ is the regularization function penalizing small coefficients. In this case, the regularization function $\mathcal{R}$ corresponds to the ℓ_0 "norm."

Due to the fact that the ℓ_0 "norm" is non-convex and NP-hard to solve (Natarajan 1995), a convex relaxation can be employed by replacing the ℓ_0 "norm" to the ℓ_1-norm. The ℓ_1-norm is convex and computationally tractable while still promoting sparsity. Therefore, it is a good approximation of the ℓ_0 "norm" for sparse decomposition. This technique is known as LASSO (least absolute shrinkage and selection operator) (Tibshirani 1996) or basis pursuit denoising (Chen et al. 2001).

One of the most popular methods for solving the LASSO problem is the Iterative Shrinkage-Thresholding Algorithm (ISTA), which is a generalized gradient descent method for non-smooth functions. ISTA is based on applying a gradient descent step followed by a proximal operator to project the gradient onto a convex set while minimizing the regularization term. The proximal operator of a given convex function f is:

$$\operatorname{prox}_f(\mathbf{x}) = \arg\min_{\mathbf{z}} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 + f(\mathbf{z}) \right\}. \tag{2.13}$$

When the regularization function is ℓ_1-norm, the solution of the proximal operator is a simple shrinkage operator $\mathcal{S}$ which refers to *soft-thresholding*:

$$\mathcal{S}_\lambda(x_i) = \operatorname{sgn}(x_i) \max(|x_i| - \lambda, 0) = x_i \left(1 - \frac{\lambda}{|x_i|}\right)^+ \tag{2.14}$$

where $\lambda > 0$ is the threshold parameter, $\operatorname{sgn}(x)$ is the sign function, $(x)^+ = \max(x, 0)$. The shrinkage operator has the property of setting small values of x_i to zero, thus promoting sparsity in the solution. The procedure is summarized in Algorithm 2.

Algorithm 2. Iterative Shrinkage-Thresholding Algorithm (ISTA)

Input: input signal $\mathbf{y}$, synthesis operator $\mathbf{\Phi}$, regularization parameter λ, shrinkage operator $\mathcal{S}$

Output: coefficients of atoms $\mathbf{x}$

Initialization: $n = 1, \mathbf{x}^{(0)} = \mathbf{0}, \gamma = \|\mathbf{\Phi}\mathbf{\Phi}^{H}\|$

1: **repeat**

2: $\quad \nabla_{\mathbf{x}}^{(n)} = \mathbf{\Phi}^{H}(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{(n-1)})$

3: $\quad \mathbf{x}^{(n)} = \mathcal{S}_{\lambda/\gamma}\left(\mathbf{x}^{(n-1)} + \frac{1}{\gamma}\nabla_{\mathbf{x}}^{(n)}\right)$

4: $\quad n = n + 1$

5: **until** stopping condition

Furthermore, the ISTA algorithm can be extended to the accelerated proximal gradient algorithm, known as the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) (Beck and Teboulle 2009). FISTA provides faster convergence and improves the computational efficiency of ISTA by incorporating a momentum term (τ in Algorithm 3), while maintaining the accuracy of the sparse decomposition. Algorithm 3 describes the FISTA procedure.

Algorithm 3. Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

Input: input signal $\mathbf{y}$, synthesis operator $\mathbf{\Phi}$, regularization parameter λ, shrinkage operator $\mathcal{S}$

Output: coefficients of atoms $\mathbf{x}$

Initialization: $n = 1, \mathbf{x}^{(0)} = \mathbf{0}, \mathbf{z}^{(0)} = \mathbf{0}, \gamma = \|\mathbf{\Phi}\mathbf{\Phi}^{H}\|, \tau^{(0)} = 1$

1: **repeat**

2: $\quad \nabla_{\mathbf{x}}^{(n)} = \mathbf{\Phi}^{H}(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{(n-1)})$

3: $\quad \mathbf{z}^{(n)} = \mathcal{S}_{\lambda/\gamma}\left(\mathbf{x}^{(n-1)} + \frac{1}{\gamma}\nabla_{\mathbf{x}}^{(n)}\right)$

4: $\quad \tau^{(n)} = \frac{1}{2}\left(1 + \sqrt{1 + (2\tau^{(n-1)})^2}\right)$

5: $\quad \mathbf{x}^{(n)} = \mathbf{z}^{(n)} + \frac{\tau^{(n-1)}-1}{\tau^{(n)}}(\mathbf{z}^{(n)} - \mathbf{z}^{(n-1)})$

6: $\quad n = n + 1$

7: **until** stopping condition

2.1.4 Structured sparsity

Sparse decomposition is a powerful technique for effectively representing signals with a smaller number of non-zero coefficients. However, simple sparse decomposition methods may fall short when it comes to capturing the underlying structural information present in signals. One approach to addressing this limitation is through the utilization of *structured sparsity* (Siedenburg and Dörfler 2011). Structured sparsity aims to incorporate prior knowledge from the signals into the decom-

position process, enabling a more meaningful representation of signals under multiple applications (Kereliuk 2013).

The structure information could be integrated into the norm by extending it to summations within and outside the group of atoms. The idea leads to a technique named *mixed norm* (Kowalski and Torrésani 2009):

$$\|\mathbf{x}\|_{p,q} = \left(\sum_{g=1}^{G} \left(\sum_{m=1}^{M} |x_{g,m}|^p \right)^{q/p} \right)^{1/q} \tag{2.15}$$

where $x_{g,m}$ is a double-indexed element of $\mathbf{x} \in \mathbb{R}^N$, G is the number of groups, and M is the number of members in each group, so that $G \times M$ is the number of atoms. The p and q determine the norm behaviors within and between groups, respectively. When $p > 2$ and $q < 2$, minimizing $\|\mathbf{x}\|_{p,q}$ promotes inter-group sparsity, implying that only fewer groups are selected, while when $p < 2$ and $q > 2$, it promotes intra-group sparsity, implying that fewer members of each group are selected (Balazs et al. 2013). The problem reduces to a standard LASSO when $p = q = 1$.

One particular case is the *group-lasso* problem, in which $p = 2$ and $q = 1$, forming the $\ell_{2,1}$ norm (Kowalski and Torrésani 2009). The problem can be denoted by the following optimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_{2,1} \right\}. \tag{2.16}$$

The proximal operator of the group-lasso problem is given by:

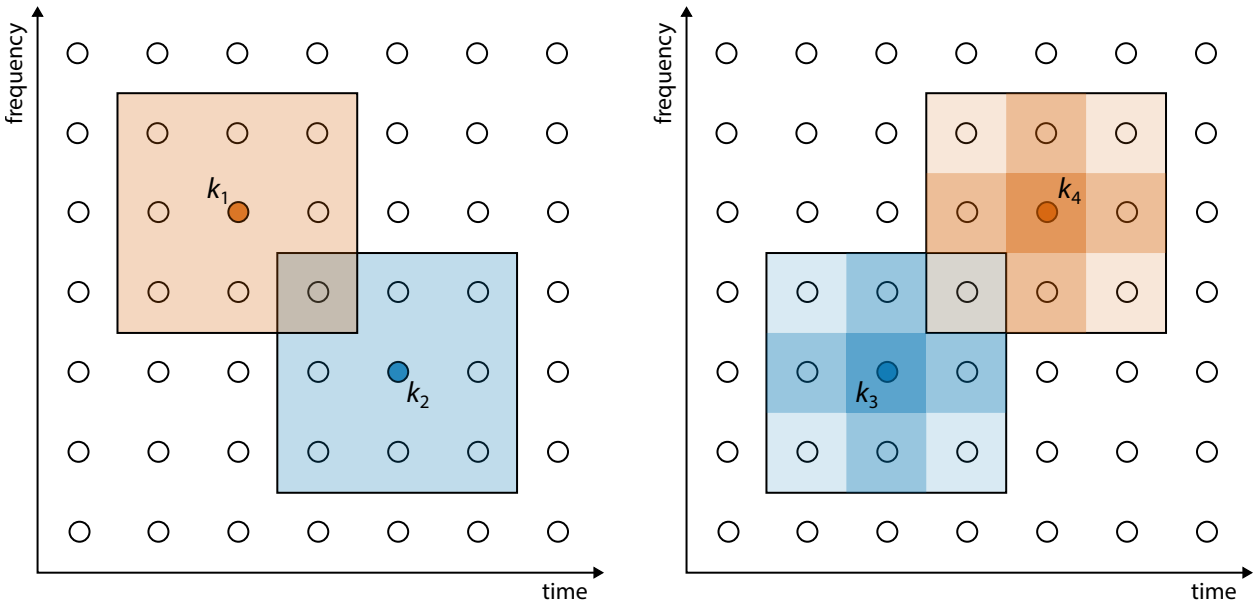
$$\text{prox}_{\|\cdot\|_{2,1}(\lambda)}(x_{g,m}) = x_{g,m} \left(1 - \frac{\lambda}{\|\mathbf{x}_g\|_2} \right)^+ \tag{2.17}$$

where $\mathbf{x}_g$ represents the vector containing all members m in the group g.

The group-lasso problem can be applied to extract structured information in the TF domain. For example, if atoms in the TF domain are grouped by time so that all frequencies in each time bin are its members, then the solution of the corresponding group-lasso will include fewer groups, and all members of the group will not shrink. This corresponds to extracting the transients of the signal, which are short and non-stationary components. In contrast, if atoms are grouped by frequency so that all times in each frequency bin are its members, this corresponds to extracting the tonal part of the signal, which corresponds to sustained and stationary part.

However, the limitation of group-lasso is that its groups are global rather than local, so it is challenging to handle signals that change over time. For example, a signal may have different transient and tonal components at different segments, and applying a fixed grouping scheme may not capture these variations.

To address this constraint, an alternative approach can be employed by constructing overlapping groups utilizing the neighborhood associated with each atom. This technique, known as *social sparsity*, involves selecting atoms based on the weighting of coefficients within their respective neighborhoods (Kowalski et al. 2013). The neighborhood $\mathcal{N}(k)$ of an atom with index k is defined as a set of indices k' that are near the atom k. The neighborhood can be of an arbitrary shape and can be weighted for more flexibility, as shown in Figure 2.1. The weights $w_{k'|k}$ of the index k' at the k-th atom should satisfy that $w_{k|k} > 0$, $w_{k'|k} \geq 0$ for all $k' \in \mathcal{N}(k)$, and $\sum_{k' \in \mathcal{N}(k)} \left(w_{k'|k}\right)^2 = 1$.



(a) evenly weighted **(b)** unevenly weighted

Figure 2.1 Two neighborhood configurations in the time-frequency plane. The circles represent atoms, the square borders represent the neighborhood $\mathcal{N}(k)$ of the corresponding index k (denoted by solid circles), and the shades of color represent the magnitude of the weights.

After defining the neighborhood, the shrinkage operator can be built. One possible shrinkage operator is the windowed-group-lasso (WGL), which is given by:

$$\mathcal{S}_{\lambda}^{\mathrm{WGL}}(x_k) = x_k \left(1 - \frac{\lambda}{\sqrt{\sum_{k' \in \mathcal{N}(k)} w_{k'|k} |x_{k'}|^2}}\right)^{+} \tag{2.18}$$

From this equation, we can observe that only the atoms with the weighted sums of the coefficients of their neighborhood larger than λ will be selected. Thus, the method is capable of suppressing isolated atoms with large coefficients while extracting structured information, depending on the configuration of the neighborhood.

2.1.5 Applications

Sparse decomposition can be utilized to address various signal processing tasks by redefining the underlying optimization problem. In this section, we will explore two specific examples where modified sparse decomposition problems are employed to tackle practical issues. These applications incorporate prior knowledge and introduce additional constraints to guide the sparse decomposition process toward estimations of the original signal.

One instance is to estimate the original signal from its degraded observation. The degradation can be formulated as:

$$\hat{\mathbf{y}} = \mathfrak{D}\mathbf{y} + \varepsilon. \tag{2.19}$$

Here, $\mathbf{y}$ represents the original signal, $\hat{\mathbf{y}}$ corresponds to the observed signal, $\mathfrak{D}$ denotes a degradation operator, and ε an additive noise. In many cases, the degradation operators are nonlinear, making the task of finding their inverse challenging. Furthermore, the degraded signal is often not as sparse as the original signal. Consequently, sparse decomposition techniques can be employed to approximate the original signal. Figure 2.2 demonstrates how different types of degradation affect the sparsity of the signal in distinct ways.

To address specific types of signal degradation, different constraints are introduced to guide the sparse decomposition process (Mokrý et al. 2020). One type of degradation is that part of the signal is missing or highly degraded. We call this degradation a *gap*. In this case, the reliable part of the observed signal should remain unchanged, while the reconstruction signal to replace the unreliable part should be sparse. This can be expressed as a modified sparse decomposition problem with an additional constraint that forces the reconstructed signal to match the reliable part of the observed signal (Adler et al. 2012). The problem can be written as:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{M}_\mathrm{R}\hat{\mathbf{y}} - \mathbf{M}_\mathrm{R}\mathbf{\Phi}\mathbf{x}\|_2^2 \leq \varepsilon. \tag{2.20}$$

Another type of degradation is clipping, where the amplitude is limited between a lower bound $-\theta_\mathrm{clip}$ and an upper bound $+\theta_\mathrm{clip}$, where θ_clip is referred to as the clipping threshold. In this scenario, the part of the signal that stays within the threshold is the reliable part and should be unchanged, while the other parts should exceed the clipping threshold and end up with larger amplitudes (Adler et al. 2012). The problem can be formulated as:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \ \text{ subject to } \ \|\mathbf{M}_\mathrm{R}\hat{\mathbf{y}} - \mathbf{M}_\mathrm{R}\mathbf{\Phi}\mathbf{x}\|_2^2 \leq \varepsilon \ \text{ and } \ |\mathbf{M}_\mathrm{C}\mathbf{\Phi}\mathbf{x}| \geq |\theta_\mathrm{clip}| \tag{2.21}$$

where $\mathbf{M}_\mathrm{C} \in \mathbb{R}^{N\times N}$ is a square diagonal matrix where the k-th diagonal element is 1 if the corresponding k-th sample of the original signal is clipped, and 0 otherwise, so that $\mathbf{M}_\mathrm{C}\mathbf{y}$ will contain all clipped samples.

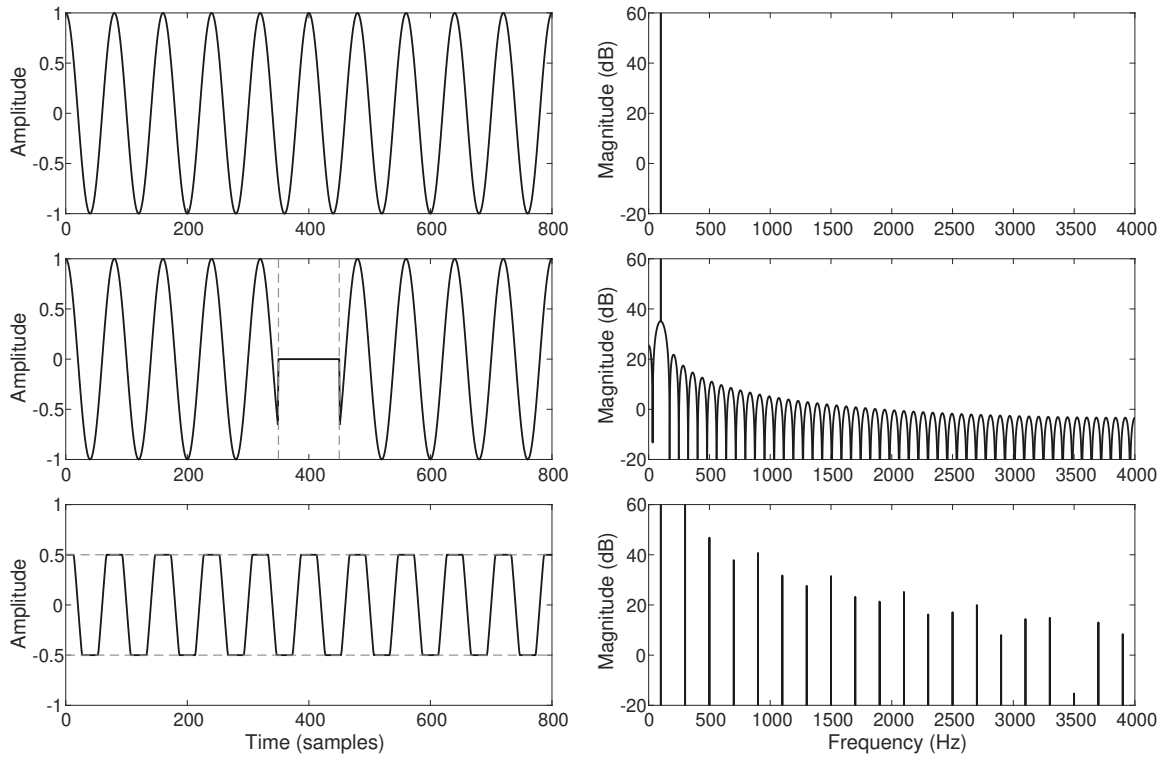


Figure 2.2 Time domain (left side) and frequency domain (right side) representations of a 100 Hz sine wave with different degradations. From top to bottom: original sine wave; sine wave with a 100-sample gap (between the two dashed lines); sine wave with clipping at an amplitude of 0.5. The sampling rate is set to $f_s = 8000$ Hz.

In addition to restoring various signal from the degradations, sparse decomposition is capable of separating a signal into different components. This technique, known as *multilayered expansion*, involves decomposing a given signal into distinct layers or components, each representing a specific aspect of the signal (Kowalski and Torrésani 2009; Kereliuk and Depalle 2011). A common expansion of the audio signal is to decompose it into *tonal*, *transient*, and *noise* components, so that the signal $\mathbf{y}$ can be represented by the summation of these components:

$$\mathbf{y} = \mathbf{y}_{\text{tonal}} + \mathbf{y}_{\text{transient}} + \mathbf{y}_{\text{noise}} \tag{2.22}$$

where $\mathbf{y}_{\text{tonal}}$, $\mathbf{y}_{\text{transient}}$, and $\mathbf{y}_{\text{noise}}$ are the tonal, transient, and noise layer respectively.

To achieve this tonal + transient + noise expansion, two different Gabor frames are utilized to represent the signal components that are suitable their signal features. The frame $\mathbf{\Phi}_{\text{long}}$ has a long duration window and is adapted to the tonal layer, while the frame $\mathbf{\Phi}_{\text{short}}$ has a short duration window and is adapted to the transient layer. Consequently, the multilayered expansion can be

formulated as:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}_\cap} \|\mathbf{x}_\cap\|_0 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{\Phi}_{\text{long}}\mathbf{x}_{\text{tonal}} - \mathbf{\Phi}_{\text{short}}\mathbf{x}_{\text{transient}}\|_2^2 \leq \varepsilon \tag{2.23}$$

where $\mathbf{x}_\cap = [\mathbf{x}_{\text{tonal}}, \mathbf{x}_{\text{transient}}]$. The noise component can be obtained from the residual of the decomposition:

$$\mathbf{y}_{\text{noise}} = \mathbf{y} - \mathbf{\Phi}_{\text{long}}\mathbf{x}_{\text{tonal}} - \mathbf{\Phi}_{\text{short}}\mathbf{x}_{\text{transient}}. \tag{2.24}$$

2.1.6 Sparsity-based inpainting models

This section introduces two sparsity-based inpainting approaches that are considered state-of-the-art. These approaches differ in their sparsity measures and optimization strategies, which lead to different outcomes. The first approach, SPAIN, minimizes the ℓ_0-norm of the coefficients, which results in a non-convex problem. The second approach, reweighted inpainting, uses the convex ℓ_1-norm relaxation and applies weighting to the coefficients. Further discussion and comparison of these approaches will be presented in the subsequent sections.

2.1.6.1 SPAIN

The SParse Audio INpainter (SPAIN) is an audio inpainting method that adapts the SPADE algorithm by Kitić et al. (2015), which was originally developed for audio declipping. The SPAIN algorithm is based on the ℓ_0-norm minimization problem, which enhances sparsity. However, this problem is non-convex and computationally intractable. To overcome this challenge, SPAIN incorporates the Alternating Direction Method of Multipliers (ADMM) optimization scheme to approximate a local-optimal solution (Mokrý et al. 2019). The analysis and synthesis variants of SPAIN are described in Algorithm 4 and Algorithm 5.

The relaxation step size σ and step rate ρ should be positive integers, and $\mathcal{H}_l(\mathbf{x})$ denotes the hard thresholding operator, which retains only the l largest magnitude coefficients of $\mathbf{x}$ while setting the rest to zero.

Experiments show that both variants achieve state-of-the-art reconstruction quality in terms of objective measures such as signal-to-noise ratio (SNR) and PEMO-Q (Huber and Kollmeier 2006), especially for gaps less than 25 ms (Mokrý et al. 2019; Mokrý and Rajmic 2020). In most cases, the analysis variant slightly outperforms the synthesis variant (Mokrý et al. 2019).

Algorithm 4. A-SPAIN

Input: input signal $\mathbf{y}$, analysis operator $\mathbf{\Phi}^{\mathrm{H}}$, reliable mask $\mathbf{M}_{\mathrm{R}}$, relaxation step size σ, relaxation step rate ρ

Output: estimated signal $\mathbf{x}$

Initialization: $\mathsf{n} = 1$, $\mathbf{x}^{(0)} = \mathbf{y}$, $\mathbf{u}^{(0)} = \mathbf{0}$, $l = \sigma$, $\mathbf{z} = \mathbf{0}$

repeat
  $\mathbf{v}^{(\mathsf{n})} = \mathcal{H}_l(\mathbf{\Phi}^{\mathrm{H}}\mathbf{x}^{(\mathsf{n}-1)} + \mathbf{u}^{(\mathsf{n}-1)})$
  $\mathbf{x}^{(\mathsf{n})} = \arg\min_{\mathbf{z}} \|\mathbf{\Phi}^{\mathrm{H}}\mathbf{z} - \mathbf{v}^{(\mathsf{n})} + \mathbf{u}^{(\mathsf{n}-1)}\|_2^2 \quad \text{subject to} \quad \mathbf{M}_{\mathrm{R}}\mathbf{z} = \mathbf{M}_{\mathrm{R}}\mathbf{y}$
  $\mathbf{u}^{(\mathsf{n})} = \mathbf{u}^{(\mathsf{n}-1)} + \mathbf{\Phi}^{\mathrm{H}}\mathbf{x}^{(\mathsf{n})} - \mathbf{v}^{(\mathsf{n})}$
  $\mathsf{n} = \mathsf{n} + 1$
  if $\mod(\mathsf{n}, \rho) = 0$ **then**
    $l = l + \sigma$
  end if
until stopping condition

Algorithm 5. S-SPAIN

Input: input signal $\mathbf{y}$, synthesis operator $\mathbf{\Phi}$, reliable mask $\mathbf{M}_{\mathrm{R}}$, relaxation step size σ, relaxation step rate ρ

Output: coefficients of atoms $\mathbf{x}$

Initialization: $\mathsf{n} = 1$, $\mathbf{x}^{(0)} = \mathbf{\Phi}^{\mathrm{H}}\mathbf{y}$, $\mathbf{u}^{(0)} = \mathbf{0}$, $l = \sigma$, $\mathbf{z} = \mathbf{0}$

repeat
  $\mathbf{v}^{(\mathsf{n})} = \mathcal{H}_l(\mathbf{\Phi}^{\mathrm{H}}\left(\mathbf{x}^{(\mathsf{n}-1)} - \mathbf{u}^{(\mathsf{n}-1)}\right))$
  $\mathbf{x}^{(\mathsf{n})} = \arg\min_{\mathbf{z}} \|\mathbf{\Phi}\mathbf{v}^{(\mathsf{n})} - \mathbf{z} + \mathbf{u}^{(\mathsf{n}-1)}\|_2^2 \quad \text{subject to} \quad \mathbf{M}_{\mathrm{R}}\mathbf{z} = \mathbf{M}_{\mathrm{R}}\mathbf{y}$
  $\mathbf{u}^{(\mathsf{n})} = \mathbf{u}^{(\mathsf{n}-1)} + \mathbf{\Phi}\mathbf{v}^{(\mathsf{n})} - \mathbf{x}^{(\mathsf{n})}$
  $\mathsf{n} = \mathsf{n} + 1$
  if $\mod(\mathsf{n}, \rho) = 0$ **then**
    $l = l + \sigma$
  end if
until stopping condition

2.1.6.2 Reweighted inpainting

Reweighted inpainting is another sparsity-based audio inpainting approach that solves a weighted ℓ_1-norm relaxation problem, which is convex and practically solvable using splitting algorithms (Mokrý and Rajmic 2020). This approach introduces a weight assignment to the coefficients, allowing coefficients with higher weights to resist shrinkage and preserve more energy within the gap region. The analysis and synthesis variants of the relaxation problem are formulated in Eq. (2.25) and Eq. (2.26):

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \|\mathbf{w} \odot \mathbf{\Phi}^{\mathrm{H}}\mathbf{s}\|_1 \quad \text{subject to} \quad \|\mathbf{M}_{\mathrm{R}}\mathbf{y} - \mathbf{M}_{\mathrm{R}}\mathbf{s}\|_2^2 \leq \varepsilon \tag{2.25}$$

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{w} \odot \mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{M}_{\mathrm{R}}\mathbf{y} - \mathbf{M}_{\mathrm{R}}\mathbf{\Phi}\mathbf{x}\|_2^2 \leq \varepsilon \tag{2.26}$$

where $\mathbf{w}$ represents the weight vector that each value w_i of the vector should be greater than 0, and $\odot$ denotes the element-wise multiplication.

Various approaches can be employed to define these weights. One approach is based on the reliability of atoms, where coefficients associated with atoms overlapping with the gap region should have lower weights than those fully located within the reliable part. The weights are defined as follows:

$$w_i = \frac{\|\mathbf{M}_{\mathrm{R}}\boldsymbol{\phi}_i\|_p^q}{\|\boldsymbol{\phi}_i\|_p^q} \tag{2.27}$$

where $\boldsymbol{\phi}_i$ is the i-th atom, $\mathbf{M}_{\mathrm{R}}\boldsymbol{\phi}_i$ is the reliable part of the atom, p and q are parameters that influence the norm behavior. The choices for p and q can be $p = q = 0$ (support based), $p = q = 1$ (ℓ_1-norm based), $p = 2$, $q = 1$ (ℓ_2-norm based), and $p = q = 2$ (energy based).

The analysis and synthesis problems can be solved using Chambolle-Pock and Douglas-Rachford algorithms, respectively (Mokrý and Rajmic 2020).

Another approach of reweighting is to adjust the weights in each iteration, called iterative reweighting. This method can adaptively adjust the weights according to the magnitude of the coefficients. For small coefficients, the weights will progressively increase, making them more resistant to shrinkage. The analysis iterative reweighting method is summarized in Algorithm 6, where step 2 is solved using the Chambolle-Pock algorithm.

Experimental results demonstrate that both reweighted method improve the reconstruction quality in terms of energy preservation, particularly for gaps longer than 25 ms, compared to non-reweighted ℓ_1 relaxation methods (Mokrý and Rajmic 2020).

Algorithm 6. Analysis Iterative Reweighted Inpainting

Input: input signal $\mathbf{y}$, analysis operator $\mathbf{\Phi}^{\mathrm{H}}$, reliable mask $\mathbf{M}_{\mathrm{R}}$, weight threshold $\epsilon > 0$

Output: estimated signal $\mathbf{x}$

Initialization: $\mathsf{n} = 1$, $\mathbf{x}^{(0)} = \mathbf{y}$, $\mathbf{u}^{(0)} = \mathbf{0}$, $\mathbf{w}^{(0)} = \mathbf{1}$

1: **repeat**

2: $\quad \mathbf{x}^{(\mathsf{n})} = \arg\min_{\mathbf{z}} \|\mathbf{w}^{(\mathsf{n}-1)} \odot \mathbf{\Phi}^{\mathrm{H}}\mathbf{z}\|_1 \quad$ subject to $\quad \mathbf{M}_{\mathrm{R}}\mathbf{z} = \mathbf{M}_{\mathrm{R}}\mathbf{y}$

3: $\quad \mathbf{u}^{(\mathsf{n})} = \mathbf{\Phi}^{\mathrm{H}}\mathbf{x}^{(\mathsf{n})}$

4: $\quad w_i^{(\mathsf{n})} = 1/(|u_i^{(\mathsf{n})}| + \epsilon)$

5: $\quad \mathsf{n} = \mathsf{n} + 1$

6: **until** stopping condition

2.1.7 Limitation of sparsity-based models

Despite the good performance of sparsity-based inpainting methods. There are still some limitation that affect the quality of the reconstructed signals. This section addresses two primary aspects of these limitation: energy loss and artifacts induced by stationary assumptions.

One of the common problems of sparsity-based inpainting methods is the energy loss for the reconstructed signal when the gaps get longer. This is partly due to the regularization process that shrinks the coefficients of the signal, resulting in a lower energy level in the gap region. This problem has been reduced by using the reweighted method proposed by Mokrý and Rajmic (2020), which assigns weights to the coefficients according to their reliability to resist shrinkage. However, both SPAIN and reweighted inpainting methods process the signal segment by segment, in an overlap-add approach. This leads to another problem: when processing windowed segments with small number of reliable samples, the algorithm struggles to make a good estimate based on very limited reliable information, thus resulting in a reconstructed segment with very low energy. One demonstration of this problem is shown in Figure 2.3. Some sparsity-based inpainting methods, such as the one proposed by Taubock et al. (2021), attempt to avoid the segmentation problem by processing the entire signal at once, but it still relies on maximal sparsity along frequency for all time bins, thereby neglecting the temporal context.

Another problem is the unsatisfactory reconstruction quality for non-stationary signals, such as those containing noise or fast time-varying components like modulations. Noise is inherently not sparse in the TF domain, which contradicts the core assumption of sparse decomposition algorithms that the signal can be represented by a small amount of atoms. Consequently, sparsity-based inpainting methods may erroneously select atoms and propagate them to the gap because of the ambiguity of representing noise with mismatched dictionaries, which makes the reconstruction look

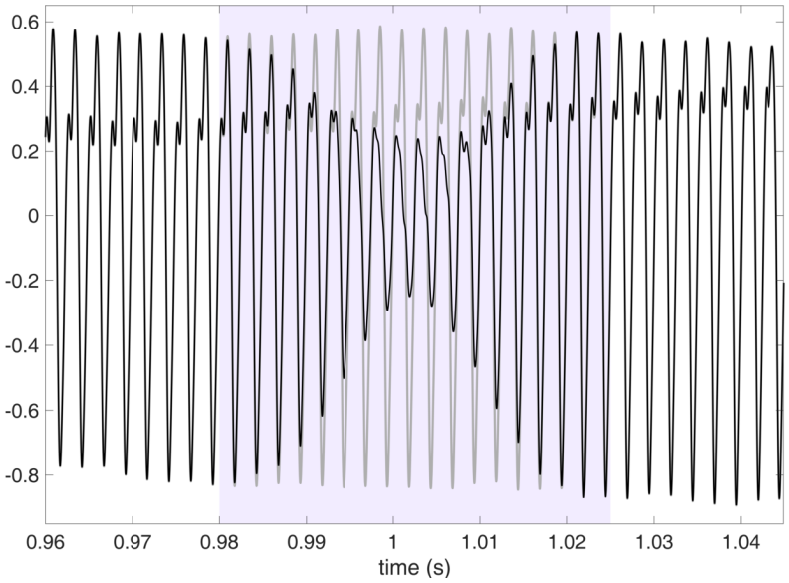
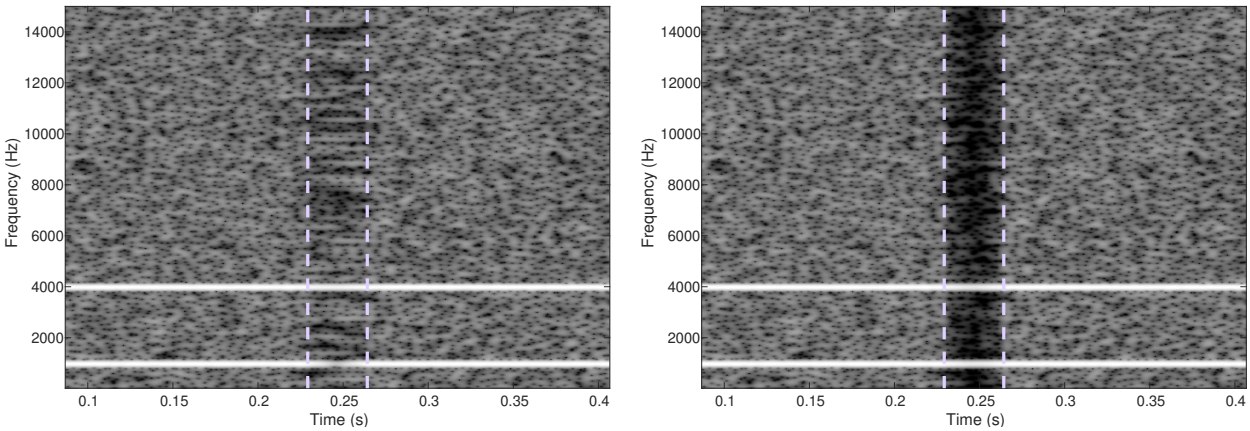


Figure 2.3 Energy drop of the reconstructed signal in the gap using the sparsity-based inpainting methods. The solid gray line represents the original signal, the solid black line represents the reconstructed signal, and the light shaded area denotes the gap area. The result is obtained by applying the A-SPAIN method.

like a stationary signal with randomly selected atoms, or sound like the "freezing" of noise in the gap region. In some cases, some methods may discard the noise component, as a way of denoising, leading to a noticeable decrease in energy. Figure 2.4 illustrates these two scenarios respectively.



(a) "freezing" of noise **(b)** discarded noise

Figure 2.4 Noise artifacts of reconstructed signals with noise in the gap using the sparsity-based inpainting methods. The two light colored dashed lines indicate the beginning and end of the gap. The left panel shows the result obtained by using the weighted Douglas-Rachford algorithm, whereas the right panel shows the result obtained by using the weighted Chambolle-Pock algorithm.

Similarly, fast time-varying signals, even though they may be sparse in some TF representations, are "forced" to propagate stationarily within the gap by sparsity-based methods. This results in a reconstruction that looks like a jump from the left reliable part to the right reliable part with a cross-fade rather than a smooth continuity within the gap region (shown in Figure 2.5). This problem has been partially solved by employing dictionary learning technique to obtain sparser solutions using a dictionary that is learned from the reliable part of the signal (Taubock et al. 2021). However, these methods still use relatively short atoms that is challenging to capture the time-varying characteristics of such signals.

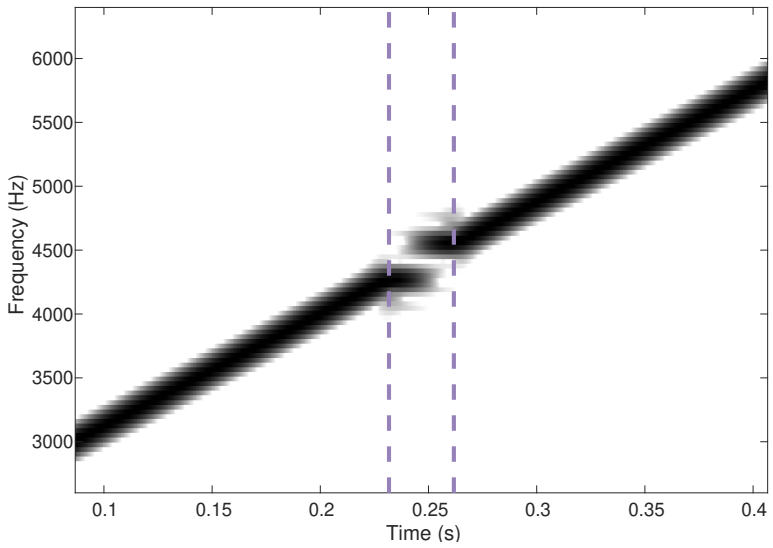


Figure 2.5 Frequency jump of reconstructed fast time-varying signal in the gap using the sparsity-based inpainting methods. The two light colored dashed lines indicate the beginning and end of the gap. The result is obtained by using the iteratively reweighted Chambolle-Pock algorithm.

2.2 Sinusoidal Modeling

Sinusoidal modeling is a technique that represents an audio signal as a sum of sinusoids with slowly evolving amplitudes and frequencies (McAulay and Quatieri 1986). The time-varying sinusoids are called *partials*. The model can be formulated as:

$$y[n] = \Re\left\{\sum_{p=1}^{P} A_p[n] \exp\left(j\left(\varphi_p + 2\pi \sum_{k=0}^{n} f_p[k]\right)\right)\right\} \tag{2.28}$$

where A_p and f_p represent the instantaneous amplitude and frequency of partial p, φ_p is the initial phase of partial p, and $\Re\{x\}$ is the real part of $x \in \mathbb{C}$.

Sinusoidal modeling involves two processes: analysis and synthesis. The analysis process consists of estimating the sinusoidal parameters and decomposing the signal into a set of independent partials, also known as *partial tracking*.

Parameter estimation is a key step of sinusoidal analysis, which involves extracting the amplitudes, frequencies, and phases of the sinusoids from the observed signals. For non-stationary signals, these parameters are not trivial to estimate, and various methods have been proposed for this case. Section 2.2.1 introduces two parameter estimation techniques: the STFT-based method and the Distribution Derivative Method (Betser 2009).

Partial tracking also plays an important role in analysis, which involves connecting the detected peaks across the slices of an audio signal appropriately to form the partials. Ideally, each tracked partial should form a smooth trajectory. Section 2.2.2 introduces a few partial tracking approaches.

The synthesis process generates reconstructed signals from parameters obtained during the analysis. It can be achieved using a bank of oscillators that generate the sinusoids according to the parameters estimated during the analysis. This process is described in Section 2.2.3.

Sinusoidal modeling can be employed for audio inpainting by filling the gap with a resynthesized signal. The resynthesized signal is constructed based on the parameters extracted in the neighborhood of the gap. Section 2.2.4 explains this method and Section 2.2.5 discusses its limitation.

2.2.1 Parameter estimation

The sinusoidal parameter estimation methods involve two steps: obtaining the coefficients in the time-frequency transformed domain and locating the spectral peaks from these coefficients. Short-time Fourier transform (STFT) is commonly used to derive the time-varying parameters, and the spectral peaks are then extracted from each slice of the STFT. The STFT of a signal $y[n]$ is formalized by:

$$Y(m,k) = \sum_{n=-\infty}^{+\infty} y[n]\psi[n - Hm]e^{-2\pi jnk/K} \tag{2.29}$$

where m represents the slice number, k is the frequency channel to evaluate, ψ is the window function, H is the hop size, and K is the number of frequency channels.

In this section, two different approaches are introduced, among the existing ones. One approach is the McAulay and Quatieri method (1986), which finds the spectral peaks from the spectrogram $P(m,k)$ of each STFT slice:

$$P(m,k) = |Y(m,k)|^2 \tag{2.30}$$

where $Y(m,k)$ is the STFT of signal $y[n]$. Spectral peaks are identified as local maxima in the spectrogram. The frequencies are estimated from the location of the peaks, while the amplitude and phase of each peak are obtained from the magnitude and argument of the STFT coefficients.

However, this method has a weakness due to the limited resolution of the spectrogram. Frequency estimation is quantized by the frequency bin size, which reduces the estimation accuracy. Moreover, sinusoids falling between frequency bins result in energy spreading over adjacent bins, causing reduced peak amplitudes and biased phases.

This issue can be improved by using a quadratic interpolation of the spectral magnitude around each peak, referred to as the quadratically interpolated FFT method (Smith and Serra 1987). This technique approximates the true peak location and magnitude by fitting a parabola to three points: the peak itself and its two neighboring bins.

Another advanced approach is the Distribution Derivative Method (DDM). The DDM is based on the exponential polynomial sinusoidal model. In this model, a partial, slowly evolving frequency and amplitude sinusoid (in the continuous time) can be defined as:

$$y(t) = \exp\left(\sum_{k=0}^{Q} \alpha_k t^k\right) \tag{2.31}$$

where α_k are complex sinusoidal parameters and Q is the polynomial order.

The instantaneous log-amplitude and phase of this component are:

$$a(t) = \sum_{k=0}^{Q} \Re\{\alpha_k\} t^k \tag{2.32}$$

$$\varphi(t) = \sum_{k=0}^{Q} \Im\{\alpha_k\} t^k \tag{2.33}$$

and the instantaneous frequency can be calculated by taking the time derivative of the phase:

$$f(t) = \frac{\varphi'(t)}{2\pi} = \frac{1}{2\pi} \sum_{k=1}^{Q-1} \Im\{\alpha_k\} k t^{k-1}. \tag{2.34}$$

If we weight the component using an arbitrary waveform (atom) ϕ that is constant 0 outside a finite range and is conjugated in order to prepare the expression of a dot product, the weighted signal y_ϕ can be represented as:

$$y_\phi(t) = y(t)\bar{\phi}(t) \tag{2.35}$$

and its derivative is:

$$\frac{dy_\phi}{dt}(t) = y'(t)\bar{\phi}(t) + y(t)\bar{\phi}'(t) = \left(\sum_{k=1}^{Q} \alpha_k u_k'(t)\right) y(t)\bar{\phi}(t) + y(t)\bar{\phi}'(t) \tag{2.36}$$

where $u_k(t) = t^k$.

By integrating the derivative over time, we get:

$$\int_{-\infty}^{+\infty} \frac{dy_\phi}{dt}(t)\,\mathrm{d}t = \sum_{k=1}^{Q} \alpha_k \int_{-\infty}^{+\infty} u_k'(t) y(t) \bar{\phi}(t)\,\mathrm{d}t + \int_{-\infty}^{+\infty} y(t)\bar{\phi}'(t)\,\mathrm{d}t. \tag{2.37}$$

Since the weighted signal $y_\phi(t)$ is 0 while approaching to infinity, the integration of its derivative should be 0:

$$\int_{-\infty}^{+\infty} \frac{dy_\phi}{dt}(t)\,\mathrm{d}t = 0. \tag{2.38}$$

This equation can be rewritten as a linear system of equations:

$$-\langle \mathbf{y}, \phi_m' \rangle = \sum_{k=1}^{Q} \alpha_k \langle \mathbf{y} \odot \mathbf{u}_k, \boldsymbol{\phi}_m \rangle \tag{2.39}$$

where $\odot$ is the element-wise multiplication.

In order to obtain all α_k, at least Q different equations are necessary to derive the unique solution of the equations.

Meanwhile, the value of α_0, which determines the initial amplitude and phase, cannot be estimated using the above method. An approximate least square estimation method is proposed for that reason. This method starts by finding the atom $\boldsymbol{\phi}_m$ with the highest correlation to the signal $\mathbf{y}$:

$$\boldsymbol{\phi}_m = \arg\max_{\boldsymbol{\phi}} |\langle \mathbf{y}, \boldsymbol{\phi} \rangle|. \tag{2.40}$$

Then the corresponding α_0 is:

$$\alpha_0 = \log(\langle \mathbf{y}, \boldsymbol{\phi}_m \rangle) - \log(\langle \exp(\mathbf{U}\boldsymbol{\alpha}), \boldsymbol{\phi}_m \rangle) \tag{2.41}$$

where $\mathbf{U} = (\mathbf{u}_1, \cdots, \mathbf{u}_Q)$ and $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_Q)^T$.

It should be noted that there are various parameter estimation methods for an exponential polynomial signal model, each with its advantages and limitation (Hamilton and Depalle 2012). A comprehensive review and comparison of these methods, evaluating their accuracy and efficiency for different signal types, can be found in Hamilton and Depalle (2012). It is worth mentioning that there are other types of parameter estimation methods, such as Prony's method (Hildebrand 1956), Pisarenko's method (Pisarenko 1973), and ESPRIT (Roy and Kailath 1989). However, they are out of the scope of this thesis.

2.2.2 Partial tracking

Partial tracking (PT) is a technique that aims to build partial trajectories by linking the spectral peaks with estimated sinusoidal parameters across slices. This technique plays a significant role in various applications such as sound synthesis (McAulay and Quatieri 1986), music transcription (Klapuri and Davy 2006), and audio inpainting (Lagrange et al. 2005). The first PT method was proposed by McAulay and Quatieri (1986) for speech analysis and synthesis. The McAulay-Quatieri partial tracking (MQ-PT) method applies a heuristic greedy algorithm to find the nearest peak in frequency in the next slice to connect. In this process, some unmatched partials at the next slice are considered as "birth", while some partials that cannot be continued to the next slice are considered as "death". However, this MQ-PT method does not guarantee an optimal solution for the partial tracking problem, and may result in incorrect trajectories. Recently, Neri and Depalle (2018) proposed a fast algorithm for partial tracking based on linear programming. The approach treats partial tracking as a combinatorial optimization problem to obtain the optimal connections between peaks by minimizing connection costs, instead of finding local optimal solutions as in the greedy MQ-PT method. This approach improves both tracking accuracy and robustness compared to the greedy MQ-PT method. We will describe these two methods in detail in the following sections.

2.2.2.1 Greedy approach

The MQ-PT method relies on a heuristic greedy algorithm to assign spectral peaks from one slice to the next based on a cost of assignment (McAulay and Quatieri 1986). The cost of assigning a peak i with frequency $f_i^{[k-1]}$ at slice $k-1$ to peak j with frequency $f_j^{[k]}$ at slice k is defined as:

$$C_{ij} = \left| f_i^{[k-1]} - f_j^{[k]} \right|, \tag{2.42}$$

which measures the frequency difference between the peaks. A matching threshold Δ_f will be defined to determine whether there is a continuation of a partial or not.

The partial tracking process is described as follows:

- **Search:** For each peak i at slice $k-1$, search for peaks at slice k such that the costs of assigning them to the peak i are lower than the predefined matching threshold Δ_f.
- **Death:** If none of the peaks at slice k satisfy this condition, then the partial trajectory associated with peak i is considered "dead" upon entering slice k.
- **Match:** If one or more peaks satisfy this condition. The peak j at slice k with the smallest cost C_{ij} among these peaks is selected to form a candidate match (i, j) between peak i and peak j.

- **Selection:** For each candidate match (i, j), if there exists another peak h at slice $k-1$ such that $C_{hj} < C_{ij}$, then peak j should be assigned to peak h instead of peak i, since peak h is closer to peak j in frequency. In this case, the candidate match (i, j) fails and peak i has to seek other potential assignments that satisfy previous steps or is considered "dead".
- **Birth:** After checking all candidate matches, if there remain unmatched peaks at slice k, these remaining peaks are considered as "born" and their partial trajectories start.

One problem with this method is that it does not guarantee an optimal solution for the partial tracking problem. For example, consider three parallel ascending chirps (sweeping signals), as shown in Figure 2.6. The greedy algorithm will fail to match the bottom left and top right peaks, and will connect the two middle peaks to the wrong partials, resulting in incorrect trajectories.

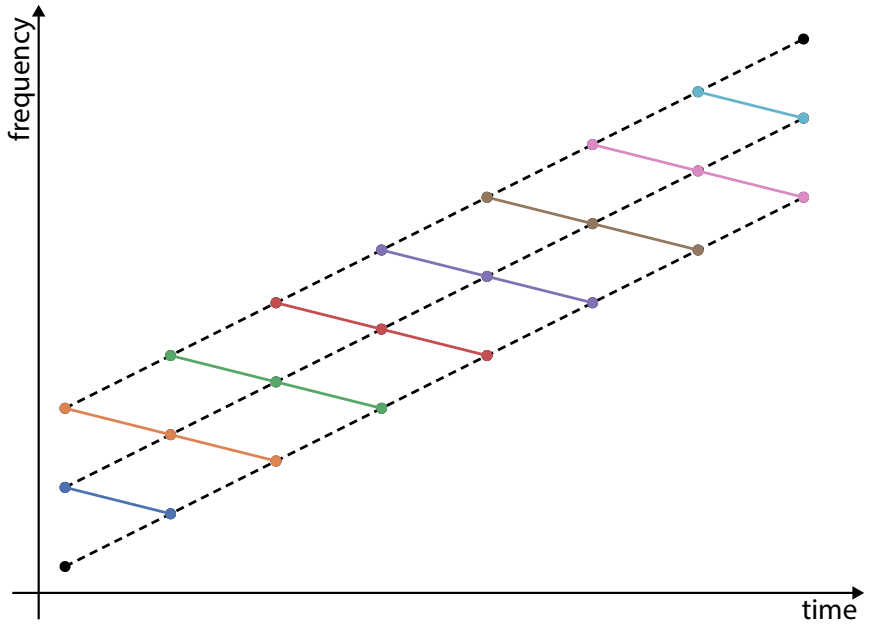


Figure 2.6 Incorrect trajectories in the MQ-PT greedy-based partial tracking process. The circles denote the estimated spectral peaks, dash lines are the true partials, which represent three parallel ascending chirps, the colored lines are the found tracked partials using the heuristic greedy algorithm.

This issue can be partially solved by alternative methods, such as the hidden Markov model-based method (Depalle et al. 1993) and the linear programming-based method (Neri and Depalle 2018). These alternative techniques improve both tracking accuracy and robustness compared to the greedy approach.

2.2.2.2 Combinatorial optimization approach

The partial tracking can be considered as an assignment problem, which is a combinatorial optimization problem that aims to find the optimal assignment of elements from one set to another set based on their assignment costs (Neri and Depalle 2018). Suppose that the first set S_1 contains N_1 elements, the second set S_2 contains N_2 elements. The assignment problem can be formalized as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} C_{ij}X_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{N_1} X_{ij} = 1 \qquad j = 1, \dots, N_2 \\ & \sum_{j=1}^{N_2} X_{ij} = 1 \qquad i = 1, \dots, N_1 \end{aligned} \tag{2.43}$$

where C_{ij} is the cost of assigning element i in set S_1 to element j in set S_2, X_{ij} is a binary variable indicating the assignment, which is set to 1 if element i is assigned to element j and 0 otherwise.

The optimal solution of the assignment problem can be obtained by the Hungarian algorithm in polynomial time (Kuhn 1955).

The method introduces continuity constraints between the midpoints of consecutive slices by incorporating the frequency and amplitude differences between the peaks at the midpoint of the slices in the cost function. The frequency and amplitude differences between peak i and j in slice k is defined as:

$$\Delta f_{ij}^{[k]} = f_i^{[k-1]}[H/2] - f_j^{[k]}[-H/2] \tag{2.44}$$

$$\Delta a_{ij}^{[k]} = a_i^{[k-1]}[H/2] - a_j^{[k]}[-H/2] \tag{2.45}$$

where H is the hop size, $f_i^{[k]}[n]$ and $a_i^{[k]}[n]$ are the instantaneous frequency and log-amplitude of partial i in slice k, respectively.

These constraints lead to two types of assignments: *useful* assignments and *spurious* assignments. Useful assignments are those that satisfy the continuity constraints, while spurious assignments are those that do not satisfy them and are thus ignored in the partial tracking process.

The cost of a useful assignment from peak i in slice $k-1$ to peak j in slice k is defined as:

$$C_{ij}^{\text{useful}[k]} = 1 - \exp\left(-\frac{{\Delta f_{ij}^{[k]}}^2}{2\sigma_f^2} - \frac{{\Delta a_{ij}^{[k]}}^2}{2\sigma_a^2}\right). \tag{2.46}$$

The parameters σ_f^2 and σ_a^2 are the variances of the frequency and amplitude distributions, respectively, which are computed as:

$$\sigma_f^2 = \frac{\zeta_f^2}{2\ln(\delta-2) - 2\ln(\delta-1)} \tag{2.47}$$

and

$$\sigma_a^2 = \frac{\zeta_a^2}{2\ln(\delta-2) - 2\ln(\delta-1)} \tag{2.48}$$

where ζ_f and ζ_a are predefined thresholds that control the range of the frequency and amplitude matching, respectively. δ is the parameter that controls the trade-off between useful and spurious assignments.

The cost of a spurious assignment is defined as:

$$C_{ij}^{\text{spurious}} = 1 - (1-\delta)C_{ij}^{\text{useful}}. \tag{2.49}$$

To obtain both useful and spurious assignments using the Hungarian algorithm, the cost matrix can be defined as:

$$C_{ij} = \min\{C_{ij}^{\text{useful}}, C_{ij}^{\text{spurious}}\}. \tag{2.50}$$

Consequently, assignments $X_{ij} = 1$ with $C_{ij} = C_{ij}^{\text{useful}}$ are considered as useful assignments, while those with $C_{ij} = C_{ij}^{\text{spurious}}$ are categorized as spurious assignments.

If a useful assignment is not connected to any previous trajectories, this assignment is considered as a born partial. If a previous trajectory does not correspond to any useful assignments in the current slice, the partial is considered as dead.

This method achieves high-quality partial tracking results across various types of signals with more robustness and less computational complexity than previous methods (Neri and Depalle 2018).

2.2.3 Sinusoidal synthesis

After obtaining the partials from the partial tracking algorithm described in Section 2.2.2, the signal can be reconstructed across slices using the overlap-add technique. The reconstructed signal in the k-th slice is given by:

$$\hat{y}^{[k]}[n] = \Re\left\{\sum_{p=1}^{P^{[k]}} \hat{a}_p^{[k]} \exp\left(j\left(2\pi n \hat{f}_p^{[k]} + \hat{\varphi}_p^{[k]}\right)\right)\right\} \tag{2.51}$$

and then each frame is overlap-added to its overlapping neighbors.

However, this simple reconstruction process suffers from a limitation. The sinusoidal parameters are fixed within each slice, which can lead to poor reconstruction quality. Therefore, interpolation of sinusoidal parameters is required for better reconstruction.

In this section, we describe the synthesis technique for sinusoidal modeling proposed by McAulay and Quatieri (1986), which involves the interpolation of both amplitude and phase parameters. The linear interpolation of amplitude is performed for each slice using:

$$\hat{a}_p^{[k]}[n] = \hat{a}_p^{[k]} + \frac{n}{H}(\hat{a}_p^{[k+1]} - \hat{a}_p^{[k]}) \tag{2.52}$$

where H represents the hop size, and n varies from 0 to $H-1$.

For phase interpolation, a cubic polynomial is used in continuous time*:

$$\hat{\varphi}_p^{[k]}(t) = \varphi_p^{[k]} + \omega_p^{[k]} t + \alpha_p^{[k]} t^2 + \beta_p^{[k]} t^3. \tag{2.53}$$

Four equations are constructed from the estimated frequencies and phases at the beginning and end of the slice to determine the parameters of the cubic polynomial:

$$\hat{\varphi}_p^{[k]}(0) = \varphi_p^{[k]} = \hat{\varphi}_p^{[k]} \tag{2.54}$$

$$\hat{\varphi}_p'^{[k]}(0) = \omega_p^{[k]} = 2\pi \hat{f}_p^{[k]} \tag{2.55}$$

$$\hat{\varphi}_p^{[k]}(T) = \varphi_p^{[k]} + \omega_p^{[k]} T + \alpha_p^{[k]} T^2 + \beta_p^{[k]} T^3 = \hat{\varphi}_p^{[k+1]} + 2\pi M \tag{2.56}$$

$$\hat{\varphi}_p'^{[k]}(T) = \omega_p^{[k]} + 2\alpha_p^{[k]} T + 3\beta_p^{[k]} T^2 = 2\pi \hat{f}_p^{[k+1]} \tag{2.57}$$

where T is the end time of a slice (H times the sample period), M is an integer used for phase unwrapping, $\hat{\varphi}_p'^{[k]}$ represents the first-order derivative of $\hat{\varphi}_p^{[k]}$.

To determine the appropriate value of M, a smoothness constraint is applied to the frequency function. This ensures that the optimal M leads to a maximally smooth frequency trajectory, or in other words, a trajectory with the minimum frequency variance. The optimization problem is formalized as:

$$\tilde{M} = \arg\min_M \int_0^T \left(\hat{\varphi}_p''^{[k]}(t)\right)^2 dt \tag{2.58}$$

where $\hat{\varphi}_p''^{[k]}$ denotes the second-order derivative of $\hat{\varphi}_p^{[k]}$. The solution $\tilde{M}$ is a real number, and the optimal M is chosen as the closest integer to $\tilde{M}$.

By incorporating both the linear amplitude interpolation and the cubic phase interpolation with maximal smoothness, the final synthesized signal in slice k is given by:

$$\hat{y}[n] = \Re\left\{\sum_{p=1}^{P^{[k]}} \hat{a}_p^{[k]}[n] \exp\left(j\hat{\varphi}_p^{[k]}[n]\right)\right\} \qquad kH \le n < (k+1)H. \tag{2.59}$$

*The time t is not the continuous time of the signal, but is normalized between 0 and H times the sample period.

This synthesis technique is capable of reconstructing diverse types of signals (McAulay and Quatieri 1986), and it will be used in our hybrid approach.

2.2.4 Sinusoidal inpainting models

One of the earliest methods that applies sinusoidal modeling to solve audio inpainting problem was proposed by Maher (1993). This method matches all the partials on both sides of the gap based on the linear extrapolation, and employs cubic polynomial extrapolation to estimate the amplitude and frequency of a partial across the gap. However, this straightforward approach does not consider the possibility of partial birth and death in the gap, and can produce false matches if the parameters of the partials are modulated.

A more advanced method is described and will be used in our hybrid approach, which employs linear prediction to estimate the parameters of partials in the gap, and interpolates matched partials or extrapolates unmatched partials based on a specific technique (Lagrange et al. 2005). This approach allows for the birth and death of partials and handles the phase discontinuity problem at the gap boundaries. The method consists of three main steps: partial extension, partial matching, and partial interpolation. The details of each step are given below.

In the partial extension stage, partials that appear on one side of the gap are extended to the other side of the gap using Burg method for linear prediction, which is a technique that estimates the optimal autoregressive parameters of a signal by minimizing the forward and backward prediction errors recursively. The Burg method will be explained in Section 2.3.

In the partial matching stage, the extended partials from both sides of the gap are matched based on the normalized Euclidean distance between two predicted partials in the gap. The normalized Euclidean distance of frequency between the predicted partials $\hat{p}_i$ and $\hat{p}_j$ is defined as:

$$\bar{d}_f(\hat{p}_i, \hat{p}_j) = \frac{\|\hat{\mathbf{f}}_i - \hat{\mathbf{f}}_j\|_2 \,/\sqrt{K_R - K_L + 1}}{1 + \sigma(\hat{\mathbf{f}}_i) + \sigma(\hat{\mathbf{f}}_j)} \tag{2.60}$$

where K_L and K_R are the first and last unreliable slices in the gap, $\hat{\mathbf{f}}_i = \left[\hat{f}_i^{[K_L]}, \ldots, \hat{f}_i^{[K_R]}\right]^T$, and $\sigma(\hat{\mathbf{f}}_i)$ is the standard deviation of the vector $\hat{\mathbf{f}}_i$. The normalized Euclidean distance of amplitude can be defined in a similar way.

If both normalized Euclidean distances of a partial pair (p_i, p_j) are smaller than their frequency and amplitude thresholds ($\bar{d}_f(\hat{p}_i, \hat{p}_j) < T_f$, $\bar{d}_a(\hat{p}_i, \hat{p}_j) < T_a$), these two partials are merged into one partial p_m, and the partials p_i and p_j are removed from the list.

In the partial interpolation stage, the parameters of the matched partials are interpolated in the gap using a crossfade technique with an asymmetric window function. For the merged partial

p_m, the predicted frequency $\hat{f}_m$ is obtained by crossfading between $\hat{f}_i$ and $\hat{f}_j$ with an asymmetric window function $\tilde{\psi}$, which is defined as:

$$\hat{f}_m^{[k]} = \tilde{\psi}\left(\frac{k-K_L+1}{K_R-K_L+2}\right)\hat{f}_i^{[k]} + \left(1-\tilde{\psi}\left(\frac{k-K_L+1}{K_R-K_L+2}\right)\right)\hat{f}_j^{[k]}. \tag{2.61}$$

The asymmetric window function is derived from time-wrapping a symmetric cosine window $\breve{\psi}$:

$$\breve{\psi}(t) = \frac{1+\cos(\pi(1+t))}{2} \tag{2.62}$$

where t is continuous time ranged from 0 to 1. The time-wrapping is related to the lengths of p_i and p_j, and satisfies a constraint that the midpoint of the window corresponds to the relative length of partial p_i, which is formalized by:

$$\tilde{\psi}\left(\frac{N_{p_i}}{N_{p_i}+N_{p_j}}\right) = \frac{1}{2} \tag{2.63}$$

where N_{p_i} and N_{p_j} are the lengths of partial p_i and p_j. The time-wrapping function $\varrho(N_{p_i}, N_{p_j})$ is defined as:

$$\varrho(N_{p_i}, N_{p_j}) = \frac{\ln(1/2)}{\ln\left\{\breve{\psi}\left(N_{p_i}/(N_{p_i}+N_{p_j})\right)\right\}}. \tag{2.64}$$

Therefore, the asymmetric window is defined as:

$$\tilde{\psi} = \begin{cases} \breve{\psi}(t)^{\varrho(N_{p_i},N_{p_j})} & N_{p_i} > N_{p_j} \\ 1-\left(1-\breve{\psi}(t)\right)^{\varrho(N_{p_j},N_{p_i})} & \text{otherwise} \end{cases}. \tag{2.65}$$

The amplitude of a partial usually has larger modulation than frequency, so the proposed method constrains the predicted amplitudes $\hat{a}_i$ and $\hat{a}_j$ for a smooth transition. The predicted amplitude $\hat{a}_i$ at the end of the gap should equal to the local average amplitude of partial p_j. Therefore, the constrained amplitude of partial p_i is:

$$\breve{a}_i^{[k]} = \hat{a}_i^{[k]} + \frac{k-K_L+1}{K_R-K_L+2}\left(\frac{\sum_{q=1}^{l_j}\hat{a}_j^{[K_R+q]}}{l_j} - \hat{a}_i^{[K_R+1]}\right) \tag{2.66}$$

where $l_j = \min\{N_{p_j}, L_{\text{avg}}\}$ is the number of slices for calculating the local average of partial p_j from slice K_R+1 to slice K_R+l_j, L_{avg} is a parameter for controlling the range for the calculation. Similarly, the constrained amplitude of partial p_j is:

$$\breve{a}_j^{[k]} = \hat{a}_j^{[k]} + \frac{k-K_R-1}{K_R-K_L+2}\left(\frac{\sum_{q=1}^{l_i}\hat{a}_i^{[K_L-q]}}{l_i} - \hat{a}_j^{[K_L-1]}\right). \tag{2.67}$$

The amplitude of merged partial p_m is then constructed using the same asymmetric window:

$$\hat{a}_m^{[k]} = \tilde{\psi}\left(\frac{k-K_L+1}{K_R-K_L+2}\right)\breve{a}_i^{[k]} + \left(1-\tilde{\psi}\left(\frac{k-K_L+1}{K_R-K_L+2}\right)\right)\breve{a}_j^{[k]}. \tag{2.68}$$

The phase of a partial p_m in the gap is constructed iteratively from left to right:

$$\hat{\varphi}_m^{[K_L]} = \varphi_m^{[K_L-1]} + \frac{H}{f_s}\pi\left(f_m^{[K_L-1]} + \hat{f}_m^{[K_L]}\right) \tag{2.69}$$

$$\hat{\varphi}_m^{[K_L+k]} = \hat{\varphi}_m^{[K_L+k-1]} + \frac{H}{f_s}\pi\left(\hat{f}_m^{[K_L+k-1]} + \hat{f}_m^{[K_L+k]}\right) \tag{2.70}$$

where $k \in [1, K_R - K_L + 1]$, f_s is the sample rate. However, the predicted phase at slice $K_R + 1$ may not equal to the actual phase from the partial p_j, leading to a phase discontinuity. The phase prediction error at $K_R + 1$ is defined as:

$$e_{\varphi_m} = \mathrm{wrap}(\hat{\varphi}_m^{[K_R+1]} - \varphi_m^{[K_R+1]}) \tag{2.71}$$

where $\mathrm{wrap}(\varphi)$ is the phase wrapping function that maps any angle φ to the range $[-\pi, \pi)$. An error spreading method is proposed to spread the prediction error into each slice in the gap to reduce the discontinuity. The phases after spreading are computed by:

$$\tilde{\varphi}_m^{[k]} = \hat{\varphi}_m^{[k]} + \frac{k-K_L+1}{K_R-K_L+2}\,\mathrm{wrap}(e_{\varphi_m}). \tag{2.72}$$

For the unmatched partials, the amplitude should decay to zero in the gap if there is a partial death, which can be formalized by:

$$\tilde{a}_i^{[k]} = \hat{a}_i^{[k]} - \frac{k-K_L+1}{l_D}\max\{\hat{a}_i^{[K_L+l_D-1]}, 0\} \tag{2.73}$$

where l_D is the maximum length of partial death. Similarly, the amplitude should increase from zero in the gap if there is a partial birth, which can be formalized by:

$$\tilde{a}_j^{[k]} = \hat{a}_j^{[k]} - \frac{K_R-k+1}{l_B}\max\{\hat{a}_j^{[K_R-l_B+1]}, 0\} \tag{2.74}$$

where l_B is the maximum length of partial birth. The frequency and phase of the unmatched partials are not modified in the gap.

The study demonstrates that the proposed method achieves improved reconstruction quality for gaps up to hundreds of milliseconds based on subjective listening tests, especially for complex polyphonic signals with modulations (Lagrange et al. 2005).

A more recent method extends the previous work by interpolating both the sinusoidal and the noisy residual components of the audio signals, which makes the method more suitable for a wider range of audio signals (Lukin and Todd 2008). The method also proposes a technique to interpolate the partials based on decomposing the frequency and amplitude trajectories of a partial to a sum of sinusoids, and use one interpolated partial with different frequency-shifts to build consistent harmonic structure. This method improves reconstruction quality and robustness for noisy and harmonic signals (Lukin and Todd 2008).

2.2.5 Limitation of sinusoidal inpainting models

The sinusoidal inpainting method utilizes prior knowledge to extract higher-level structural information (partials) from audio signals than atoms in sparse decomposition, enabling it to reconstruct longer gaps without energy loss and to handle modulated signals. However, there are still some limitations related to this type of approaches.

One limitation is that the sinusoidal modeling is not suitable for reconstructing the noise component of the audio signals. Some methods only synthesize the sinusoidal component and disregard the noise component, which can cause perceptual artifacts (Lagrange et al. 2005). Some methods use alternative techniques to deal with the noise, such as autoregressive modeling (Lukin and Todd 2008), which will be discussed in Section 2.3.

Another limitation is that some sinusoidal inpainting methods using linear prediction to extend the partial trajectory struggle to interpolate chirp-like partials, which frequency variations do not have an autoregressive structure. This is because these methods assume that the partials are stationary or slowly varying, while a chirp contains a trend that is hard to capture by autoregressive models. This can result in inaccurate interpolation results, as illustrated in Figure 2.7.

Moreover, the performance of the sinusoidal inpainting method relies heavily on the quality of parameter estimation and partial tracking algorithms. For signals with crossing partials or significant noise, the tracked partials are often fragmented. In this case, a partial is split into multiple partials and each has erroneous values at both ends. This makes it difficult to predict the partials in the gap using less reliable information from shortened partials due to fragmentation, which can impair the quality of reconstruction. The fragmentation of tracked partials is shown in Figure 2.8.

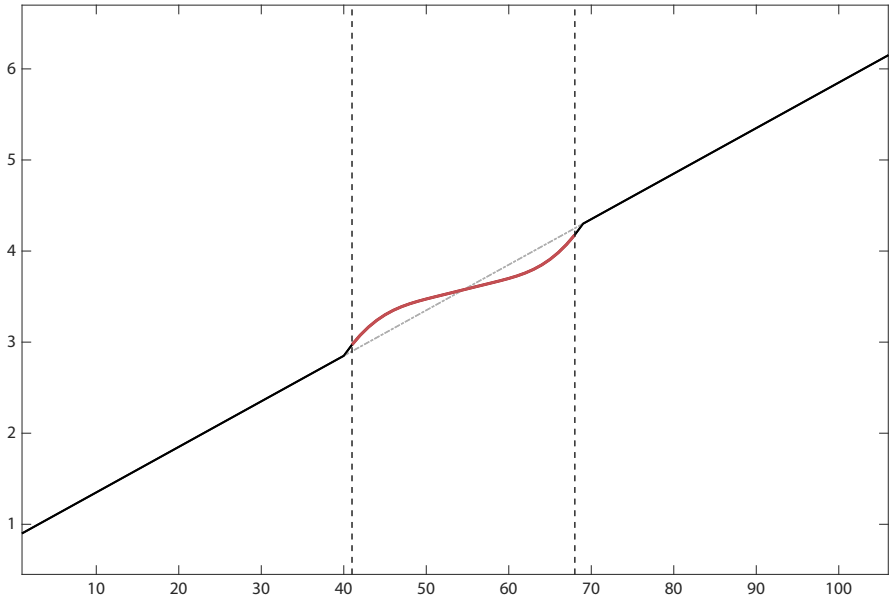


Figure 2.7 Interpolation of a linearly increasing curve with a gap using the Burg method with 32 poles. The dash lines indicate the start and end of the gap. The red solid line in the gap is the prediction result using the Burg method, which shows an autoregressive structure that oscillates around it.

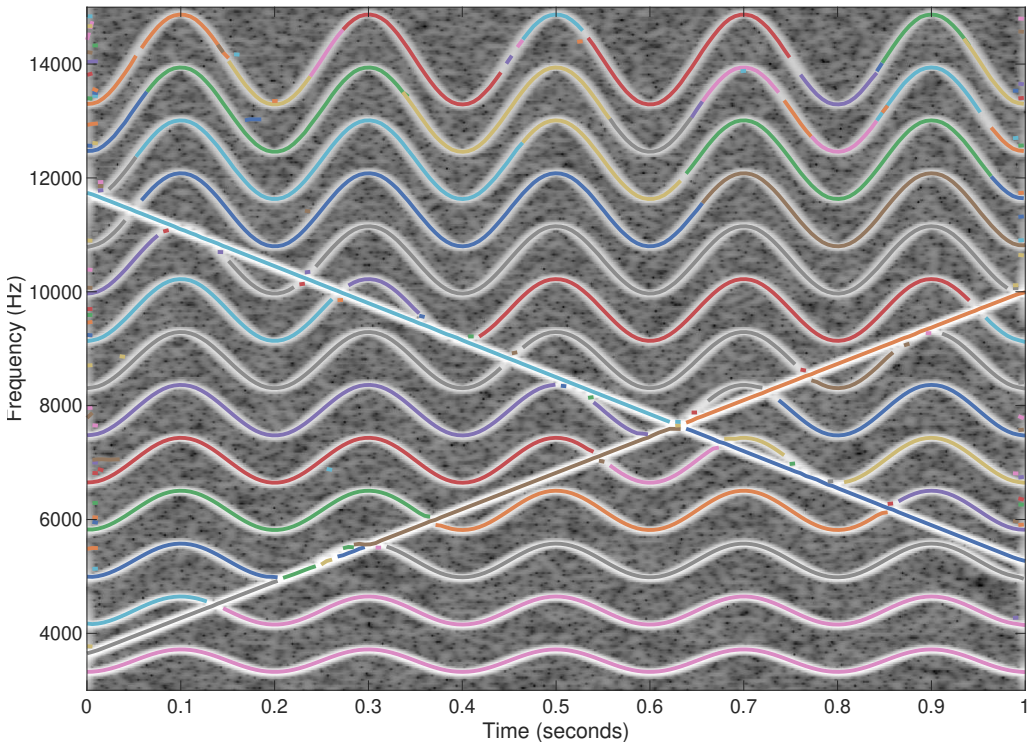


Figure 2.8 Fragmentation of tracked partials in a synthesized audio signal consisting of a harmonic signal with a vibrato, mixed with chirps plus background noise. All parameters are unchanged as in Neri and Depalle (2018).

2.3 Autoregressive Modeling

Autoregressive (AR) modeling is a model for both analysis and synthesis of audio signals. The model assumes that each sample of a signal is a linear combination of its past samples. A p-th order AR model is defined as:

$$\hat{y}[n] = -\sum_{k=1}^{p} a_k y[n-k] \tag{2.75}$$

where a_k are the AR coefficients and $\hat{y}[n]$ is the modeled signal.

By incorporating an input signal, the autoregressive structure defines the structure of a system that constrains how an input signal evolves over time. A p-th order AR structure is defined as:

$$y[n] = x[n] - \sum_{k=1}^{p} a_k y[n-k] \tag{2.76}$$

where $x[n]$ is the input signal, and $y[n]$ is the output signal.

In the deterministic context, the AR structure is an infinite impulse response (IIR) filter processing an input signal $x[n]$. In the stochastic context, the AR structure represents a random process that assumes that each sample in the sequence is a linear combination of its previous values plus a random variable. This random variable represents the unpredictable part of the signal, and it is usually modeled as white noise $\epsilon[n]$.

If an observed signal $y[n]$ is modeled by an AR model $\hat{y}[n]$, the error (or residual) $e[n]$ of the AR model can be calculated by taking the difference between the actual signal $y[n]$ and its model $\hat{y}[n]$:

$$e[n] = y[n] - \hat{y}[n] = y[n] + \sum_{k=1}^{p} a_k y[n-k]. \tag{2.77}$$

If we compare this equation with the AR structure in Eq. (2.76), the error can be considered as the input $x[n]$ (or a white noise $\epsilon[n]$) of the system.

In addition, the AR structure can be expressed in the frequency domain. Using z-transform, Eq. (2.76) can be rewritten as:

$$Y(z) = \frac{X(z)}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{2.78}$$

where $Y(z)$ and $X(z)$ are the z-transforms of $y[n]$ and $x[n]$, respectively. Therefore, the p-th order AR structure can be considered as an all-pole filter with p poles. The transfer function $H(z)$ of the all-pole filter is:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.79}$$

AR modeling can be considered from two perspectives, which will be explained in Section 2.3.1. The techniques of obtaining the AR coefficients will be described in Section 2.3.2 and 2.3.3. Section 2.3.4 provides two AR-based models for addressing the audio inpainting problem.

2.3.1 Dual perspectives of AR modeling

Autoregressive modeling can be applied from two different perspectives for achieving different goals: synthesis and analysis. In the synthesis perspective, the goal is to generate a signal $y[n]$ from an input signal $x[n]$ using a given AR structure, which is parameterized in terms of coefficients a_k. The output signal is thus obtained by filtering the input signal $x[n]$ with the IIR filter which transfer function is $H(z)$ in Eq. (2.79). In practice, we usually choose the input signal to be a white noise $\epsilon[n]$ with zero mean and unit variance. Signal $y[n]$ can be formulated as:

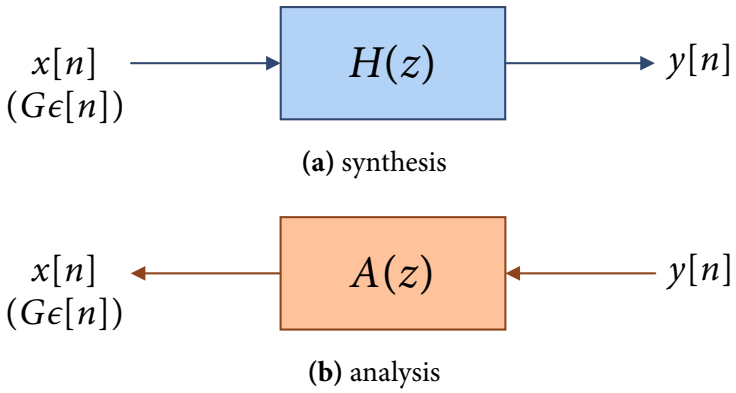
$$y[n] = G\epsilon[n] - \sum_{k=1}^{p} a_k y[n-k]. \tag{2.80}$$

Therefore, a scaling factor G should be added to scale the input noise to change the variance of the output signal. This procedure is illustrated in Figure 2.9a.

One application of this synthesis perspective is the source-filter synthesis model. Source-filter consists of synthesizing a sound by filtering a spectrally rich input sound source. It was originally proposed to explain the production of speech (Fant 1971), and then was extended to speech or musical instrument sound synthesis (Makhoul 1975; Caetano and Rodet 2012). The sound source can be various types of signals, such as a pulse train or noise. The filters are usually obtained by analyzing the acoustic features of a target sound. In this thesis, this source-filter technique is applied to reconstruct the time-varying noise component within the gap.

In the analysis perspective, we assume that a given signal $y[n]$ is generated by an input signal $x[n]$ filtered with a specific AR structure, and the goal is to estimate the AR structure. Suppose that the p-th-order AR structure is estimated with the right parameters a_k. Then, we can write the prediction error signal $e[n]$ in the form of Eq. (2.77), which ideally should be white noise with a mean of zero and a variance of G^2 (G is the scaling factor used in the synthesis perspective). In other words, the error signal $e[n]$ (or input signal $x[n]$) can be obtained by filtering the given signal $y[n]$ with an all-zero finite impulse response (FIR) filter, which is known as the *inverse filter* (Makhoul 1975). The transfer function of this FIR filter $A(z)$ can be deduced from rewriting Eq. (2.77) using the z-transform:

$$A(z) = \frac{E(z)}{Y(z)} = 1 + \sum_{k=1}^{p} a_k z^{-k}. \tag{2.81}$$



(a) synthesis

(b) analysis

Figure 2.9 Dual perspectives of AR modeling.

This procedure is illustrated in Figure 2.9b.

One application of this analysis perspective is the spectral envelope estimation, which is to analyze a signal to approximate its spectral envelope from estimated AR coefficients (Makhoul 1975).

By reorganizing Eq. (2.81) and replacing z with $e^{j\omega}$, we get:

$$Y(e^{j\omega}) = \frac{E(e^{j\omega})}{1 + \sum_{k=1}^{p} a_k e^{-j\omega k}}. \tag{2.82}$$

The power spectral density (PSD) of the output signal $y[n]$ is then defined as:

$$P_y(\omega) = |Y(e^{j\omega})|^2 = \frac{\left|E(e^{j\omega})\right|^2}{\left|1 + \sum_{k=1}^{p} a_k e^{-j\omega k}\right|^2}. \tag{2.83}$$

As the transfer function of the estimated AR structure is:

$$H(z) = \frac{G}{A(z)} = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.84}$$

The estimated PSD from the AR structure filtering the white noise of variance 1 is given by:

$$\hat{P}_y(\omega) = |H(e^{j\omega})|^2 = \frac{G^2}{\left|1 + \sum_{k=1}^{p} a_k e^{-j\omega k}\right|^2}. \tag{2.85}$$

The optimal AR structure, which is obtained by minimizing the variance of the error, can be reformulated in the spectral domain:

$$\mathrm{SSE}_y = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(e^{j\omega})|^2 \, d\omega = \frac{G^2}{2\pi} \int_{-\pi}^{\pi} \frac{P_y(\omega)}{\hat{P}_y(\omega)} \, d\omega. \tag{2.86}$$

According to this equation, minimizing the error corresponds to minimizing the ratio of the integrated PSD of the signal to its approximation on the unit circle, thereby providing an estimation of the spectral envelope of signal $y[n]$ with a smooth curve with $\lceil p/2 \rceil$ peaks (Hayes 1996).

However, the shape of the estimated spectral envelope depends on the method used to estimate the AR coefficients (Kay and Marple 1981). Therefore, one should choose an appropriate estimation method and model order according to their application and the characteristics of the input signal. Section 2.3.3 will discuss some of these estimation methods.

2.3.2 Evaluating AR coefficients from the correlation of the signal

In this section, we will derive the AR coefficients a_k of an AR model. Suppose that the input signal is a white noise $\epsilon[n]$ with zero mean and unit variance. The AR structure can be compactly rewritten as:

$$\sum_{k=0}^{p} a_k y[n-k] = \epsilon[n] \tag{2.87}$$

where $a_0 = 1$.

Multiplying both sides by $y^*[n-l]$ for $l = 0, 1, \dots, p$ and taking the expectation, we obtain:

$$\sum_{k=0}^{p} a_k \mathrm{E}\{y[n-k]y^*[n-l]\} = \mathrm{E}\{\epsilon[n]y^*[n-l]\} \tag{2.88}$$

where $\mathrm{E}\{x\}$ is the expectation of x, and $y^*[n]$ is the complex conjugate of $y[n]$. Since $\epsilon[n]$ is uncorrelated with past values of $y[n]$, we have $\mathrm{E}\{\epsilon[n]y[n-l]\} = 0$ for any $l > 0$. This equation can be rewritten with the autocorrelation function:

$$\sum_{k=0}^{p} a_k r_y(l-k) = \begin{cases} \sum_{k=0}^{p} r_y(-k) & \text{for } l = 0 \\ 0 & \text{for } l > 0 \end{cases} \tag{2.89}$$

where $r_y(l)$ is the autocorrelation of signal $y[n]$ with lag l. These equations are referred to as the *Yule-Walker equations*. The equations can also be written in matrix form as:

$$\underbrace{\begin{bmatrix} r_y(0) & r_y(-1) & \cdots & r_y(-p) \\ r_y(1) & r_y(0) & \cdots & r_y(-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ r_y(p) & r_y(p-1) & \cdots & r_y(0) \end{bmatrix}}_{\mathbf{R_y}} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} \sum_{k=0}^{p} r_y(-k) \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{b}} \tag{2.90}$$

or compactly:

$$\mathbf{R}_y \mathbf{a} = \mathbf{b}. \tag{2.91}$$

Since the autocorrelation matrix $\mathbf{R}_y$ is a Toeplitz matrix, the *Levinson-Durbin recursion* can be applied to solve the AR coefficients $\mathbf{a}$ efficiently (Kay 1988).

2.3.3 Estimating AR coefficients from the observed signal samples

In practice, the autocorrelation $r_y(l)$ is unknown, since we do not have all the possible realizations of the signal $y[n]$. Therefore, AR coefficients cannot be obtained and can only be estimated. We usually estimate the AR coefficients from a finite number of samples from one specific realization of a random process. This technique is referred to as *linear prediction*. In this section, we will introduce two methods for estimating the AR coefficients: the autocorrelation method and the Burg method. These methods will be used in subsequent sections.

The autocorrelation method directly estimate the autocorrelation $r_y(l)$ based on the observed N samples of the signal $y[n]$ and use the estimated autocorrelation to solve the Yule-Walker equations. The estimated autocorrelation $\hat{r}_y(l)$ is defined as:

$$\hat{r}_y(l) = \frac{1}{N} \sum_{n=0}^{N-1} y[n] y[n-l]. \tag{2.92}$$

The Burg method, on the other hand, does not rely on solving the Yule-Walker equation but rather expresses the AR structure (inverse filter $A(z)$) as a lattice structure. The estimated AR coefficients are derived from the estimation of reflection coefficients of the lattice structure by minimizing the forward and backward prediction errors iteratively (Hayes 1996).

The forward prediction error $e_k^{\rightarrow}[n]$ at order k is defined as the difference between signal $y[n]$ and its prediction $\hat{y}[n]$:

$$e_k^{\rightarrow}[n] = y[n] - \hat{y}[n] = y[n] + \sum_{l=1}^{k} a_k[l] y[n-l], \tag{2.93}$$

where $a_k[l]$ represents the l-th AR coefficient (a_l) at order k.

The backward prediction error $e_k^{\leftarrow}[n]$ at order k is then defined based on the Levinson-Durbin recursion:

$$e_k^{\leftarrow}[n] = y[n-k] - \hat{y}[n-k] = y[n-k] + \sum_{l=1}^{k} a_k^*[l] y[n-k+l]. \tag{2.94}$$

Using the two prediction errors from the previous order, we can build a cell of lattice structure, which is illustrated in Figure 2.10. The prediction errors for the next order can be calculated by the

following equations:

$$e^{\rightarrow}_{k+1}[n] = e^{\rightarrow}_{k}[n] + \gamma_{k+1} e^{\leftarrow}_{k}[n-1] \tag{2.95}$$

$$e^{\leftarrow}_{k+1}[n] = e^{\leftarrow}_{k}[n-1] + \gamma^{*}_{k+1} e^{\rightarrow}_{k}[n] \tag{2.96}$$

where $e^{\rightarrow}_{0}[n] = e^{\leftarrow}_{0}[n] = y[n]$, and γ_k is referred to as the k-th *reflection coefficient*, which determines the coefficients of the k-th lattice filter.

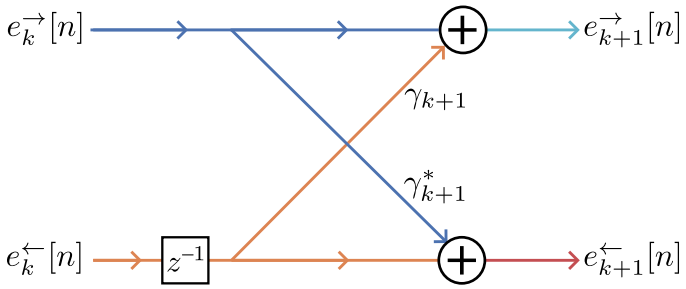


Figure 2.10 Single cell of a FIR lattice structure.

The Burg method aims to estimate the reflection coefficients γ_k based on the average of the sum of the squares of the forward and backward errors:

$$\bar{e}^{\leftrightarrow}_{k} = \frac{1}{2(N-k)} \sum_{n=k}^{N-1} \left(|e^{\rightarrow}_{k}[n]|^2 + |e^{\leftarrow}_{k}[n]|^2 \right) \tag{2.97}$$

where N is the length of signal $y[n]$.

The reflection coefficients γ_k are obtained by differentiating $\bar{e}^{\leftrightarrow}_{k}$ with respect to γ_k, which yields

$$\gamma_k = \frac{-2 \sum_{n=k}^{N-1} e^{\rightarrow}_{k-1}[n] e^{\leftarrow *}_{k-1}[n-1]}{\sum_{n=k}^{N-1} \left(|e^{\rightarrow}_{k-1}[n]|^2 + |e^{\leftarrow}_{k-1}[n-1]|^2 \right)}. \tag{2.98}$$

Then, the AR coefficients a_k can be calculated from the reflection coefficients as follows:

$$a_k[n] = \begin{cases} a_{k-1}[n] + \gamma_k a^{*}_{k-1}[k-n] & n = 1, 2, \dots, k-1 \\ \gamma_k & n = k \end{cases}. \tag{2.99}$$

Since $|\gamma_k| \leq 1$, the Burg method ensures that all estimated poles are on or inside the unit circle, thus the estimated all-pole filter is always stable (Kay 1988).

2.3.4 AR-based inpainting models

One possible use of the AR modeling is to predict the future values of a signal from its past observation, based on estimating the AR coefficients that best capture the underlying temporal relationships within the signal (Makhoul 1975). This predictive aspect of AR modeling in the time domain enables the interpolation and extrapolation of a signal, which can be useful for applications such as audio inpainting. In this section, we will describe two distinct methods for AR-based audio inpainting (Janssen et al. 1986; Etter 1996).

The first method, proposed by Janssen et al. (1986), assumes that the signal is stationary around the missing samples, which allows us to regenerate them based on the AR parameters that are estimated from the reliable neighborhood.

Let $\mathbf{y} = [y[0], y[1], \dots, y[N-1]]^T$ be a signal with some missing samples. The missing samples are denoted by $\mathbf{m} = [y[m_1], y[m_2], \dots, y[m_M]]^T$. Since an AR model can represent one missing sample by other previous samples, the sum of the squares of the prediction error of the AR model is given by:

$$\mathrm{SSE}(\mathbf{a}, \mathbf{m}) = \sum_{n=p}^{N-1} \left| \sum_{l=0}^{p} a_l y[n-l] \right|^2 . \tag{2.100}$$

where $\mathbf{a} = [a_1, \dots, a_p]^T$, $a_0 = 1$.

The goal of this method is to minimize $\mathrm{SSE}(\mathbf{a}, \mathbf{m})$ with respect to both AR parameters $\mathbf{a}$ and missing samples $\mathbf{m}$. However, this is a nontrivial problem. This method seeks to obtain a suboptimal solution based on an iterative approach. At the k-th iteration, we obtain an estimate of $\hat{\mathbf{a}}^{(k)}$ from the previous estimate of $\hat{\mathbf{m}}^{(k-1)}$, and then use that $\hat{\mathbf{a}}^{(k)}$ to obtain the estimate of $\hat{\mathbf{m}}^{(k)}$ for this iteration. This can be achieved by solving the following two equations:

$$\hat{\mathbf{a}}^{(k)} : \frac{\partial \,\mathrm{SSE}(\mathbf{a}, \hat{\mathbf{m}}^{(k-1)})}{\partial \mathbf{a}} = 0 \tag{2.101}$$

$$\hat{\mathbf{m}}^{(k)} : \frac{\partial \,\mathrm{SSE}(\hat{\mathbf{a}}^{(k)}, \mathbf{m})}{\partial \mathbf{m}} = 0. \tag{2.102}$$

This procedure is initialized with $\mathbf{m}^{(0)} = \mathbf{0}$, and terminated when convergence is reached.

The second method, proposed by Etter (1996), uses two separate AR models for the left-sided and right-sided neighborhood, and crossfades two extrapolations to restore the missing samples.

Suppose the signal $\mathbf{y}$ is separated into three segments: left reliable part $\mathbf{y}_L = [y[0], \dots, y[m_L - 1]]^T$, missing part $\mathbf{y}_M = [y[m_L], \dots, y[m_R]]^T$, and right reliable part $\mathbf{y}_R = [y[m_R+1], \dots, y[N-1]]^T$. We estimate the AR parameters $a_k^{\rightarrow}$ for the left reliable part, then calculate the forward prediction

of the missing samples as follows:

$$\widehat{y}^{\rightarrow}[m] = \begin{cases} \sum_{k=1}^{p} a_k^{\rightarrow} \widehat{y}^{\rightarrow}[m-k] & m_L \leq m \leq m_R \\ y[m] & \text{otherwise} \end{cases}. \tag{2.103}$$

Similarly, we estimate the AR parameters $a_k^{\leftarrow}$ for the right reliable part from right to left, then calculate the backward prediction as follows:

$$\widehat{y}^{\leftarrow}[m] = \begin{cases} \sum_{k=1}^{p} a_k^{\leftarrow} \widehat{y}^{\leftarrow}[m+k] & m_L \leq m \leq m_R \\ y[m] & \text{otherwise} \end{cases}. \tag{2.104}$$

The estimated signal is obtained by crossfading the forward and backward prediction:

$$\widehat{y}[m] = \psi\left(\frac{1}{2} + \frac{m - m_L}{2(m_R - m_L + 1)}\right)\widehat{y}^{\rightarrow}[m] + \psi\left(\frac{m - m_L}{2(m_R - m_L + 1)}\right)\widehat{y}^{\leftarrow}[m] \tag{2.105}$$

where $\psi(t) = \frac{1}{2}\left(1 - \cos(2\pi t)\right)$, $t \in [0, 1)$ is the raised-cosine window.

This method does not require the signal to be stationary over the entire missing segment, which leads to better performance for various types of signals, compared with Janssen's method (Etter 1996).

3

Hybrid Inpainting Approach

In the previous chapter, we explored and discussed various existing methods that have been commonly used in the context of audio inpainting. These methods, while providing valuable insights and advancements, often exhibit certain limitations when applied to various audio signals.

In this chapter, we propose a hybrid approach that views an audio signal as a mixture of three components: tonal, transient, and noise. By integrating and refining previous methods for treating the different components, our proposed approach aims to overcome the limitations observed in individual techniques and achieve better reconstruction quality. Figure 3.1 illustrates the overall process of our method.

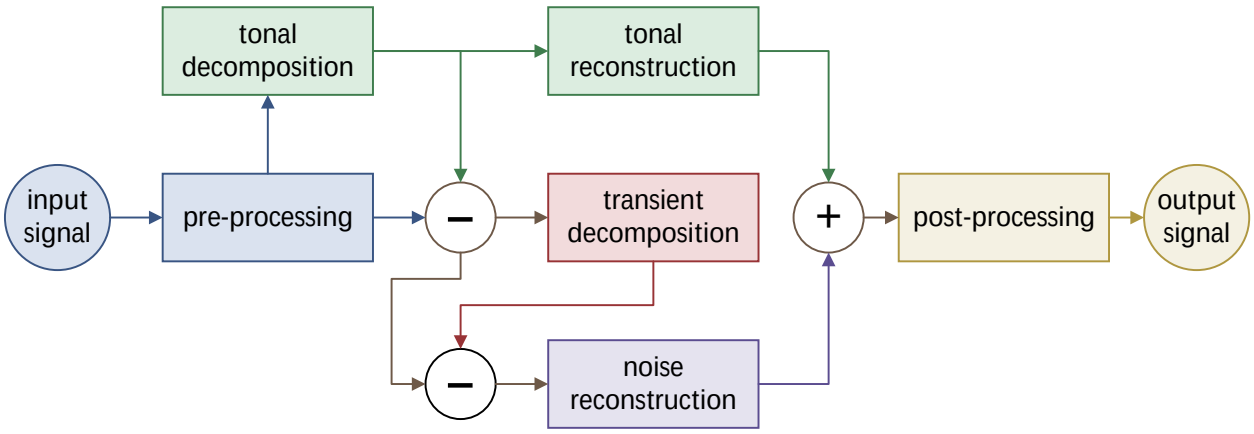


Figure 3.1 The overview structure of the proposed hybrid approach.

The rest of this chapter is organized as follows: Section 3.1 describes the pre-processing procedure for this approach. Section 3.2 discusses the methods used to decompose an audio signal into its tonal component and the remainder. Section 3.3 illustrates how we reconstruct the deterministic (tonal) part using partial analysis and prediction. Section 3.4 explains how to separate the transient and stochastic components of the residual signal. Section 3.5 provides two techniques for restor-

ing the noise component. Section 3.6 demonstrates how to construct the output signal using the previously predicted components.

3.1 Pre-processing

The pre-processing involves shortening and aligning the input signal for future processes. The length of the shortened signal is determined by the signal type and the length of the gap. The center of the gap should be aligned with the window so that the reconstruction energy will be symmetric (Mokrý and Rajmic 2020). The amount of extra shift is referred to as the offset. There are two configurations of offsets: the full offset aligns the center of the gap with the center of a Gabor window, whereas the half offset places the gap's center just in the midpoint of two adjacent windows (Mokrý and Rajmic 2020). In our method, the half offset configuration will be chosen.

The minimum length of the shortened signal is determined based on the offset, window size, and time shift of the window (Rajmic et al. 2015). In order to better estimate time-varying signals, we will set the length longer than the minimal support. The extended length of the left and right neighborhoods N_{neighbor} will be an integer multiple of the time shift of the window.

3.2 Sparse Decomposition of Tonal Part

In our hybrid approach, we will process the three components of audio signals in a different way. In order to process these components, we first decompose the signal into a deterministic part and a residual part by using sparse decomposition with an iterative re-weighting method.

3.2.1 Model selection

We use the analysis variant* of social sparsity for the decomposition of the tonal part. The goal is to solve an optimization problem of the following form:

$$\arg\min_{\mathbf{x}} \left\{ \frac{1}{2}\|\mathbf{M}_{\mathrm{R}}\mathbf{x} - \mathbf{M}_{\mathrm{R}}\mathbf{y}\|_2^2 + \lambda\|\mathbf{\Phi}^{\mathrm{H}}\mathbf{x}\|_1 \right\}. \tag{3.1}$$

The Loris-Verhoeven (LV) algorithm could be applied to solve this problem (Záviška and Rajmic 2022). The algorithm is summarized in Algorithm 7.

*The method of calculating the neighborhood of the social sparsity has been changed, and the details are described in Section 3.2.3.

Algorithm 7. Loris-Verhoeven (LV) algorithm for audio inpainting, where $\mathbf{u}^{(n-1/2)}$ designates the intermediate state of $\mathbf{u}^{(n)}$ in each iteration.

Input: input signal $\mathbf{y}$, reliable mask $\mathbf{M}_{\mathrm{R}}$, synthesis operator $\mathbf{\Phi}$, analysis operator $\mathbf{\Phi}^{\mathrm{H}}$, weights $\mathbf{w}$, parameters σ, τ, ρ

Output: estimated signal $\mathbf{x}$

Initialization: $n = 1, \mathbf{x}^{(0)} = \mathbf{y}, \mathbf{u}^{(0)} = \mathbf{\Phi}^{\mathrm{H}}\mathbf{y}$

1: **repeat**
2: $\quad \mathbf{g}^{(n-1)} = \mathbf{M}_{\mathrm{R}}^{\mathrm{H}}(\mathbf{M}_{\mathrm{R}}\mathbf{x}^{(n-1)} - \mathbf{M}_{\mathrm{R}}\mathbf{y})$
3: $\quad \mathbf{v}^{(n-1)} = \mathbf{u}^{(n-1)} + \sigma\mathbf{\Phi}^{\mathrm{H}}(\mathbf{x}^{(n-1)} - \tau\mathbf{g}^{(n-1)} - \tau\mathbf{\Phi}\mathbf{u}^{(n-1)})$
4: $\quad \mathbf{u}^{(n-1/2)} = \mathbf{v}^{(n-1)} - \sigma\mathcal{S}_{\lambda/\sigma,\mathbf{w}}(\mathbf{v}^{(n-1)}/\sigma)$
5: $\quad \mathbf{x}^{(n)} = \mathbf{x}^{(n-1)} - \rho\tau(\mathbf{g}^{(n-1)} + \mathbf{\Phi}\mathbf{u}^{(n-1/2)})$
6: $\quad \mathbf{u}^{(n)} = \mathbf{u}^{(n-1)} + \rho(\mathbf{u}^{(n-1/2)} - \mathbf{u}^{(n-1)})$
7: $\quad n = n + 1$
8: **until** stopping condition

3.2.2 Reweighting method

We propose an iterative reweighting method for the sparse decomposition process to penalize unwanted frequencies in noise and preserve more energy of the partials. The shrinkage thresholds for each atom depend on both λ and the weight w_k. For example, the Persistent Empirical Wiener (PEW) shrinkage function with weight coefficient w_k (Siedenburg et al. 2014) is:

$$\mathcal{S}_{\lambda,\mathbf{w}}(x_k) = x_k \left(1 - \frac{\lambda^2 \cdot w_k}{\|\mathcal{N}(x_k)\|_2^2}\right)^+ \tag{3.2}$$

To implement the iterative reweighting method, we calculate the autocorrelation of both left (x_{L}) and right (x_{R}) parts of the reliable neighborhood:

$$r_x(n) = \frac{1}{N}\mathrm{E}\{x_{n+m}x_m^*\} = \begin{cases} \frac{1}{N}\sum_{m=0}^{N-n+1} x_{n+m}x_m^* & m \geq 0 \\ r_x^*(-m) & m < 0 \end{cases} \tag{3.3}$$

where $\mathrm{E}\{x\}$ is the expected value of x. The range of the autocorrelation signal is from $-(N-1)$ to $N-1$.

Then we estimate the PSD of these two signals, using the Burg's AR method with p_{tonal} poles (see Section 2.3). The corresponding PSD will be obtained by calculating the frequency response from the AR coefficients. The number of data points of the frequency response should match the number of the frequency bins of a frame. The PSD curve $p[k]$ is shown in Figure 3.2a.

Since the energy of high frequency components is low and we want a flat reweighting curve, we flatten the PSD curve. To achieve that, we track the prominent peaks of the PSD whose amplitude is above the amplitude threshold a_{peak}. Then we use linear interpolation to draw a curve $q[k]$ that connects these peaks (Figure 3.2b). For the parts without peaks on the left and right sides, the values should be the amplitudes of the nearest peaks. If there are no peaks detected, the value of this curve is set to 1. The flattened curve is the PSD curve $p[k]$ divided by the peak curve $q[k]$, and the values smaller than the threshold $a_{\text{min}}^{\text{flatten}}$ are set to the threshold itself to avoid very small weights (Figure 3.2c):

$$w_{\text{flatten}}[k] = \max\left\{\frac{p[k]}{q[k]}, a_{\text{min}}^{\text{flatten}}\right\}. \tag{3.4}$$

After obtaining the flattened curve, we build the reweighting curve $w_{\text{re}}[k]$ by shifting the flattened curve by $-a_{\text{shift}}$ and then rescaling it between two bounds $\mu_{\text{low}} < 1$ and $\mu_{\text{high}} > 1$ (Figure 3.2d), so that most values of this curve are below 1. The upper and lower bounds converge to 1 as the number of iterations increases:

$$\mu_{\text{low}}^{(\mathsf{n})} = 1 - \frac{1 - \mu_{\text{low}}^{(0)}}{\mathsf{n}} \tag{3.5}$$

$$\mu_{\text{high}}^{(\mathsf{n})} = 1 + \frac{\mu_{\text{high}}^{(0)} - 1}{\mathsf{n}} \tag{3.6}$$

where n is the iteration number.

After building the reweighting curves for the left part (w_{L}) and right part (w_{R}), we use that curve to re-adjust our weighting matrix. Suppose that the weighting matrix $\mathbf{W}$ contains K rows (for frequency bins) and M columns (for time bins). We split the matrix into a left and right subpart, such that the left part $\mathbf{W}_{\text{L}}$ contains the first $\lceil M/2 \rceil$ columns of $\mathbf{W}$ and the right part $\mathbf{W}_{\text{R}}$ contains the remaining columns ($M - \lceil M/2 \rceil$). The updated weighting matrix will be:

$$\mathbf{W}^{(\mathsf{n}+1)} = \left[\mathbf{W}_{\text{L}}^{(\mathsf{n}+1)}, \mathbf{W}_{\text{R}}^{(\mathsf{n}+1)}\right] \tag{3.7}$$

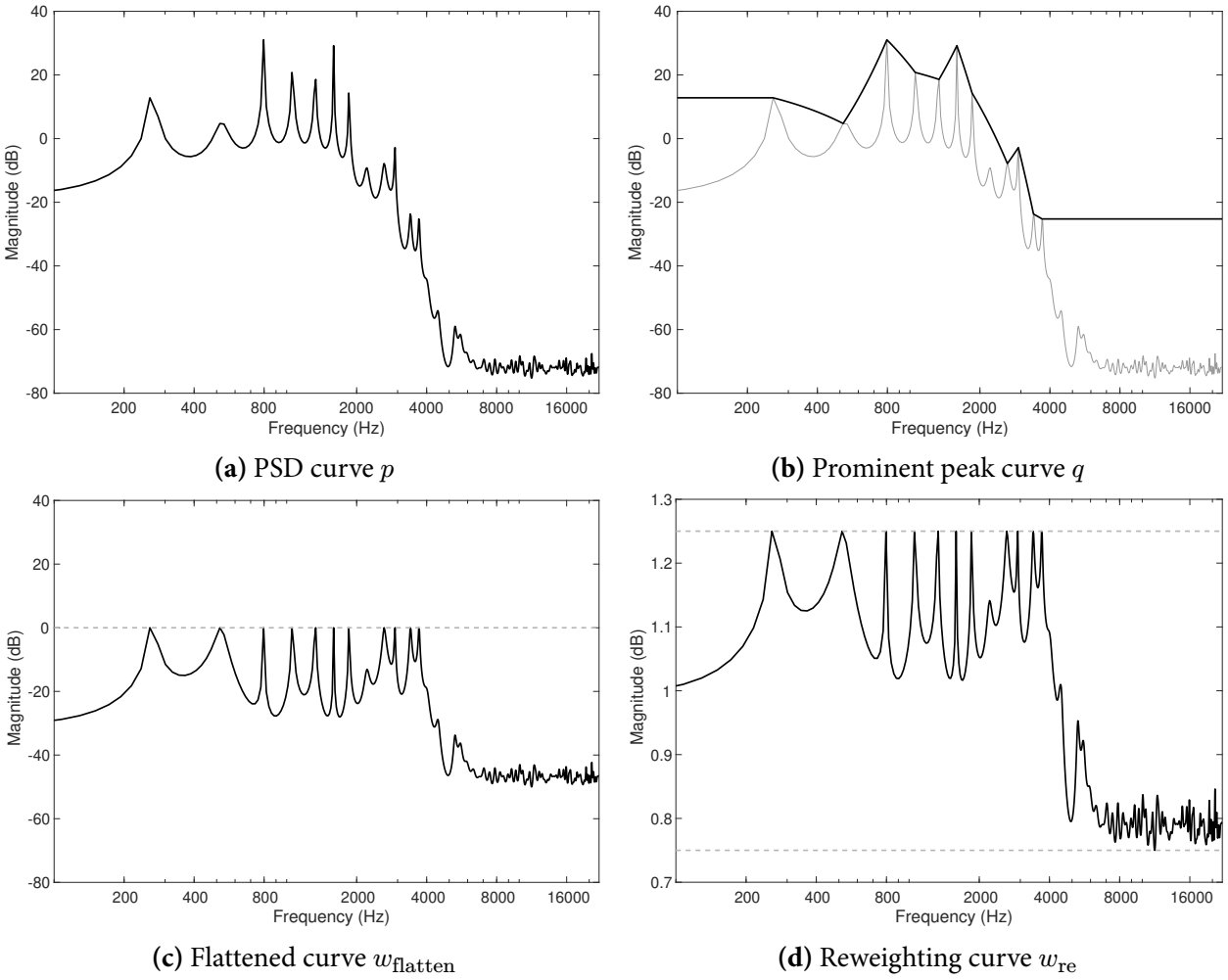
$$= \left[\mathbf{W}_{\text{L}}^{(\mathsf{n})} \odot [\mathbf{w}_{\text{L}}, \cdots, \mathbf{w}_{\text{L}}], \mathbf{W}_{\text{R}}^{(\mathsf{n})} \odot [\mathbf{w}_{\text{R}}, \cdots, \mathbf{w}_{\text{R}}]\right] \tag{3.8}$$

where the superscript $\cdot^{(\mathsf{n})}$ represents the iteration number, $\odot$ is the element-wise multiplier, $[\mathbf{w}_{\text{L}}, \cdots, \mathbf{w}_{\text{L}}]$ represents a matrix that has the same shape as $\mathbf{W}_{\text{L}}$, and each column is $\mathbf{w}_{\text{L}}$.

To limit the weight, we set an upper bound $w_{\max}$ for the weighting matrix, so that for each element w_k in $\mathbf{w}_{\text{L}}$ or $\mathbf{w}_{\text{R}}$:

$$w_k := \min\{w_k, w_{\max}\}. \tag{3.9}$$

Figure 3.2 illustrates the curves constructed at each stage of the reweighting method.



(a) PSD curve p

(b) Prominent peak curve q

(c) Flattened curve w_{flatten}

(d) Reweighting curve w_{re}

Figure 3.2 PSD and related curves created at each stage of the reweighting method.

3.2.3 Neighborhood definition

In addition, we change the neighborhood calculation strategy of the social sparsity (see Section 2.1.4). Instead of using a constant 2D kernel as the neighborhood weights for the 2D convolution to calculate the coefficient sum of each atom's neighborhood in the TF plane, we use an *order-statistic filter*. The order-statistic filter is a non-linear spatial filter that sets the coefficient based on the k-th smallest value among the defined neighbors (Pitas and Venetsanopoulos 1992). For example, a 4th-order statistic filter with a 3×3 square neighborhood with values from 1 to 9 sets the coefficient to 4, which is the 4th smallest value among its 3×3 neighbors. In our method, we define a neighborhood $\mathcal{N}$ with 21 members as in Figure 3.3:

We set the order of the filter to $\lfloor (N_{\mathcal{N}} - 1)/2 \rfloor$, where $N_{\mathcal{N}}$ is the number of non-zero element in the neighborhood $\mathcal{N}$ ($N_{\mathcal{N}} = 21$ in our definition), so that it will be just smaller than using the median value to filter out more noise to make the result sparser.

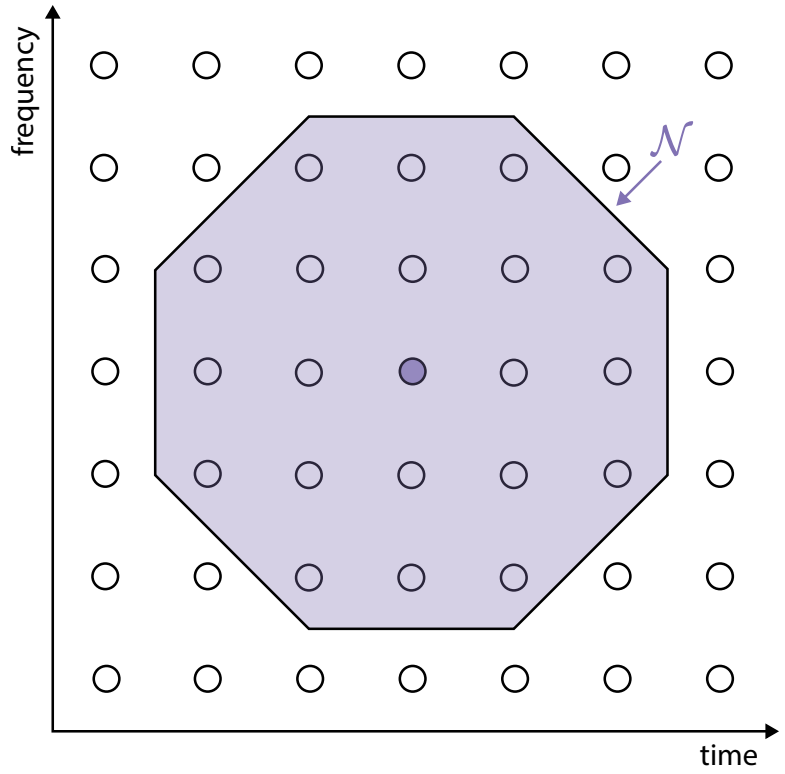


Figure 3.3 The neighborhood configuration in our hybrid approach. The dark solid circle represents the center point and the colored shaded area represents the neighborhood of the center point.

3.2.4 Tuning of lambda

The parameter λ, which controls the sparsity, is automatically tuned based on the following procedure.

A short-time Fourier transform (STFT) of the signal is performed using a Hann window of length 512 and a hop size of 128. The gradient magnitude at the location of each TF atom is computed, which reflects their local variability:

$$|\nabla x_{m,k}| = \sqrt{\left|\frac{\partial x_{m,k}}{\partial m}\right|^2 + \left|\frac{\partial x_{m,k}}{\partial k}\right|^2} \tag{3.10}$$

where $x_{m,k}$ represents the STFT coefficient of a signal at frame number m and frequency channel k.

The gradient in discrete time can be estimated using the *central difference* for inner values:

$$\frac{\partial x_{m,k}}{\partial m} = \frac{x_{m+1,k} - x_{m-1,k}}{2} \tag{3.11}$$

$$\frac{\partial x_{m,k}}{\partial k} = \frac{x_{m,k+1} - x_{m,k-1}}{2} \tag{3.12}$$

and *single-sided differences* for values along the outermost edges of the matrix:

$$\frac{\partial x_{1,k}}{\partial m} = \frac{x_{2,k} - x_{1,k}}{2} \tag{3.13}$$

$$\frac{\partial x_{M,k}}{\partial m} = \frac{x_{M,k} - x_{M-1,k}}{2} \tag{3.14}$$

$$\frac{\partial x_{m,1}}{\partial k} = \frac{x_{m,2} - x_{m,1}}{2} \tag{3.15}$$

$$\frac{\partial x_{m,K}}{\partial k} = \frac{x_{m,K} - x_{m,K-1}}{2} \tag{3.16}$$

where M and K are the number of time and frequency bins, respectively.

The randomness of the gradient magnitudes is quantified by evaluating their spectral flatness:

$$\text{flatness}_{\nabla}(m) = \frac{\left(\prod_{k=1}^{K} |\nabla x_{m,k}|\right)^{1/K}}{\frac{1}{K}\sum_{k=1}^{K} |\nabla x_{m,k}|} = \frac{\exp\left(\frac{1}{K}\sum_{k=1}^{K} \ln |\nabla x_{m,k}|\right)}{\frac{1}{K}\sum_{k=1}^{K} |\nabla x_{m,k}|}. \tag{3.17}$$

The relative noisiness level of the STFT of the original signal is determined by taking the median value of the spectral flatness of the gradient magnitudes.

Since spectral flatness is a relative value and λ is an absolute value for the coefficients of the atoms, we use spectral flatness and root-mean-square energy (RMSE) of the input signal together to predict the value of λ. λ is calculated from the following equation, obtained by fitting a linear regression between $\log_{10}(\lambda)$ and the $\log_{10}$ of the product of the spectral flatness and the RMSE. The data points for linear regression come from an experiment. In this experiment, we created a synthesized signal consisting of frequency-modulated sinusoids and linear chirps. The number of atoms with non-zero coefficients after decomposing the signal (without introducing the sparsity constraint, i.e., $\lambda = 0$) is used as the baseline to measure the sparsity. Then, we add white noise with different noise levels to the synthesized signal and decompose the signal with different λ. The smallest λ that reaches the baseline (times a factor that is slightly greater than 1) is considered the optimal λ, and will be used for the linear regression. The regression result is formulated as:

$$\log_{10} \lambda = 1.55465 \log_{10}(\theta_{50\%} \cdot \mu_{\mathrm{R}}) + 0.51171 \tag{3.18}$$

where $\theta_{50\%} = \text{median}\{\text{flatness}_{\nabla}\}$ is the median value (50th percentile) of the spectral flatness values flatness_{∇}, and $\mu_{\mathrm{R}} = \text{RMSE}(\mathbf{M}_{\mathrm{R}}\mathbf{y})$ is the root-mean-square energy of the reliable parts of signal $\mathbf{y}$.

3.3 Inpainting of the Deterministic Part

The deterministic component resulting from the sparse decomposition is mainly composed of time-varying sinusoids, also known as partials. Therefore, the techniques for analyzing and re-synthesizing partials can be applied to reconstruct the deterministic part in the gap region. This goal is achieved by performing the following steps: first, the signal is analyzed to extract the partials and the partials are further processed to be more reliable and consistent; second, the corresponding partials on both sides of the gap are matched and predicted; finally, the partials are resynthesized to obtain the reconstructed signal.

3.3.1 Partial tracking

The partial tracking method is primarily based on the approach proposed by Neri and Depalle (2018), with modifications on the treatment of the gap and of the cost matrix.

3.3.1.1 Treatment of the gap

Gaps in signal usually lead to discontinuities in the time domain and artifacts in the spectrum, which can confuse the partial tracking algorithm and result in inaccurate estimates of frequency and amplitude, and even in many fragmented or "fake" partials at the beginning and end of the gap. That degrades the quality and accuracy of subsequent analysis and synthesis.

To address this problem, we will skip analyzing the frames where more than 25% of the samples are missing. Therefore, samples located near the gap region will not be analyzed.

3.3.1.2 Cost matrix

A fixed frequency threshold in Hz may be difficult to track for partials in high frequency, since the frequency variation is greater at high frequencies than at low frequencies. Therefore, we calculate the mel-scale frequency difference instead, so that:

$$\Delta f_{ij} = \mathrm{mel}\left(f_i^{[k-1]}(H/2)\right) - \mathrm{mel}\left(f_j^{[k]}(-H/2)\right) \tag{3.19}$$

where the definition $\mathrm{mel}(f) = 2595\log_{10}(1 + f/700)$ is from O'Shaughnessy (2000), and $f_i^{[k]}(n)$ represent the instantaneous frequency of the i-th partial at time n over frame k.

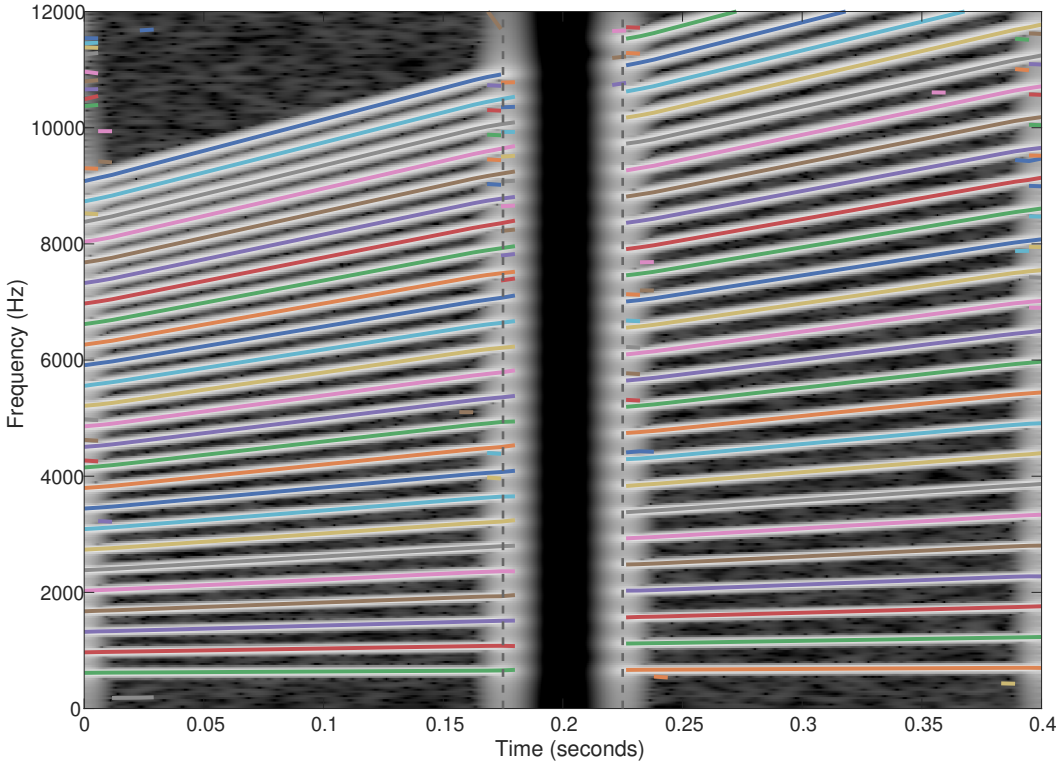


Figure 3.4 Tracking of partials in a synthesized audio signal consisting of multiple chirps plus background noise with a gap (between the dashed lines). Parameters of the tracking method are unchanged as in Neri and Depalle (2018).

In order to suppress tracking of partials with abrupt changes in frequency, we add a constraint related to the frequency derivative (also called the chirp rate):

$$\Delta\beta_{ij} = \frac{f_i'^{[k-1]}(H/2)}{\log_{10}\left(f_i^{[k-1]}(H/2)\right)} - \frac{f_j'^{[k]}(-H/2)}{\log_{10}\left(f_j^{[k]}(-H/2)\right)} \tag{3.20}$$

where f' is the first-order derivative of f.

The costs for useful and spurious assignments will be:

$$C_{ij}^{\text{useful}} = 1 - \exp\left(-\frac{(\Delta a_{ij})^2}{2\sigma_a^2} - \frac{(\Delta f_{ij})^2}{2\sigma_f^2} - \frac{(\Delta\beta_{ij})^2}{2\sigma_\beta^2}\right) \tag{3.21}$$

$$C_{ij}^{\text{spurious}} = 1 - (1 - \delta^{\text{track}})C_{ij}^{\text{useful}} \tag{3.22}$$

where:

$$\Delta a_{ij} = a_i^{[k-1]}(H/2) - a_j^{[k]}(-H/2) \tag{3.23}$$

$$\sigma_a^2 = \frac{\zeta_a^2}{2\ln(\delta^{\text{track}} - 2) - 2\ln(\delta^{\text{track}} - 1)} \tag{3.24}$$

$$\sigma_f^2 = \frac{\zeta_f^2}{2\ln(\delta^{\text{track}} - 2) - 2\ln(\delta^{\text{track}} - 1)} \tag{3.25}$$

$$\sigma_\beta^2 = \frac{\zeta_\beta^2}{2\ln(\delta^{\text{track}} - 2) - 2\ln(\delta^{\text{track}} - 1)}. \tag{3.26}$$

Parameters ζ_a, ζ_f, and ζ_β control the threshold values for amplitude, frequency, and frequency slope change, respectively.

The cost matrix will be:

$$C_{ij} = \min\{C_{ij}^{\text{useful}}, C_{ij}^{\text{spurious}}\}. \tag{3.27}$$

The Hungarian algorithm is then used to obtain the optimal assignment matrix by providing the cost matrix $\mathbf{C}$ (Neri and Depalle 2018).

3.3.2 General partial prediction method

A general prediction method is proposed in this section, which will be extensively applied to the subsequent processing of partials.

Consider a signal that contains a vibrato and which average frequency increases linearly, such as a combination of a vibrato and a chirp. The instantaneous frequency of this partial should be:

$$f_i(t) = (f_0 + \beta t) + A\cos(2\pi f_m t) \tag{3.28}$$

where f_0 is the fundamental frequency, β is the chirp rate, A is the modulation depth, and f_m is the modulation frequency.

The signal contains a long-term trend, and a short-term periodicity, which makes it challenging to be predicted only by an autoregressive model. Therefore, we assume that a partial may contain both trend and periodicity, or only one of them, and predict these components separately.

In order to predict the trend component, a linear regression of the frequency (or amplitude) of the partial is performed. The coefficient of determination (R^2) for the regression result, which measures how well the regression model fits the data, is calculated by the following equation:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{3.29}$$

where y_i is the observed data (analyzed frequency or amplitude values), $\hat{y}_i$ is the predicted data, and $\bar{y}$ is the mean of the observed data.

If R^2 is greater than a threshold (R_f^2 for frequency and R_a^2 for amplitude), the trend is considered as it exists, and the linear regression model is used to predict the trend value. If R^2 is less than or equal to the threshold, the trend is considered as non-existent, and the trend component is set to 0.

Burg's AR model is employed to predict the periodicity component. The order (or the number of poles) of the AR model is set to p_{predict}.

The final result is obtained by adding the trend prediction and the periodicity prediction. This allows both long-term and short-term variations in partial frequency and amplitude to be captured.

3.3.3 Partial reconnection

In some cases, a partial that is originally continuous may be segmented into several partials due to noise or other factors. The length of the partials affects the accuracy of the prediction, as longer partials contain more information than shorter partials that are fragmented (shown in Figure 2.8). To address this issue, we propose a method to reconnect these fragmented partials based on their frequency and amplitude continuity.

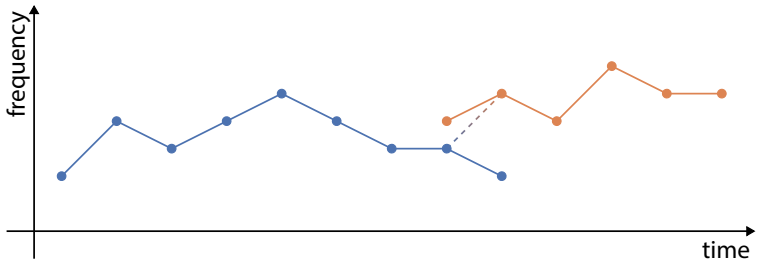
The proposed method can be applied to the partials that fall into the following two scenarios. The first scenario is when the two partials overlap in time by a small amount. The second scenario is when the two partials do not overlap in time, but are close together, which means that there is a small gap between their end and start points. Figure 3.5 illustrates these two cases accordingly.

The proposed method is based on the following criteria:

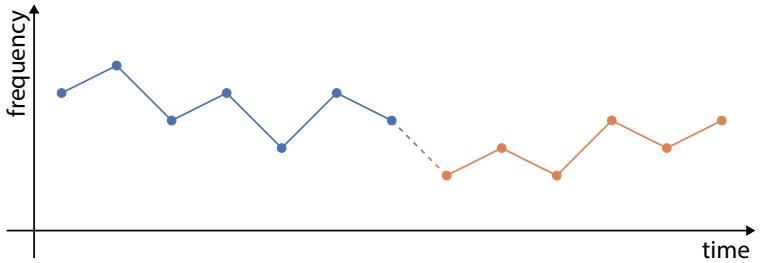
- The length of a partial should not be smaller than $l_{\text{min}}^{\text{connect}}$, as the prediction error is very large for very short partials.
- The two partials should have similar boundary frequencies, as this indicates that they likely belong to the same original partial.
- The two partials cannot overlap more than $l_{\text{max}}^{\text{overlap}}$ windows or jump more than $l_{\text{max}}^{\text{jump}}$ windows, as this may indicate that they are two separate partials instead of one.
- The cost of connecting two partials should be small enough.

The process starts by selecting the longest partial and comparing it with all other nearby partials. Nearby partials are defined by their relative boundary frequency difference, with the long partial being less than a threshold $\Delta_f^{\text{boundary}}$.

For each pair of selected partials, the long one needs to be extrapolated for comparison. There are two cases of extrapolation of the long partial: if they overlap, the overlapping data points are



(a) overlap



(b) no-overlap

Figure 3.5 Two scenarios of potential partial connection. The solid lines represent analyzed partials, the solid points represent data points (per frame), and the dashed lines with gradient color represent potential partial connections.

removed and then the long partial is extrapolated using the proposed prediction method. If they do not overlap, the full range of data points is used to extrapolate the long partial.

The costs of connection are then calculated for each pair of partials. Suppose the long partial is denoted by p_i and the other partial is denoted by p_j. Then the cost for connecting p_j to p_i is:

$$C^{\text{connect}}_{p_i \leftarrow p_j} = \delta^{\text{connect}} \frac{\bar{d}_f(p_i, p_j)}{\zeta^{\text{connect}}_f} + (1 - \delta^{\text{connect}}) \frac{\bar{d}_a(p_i, p_j)}{\zeta^{\text{connect}}_a} \tag{3.30}$$

where ζ^{connect}_f and ζ^{connect}_a are the thresholds for frequency and amplitude, $0 \leq \delta^{\text{connect}} \leq 1$ is a parameter that controls the influence of the two metrics on the cost. $\bar{d}_f(p_i, p_j)$ and $\bar{d}_a(p_i, p_j)$ represent the normalized frequency (in the mel scale) or amplitude (in dB scale) Euclidean distances between two partials p_i and p_j in the range of p_j, which means the frequency and amplitude of p_i in this range is mostly extrapolated. We use the definition of normalized Euclidean distance from (Lagrange et al. 2005):

$$\bar{d}_f(\hat{p}_i, p_j) = \frac{\|\text{mel}(\hat{\mathbf{f}}_i) - \text{mel}(\mathbf{f}_j)\|_2 \Big/ \sqrt{N_{p_j}}}{1 + \sigma\{\text{mel}(\hat{\mathbf{f}}_i))\} + \sigma\{\text{mel}(\mathbf{f}_j)\}} \tag{3.31}$$

$$\bar{d}_a(\hat{p}_i, p_j) = \frac{\|\hat{\mathbf{a}}_i - \mathbf{a}_j\|_2 \Big/ \sqrt{N_{p_j}}}{1 + \sigma\{\hat{\mathbf{a}}_i\} + \sigma\{\mathbf{a}_j\}} \tag{3.32}$$

where $\hat{\mathbf{f}}_i$ and $\hat{\mathbf{a}}_i$ are the (predicted) frequency and amplitude of partial p_i in the range of partial p_j, N_{p_j} is the length (in frames) of partial p_j, and $\sigma\{\mathbf{x}\}$ is the standard deviation of $\mathbf{x}$.

The partial with minimal connection cost (C^{connect}) is determined only if the minimal cost is less than 1 ($\min\{C^{\text{connect}}\} < 1$). If all costs are greater than 1, the long partial is unable to connect to any other partials. The selected partial is then merged to the long partial. If they overlap, crossfading is used in the overlapping area to smooth the transition. If they do not overlap, they are simply concatenated. After merging these two partials, the shorter partial is removed from the list. The process is repeated until all valid partials are processed.

The reconnection method can reduce the total number of partials, which may result in more accurate and consistent matching and prediction results.

3.3.4 Partial matching

The next step is to determine which partial near the left boundary of the gap should be connected to which partial near the right boundary of the gap in order to form a merged partial. To achieve this, a method for matching two partials before and after the gap is proposed.

First, all partials with enough length (more than a threshold $l_{\min}^{\text{match}}$) that are near the gap are selected as candidates for matching. The definition of "near" is from the last fully reliable frame (no missing samples) to the last analyzed frame before the gap.

Then, all candidate partials in the gap region are extrapolated using the general prediction method described in the previous section. For the left reliable part, any sudden change, such as a fast attack, is detected and removed based on a slope ratio threshold β_a^{attack}. The frequency extrapolation uses either an AR model or an AR model plus a linear regression, depending on the existence of a trend. The amplitude extrapolation does not use amplitude data from semi-reliable frames that contains missing samples for prediction. For the right reliable part, no sudden change detection is performed because we assume that the region immediately to the right of the gap does not contain any fast attack followed by a decay.

Next, the normalized Euclidean distances (defined in Section 3.3.3) between the left and right predictions for each pair of candidate partials are calculated. Instead of calculating the distance between the two predictions of a pair of partials in the gap equally, we focus on the prediction distance at the gap's bounds. To achieve this, we define the forward and backward weighted distance differences and use them to compute the weighted Euclidean distances. The weighted normalized

Euclidean distances of frequency (in mel scale) and amplitude (in dB scale) are calculated as follows:

$$\bar{d}_f(\hat{p}_i, \hat{p}_j) = \frac{\min\{\|\hat{\mathbf{f}}_{i\to j}\|_2, \|\hat{\mathbf{f}}_{i\leftarrow j}\|_2\}/\sqrt{l_{\text{gap}}}}{1 + \sigma(\text{mel}(\hat{\mathbf{f}}_i)) + \sigma(\text{mel}(\hat{\mathbf{f}}_j))} \tag{3.33}$$

$$\bar{d}_a(\hat{p}_i, \hat{p}_j) = \frac{\min\{\|\hat{\mathbf{a}}_{i\to j}\|_2, \|\hat{\mathbf{a}}_{i\leftarrow j}\|_2\}/\sqrt{l_{\text{gap}}}}{1 + \sigma(\hat{\mathbf{a}}_i) + \sigma(\hat{\mathbf{a}}_j)} \tag{3.34}$$

where the hat above the symbols represents predicted values in the gap region, l_{gap} is the number of frames that contains the gap, and $\hat{\mathbf{f}}_{i\to j} = \mathbf{w}_{\to} \odot \left(\text{mel}(\hat{\mathbf{f}}_i) - \text{mel}(\hat{\mathbf{f}}_j)\right)$ and $\hat{\mathbf{f}}_{i\leftarrow j} = \mathbf{w}_{\leftarrow} \odot \left(\text{mel}(\hat{\mathbf{f}}_i) - \text{mel}(\hat{\mathbf{f}}_j)\right)$ are the weighted forward and backward differences of $\hat{p}_i$ and $\hat{p}_j$ in frequency on the mel scale. The weighting vectors $\mathbf{w}_{\to}$ and $\mathbf{w}_{\leftarrow}$ are defined as:

$$\mathbf{w}_{\to} = \exp\left(\frac{-\ln(l_{\text{gap}})}{l_{\text{gap}}}[0, 1, \dots, l_{\text{gap}} - 1]^T\right), \tag{3.35}$$

$$\mathbf{w}_{\leftarrow} = \exp\left(\frac{-\ln(l_{\text{gap}})}{l_{\text{gap}}}[l_{\text{gap}} - 1, l_{\text{gap}} - 2, \dots, 0]^T\right). \tag{3.36}$$

Similarly, $\hat{\mathbf{a}}_{i\to j}$ and $\hat{\mathbf{a}}_{i\leftarrow j}$ are defined as:

$$\hat{\mathbf{a}}_{i\to j} = \mathbf{w}_{\to} \odot \left(\hat{\mathbf{a}}_i - \hat{\mathbf{a}}_j\right), \tag{3.37}$$

$$\hat{\mathbf{a}}_{i\leftarrow j} = \mathbf{w}_{\leftarrow} \odot \left(\hat{\mathbf{a}}_i - \hat{\mathbf{a}}_j\right). \tag{3.38}$$

This distance measures how well these two partials match in terms of frequency and amplitude. A cost matrix based on the normalized Euclidean distances is constructed as follows, similar to the matrix in Section 3.3.1:

$$C_{ij} = \min\{C_{ij}^{\text{match}}, C_{ij}^{\text{mismatch}}\} \tag{3.39}$$

and

$$C_{ij}^{\text{match}} = 1 - \exp\left(-\frac{\left(\bar{d}_a(\hat{p}_i, \hat{p}_j)\right)^2}{2\sigma_a^2} - \frac{\left(\bar{d}_f(\hat{p}_i, \hat{p}_j)\right)^2}{2\sigma_f^2}\right) \tag{3.40}$$

$$C_{ij}^{\text{mismatch}} = 1 - (1 - \delta^{\text{match}})C_{ij}^{\text{match}}. \tag{3.41}$$

Finally, the Hungarian algorithm (Neri and Depalle 2018) is employed to determine the optimal matching that minimizes the total cost. This matching indicates which partials should be connected across the gap.

3.3.5 Partial prediction

After matching the partials near the gap, further extrapolation of these partials is required for inpainting. All partials involved in the partial matching process are further partitioned into three groups: *matched* partials, unmatched *born* partials, and unmatched *dead* partials. Different extrapolation (or interpolation) strategies will be applied to these three groups of partials.

3.3.5.1 Frequency extrapolation

We use the frequency interpolation method based on asymmetric crossfading in (Lagrange et al. 2005). However, the proposed general partial prediction method is utilized for predicting the frequency instead of directly applying linear prediction.

3.3.5.2 Amplitude extrapolation

For the amplitude extrapolation, different methods are applied to the three groups of partials.

The matched partials are interpolated using the amplitude constraint and asymmetric crossfading method proposed by Lagrange et al. (2005), but with our partial prediction method instead.

The unmatched born partials are further separated into two types based on their slope of the trend line calculated from the general partial prediction method.

If the slope is positive, indicating an increasing amplitude in the long term, the predicted amplitude at the beginning of the gap is checked against a certain threshold. If it is below (or equal to) the threshold, the predicted amplitude is unchanged. If it is above the threshold, a linear attack is added so that the amplitude at the beginning of the gap reaches the threshold. The amplitude after the linear attack is given by:

$$a^{[k]} = \tilde{a}^{[k]} + \frac{k_{\mathrm{R}} - k}{l_{\mathrm{birth}}} T_a \tag{3.42}$$

where $a^{[k]}$ is the output amplitude, $\tilde{a}^{[k]}$ is the predicted amplitude, k_{R} is the frame index of the first reliable amplitude after the gap, k is the current frame index, l_{birth} is the length of the attack, and T_a is the minimal amplitude threshold.

If the slope is negative or zero, indicating a decreasing or constant amplitude in the long term, we assume the partial is attacked and then decayed during the gap in this scenario. Therefore, the amplitude curve is constructed by replacing the linear trend with a parabola that satisfies three conditions:

- the first reliable amplitude after the gap should be on the parabola,

- the derivative of the parabola at the first reliable data point after the gap should be the same as the slope of the linear trend, if the slope of the linear trend is not too steep, and
- the amplitude at the start of the partial should be below the minimal amplitude threshold.

The parabola in the form of ax^2+bx+c can be derived by solving the following linear equations:

$$\begin{pmatrix} k_{\mathrm{R}}^2 & k_{\mathrm{R}} & 1 \\ 2k_{\mathrm{R}} & 1 & 0 \\ (k_{\mathrm{R}}-l_{\mathrm{birth}})^2 & (k_{\mathrm{R}}-l_{\mathrm{birth}}) & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a_{\mathrm{R}} \\ \beta_{\mathrm{trend}} \\ T_a \end{pmatrix} \tag{3.43}$$

where a_{R} is the first reliable amplitude after the gap, β_{trend} is the slope of linear trend, and l_{birth} is the length of the partial birth.

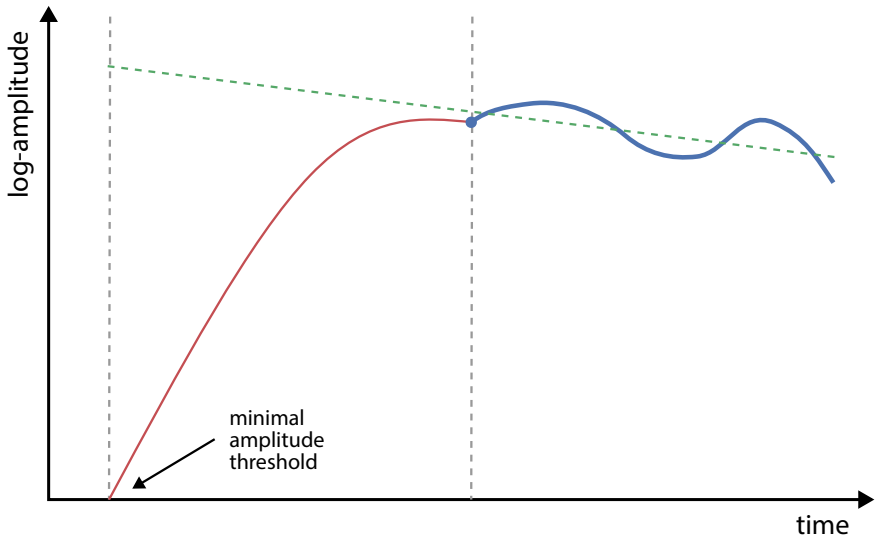


Figure 3.6 The predicted parabolic trend of the amplitude of a partial birth. The gray dashed line represents the limits of the gap, the blue solid line on the right side represents the analyzed partial, the green dashed line represents the estimated linear trend, and the red solid line on the left side represents the calculated parabolic trend.

The amplitude curve is then obtained by adding the parabola to the periodicity prediction curve.

The unmatched dead partials are treated similarly as the unmatched born partials, except that the sign of the slope condition is flipped, and parabola conditions are changed accordingly. The linear equations for this parabola are defined as:

$$\begin{pmatrix} k_{\mathrm{L}}^2 & k_{\mathrm{L}} & 1 \\ 2k_{\mathrm{L}} & 1 & 0 \\ (k_{\mathrm{L}}+l_{\mathrm{death}})^2 & (k_{\mathrm{L}}+l_{\mathrm{death}}) & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a_{\mathrm{L}} \\ \beta_{\mathrm{trend}} \\ T_a \end{pmatrix} \tag{3.44}$$

where k_{L} is the frame index of the first reliable amplitude before the gap, a_{L} is the first reliable amplitude before the gap, and l_{death} is the length of the partial death.

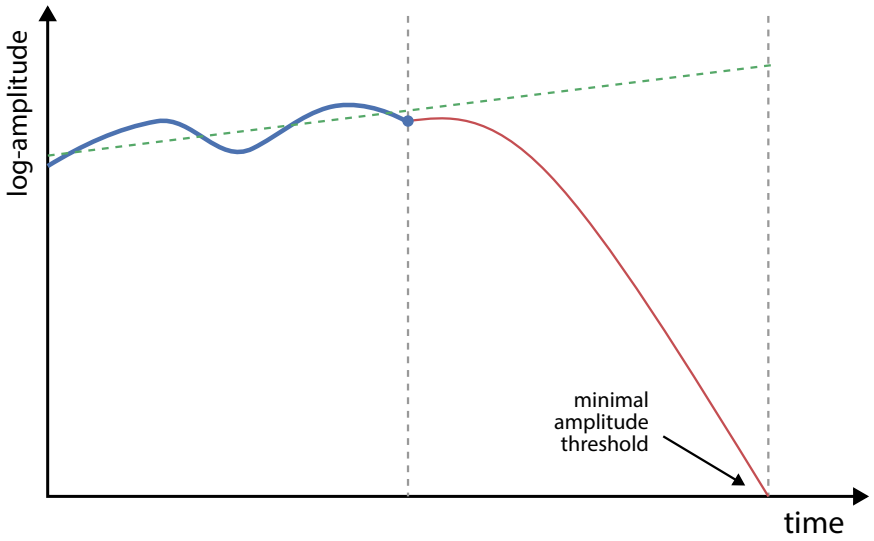


Figure 3.7 The predicted parabolic trend of the amplitude of a partial death. The gray dashed line represents the gap, the blue solid line on the left side represents the analyzed partial, the green dashed line represents the estimated linear trend, and the red solid line on the right side represents the calculated parabolic trend.

3.3.5.3 Phase extrapolation

The phase is reconstructed in the same way as the phase interpolation method in Lagrange et al. (2005), which is based on the method of McAulay and Quatieri (1986) and further spreads the phase error over the whole gap.

3.3.5.4 Partial re-synthesis

The signal $\hat{y}_{\mathrm{tonal}}$ with all partials is reconstructed using the synthesis method in McAulay and Quatieri (1986), which is described in Section 2.2.3.

3.4 Decomposition of Transient and Stochastic Parts

After obtaining the residual signal without most of the deterministic part, the next step is to further decompose it into a transient and a stochastic component.

Before starting the decomposition, the residual signal is further shortened, because no long context is needed for analyzing the transient component that is short in duration. In our thesis, we

specify that the length of each reliable neighborhood of the shortened signal should be no less than the maximum of the following two values: 1/4 of the gap length and the length of the window used for the tonal decomposition (Ψ_{tonal}).

The plain analysis variant of social sparse decomposition is used, without using the reweighting method or special neighborhood definition, compared with the model in Section 3.2. The neighborhood is defined as an equal-weight rectangle with 1 time-bin width and 25 frequency-bin height, which promote the selection of transient-like structure.

Parameter λ is obtained in a similar way as in Section 3.2.4. To better analyze the transients, the window length and hop size of STFT are changed to 128 and 32, respectively. Instead of calculating the spectral flatness of the gradient magnitudes, we calculate it from the gradient along the frequency direction, because the gradient along the frequency may be less affected by transient signals with sudden changes in frequency components. The spectral flatness along frequency direction is formalized as follows:

$$\text{flatness}_f(m) = \frac{\left(\prod_{k=1}^{K}\left|\frac{\partial x_{m,k}}{\partial k}\right|\right)^{1/K}}{\frac{1}{K}\sum_{k=1}^{K}\left|\frac{\partial x_{m,k}}{\partial k}\right|} = \frac{\exp\left(\frac{1}{K}\sum_{k=1}^{K}\ln\left|\frac{\partial x_{m,k}}{\partial k}\right|\right)}{\frac{1}{K}\sum_{k=1}^{K}\left|\frac{\partial x_{m,k}}{\partial k}\right|}. \tag{3.45}$$

The parameter λ can be calculated by the following equation:

$$\log_{10}\lambda = 1.06324\log_{10}(\theta_{90\%}\cdot\mu_{\text{R}}) + 0.51639 \tag{3.46}$$

where $\theta_{90\%}$ is the 90th percentile of the spectral flatness values flatness_f, $\mu_{\text{R}} = \text{RMSE}(\mathbf{M}_{\text{R}}\mathbf{y}_{\text{res}})$ is the root-mean-square energy of the reliable parts of the tonal residual signal $\mathbf{y}_{\text{res}}$.

The decomposition result is the transient part of the residual signal. The residual of the decomposition is considered to be the stochastic part, which will be analyzed and reconstructed in a subsequent process.

3.5 Noise Reconstruction

In order to analyze the stochastic part and to reconstruct the noise from the residual, the region near the boundary of the gap is set to be unreliable for the analysis of the stochastic part. Because the previous sparse decomposition of the tonal part may not be accurate enough near the boundary, and may leak some energy of tonal part to the residual signal. The length of the unreliable part to ignore for the analysis is determined by a parameter N_{ignore}.

Burg's AR model is used to estimate the PSD of the left and right reliable neighborhoods that consist of noise. The number of poles for the AR model is set to p_{noise}. Based on the PSD estimation, two reconstruction methods are proposed for different scenarios.

If the PSD change is small over time, two noise signals that have the same PSDs as the left and right reliable neighborhoods' are generated by filtering a normalized Gaussian noise with linear prediction coefficients calculated from Burg's method. Then, a cosine window with length Ψ_{noise} is used to crossfade the left and right noise signals to obtain a smooth transition.

If the PSD change is large over time such as a moving resonance, the Log Area Ratio (LAR) coefficients between the left and right reliable neighborhoods' are calculated and linearly interpolated and converted back to filter coefficients using Levinson's recursion (Hayes 1996).

$$\text{LAR}(\gamma_i) = \ln\left(\frac{1-\gamma_i}{1+\gamma_i}\right) \tag{3.47}$$

where γ_i is the i-th reflection coefficient estimated with the Burg's method.

Then, a normalized Gaussian noise is filtered with these coefficients to generate a noise signal that matches the PSD variation. The reconstructed noise will be placed in the unreliable region.

3.6 Output Generation

The complete reconstructed signal $\hat{y}_{\text{rec}}$ is obtained by superimposing the partial signal $\hat{y}_{\text{tonal}}$ and the noise signal $\hat{y}_{\text{noise}}$ together. In order to keep the reliable part of the original signal unchanged, only the gap region will be replaced by the reconstructed signal, with a short crossfade (length $N_{\text{crossfade}}$) at the boundaries of the gap to suppress potential discontinuity.

Suppose that the observed signal is $\hat{y}[n]$, the window functions for crossfading are $\psi_{\nearrow}[n]$ (for ramping up) and $\psi_{\searrow}[n]$ (for ramping down), the degraded part starts from n_L to n_R. The output signal y_{rec} can be formulated as:

$$y_{\text{rec}}[n] = \begin{cases} \hat{y}[n] & \text{for } n \in [0, n_{LL}) \\ \hat{y}[n] \cdot \psi_{\searrow}[n - n_{LL}] + \hat{y}_{\text{rec}}[n] \cdot \psi_{\nearrow}[n - n_{LL}] & \text{for } n \in [n_{LL}, n_L) \\ \hat{y}_{\text{rec}}[n] & \text{for } n \in [n_L, n_R] \\ \hat{y}[n] \cdot \psi_{\nearrow}[n - n_R - 1] + \hat{y}_{\text{rec}}[n] \cdot \psi_{\searrow}[n - n_R - 1] & \text{for } n \in (n_R, n_{RR}] \\ \hat{y}[n] & \text{for } n \in (n_{RR}, N] \end{cases} \tag{3.48}$$

where $n_{LL} = n_L - N_{\text{crossfade}}$, $n_{RR} = n_R + N_{\text{crossfade}}$, $\psi_{\nearrow}[n] \neq 0$ for $0 \leq n < N_{\text{crossfade}}$, and $\psi_{\searrow}[n] \neq 0$ for $0 \leq n < N_{\text{crossfade}}$.

4

Experiments

This chapter presents a comprehensive evaluation and analysis of the hybrid inpainting approach proposed in Chapter 3. A parameter selection strategy for optimizing the hybrid approach is proposed in Section 4.1, which will be applied to subsequent experiments. The evaluation metrics employed include signal-to-noise ratio (SNR), Itakura-Saito distance (ISD), and objective difference grade (ODG), and are introduced in Section 4.2. Three distinct experiments are conducted to assess the performance and effectiveness of each component of the hybrid approach in various scenarios, described in Section 4.3, 4.4 and 4.5, respectively. Furthermore, the hybrid approach is compared with other state-of-the-art inpainting methods using real audio signals in Section 4.6.

4.1 Parameter Selection Strategy

This section describes the strategy for choosing the parameters of various methods used in the stages of our hybrid approach in Chapter 3. These parameters were determined before conducting the experiments.

4.1.1 Pre-processing parameters

We extend the lengths of the left and right reliable neighborhoods beyond the minimum support to provide a larger context for better extrapolating time-varying partials. The length of the each expanded neighborhood is an integer multiple of the time shift a_{tonal} for tonal decomposition. The extended length of each reliable neighborhood is defined as:

$$N_{\text{neighbor}} = a_{\text{tonal}} \cdot \max\left\{\left\lceil \frac{N_{\text{gap}}}{a_{\text{tonal}}} \right\rceil, 32\right\} \tag{4.1}$$

where N_{gap} is the number of samples of the gap. The length can be shorter if the signal around the gap is stationary.

4.1.2 Sparse decomposition parameters

The sparse decomposition technique is employed in two stages of our hybrid approach: decomposing the tonal part and the transient part. For the tonal decomposition, we use a long window and a high degree of redundancy, which provide a high frequency resolution for locating spectral peaks and reliable estimated time-varying parameters for partial processing. The window size should range from 1024 to 4096 samples, and the redundancy should be no less than 4. For the transient decomposition, we use a short window and low degree of redundancy for detecting rapid changes in the amplitude or frequency of the signal. The window size in this scenario should be less than 128 samples, and the redundancy should be small but not exceed the limit of the perfect reconstruction condition for the window used.

In order to reduce the spectral leakage and to promote the sparsity of the tonal component, we use a Nuttall window in the tonal decomposition. The Nuttall window is a type of window function that consists of up to five trigonometric terms and continuous derivatives, which can provide very low peak side-lobe and fast side-lobe roll-off rate (Nuttall 1981). The Nuttall window used in tonal decomposition is defined as:

$$\psi_{\text{nuttall}}[n] = \frac{1}{\Psi} \sum_{k=0}^{3} \alpha_k \cos(2\pi k n / \Psi) \tag{4.2}$$

where Ψ is the window size, $\alpha_0 = 0.338946$, $\alpha_1 = 0.481973$, $\alpha_2 = 0.161054$, and $\alpha_3 = 0.018027$.

Meanwhile, we use the Hann window in the transient decomposition for its low redundancy for perfect reconstruction. The Hann window is defined as:

$$\psi_{\text{hann}}[n] = \frac{1}{\Psi} \left(\frac{1}{2} + \frac{\cos(2\pi n / \Psi)}{2} \right). \tag{4.3}$$

Table 4.1 and 4.2 summarize the parameters used to decompose the tonal and transient components, respectively. The window sizes (Ψ_{tonal} and $\Psi_{\text{transient}}$) and hop sizes (H_{tonal} and $H_{\text{transient}}$) control the trade-off between time and frequency resolutions of the tonal and transient decompositions. The error tolerance parameters ($\varepsilon_{\text{tonal}}$ and $\varepsilon_{\text{transient}}$) are the stopping conditions of the decomposition algorithms. The three hyperparameters in LV algorithm (σ, τ, and ρ) usually do not need to be changed. The AR order p_{tonal} determines the number of peaks in the estimated spectral envelope, which should not be too small (unable to flatten small peaks) or too large (long computing time). The parameters a_{peak}, $a_{\text{min}}^{\text{flatten}}$, $\mu_{\text{low}}^{(0)}$, $\mu_{\text{high}}^{(0)}$, and a_{shift} jointly influence the shape of the

weight rescale curve. The optimal combination of parameters should result in a curve that flattens all frequencies with partials while discarding all frequencies with noise. The maximum weight $w_{\max}$ provides an upper limit to the curve to prevent preserving energy from noise at specific frequencies.

Table 4.1 Parameters for the sparse decomposition of tonal component.

Parameter	Symbol	Value
Window size (in sample)	Ψ_{tonal}	2048
Time shift / hop size (in sample)	H_{tonal}	256
Error tolerance	$\varepsilon_{\text{tonal}}$	0.5
Step size of variable $\mathbf{u}$ in LV algorithm	σ	$2/3$
Step size of variable $\mathbf{v}$ in LV algorithm	τ	1.5
Relaxation parameter in LV algorithm	ρ	1
AR order of Burg's model	p_{tonal}	128
Amplitude threshold for prominent peaks (in dB)	a_{peak}	-50
Minimum amplitude of flattened curve (in dB)	$a_{\text{min}}^{\text{flatten}}$	-100
Initial lower bound of weight rescale curve	$\mu_{\text{low}}^{(0)}$	0.75
Initial upper bound of weight rescale curve	$\mu_{\text{high}}^{(0)}$	1.25
Amplitude shift of weight rescale curve (in dB)	a_{shift}	-30
Maximum weight	$w_{\max}$	100

Table 4.2 Parameters for the sparse decomposition of transient component.

Parameter	Symbol	Value
Window size (in sample)	$\Psi_{\text{transient}}$	64
Time shift / hop size (in sample)	$H_{\text{transient}}$	16
Error tolerance	$\varepsilon_{\text{transient}}$	0.5
Step size of variable $\mathbf{u}$ in LV algorithm	σ	$2/3$
Step size of variable $\mathbf{v}$ in LV algorithm	τ	1.5
Relaxation parameter in LV algorithm	ρ	1

4.1.3 Partial processing parameters

The partial processing procedure aims to reconstruct the tonal part of the signal in the gap region using techniques such as partial tracking, reconnection, matching, and prediction. For the partial

tracking method, we use a small amplitude threshold $a_{\text{peak}}^{\text{track}}$ to select spectral peaks that captures more partials at high frequency. However, a too small threshold may also select some noise as partials, resulting in distortion. The polynomial order Q of the exponential sinusoidal model used in DDM is set to 2 to capture the linear variation of frequency and to simplify the prediction using linear prediction. Parameters ζ_a, ζ_f, ζ_β control the cost of connecting two spectral peaks into a partial. The larger these three parameters are, the more likely it is for the peaks to be connected, even if their amplitudes, frequencies, and frequency slopes are far apart. On the other hand, very small parameters may prevent the partials from connecting, leading to overly fragmented partials. The parameter δ^{track} adjusts the preference between useful and spurious assignments, where all assignments are spurious if $\delta^{\text{track}} = 0$ and useful if $\delta^{\text{track}} = 1$. The determined partial tracking parameters are shown in Table 4.3.

Table 4.3 Parameters for partial tracking.

Parameter	Symbol	Value
Window size (in sample)	Ψ_{partial}	2048
Amplitude threshold for peak picking (in dB)	$a_{\text{peak}}^{\text{track}}$	-50
Order of exponential sinusoidal model in DDM	Q	2
Preference between useful and spurious assignments	δ^{track}	0.25
Assignment threshold for amplitude change (in dB)	ζ_a	15
Assignment threshold for frequency change (in mel)	ζ_f	15
Assignment threshold for frequency slope change	ζ_β	0.002

In the process of partial re-connection, we iterate through each partial in order from longest to shortest, and compare that partial to all other partials in each iteration for reconnection. When looking for potential reconnections, we exclude short partials of length less than $l_{\text{min}}^{\text{connect}}$ from the outer loop because they are hard to predict. Parameter $l_{\text{min}}^{\text{connect}}$ should not be too large, which would prohibit the reconnection of fragmented partials. Parameter $\Delta_f^{\text{boundary}}$ controls the number of candidate partials for the inner loop, which should be set as small as possible without affecting potential reconnections. The maximum length of partial overlap $l_{\text{max}}^{\text{overlap}}$ and $l_{\text{max}}^{\text{jump}}$ should not be too large to avoid misconnecting two independent partials. The parameter configuration is in Table 4.4.

For the partial matching method, we ignore short partials of length less than $l_{\text{min}}^{\text{match}}$ near the gap for matching. The amplitude slope ratio β_a^{attack} used to detect attacks should be large enough, otherwise it may wrongly consider the rising part of a periodic signal as an attack. The parameters are summarized in Table 4.5.

Table 4.4 Parameters for partial reconnection.

Parameter	Symbol	Value
Partial length threshold for potential reconnection (in window)	$l_{\min}^{\text{connect}}$	6
Minimum relative frequency difference between partial bounds	$\Delta_f^{\text{boundary}}$	0.06
Assignment threshold for amplitude change (in dB)	ζ_a^{connect}	12
Assignment threshold for frequency change	ζ_f^{connect}	0.02
Preference between useful and spurious assignments	δ^{connect}	0.5
Maximum length of partial overlap (in window)	$l_{\max}^{\text{overlap}}$	5
Maximum length of partial jump (in window)	$l_{\max}^{\text{jump}}$	3

Table 4.5 Parameters for partial matching.

Parameter	Symbol	Value
Partial length threshold for potential matching (in window)	$l_{\min}^{\text{match}}$	4
Amplitude slope ratio threshold for attack detection	β_a^{attack}	2.5
Assignment threshold for amplitude change (in dB)	ζ_a^{match}	10
Assignment threshold for frequency change	ζ_f^{match}	0.012
Preference between useful and spurious assignments	δ^{match}	0.6

For the partial prediction, we use two R^2 thresholds to examine the existence of frequency and amplitude trends. The threshold of frequency trend R_f^2 should be larger than that of amplitude trend R_a^2 because the amplitude fluctuates more than frequency. The AR order of Burg's method p_{predict} should be large for better prediction, which is set to $\lceil l_p/2 \rceil$ in our implementation, where l_p is the length of the partial used for prediction and determinated by the method. The minimum amplitude T_a of the parabola trend should be a very small value so that a partial can hardly be perceived at its birth or death. The values of these parameters are shown in Table 4.6, where l_{gap} is the length of the gap (in window), estimated by the method.

4.1.4 Noise reconstruction and post-processing parameters

To analyze the noise component, we removed small segments of signal near the boundaries of the gap, where the tonal decomposition may have energy leakage. The length N_{ignore} of these segments should not be too long, as this will reduce the amount of reliable information near the gap. Similarly, the two crossfade lengths for noise (Ψ_{noise}) and reliable signals ($N_{\text{crossfade}}$) should be small. The pole number p_{noise} of the AR model for estimating the spectral envelope should be sufficient to

Table 4.6 Parameters for partial prediction.

Parameter	Symbol	Value
R^2 threshold for the existence of frequency trend	R_f^2	0.92
R^2 threshold for the existence of amplitude trend	R_a^2	0.88
AR order of Burg's model for linear prediction	p_{predict}	$\lceil l_p/2 \rceil$
Minimum amplitude threshold of a parabola curve (in dB)	T_a	-80
Length of partial birth (attack) in the gap (in window)	l_{birth}	l_{gap}
Length of partial death (decay) in the gap (in window)	l_{death}	l_{gap}

capture all prominent peaks of the noise component. Table 4.7 summarizes the configuration of these parameters in our experiments.

Table 4.7 Parameters for noise reconstruction and post-processing.

Parameter	Symbol	Value
Length to ignore for analysis (in sample)	N_{ignore}	$\lceil \Psi_{\text{tonal}}/4 \rceil$
AR order of Burg's model	p_{noise}	128
Cosine window size for crossfading noise signals (in sample)	Ψ_{noise}	32
Crossfade length for reliable neighborhoods (in sample)	$N_{\text{crossfade}}$	128

4.2 Evaluation Metrics

This section presents three different metrics that are used to evaluate the quality of the reconstructed signals for the proposed hybrid inpainting approach and other inpainting methods.

The first metric is the signal-to-noise ratio (SNR), which is also known as the signal-to-distortion ratio in the context of audio inpainting (Taubock et al. 2021). The SNR is defined as follows:

$$\text{SNR}(\mathbf{y}, \hat{\mathbf{y}}) = 10 \log_{10} \frac{\|\mathbf{y}\|_2^2}{\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2} \tag{4.4}$$

where $\mathbf{y}$ and $\hat{\mathbf{y}}$ represent the original signal (without gaps) and the inpainted signal, respectively. A higher SNR value indicates a better reconstruction of the audio signal. Alternatively, the SNR can be computed only on the gap region, since the reliable parts are usually unchanged in inpainting methods (except for a very short crossfade in our hybrid approach):

$$\text{SNR}_{\text{gap}}(\mathbf{y}, \hat{\mathbf{y}}) = 10 \log_{10} \frac{\|\mathbf{M}_{\text{G}}\mathbf{y}\|_2^2}{\|\mathbf{M}_{\text{G}}\mathbf{y} - \mathbf{M}_{\text{G}}\hat{\mathbf{y}}\|_2^2} \tag{4.5}$$

where $\mathbf{M}_{\mathrm{G}} \in \mathbb{R}^{N \times N}$ is a diagonal matrix that is the complement of $\mathbf{M}_{\mathrm{R}}$ such that $\mathbf{M}_{\mathrm{G}}\mathbf{y}$ extracts all unreliable (gap) samples in $\mathbf{y}$.

The second metric is the Itakura-Saito distance (ISD), more properly referred to as the Itakura-Saito divergence, which quantifies the spectral (dis)similarity between an original power spectrum P and its approximation $\hat{P}$ (Itakura and Saito 1968). Unlike SNR, ISD takes into account the frequency characteristics of the signals. The ISD is defined as follows:

$$\mathrm{ISD}(P, \hat{P}) = \frac{1}{K} \sum_{k=1}^{K} \left(\frac{P[k]}{\hat{P}[k]} - \ln \frac{P[k]}{\hat{P}[k]} - 1 \right) \tag{4.6}$$

where K represents the number of spectral data points in P. A lower ISD indicates a higher spectral similarity between the original and reconstructed signals, which implies a better quality of reconstruction.

The third metric is a perceptual metric called objective difference grade (ODG), which measures the perceptual similarity between the original and reconstructed signals. The ODG corresponds to the subjective difference grade obtained from subjective listening tests, which is specified in the ITU-R recommendation BS.1387 (ITU-R 1998). The ODG ranges from 0 to –4 and can be interpreted as shown in Table 4.8.

Table 4.8 Interpretation of ODG levels.

Description	ODG
Imperceptible	0
Perceptible but not annoying	-1
Slightly annoying	-2
Annoying	-3
Very annoying	-4

There are several methods that implement ODG, such as the Perceptual Evaluation of Audio Quality (PEAQ) method (Thiede et al. 2000). In this thesis, we use the PEMO-Q method to calculate the ODG, which has a more advanced auditory model based on a modulation filterbank and demonstrates higher prediction accuracy than PEAQ (Huber and Kollmeier 2006).

4.3 Experiment 1: Separation of the Three Components

In this section, we evaluate the performance of our proposed hybrid approach for decomposing an input signal with gaps into three components (layers): tonal, transient, and noise. The decom-

position process is illustrated in Figure 4.1, where these three components are extracted layer by layer from the input signal. First, the tonal layer is obtained by applying the tonal decomposition algorithm to the input signal. Then, the transient layer is derived by applying the transient decomposition algorithm to the residual of the tonal decomposition. Finally, the noise layer is the residual of the transient decomposition.

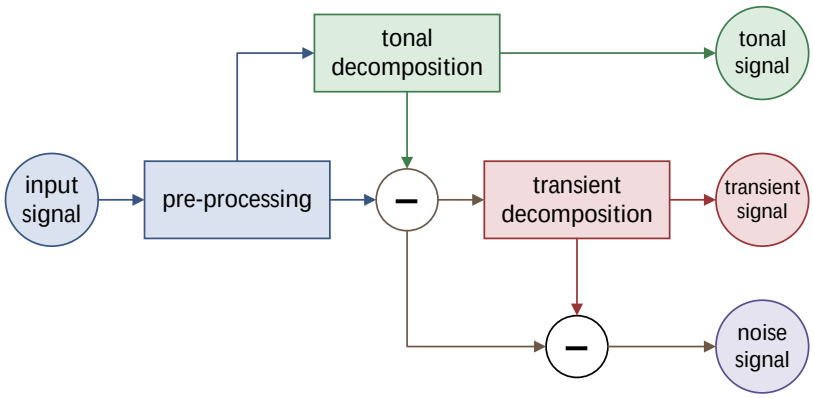
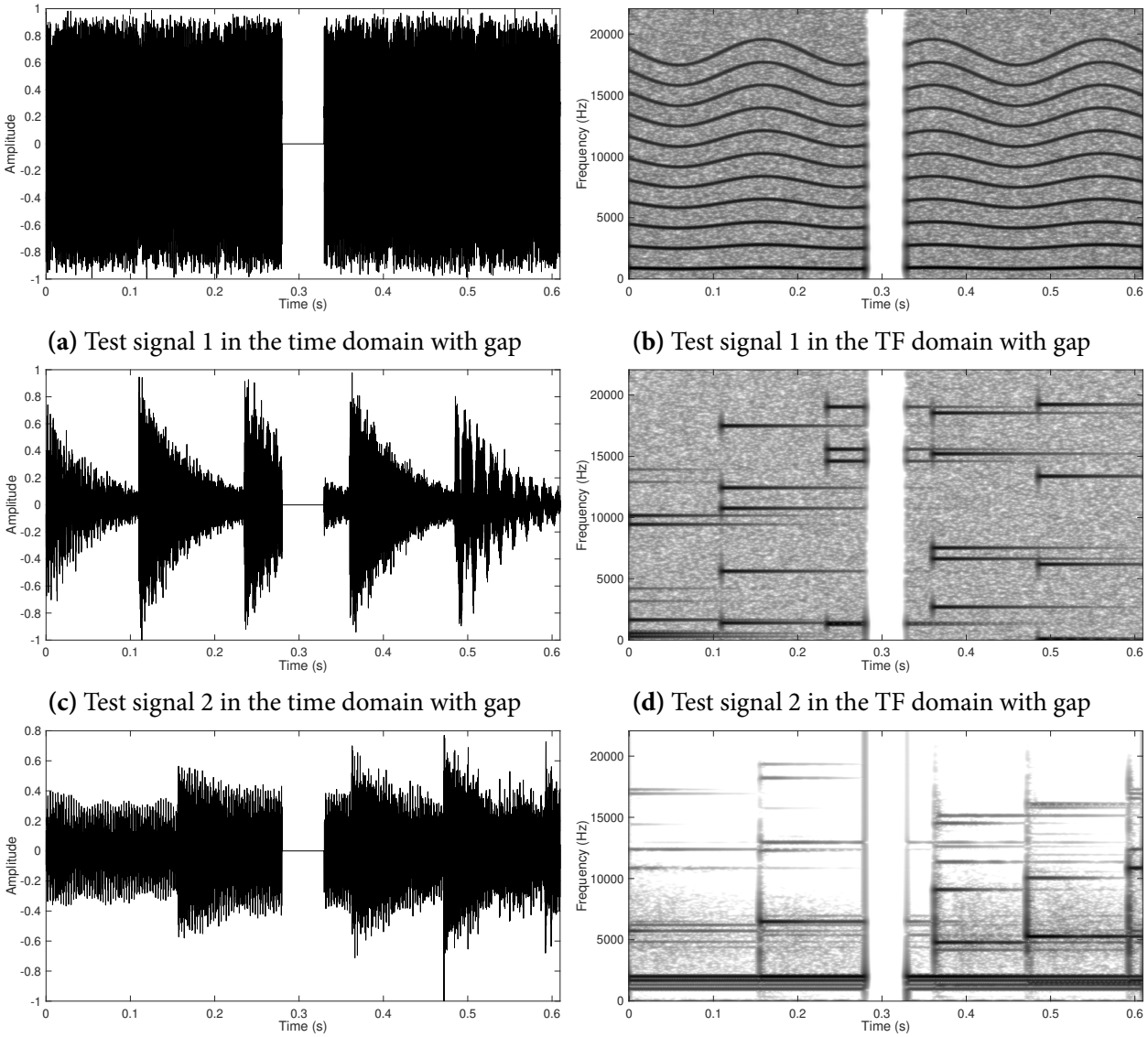


Figure 4.1 The decomposition process for Experiment 1.

In this experiment, we use three different signals to test our hybrid approach. The first signal is a synthesized harmonic signal with modulations and additive noise, which contains only tonal and noise components. The second signal is a synthesized inharmonic signal with exponentially damped sinusoids with a ramp-up attack and additive noise, which contains all three components. The third signal is a recording of a glockenspiel melodic phrase from the Sound Quality Assessment Material (SQAM) dataset (European Broadcasting Union 2008), which is a real audio signal that also contains all the three components. These signals are visualized in Figure 4.2. All test signals are mono, 44.1 kHz, and 16 bits.

Figure 4.3 shows the time domain and the time-frequency (TF) domain representations of the original signal (with a gap), the decomposed tonal component, the decomposed transient component, and the noise component for the first signal (tonal + noise). We can observe that our hybrid approach successfully separates the tonal and noise components from the input signal. No transients are extracted except for the discontinuity at the boundaries of the gap, which is expected since there are no transients in the original signal. However, we notice that for partials with greater modulation depths in the high frequency, the energy leakage of tonal decomposition is more significant, which means that some energy from the tonal layer is leaking to the noise layer.

Figure 4.4 shows the decomposition results for the second signal (tonal + transient + noise). Our hybrid approach achieves good separation of these three components from the input signal. However, we observe that there are some energy leakages from the tonal and transient decompositions



(a) Test signal 1 in the time domain with gap

(b) Test signal 1 in the TF domain with gap

(c) Test signal 2 in the time domain with gap

(d) Test signal 2 in the TF domain with gap

(e) Test signal 3 in the time domain with gap

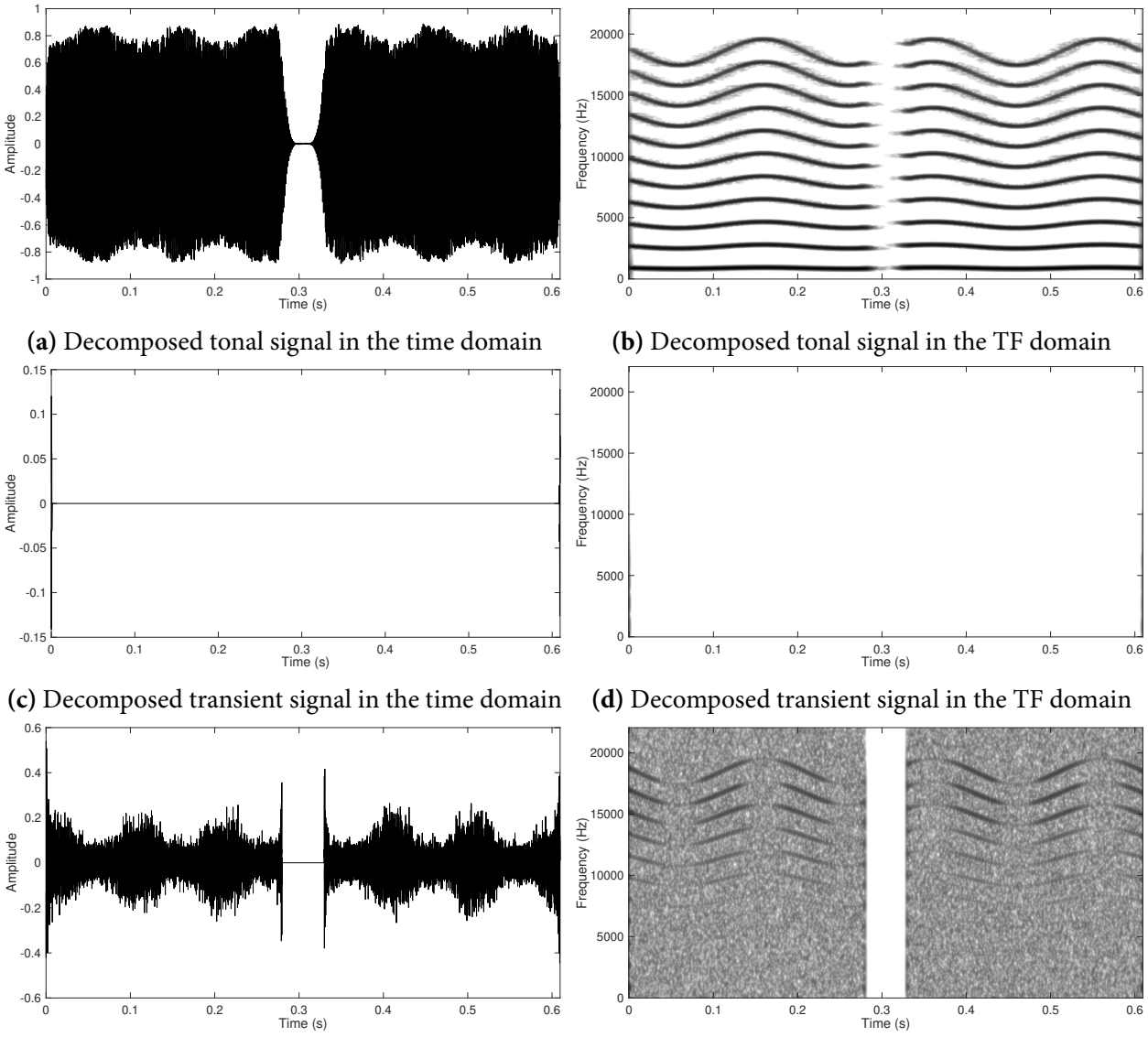
(f) Test signal 3 in the TF domain with gap

Figure 4.2 Time and time-frequency domain representations of the three test signals used for Experiment 1. The noise level for test signals 1 and 2 is at –20 dB. The length of the gap is 50 milliseconds.

to the noise layer.

Figure 4.5 shows the decomposition results for the third signal (recording with all three components). Our hybrid approach demonstrates good separation of these three components from the input signal. However, there are transients remaining in the tonal layer and some noise remaining in the transient layer*.

*The audio excerpts and supplemental figures can be accessed through the webpage: https://etosphere.github.io/hybrid-inpainting-approach-demo/.



(a) Decomposed tonal signal in the time domain

(b) Decomposed tonal signal in the TF domain

(c) Decomposed transient signal in the time domain

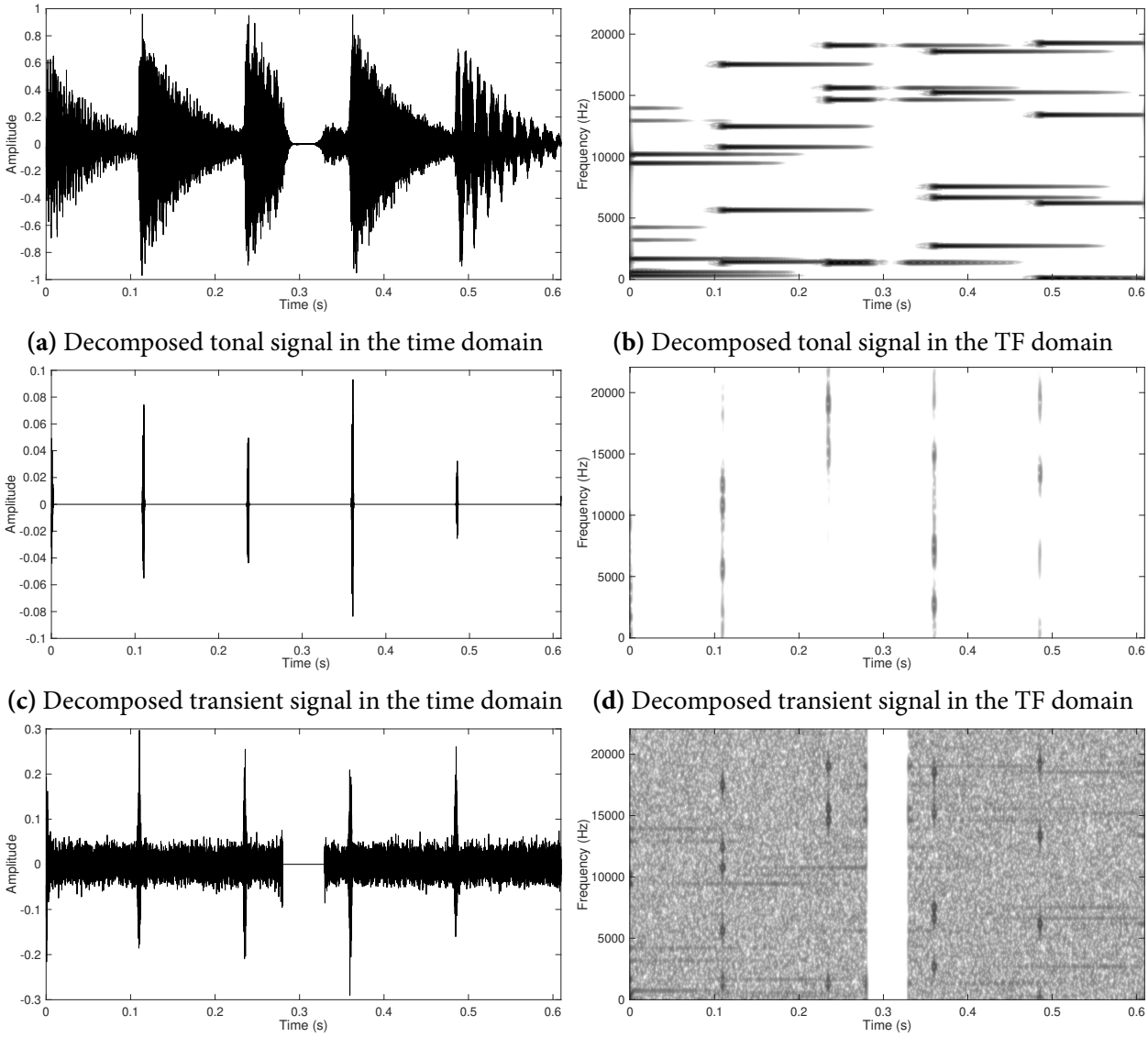
(d) Decomposed transient signal in the TF domain

(e) Decomposed noise signal in the time domain

(f) Decomposed noise signal in the TF domain

Figure 4.3 Decomposition results for test signal 1 for a noise level of -20 dB in Experiment 1.

We discuss our findings from this experiment as follows. Our hybrid approach succeeds in decomposing an input signal with gaps into three components: tonal, transient, and noise. The tonal decomposition performs better for stationary or slow-varying signals but is less ideal for signals with fast modulations. Figure 4.6 illustrates the reconstruction quality of tonal and transient components at different noise levels for the first two test signals: one with modulations and one without modulations. We can observe that the reconstruction SNR is lower for the signal with modulations compared with the more stationary signal. This may be because we build the reweighting curves based on the assumption that the left and right reliable parts are approximately stationary for estimating the spectral envelope, and this assumption does not hold for signals with fast-varying



(a) Decomposed tonal signal in the time domain

(b) Decomposed tonal signal in the TF domain

(c) Decomposed transient signal in the time domain

(d) Decomposed transient signal in the TF domain

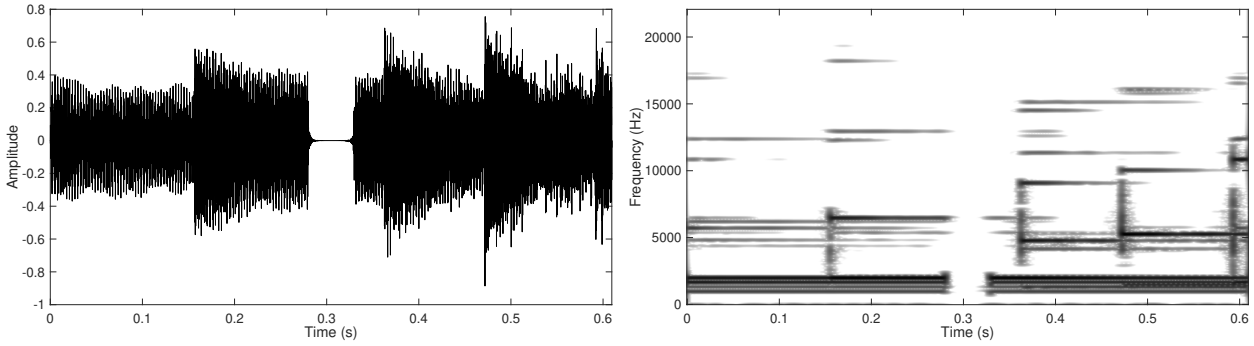
(e) Decomposed noise signal in the time domain

(f) Decomposed noise signal in the TF domain

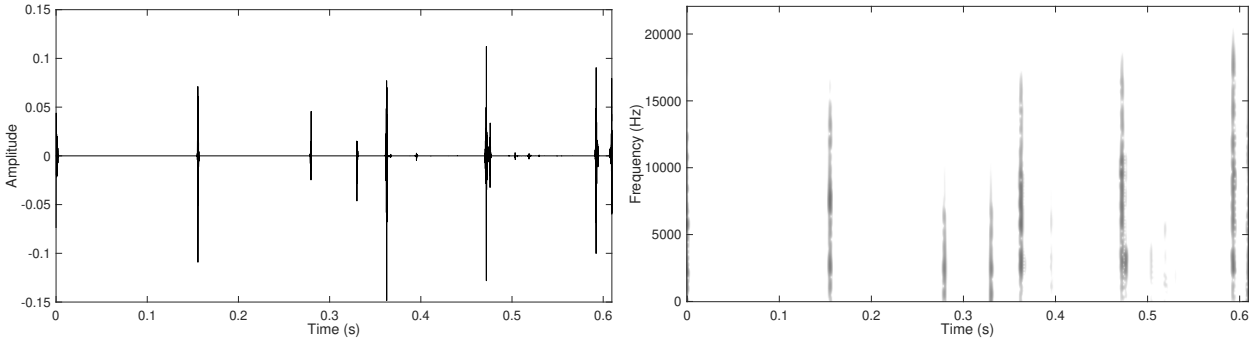
Figure 4.4 The decomposition results for test signal 2 at a noise level of -20 dB in Experiment 1.

partials.

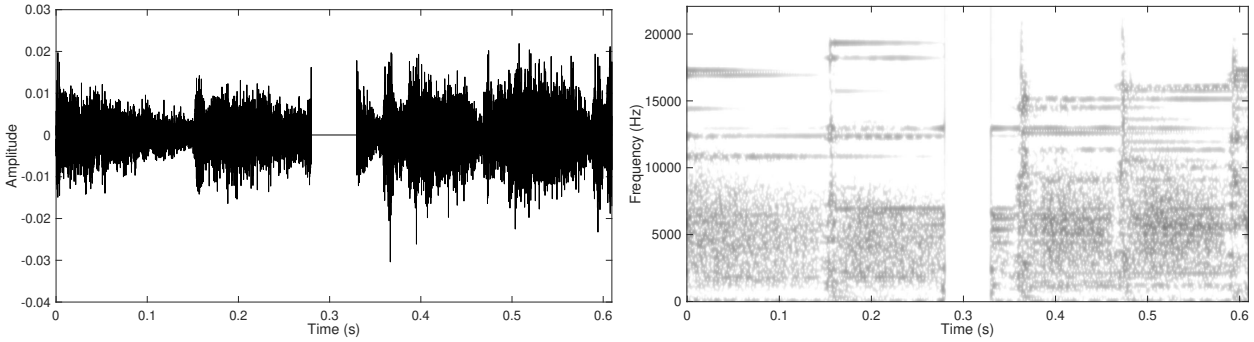
There are also some energy leakages from the decompositions to the transient and noise layers, which can influence subsequent inpainting processes. This may be because of inappropriate choices of sparsity parameters λ for tonal and transient decompositions. Too low λ keeps a portion of energy that belongs to lower layers (transient and noise) in upper layers (tonal and transient), leading to unsuccessful decompositions. On the other hand, too high λ allows some energy from the upper layers (tonal and transient) to slip into the lower layers (transient and noise), leading to excessive energy leakage.



(a) Decomposed tonal signal in the time domain **(b)** Decomposed tonal signal in the TF domain



(c) Decomposed transient signal in the time domain **(d)** Decomposed transient signal in the TF domain



(e) Decomposed noise signal in the time domain **(f)** Decomposed noise signal in the TF domain

Figure 4.5 The decomposition results for test signal 3 in Experiment 1.

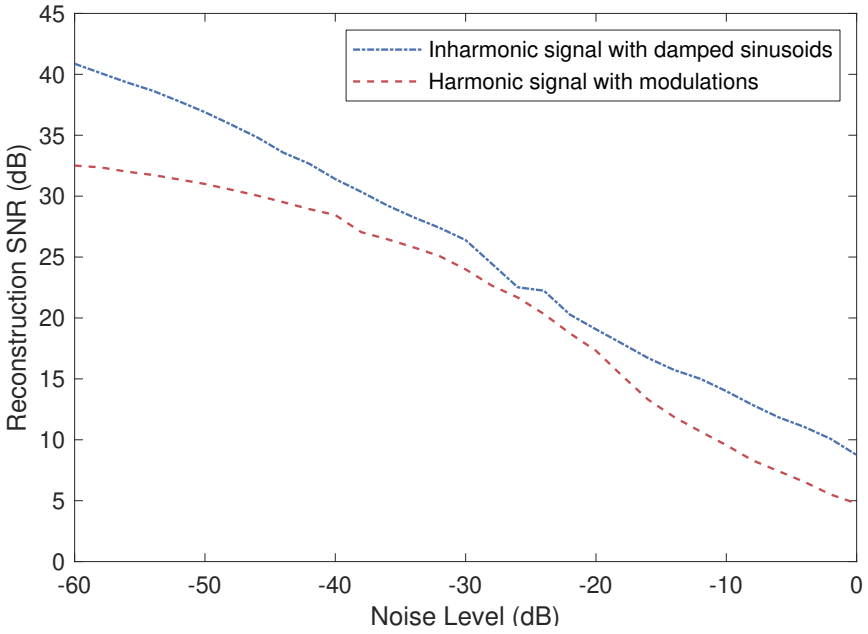


Figure 4.6 The reconstruction SNR of tonal and transient components for two signals at different noise levels. The red dashed line corresponds to test signal 1 and the blue dash-dotted line corresponds to test signal 2.

4.4 Experiment 2: Partial Processing and Tonal Reconstruction

In this section, we test the reconstruction quality of the tonal component. The reconstruction process involves partial analysis, prediction, and resynthesis. As shown in Figure 4.7, after pre-processing, the input signal is first decomposed to extract its tonal component. Then, the partials of the decomposed tonal component will be tracked, reconnected, and matched to predict the partials in the gap. Next, the tonal signal in the gap will be synthesized from the predicted partials. Finally, the inpainted tonal component will be obtained by replacing the gap region of the decomposed tonal signal with the synthesized signal, with a small amount of crossfading at the gap boundaries.

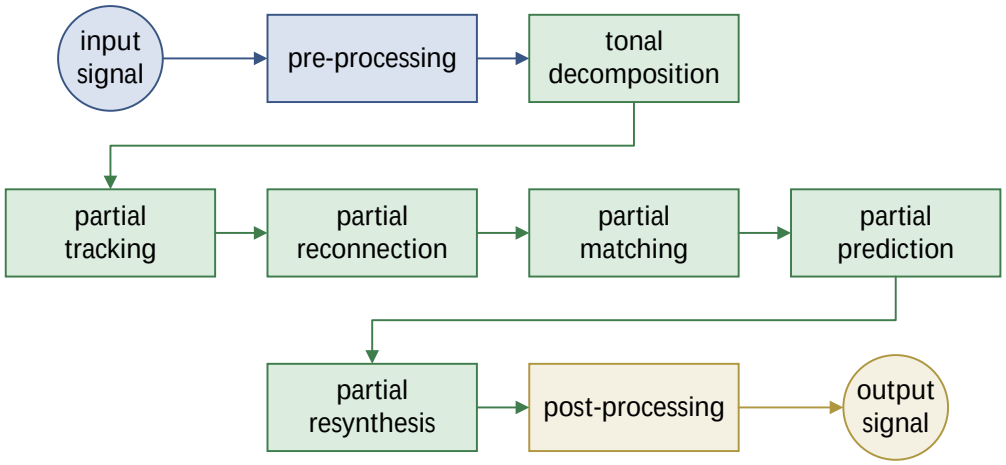
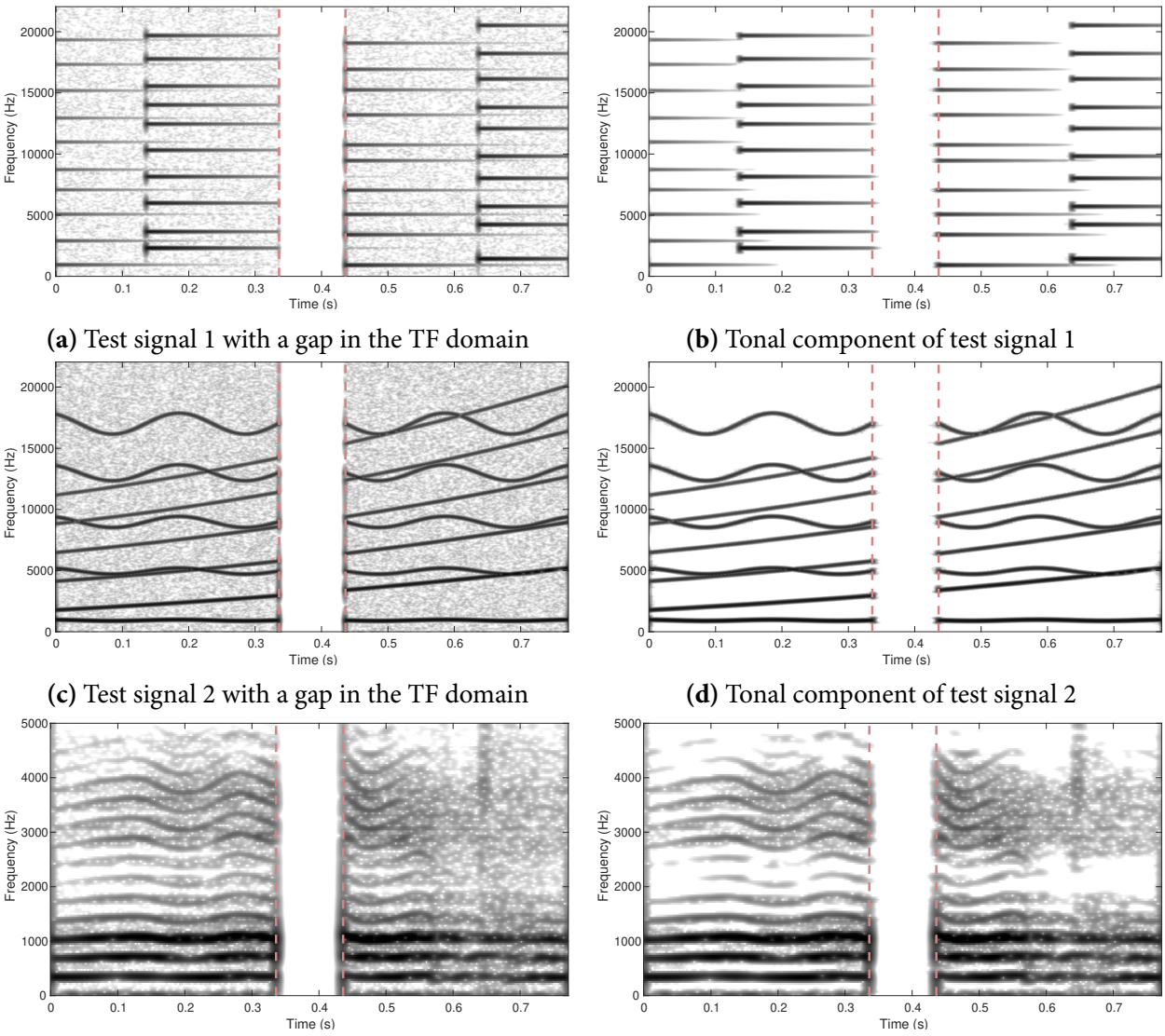


Figure 4.7 The partial reconstruction process for Experiment 2.

In this experiment, we use three signals to test our partial processing and tonal reconstruction methods. The first signal is a synthesized inharmonic signal with exponentially damped sinusoids and additive noise. The second signal is a synthesized signal with harmonic modulations, inharmonic quadratic chirps, and additive noise. The third signal is a recording of a soprano voice with vibrato from the Sound Quality Assessment Material (SQAM) dataset (European Broadcasting Union 2008). These three test signals are visualized in the left panel of Figure 4.8. All test signals are mono, 44.1 kHz, and 16 bits. The extracted tonal components of these test signals are shown in the right panel of Figure 4.8.

For the first test signal, Figure 4.9 shows the intermediate results of the process of inpainting the tonal part. We can find that partials are successfully detected from the decomposed tonal part (Figure 4.9a), and some fragmented partials with small amplitudes are reconnected (Figure 4.9b). The predicted partials in the gap are not misconnected to each other, which indicates that the partial matching method works as expected (Figure 4.9c). The reconstructed tonal part has smooth attacks and decays in the gap (Figure 4.9d).



(a) Test signal 1 with a gap in the TF domain

(b) Tonal component of test signal 1

(c) Test signal 2 with a gap in the TF domain

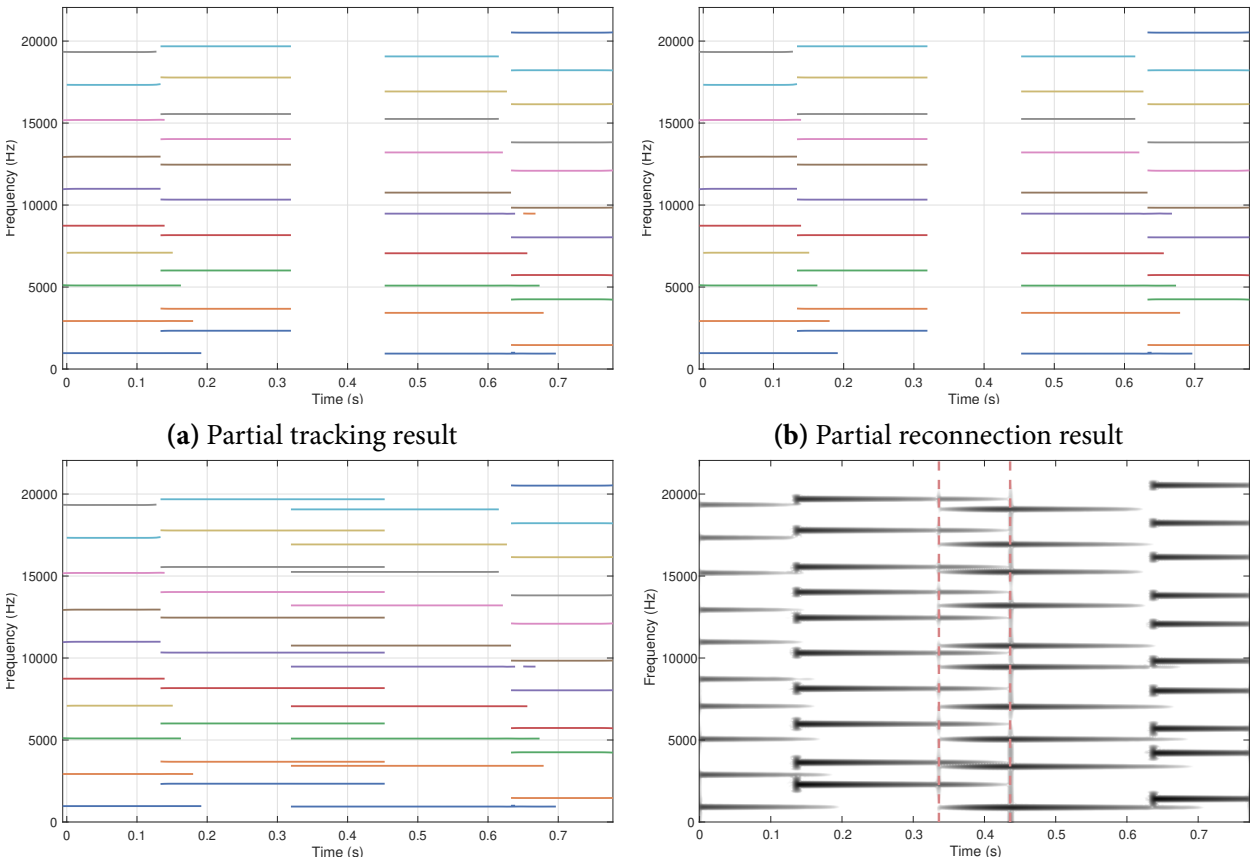
(d) Tonal component of test signal 2

(e) Test signal 3 with a gap in the TF domain

(f) Tonal component of test signal 3

Figure 4.8 Time-frequency domain representations of the three test signals and their decomposed tonal components used for Experiment 2. The noise level for test signals 1 and 2 is at –30 dB. The length of the gap is 100 milliseconds.

Figure 4.10 shows the results for the second test signal. At the partial tracking stage, the method works well for isolated partials. However, some problems may arise when partials cross or converge (Figure 4.10a). At the reconnection stage, some truncated partials are reconnected correctly, whereas some other partials are connected wrongly because of serious frequency interferences (Figure 4.10b). In general, from the partial prediction result (Figure 4.10c) and the reconstructed tonal component (Figure 4.10d), we find that most partials are predicted correctly. The prediction may become inaccurate for heavily fragmented partials and partials with higher interference. At the



(a) Partial tracking result

(b) Partial reconnection result

(c) Partial prediction result

(d) Reconstructed tonal component in the TF domain

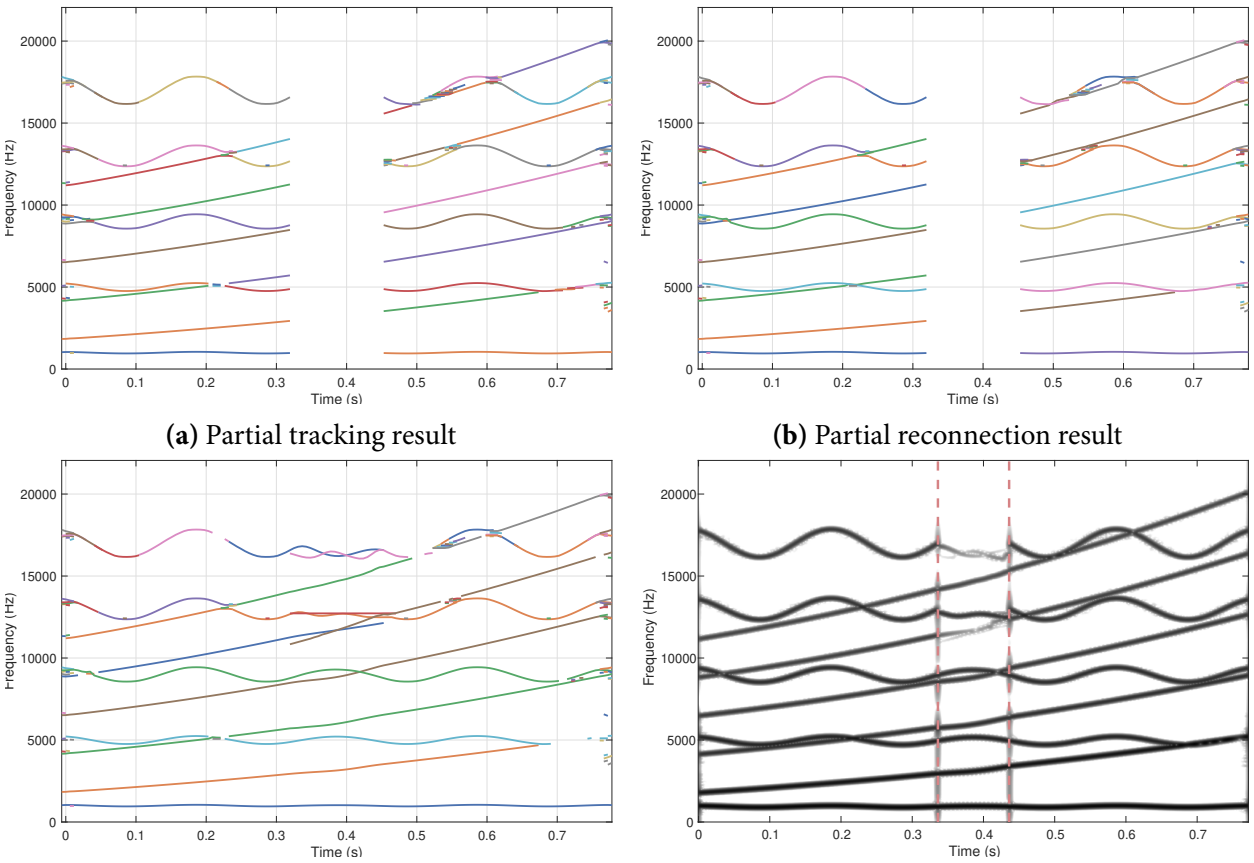
Figure 4.9 Partial reconstruction results at different stages for test signal 1 in Experiment 2.

same time, some unwanted attacks and decays and some discontinuity at the boundary of the gap may occur.

Figure 4.11 shows the tonal reconstruction results of the third test signal. We can observe that the presence of noise makes the tracked partials more fragmented, and some of these partials are composed of noise and are unpredictable (Figure 4.11a). The reconnection process connects truncated partials to overcome fragmentation, but some partials composed of noise are misconnected (Figure 4.11b). From the prediction and reconstruction results (Figure 4.11c and 4.11d), we find that the prediction quality is high for the low-frequency part, where the noise interference is small, while the prediction is inaccurate for the high-frequency part, where the noise interference is large*.

In conclusion, our tonal reconstruction method is able to predict the trajectories of partials so that their temporal evolutions are more similar to their origins than what other methods provide.

*The audio excerpts and supplemental figures can be accessed through the webpage: https://etosphere.github.io/hybrid-inpainting-approach-demo/.



(a) Partial tracking result

(b) Partial reconnection result

(c) Partial prediction result

(d) Reconstructed tonal component in the TF domain

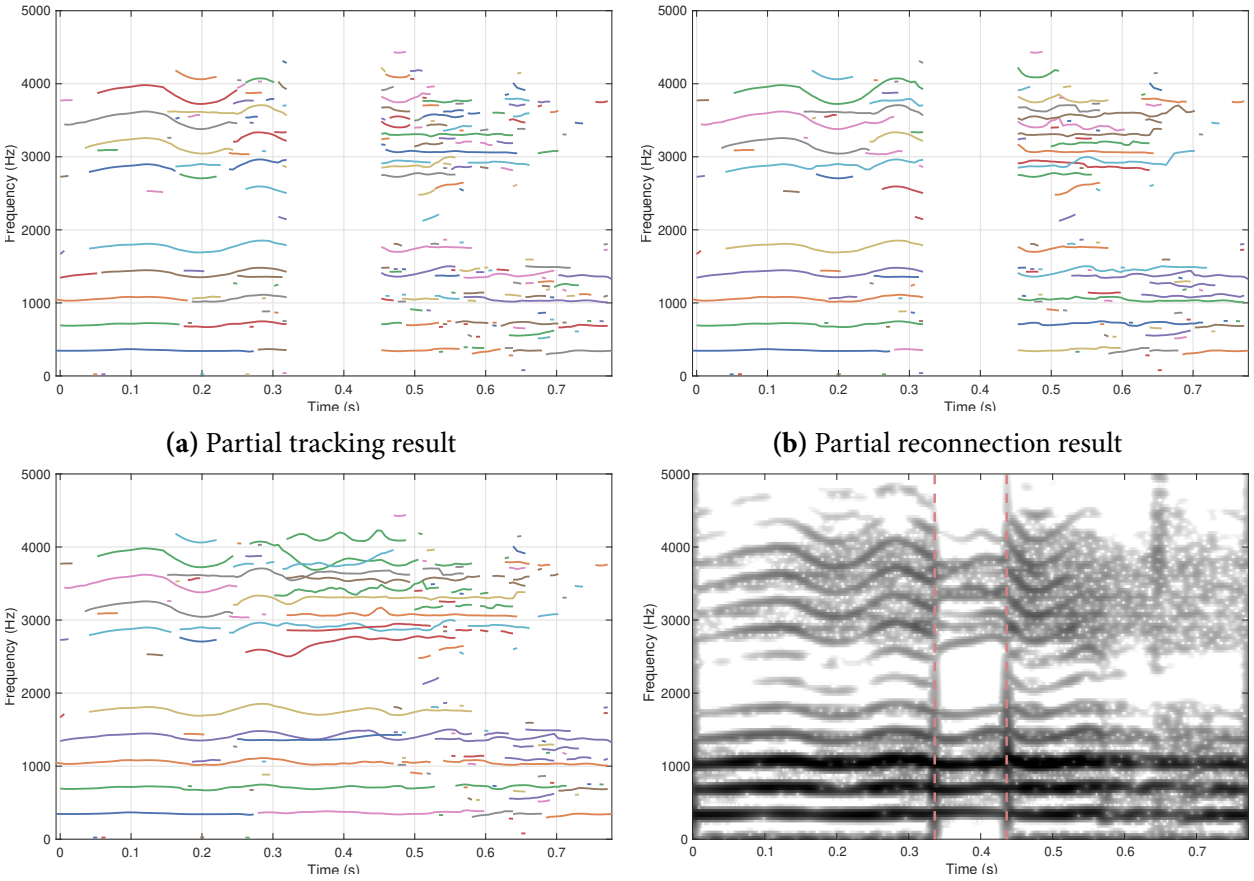
Figure 4.10 Partial reconstruction results at different stages for test signal 2 in Experiment 2.

However, some problems still exist in some cases. During the partial tracking process, we find that when two partials cross each other, they interfere with each other on both frequency and amplitude, resulting in inaccurate partial parameters estimated by the DDM, which may further lead to the following scenarios:

- The trajectories are interrupted, producing some new partials.
- Trajectories are misconnected, so that one partial may connect to another after the crossing.
- Many fragmented partials appear near the intersection.

The partial reconnection process is capable of reducing the fragmentation of partials. However, if the trajectory of a partial has been deviated by the interference or the partials are too fragmented, the results of this process are less ideal.

Our partial prediction method, which incorporates both trend and AR components, making it capable of reconstructing both partials with increasing or decreasing frequency, and partials with periodic changes in frequency. By adjusting the corresponding parameter (R^2), we can control



(a) Partial tracking result

(b) Partial reconnection result

(c) Partial prediction result

(d) Reconstructed tonal component in the TF domain

Figure 4.11 Partial reconstruction results at different stages for test signal 3 in Experiment 2.

the sensitivity of extracting the trend component. If the prediction method is too sensitive, in other words, if R^2 is too large, it will be easier to incorrectly extract the trend component from the periodic trajectories.

For the prediction of a partial's amplitude, more energy is preserved in the gap region due to the use of a parabola instead of a linear trend component. However, the parabola trend cannot simulate a fast attack with an asymmetric shape, and we do not try to predict the temporal location and amplitude of the attack based on audio information.

4.5 Experiment 3: Noise Reconstruction

In this section, we test and compare the reconstruction quality of the noise component using two different techniques: crossfade-based technique and LAR interpolation technique. The process is illustrated in Figure 4.12, where the input noise signal with a gap is analyzed and reconstructed separately using the two methods, resulting in two inpainted signals correspondingly.

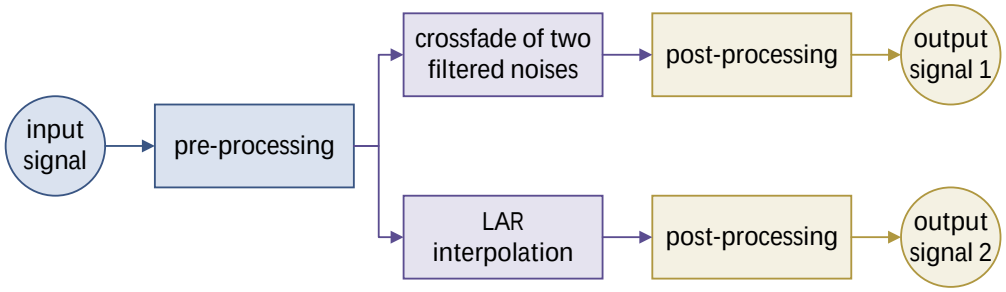


Figure 4.12 The noise reconstruction process for Experiment 3.

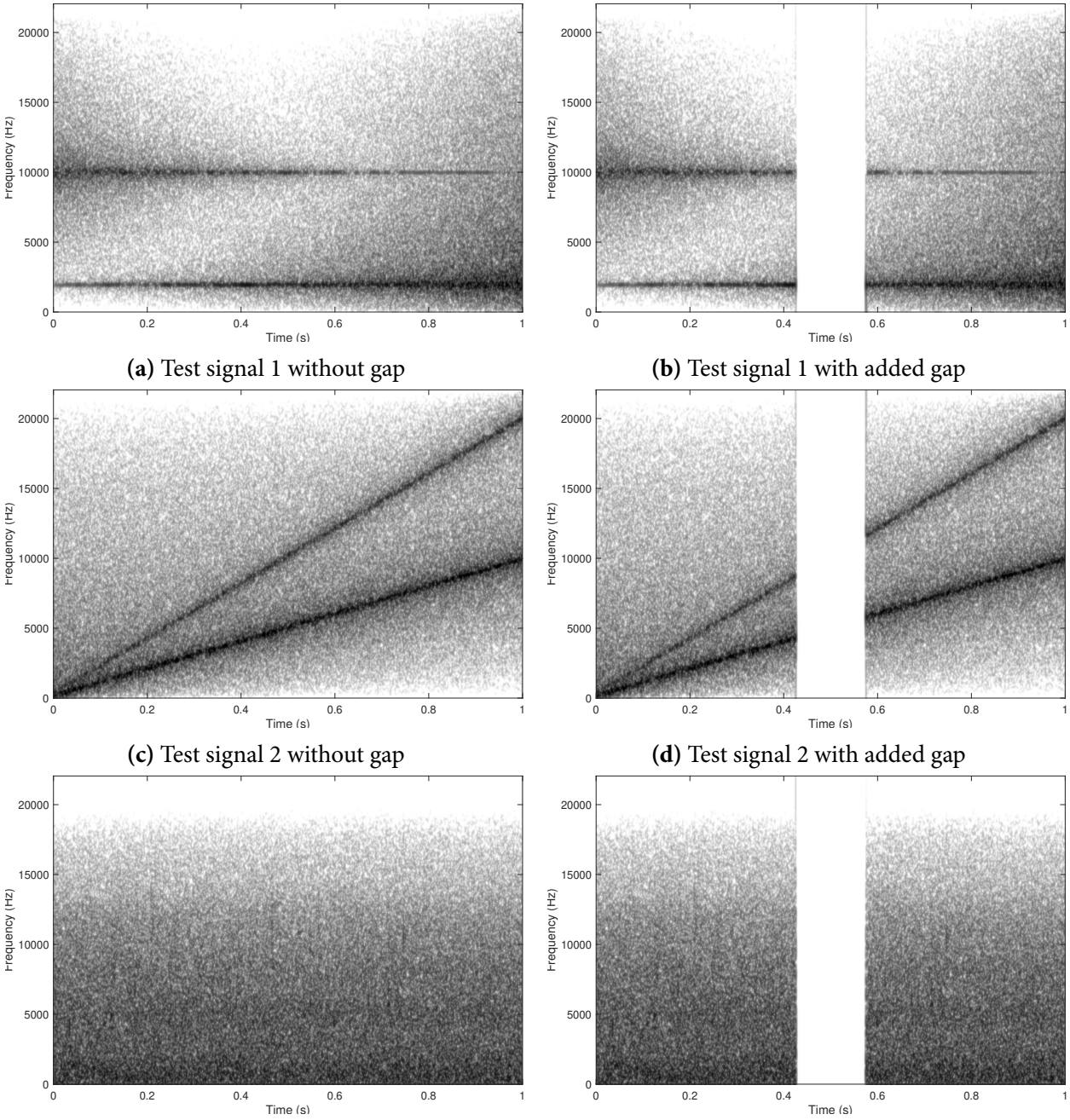
We use three test signals in this experiment, which are visualized in Figure 4.13. The first test signal is a synthesized signal with the superposition of two time-varying filtered noises, whose bandwidths are varying with time. The second test signal is also a synthesized signal with the superposition of two time-varying filtered noises, but whose center frequencies are varying with time. The third test signal is a recording of a rain sound, which is obtained from an environmental sound dataset named BDLib2 (Bountourakis et al. 2015; Bountourakis et al. 2019). All signals are mono, 44.1 kHz, and 16 bits.

Figure 4.14 illustrates the reconstruction results using the two noise reconstruction techniques for the first test signal. We can observe from the spectrograms that the reconstructed signals from both methods are similar to the original test signal without gap.

Figure 4.15 shows the reconstruction results for the second test signal. In this case, we can see that the crossfade-based technique (Figure 4.15a) ends up with stationary resonance frequencies in the gap region, while the LAR interpolation technique (Figure 4.15b) successfully reconstructs the time-varying resonance frequencies, although the reconstruction quality of the filtered noise with faster frequency variation is lower than that of the slower-varying one.

Figure 4.16 presents the reconstruction results for the third test signal. The reconstruction quality is very similar to that of the first test signal, and there is no noticeable difference between the two methods*.

*The audio excerpts and supplemental figures can be accessed through the webpage: https://etosphere.github.io/hybrid-inpainting-approach-demo/.



(a) Test signal 1 without gap

(b) Test signal 1 with added gap

(c) Test signal 2 without gap

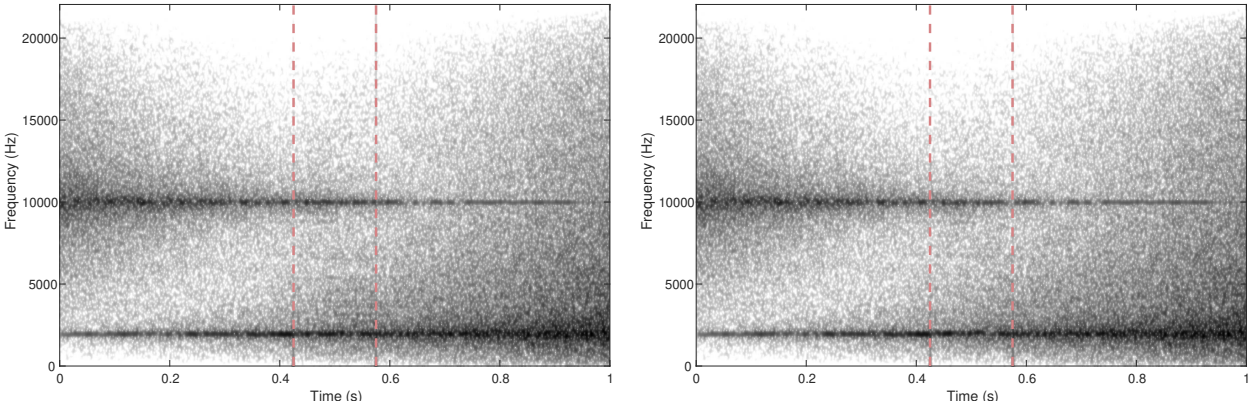
(d) Test signal 2 with added gap

(e) Test signal 3 without gap

(f) Test signal 3 with added gap

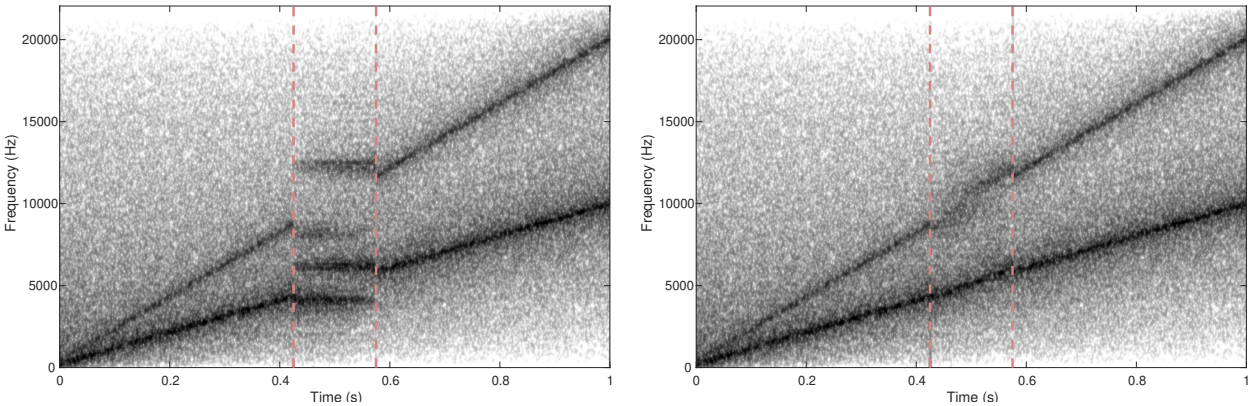
Figure 4.13 Time-frequency representations of the three test signals (with or without gap) in Experiment 3. The length of the gap is 150 milliseconds.

To evaluate the reconstruction quality of the three test signals quantitatively, we use the time-varying Itakura-Saito distance (TV-ISD) between the reconstructed and the original noise signals (without the gap) as a metric. We use a Hann window to segment the gap region of both signals, with a window size of 2048 and a hop size of 512. For each segment, we calculate the ISD between



(a) Crossfade of two filtered noises **(b)** LAR interpolation

Figure 4.14 Noise reconstruction results for test signal 1 using two different techniques in Experiment 3.



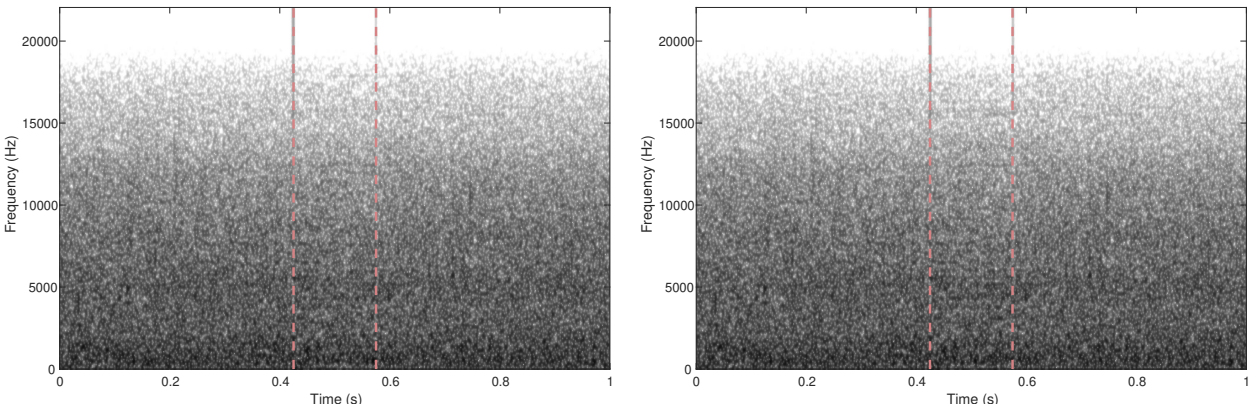
(a) Crossfade of two filtered noises **(b)** LAR Interpolation

Figure 4.15 Noise reconstruction results for test signal 2 using two different techniques in Experiment 3.

these two signals. The PSDs for each segment are estimated using the Burg method with an order of 8. The TV-ISD is a vector that contains the ISD calculated from each segment.

The evaluation result is shown in Table 4.9. We can see that for noise signals with small variations of the local maximum of their PSDs, such as test signals 1 and 3, the crossfade-based technique yields slightly better and more consistent reconstruction quality than the LAR interpolation technique. On the other hand, for noise signals with large variations at the local maximum of their PSDs, such as test signal 2, the LAR interpolation technique significantly outperforms the crossfade-based technique in terms of reconstruction quality.

Therefore, to achieve a better reconstruction of the noise component, we need to choose which technique to use based on the characteristics of the signal. For signals with stationary or slowly



(a) Crossfade of two filtered noises **(b)** LAR Interpolation

Figure 4.16 Noise reconstruction results for test signal 3 using two different techniques in Experiment 3.

Table 4.9 The time-varying Itakura-Saito distance (TV-ISD) of reconstructed signals using two noise reconstruction techniques for three test signals. The values are acquired by repeating the process a 100 times.

	Mean TV-ISD (Standard Deviation)		
Reconstruction technique	Test signal 1	Test signal 2	Test Signal 3
Crossfade of two filtered noises	**0.107 (0.042)**	1.377 (0.244)	0.041 (0.009)
Interpolation of LAR coefficients	0.443 (0.398)	**0.229 (0.048)**	0.039 (0.009)

varying PSDs, we can use the crossfade-based technique, which is simpler and faster. For signals with fast varying PSDs, we should use the LAR interpolation technique, which can capture and preserve the temporal variations better. An automatic selection between the two techniques could be achieved in future research through measuring the variability of the PSDs and selecting based on a predefined threshold.

4.6 Experiment 4: Comparison with Other Inpainting Methods

In this experiment, we compare the reconstruction quality of our hybrid approach (Hybrid) with four state-of-the-art inpainting methods using 10 real audio signals. The four inpainting methods are the analysis variant of SPAIN method (A-SPAIN) (Mokrý et al. 2019), the weighted Chambolle-Pock method (w-CP) (Mokrý and Rajmic 2020), the iteratively reweighted Chambolle-Pock method (re-CP) (Mokrý and Rajmic 2020), and the frame-wise Janssen method (Janssen) (Janssen et al. 1986). All these methods use the half offset configuration, a window size of 2800, and a hop size of 700,

as the same values in Mokrý and Rajmic (2020). For the Janssen method, we set the number of iterations to 20.

The 10 audio recordings are chosen from the Sound Quality Assessment Material (SQAM) dataset (European Broadcasting Union 2008), which contains various types of signals. The information on these test signals are summarized in Table 4.10.

Table 4.10 Information on the ten test audio signals used for comparison in Experiment 4.

Track number	Signal content	Duration (second)
09	Viola melodic phrase	10.59
14	Oboe melodic phrase	10.32
23	Horn melodic phrase	10.74
33	Gong single tone (forte)	10.82
44	Soprano solo	10.69
56	Organ solo	10.43
58	Guitar solo	9.67
60	Piano solo	10.26
65	Orchestra excerpt	10.99
69	Pop music excerpt	10.62

For each test audio signal, we divide it equally into 10 segments, and create a gap of a specified size (from 10 to 500 milliseconds) at a random location in each segment. We exclude the first and last segments from the inpainting process, since they are usually the attack and decay parts of the signal. We also ensure that the reliable neighborhoods needed for the inpainting methods are long enough so that gaps are not presented near the boundaries. If the gap length is larger than 200 milliseconds, we reduce the number of segments to 6 to avoid creating gaps that are too large compared to the segment length.

For each segment with a gap, we reconstruct the signal using our hybrid approach and the four inpainting methods*. For the hybrid approach, we use the crossfade-based technique to reconstruct the noise component. We evaluate the reconstruction quality using three metrics: SNR ($\mathrm{SNR}_{\mathrm{gap}}$), TV-ISD, and ODG. To calculate the TV-ISD, we use a Hann window with a window size of 2048 and a hop size of 512, and estimate the PSDs for each segment using the Burg method with an order of 64. In addition, we record the elapsed time for each inpainting method to produce results.

*The audio excerpts and supplemental figures can be accessed through the webpage: https://etosphere.github.io/hybrid-inpainting-approach-demo/.

For each inpainting method and each gap length, we use the interquartile range (IQR) method to remove outliers from the metrics data. The IQR is calculated by subtracting the 25th percentile (Q_1) from the 75th percentile (Q_3) of the data. Data points that are smaller than $Q_1 - 1.5\text{IQR}$ or larger than $Q_3 + 1.5\text{IQR}$ are considered as outliers and removed from the analysis.

The evaluations of different audio inpainting methods under three metrics are shown in Figure 4.17, 4.18, and 4.19.

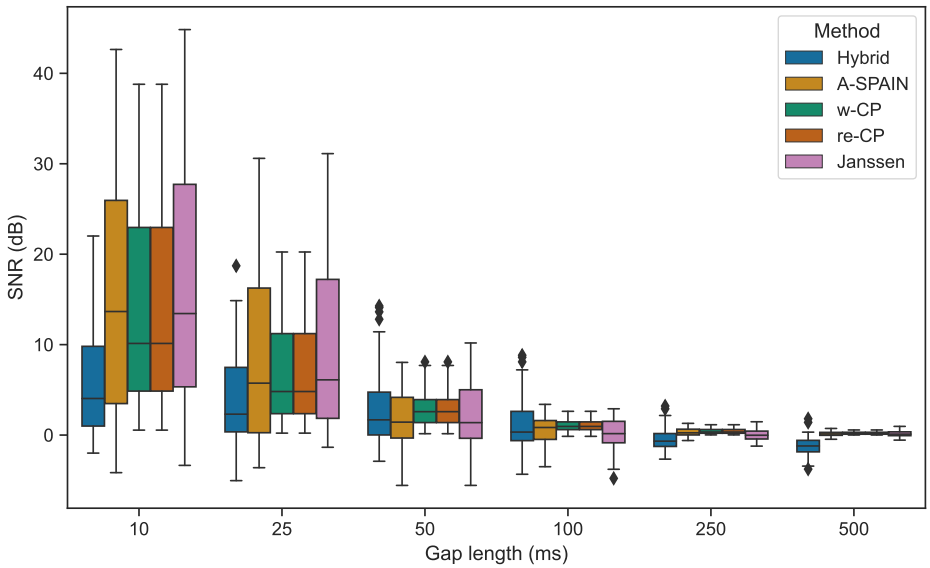


Figure 4.17 Comparison of audio inpainting methods under different gap lengths in terms of SNR.

According to Figure 4.17, when the gap length is lower than 50 ms, the hybrid approach does not have the same good SNR as other methods, and all the methods show the same overall performance when the gap length is 50 ms or longer.

In terms of TV-ISD, the hybrid approach performs more consistently across all gap lengths and beats all other approaches for gaps longer than 100 ms (Figure 4.18). This indicates that the hybrid approach is able to maintain the spectral shape over a wide range of gaps.

Comparing the results of ODG, the hybrid approach has a much better reconstruction quality than other methods when the gap is longer than 50 ms in both evaluation metrics. It is not as good as other models when the gap is lower than 50 ms (Figure 4.19).

The degradation of the reconstruction quality of our hybrid approach may be attributed to the following reasons. First, it is difficult to spread the phase error when synthesizing the partials when gap is short, which leads to a more pronounced discontinuity and lowers the ODG. Moreover, since we use preset parameters to inpaint all types of signals, the partial matching method sometimes

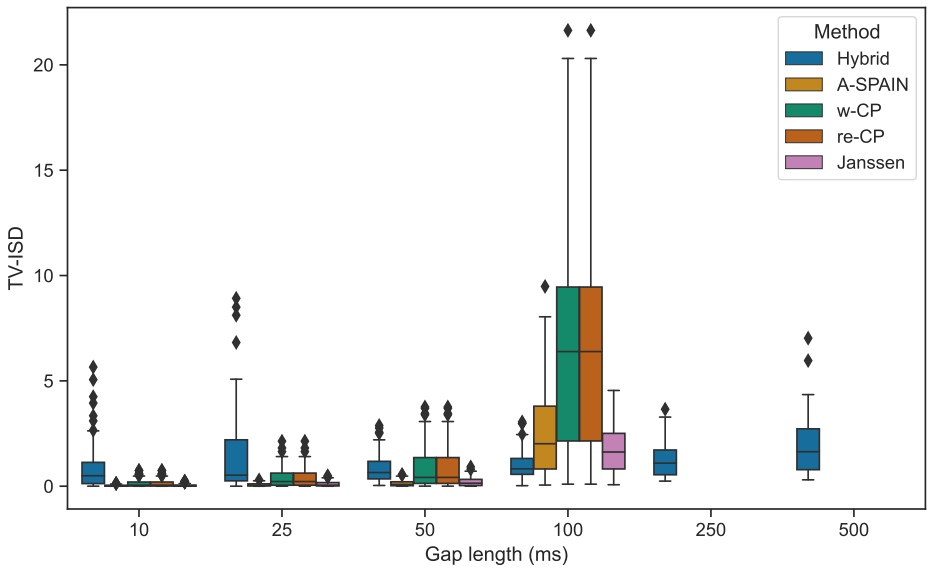


Figure 4.18 Comparison of audio inpainting methods under different gap lengths in terms of TV-ISD (lower values are better). For gaps of lengths 250 and 500 ms, the TV-ISD cannot be calculated (extremely large values of results) for methods except the hybrid approach, and is therefore not shown in the figure.

matches two partials that represent different notes together (parameters such as ζ_f^{match} and ζ_a^{match} are set too aggressive in this scenario). This predicts incorrect partial trajectories, which significantly decrease the reconstruction quality, especially for longer gaps.

We can infer from the above findings that the hybrid approach achieves better results in all three evaluation metrics when the gap is longer than 50 ms, which means that our approach is better at inpainting longer gaps (100–500 ms). Notably, when the gap is longer than 50 ms, the hybrid approach has stable performance, whereas other methods such as Janssen and re-CP start to fail. Even more, the hybrid approach shows less standard variations in longer gaps.

At the same time, the hybrid approach shows better reconstruction quality in more stationary signals, such as the Viola and the Horn signals, as shown in Figure 4.20, and far outperforms other methods for gap lengths greater than 100 ms (Figure 4.21). Furthermore, as for the running time, our hybrid approach, together with the A-SPAIN and the w-CP methods, has a much shorter running time and almost does not increase when the gap length grows, as shown in Figure 4.22.

The main advantage of our hybrid approach is that it is more adaptive and flexible than other approaches for inpainting an audio signal by integrating more controllable parameters. In practice, fine-tuning related parameters depending on the characteristics of an audio signal can result in better reconstruction.

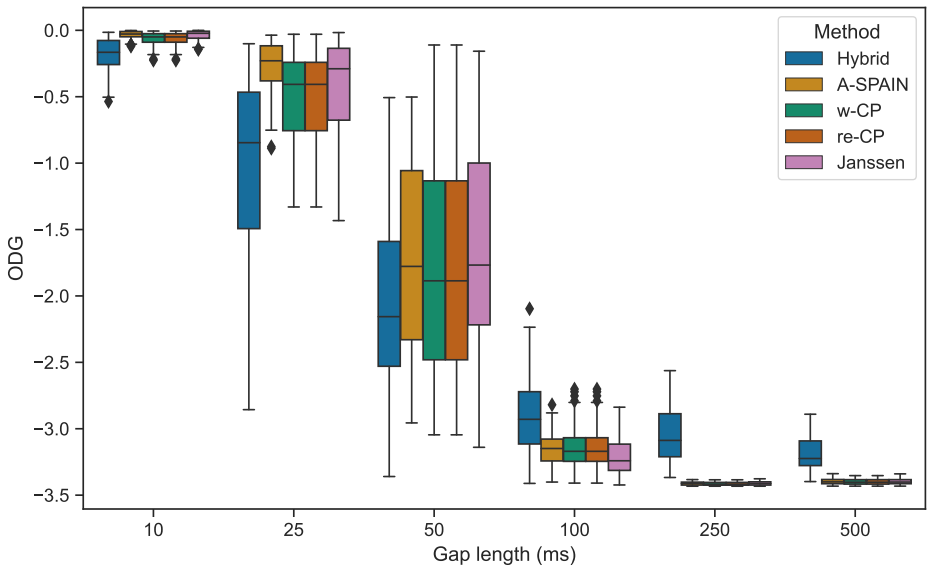


Figure 4.19 Comparison of audio inpainting methods under different gap lengths in terms of ODG (higher values are better).

We illustrate the adaptability and flexibility of our hybrid approach with two examples. The first example uses a synthesized sound with linear chirps and added noise (with noise level at –15 dB), and the second uses a soprano recording with vibrato from the Sound Quality Assessment Material (SQAM) dataset (European Broadcasting Union 2008).

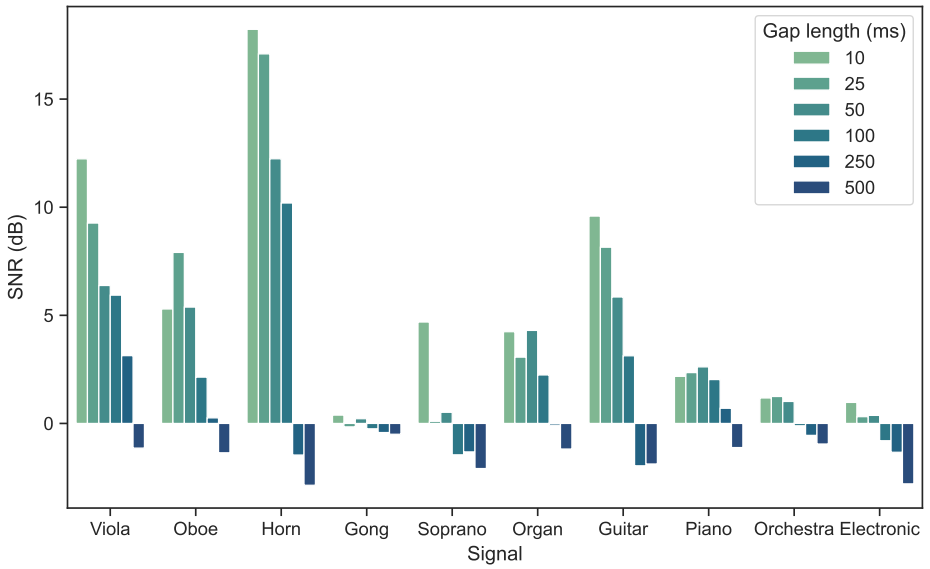


Figure 4.20 Reconstruction SNR of the hybrid approach for multiple signals under different gap lengths.

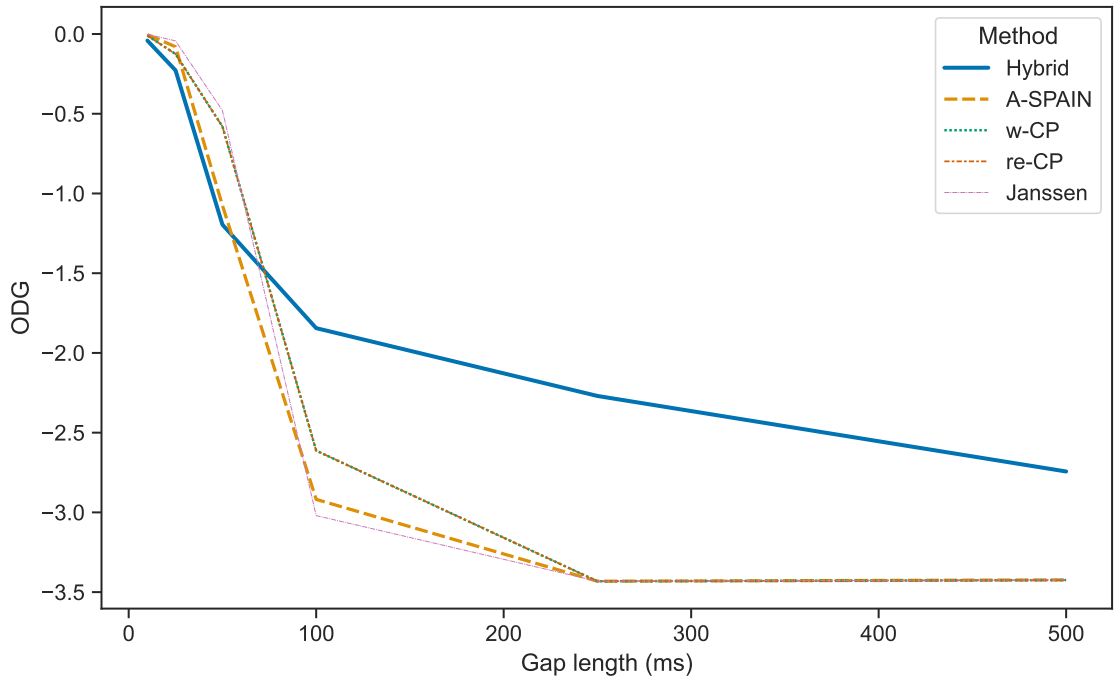


Figure 4.21 Comparison of audio inpainting methods for the Horn signal under different gap lengths in terms of ODG.

For both signals, we fine-tune some parameters in the partial tracking and matching methods. We set the window size of partial tracking Ψ_{partial} to 1024 samples to better capture the time-varying nature of these two signals. We set smaller values for the assignment thresholds for amplitude ($\zeta_a = 12$) and frequency ($\zeta_f = 12$) changes for the partial tracking method to reduce the misconnection of two independent partials. We also set smaller values for the assignment thresholds for amplitude ($\zeta_a^{\text{match}} = 0.01$) and frequency ($\zeta_a^{\text{match}} = 9$) changes for the partial matching method to reduce the mismatch of two independent trajectories. The reconstruction results for the two test signals are presented in Figures 4.23 and 4.24, respectively.

As for the synthesized signal with chirps and noise, compared to the original TF representation (Figure 4.23a), w-CP and re-CP fail to inpaint the tonal (chirp) part, and the re-CP method discards the noise in the gap. At the same time, A-SPAIN and Janssen cannot adapt to the variations of frequencies in the gap due to their stationary assumptions in the gap, leading to frequency jumps and the "freezing" of noise. However, our hybrid approach successfully captured the features from the reliable neighborhoods and accurately predicted both tonal and noise components for this signal.

As for the soprano signal with vibrato, compared to the original TF representation (Figure 4.24a), w-CP and re-CP failed to inpaint the tonal component with modulation. Meanwhile, A-SPAIN and Janssen fail to connect the correct partials. The hybrid approach shows the most similar trajectories

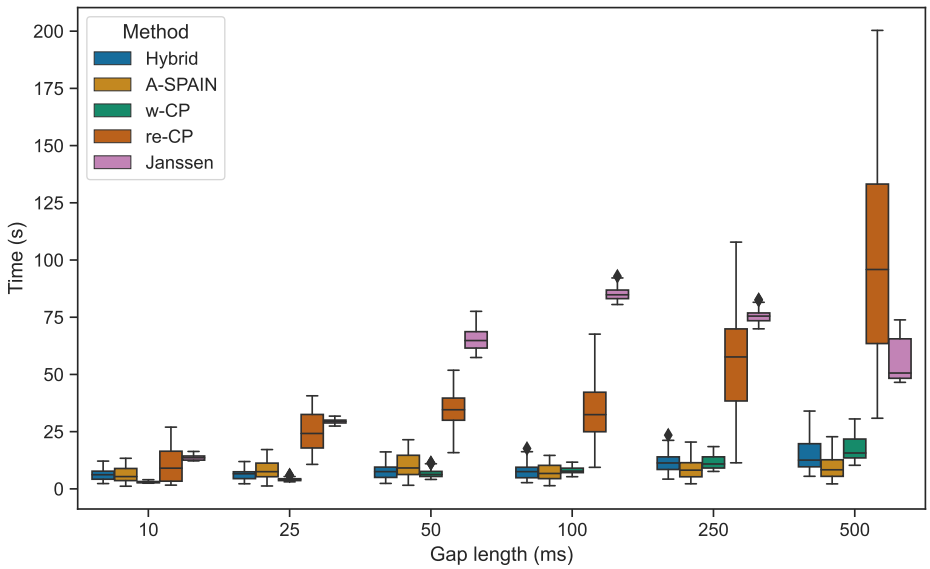
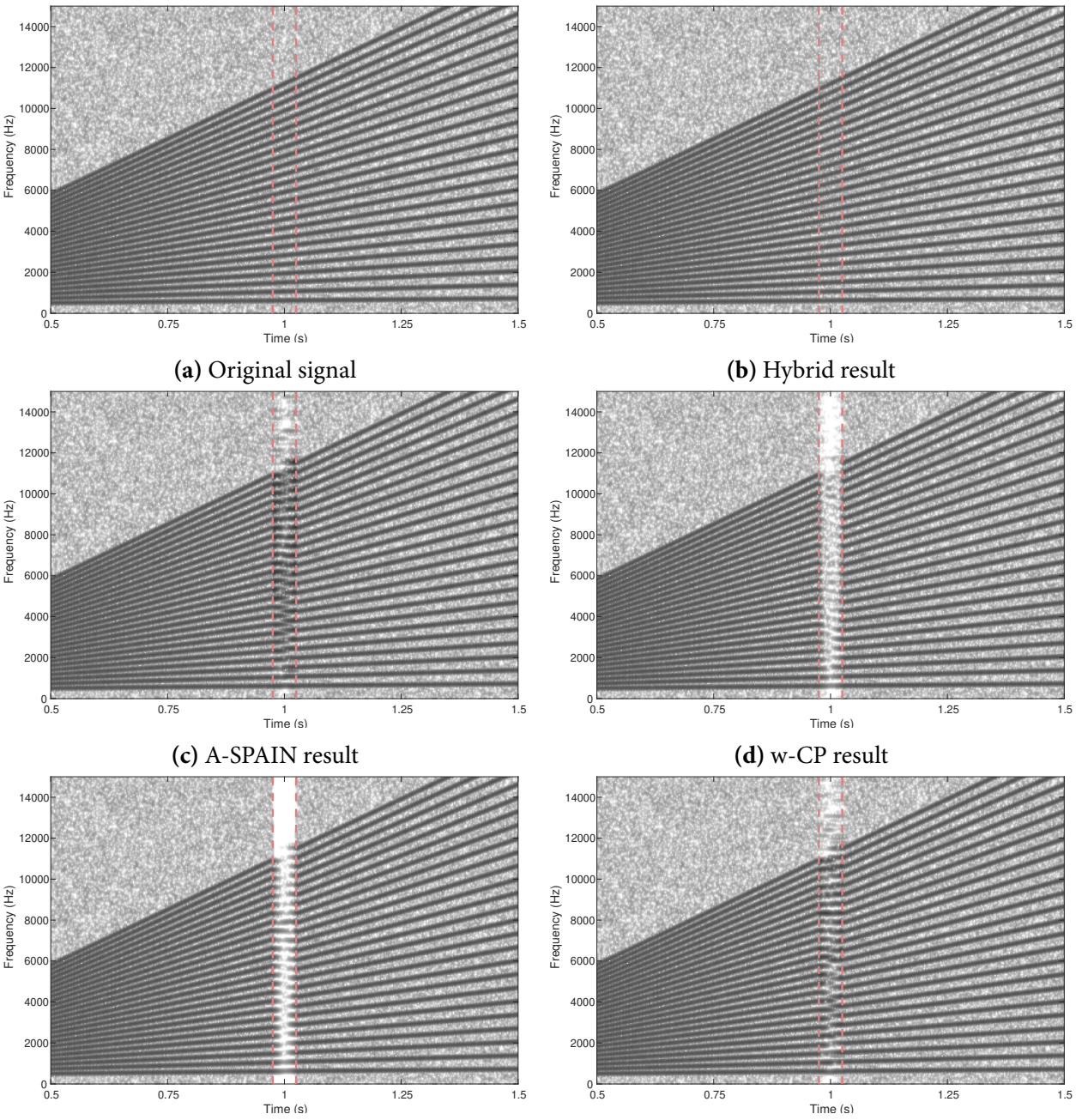


Figure 4.22 Comparison of audio inpainting methods under different gap lengths in terms of running time.

as the original audio signal with the correctly inpainted partials with modulation and captures some of the noise. These two results also confirm the "frequency jump" problem and the "freezing/discard of noise" problem of the other inpainting approaches.

In conclusion, the hybrid approach is more general and adaptive than other methods with various lengths of gaps, especially for longer gaps and stationary signals, and it can also be fine-tuned to get better results based on the characteristics of the signal in practical usage.



(a) Original signal

(b) Hybrid result

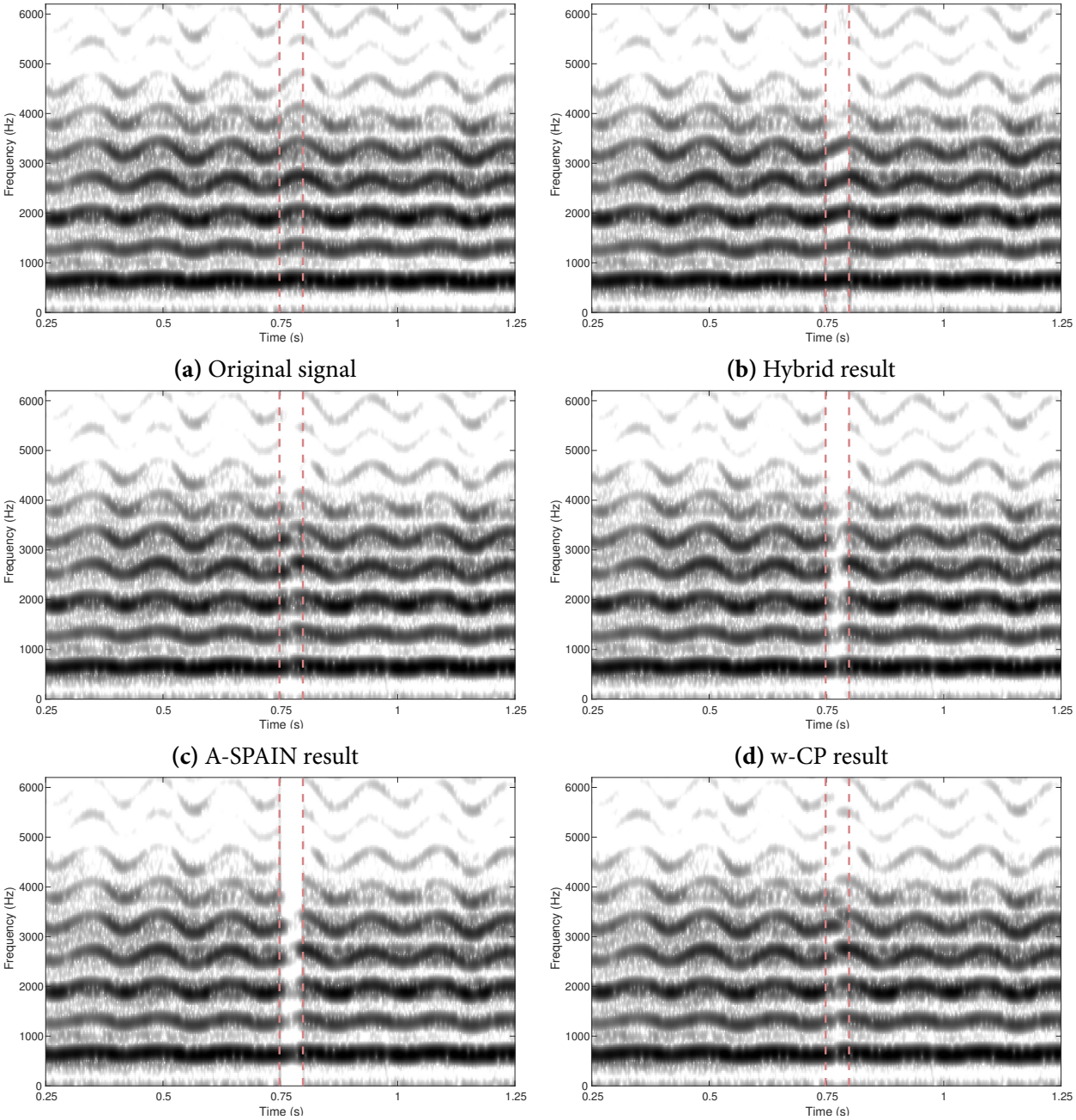
(c) A-SPAIN result

(d) w-CP result

(e) re-CP result

(f) Janssen result

Figure 4.23 Time-frequency domain representations of the reconstructions of various inpainting methods for the synthesized chirps with added noise. The length of the gap is 50 milliseconds.



(a) Original signal

(b) Hybrid result

(c) A-SPAIN result

(d) w-CP result

(e) re-CP result

(f) Janssen result

Figure 4.24 Time-frequency domain representations of the reconstructions of various inpainting methods for the soprano recording with vibrato. The length of the gap is 50 milliseconds.

5

Conclusion

5.1 Summary

This thesis proposes a novel hybrid audio inpainting approach that takes into account the diversity of audio signals. This approach solves the inpainting problem in a structured way as it decomposes the signal into tonal, transient, and noise components and reconstructs them separately using their own reliable neighborhoods. Some state-of-the-art methods are adopted and refined in the present study, such as the structural sparse decomposition, partial tracking, and extrapolation algorithms, to provide more robust inpainting results.

Comparing the inpainting results of A-SPAIN, w-CP, re-CP, and Janssen methods across the gap from 10 ms to 500 ms in terms of SNR, TV-ISD, and ODG, we found that the hybrid approach is more general and adaptive than other methods with various lengths of gaps, especially for longer gaps (longer than 50 ms) and stationary signals. Furthermore, our hybrid approach remains fast and scarcely increase the running time as the gap length grows.

5.2 Future Work

There are still some limitations and challenges that need to be addressed in future work. Some possible directions are as follows:

Recovering the transients in the gap. It is crucial for reconstructing the transient component to estimate the onset positions of the transients in the gap. A possible solution is to incorporate some beat tracking models, such as Ellis (2007), to induce the potential locations for transients based on the rhythmic structure of the signal. Another possibility is to exploit the fact that transients are typically found at the start of a sustained sound, and divide the "sustained" components in the

gap into several damped sine waves with transients. The ramped exponentially damped sinusoidal (REDS) atoms, proposed by Neri and Depalle (2017), can efficiently represent this type of signal and can be potentially used for reconstructing the transients.

Improving the robustness of the hybrid approach. Another challenge of our approach is to deal with factors such as noise that can interfere with the estimation of the parameters of a partial, which, together with the fact that partial tracking and reconnection methods often fall into a local optimum, leads to unstable experimental results. Despite proposing many strategies and using parameters to constrain the partial processing methods, the results are still undesirable for some cases. Therefore, more advanced partial tracking and prediction methods need to be proposed to address this issue.

Using supplementary materials to guide the inpainting process. A possible way to enhance our approach is to use some additional information or context from other modalities or sources to inpaint the audio. For instance, Zhou et al. (2019) proposed a vision-infused deep audio inpainting method that utilizes the modality context in the accompanying video to inpaint the audio. Similarly, it may be also possible to use musical scores to provide information (such as note and rhythm) to help guide the inpainting process for long gaps.

Extending the hybrid approach for reconstructing other degradations. The recovery of various degradations of audio signals is a more general problem than audio inpainting in the context of our thesis. The audio signals can be not only degraded in time domain, such as missing samples (gap) or limited amplitude (clipping), but also degraded in time-frequency domain, including data compression (quantization) and high-frequency removal (bandlimiting) (Mokrý et al. 2020; Jax and Vary 2002). The three-layer structured audio processing approach can be applied to recover other degradations.

Bibliography

Adler, Amir, Valentin Emiya, Maria G. Jafari, Michael Elad, Rémi Gribonval, and Mark D. Plumbley. 2012. "Audio Inpainting." *IEEE Transactions on Audio, Speech, and Language Processing* 20 (3): 922–932. https://doi.org/10.1109/TASL.2011.2168211.

Balazs, Peter, Monika Doerfler, Matthieu Kowalski, and Bruno Torresani. 2013. "Adapted and Adaptive Linear Time-Frequency Representations: A Synthesis Point of View." *IEEE Signal Processing Magazine* 30 (6): 20–31. https://doi.org/10.1109/MSP.2013.2266075.

Beck, Amir, and Marc Teboulle. 2009. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems." *SIAM Journal on Imaging Sciences* 2 (1): 183–202. https://doi.org/10.1137/080716542.

Betser, Michaël. 2009. "Sinusoidal Polynomial Parameter Estimation Using the Distribution Derivative." *IEEE Transactions on Signal Processing* 57 (12): 4633–4645. https://doi.org/10.1109/TSP.2009.2027401.

Bountourakis, Vasileios, Lazaros Vrysis, Konstantinos Konstantoudakis, and Nikolaos Vryzas. 2019. "An Enhanced Temporal Feature Integration Method for Environmental Sound Recognition." *Acoustics* 1 (2): 410–422. https://doi.org/10.3390/acoustics1020023.

Bountourakis, Vasileios, Lazaros Vrysis, and George Papanikolaou. 2015. "Machine Learning Algorithms for Environmental Sound Recognition: Towards Soundscape Semantics." In *Proceedings of the Audio Mostly 2015 on Interaction With Sound,* 1–7. Thessaloniki, Greece: ACM, 2015. https://doi.org/10.1145/2814895.2814905.

Caetano, Marcelo, and Xavier Rodet. 2012. "A Source-Filter Model for Musical Instrument Sound Transformation." In *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 137–140. Kyoto, Japan: IEEE. https://doi.org/10.1109/ICASSP.2012.6287836.

Chen, Scott Shaobing, David L. Donoho, and Michael A. Saunders. 2001. "Atomic Decomposition by Basis Pursuit." *SIAM Review* 43 (1): 129–159. https://doi.org/10.1137/S003614450037906X.

Daudet, Laurent, and Bruno Torrésani. 2002. "Hybrid Representations for Audiophonic Signal Encoding." *Signal Processing* 82 (11): 1595–1617. https://doi.org/10.1016/S0165-1684(02)00304-3.

Depalle, Philippe, Guillermo Garcia, and Xavier Rodet. 1993. "Tracking of Partials for Additive Sound Synthesis Using Hidden Markov Models." In *Proceedings of the 1993 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP),* 225–228 vol.1. Minneapolis, MN, USA: IEEE. https://doi.org/10.1109/ICASSP.1993.319096.

Dörfler, Monika. 2001. "Time-Frequency Analysis for Music Signals: A Mathematical Approach." *Journal of New Music Research* 30 (1): 3–12. https://doi.org/10.1076/jnmr.30.1.3.7124.

Ellis, Daniel P. W. 2007. "Beat Tracking by Dynamic Programming." *Journal of New Music Research* 36 (1): 51–60. https://doi.org/10.1080/09298210701653344.

Esquef, Paulo A A, Vesa Välimäki, Kari Roth, and Ismo Kauppinen. 2003. "Interpolation of Long Gaps in Audio Signals Using the Warpped Burg's Method." In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx).* London, UK.

Etter, Walter. 1996. "Restoration of a Discrete-Time Signal Segment by Interpolation Based on the Left-Sided and Right-Sided Autoregressive Parameters." *IEEE Transactions on Signal Processing* 44 (5): 1124–1135. https://doi.org/10.1109/78.502326.

European Broadcasting Union. 2008. "Sound Quality Assessment Material Recordings for Subjective Tests," 2008. Accessed April 3, 2023. https://tech.ebu.ch/publications/sqamcd.

Fant, Gunnar. 1971. *Acoustic Theory of Speech Production: With Calculations Based on X-Ray Studies of Russian Articulations.* Berlin, Boston: De Gruyter Mouton. https://doi.org/doi:10.1515/9783110873429.

Godsill, Simon J., and Peter J. W. Rayner. 1998. *Digital Audio Restoration.* London: Springer London. https://doi.org/10.1007/978-1-4471-1561-8.

Hamilton, Brian, and Philippe Depalle. 2012. "Comparisons of Parameter Estimation Methods for an Exponential Polynomial Sound Signal Model." In *Proceedings of the AES 45th International Conference.* Helsinki, Finland.

Hayes, M. H. 1996. *Statistical Digital Signal Processing and Modeling.* New York, NY, USA: John Wiley & Sons.

Hildebrand, Francis Begnaud. 1956. *Introduction to Numerical Analysis.* Second Edition. International Series in Pure and Applied Mathematics. New York, NY, USA: McGraw-Hill.

Huber, Rainer, and Birger Kollmeier. 2006. "PEMO-Q: A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception." *IEEE Transactions on Audio, Speech and Language Processing* 14 (6): 1902–1911. https://doi.org/10.1109/TASL.2006.883259.

Itakura, Fumitada, and Shuzo Saito. 1968. "Analysis Synthesis Telephony Based on the Maximum Likelihood Method." In *Proceedings of the 6th International Congress on Acoustics,* C-17–20. Tokyo, Japan.

ITU-R. 1998. *Method for Objective Measurements of Perceived Audio Quality.* Recommendation BS.1387-0. Geneva, Switzerland.

Janssen, Augustus, Raymond Veldhuis, and Lodewijk Vries. 1986. "Adaptive Interpolation of Discrete-Time Signals That Can Be Modeled as Autoregressive Processes." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34 (2): 317–330. https://doi.org/10.1109/TASSP.1986.1164824.

Jax, Peter, and Peter Vary. 2002. "An Upper Bound on the Quality of Artificial Bandwidth Extension of Narrowband Speech Signals." In *Proceedings of the 2002 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP),* I-237–240. Orlando, FL, USA. https://doi.org/10.1109/ICASSP.2002.5743698.

Kay, Steven M. 1988. *Modern Spectral Estimation: Theory and Application.* Prentice-Hall Signal Processing Series. Englewood Cliffs, NJ: Prentice Hall.

Kay, Steven M., and Stanley Lawrence Marple. 1981. "Spectrum Analysis—A Modern Perspective." *Proceedings of the IEEE* 69 (11): 1380–1419. https://doi.org/10.1109/PROC.1981.12184.

Kereliuk, Corey. 2013. "Sparse and Structured Atomic Modelling of Audio." PhD diss., McGill University.

Kereliuk, Corey, and Philippe Depalle. 2011. "Sparse Atomic Modeling of Audio: A Review." In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx),* 81–92. Paris, France.

Kitić, Srđan, Nancy Bertin, and Rémi Gribonval. 2015. "Sparsity and Cosparsity for Audio Declipping: A Flexible Non-Convex Approach." In *Proceedings of the 12th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA),* 243–250. Liberec, Czech Republic.

Klapuri, Anssi, and Manuel Davy, eds. 2006. *Signal Processing Methods for Music Transcription.* Boston, MA: Springer US. https://doi.org/10.1007/0-387-32845-9.

Kowalski, Matthieu, Kai Siedenburg, and Monika Dörfler. 2013. "Social Sparsity! Neighborhood Systems Enrich Structured Shrinkage Operators." *IEEE Transactions on Signal Processing* 61 (10): 2498–2511. https://doi.org/10.1109/TSP.2013.2250967.

Kowalski, Matthieu, and Bruno Torrésani. 2009. "Sparsity and Persistence: Mixed Norms Provide Simple Signal Models with Dependent Coefficients." *Signal, Image and Video Processing* 3 (3): 251–264. https://doi.org/10.1007/s11760-008-0076-1.

Kuhn, Harold W. 1955. "The Hungarian Method for the Assignment Problem." *Naval Research Logistics Quarterly* 2:83–97. https://doi.org/10.1002/nav.3800020109.

Lagrange, Mathieu, Sylvain Marchand, and Jean-bernard Rault. 2005. "Long Interpolation of Audio Signals Using Linear Prediction in Sinusoidal Modeling." *Journal of the Audio Engineering Society* 53 (10): 891–905.

Lukin, Alexey, and Jeremy Todd. 2008. "Parametric Interpolation of Gaps in Audio Signals." In *Proceedings of the Audio Engineering Society 125th Convention,* 891–905. San Francisco, CA.

Maher, Robert C. 1993. "A Method for Extrapolation of Missing Digital Audio Data." In *Proceedings of the Audio Engineering Society 95th Convention.* New York, NY, USA.

Makhoul, John. 1975. "Linear Prediction: A Tutorial Review." *Proceedings of the IEEE* 63 (4): 561–580. https://doi.org/10.1109/PROC.1975.9792.

Mallat, Stephane G., and Zhifeng Zhang. 1993. "Matching Pursuits with Time-Frequency Dictionaries." *IEEE Transactions on Signal Processing* 41 (12): 3397–3415. https://doi.org/10.1109/78.258082.

Marafioti, Andrés, Nicki Holighaus, Piotr Majdak, and Nathanaël Perraudin. 2019. "Audio Inpainting of Music by Means of Neural Networks." In *Proceedings of the Audio Engineering Society 146th Convention.* Dublin, Ireland.

Marafioti, Andrés, Piotr Majdak, Nicki Holighaus, and Nathanaël Perraudin. 2021. "GACELA: A Generative Adversarial Context Encoder for Long Audio Inpainting of Music." *IEEE Journal of Selected Topics in Signal Processing* 15 (1): 120–131. https://doi.org/10.1109/JSTSP.2020.3037506.

McAulay, Robert J., and Thomas F. Quatieri. 1986. "Speech Analysis/Synthesis Based on a Sinusoidal Representation." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34 (4): 744–754. https://doi.org/10.1109/TASSP.1986.1164910.

Mokrý, Ondřej, Paul Magron, Thomas Oberlin, and Cédric Févotte. 2023. "Algorithms for Audio Inpainting Based on Probabilistic Nonnegative Matrix Factorization." *Signal Processing* 206:108905. https://doi.org/10.1016/j.sigpro.2022.108905.

Mokrý, Ondřej, and Pavel Rajmic. 2020. "Audio Inpainting: Revisited and Reweighted." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28:2906–2918. https://doi.org/10.1109/TASLP.2020.3030486.

Mokrý, Ondřej, Pavel Rajmic, and Pavel Záviška. 2020. "Flexible Framework for Audio Reconstruction." In *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx),* 117–124. Vienna, Austria.

Mokrý, Ondřej, Pavel Záviška, Pavel Rajmic, and Vítězslav Veselý. 2019. "Introducing SPAIN (SParse Audio INpainter)." In *Proceedings of the 27th European Signal Processing Conference (EUSIPCO),* 1–5. A Coruña, Spain. https://doi.org/10.23919/EUSIPCO.2019.8902560.

Moliner, Eloi, and Vesa Välimäki. 2023. "Diffusion-Based Audio Inpainting," https://doi.org/10.48550/ARXIV.2305.15266.

Nam, Sangnam, Mark E. Davies, Michael Elad, and Rémi Gribonval. 2013. "The Cosparse Analysis Model and Algorithms." *Applied and Computational Harmonic Analysis* 34 (1): 30–56. https://doi.org/10.1016/j.acha.2012.03.006.

Natarajan, Balas K. 1995. "Sparse Approximate Solutions to Linear Systems." *SIAM Journal on Computing* 24 (2): 227–234. https://doi.org/10.1137/S0097539792240406.

Neri, Julian, and Philippe Depalle. 2017. "REDS: A New Asymmetric Atom for Sparse Audio Decomposition and Sound Synthesis." In *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx),* 268–275. Edinburgh, Scotland.

———. 2018. "Fast Partial Tracking of Audio with Real-Time Capability Through Linear Programming." In *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx),* 326–333. Aveiro, Portugal.

Nuttall, A. 1981. "Some Windows with Very Good Sidelobe Behavior." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29 (1): 84–91. https://doi.org/10.1109/TASSP.1981.1163506.

O'Shaughnessy, Douglas. 2000. *Speech Communications: Human and Machine.* IEEE. https://doi.org/10.1109/9780470546475.

Perraudin, Nathanael, Nicki Holighaus, Piotr Majdak, and Peter Balazs. 2018. "Inpainting of Long Audio Segments with Similarity Graphs." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26 (6): 1083–1094. https://doi.org/10.1109/TASLP.2018.2809864.

Pisarenko, Vladilen Fedorovich. 1973. "The Retrieval of Harmonics from a Covariance Function." *Geophysical Journal International* 33 (3): 347–366. https://doi.org/10.1111/j.1365-246X.1973.tb03424.x.

Pitas, Ioannis, and Anastasios N. Venetsanopoulos. 1992. "Order Statistics in Digital Image Processing." *Proceedings of the IEEE* 80 (12): 1893–1921. https://doi.org/10.1109/5.192071.

Rajmic, Pavel, Hana Bartlová, Zdeněk Průša, and Nicki Holighaus. 2015. "Acceleration of Audio Inpainting by Support Restriction." In *Proceedings of the 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT),* 325–329. Brno, Czech Republic: IEEE. https://doi.org/10.1109/ICUMT.2015.7382451.

Roy, Richard, and Thomas Kailath. 1989. "ESPRIT: Estimation of Signal Parameters via Rotational Invariance Techniques." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37 (7): 984–995. https://doi.org/10.1109/29.32276.

Siedenburg, Kai, and Monika Dörfler. 2011. "Structured Sparsity for Audio Signals." In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx),* 23–26. Paris, France.

Siedenburg, Kai, Matthieu Kowalski, and Monika Dörfler. 2014. "Audio Declipping with Social Sparsity." In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 1577–1581. Florence, Italy. https://doi.org/10.1109/ICASSP.2014.6853863.

Smith, Julius O., and Xavier Serra. 1987. "PARSHL: An Analysis/Synthesis Program for Non-Harmonic Sounds Based on a Sinusoidal Representation." In *Proceedings of the 1987 International Computer Music Conference (ICMC),* 290–297. IL, USA: Michigan Publishing.

Tantibundhit, Charturong, J. Robert Boston, Ching-Chung Li, John D. Durrant, Susan Shaiman, Kristie Kovacyk, and Amro El-Jaroudi. 2006. "Speech Enhancement Using Transient Speech Components." In *Proceedings of the 2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings (ICASSP),* 1:833–836. Toulouse, France: IEEE. https://doi.org/10.1109/ICASSP.2006.1660150.

Taubock, Georg, Shristi Rajbamshi, and Peter Balazs. 2021. "Dictionary Learning for Sparse Audio Inpainting." *IEEE Journal of Selected Topics in Signal Processing* 15 (1): 104–119. https://doi.org/10.1109/JSTSP.2020.3046422.

Thiede, Thilo, William C. Treurniet, Roland Bitto, Christian Schmidmer, Thomas Sporer, John G. Beerends, and Catherine Colomes. 2000. "PEAQ - the ITU Standard for Objective Measurement of Perceived Audio Quality." *Journal of the Audio Engineering Society* 48 (1/2): 3–29.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection Via the Lasso." *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1): 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

Verma, Tony S., and Teresa H. Y. Meng. 2000. "Extending Spectral Modeling Synthesis with Transient Modeling Synthesis." *Computer Music Journal* 24 (2): 47–59. https://doi.org/10.1162/014892600559317.

Záviška, Pavel, and Pavel Rajmic. 2022. "Audio Declipping with (Weighted) Analysis Social Sparsity." In *Proceedings of the 45th International Conference on Telecommunications and Signal Processing (TSP),* 407–412. Prague, Czech Republic, 2022. https://doi.org/10.1109/TSP55681.2022.9851269.

Zhou, Hang, Ziwei Liu, Xudong Xu, Ping Luo, and Xiaogang Wang. 2019. "Vision-Infused Deep Audio Inpainting." In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV),* 283–292. Seoul, Korea (South): IEEE. https://doi.org/10.1109/ICCV.2019.00037.