

Dissertation

2026

Simon Johannes Schwär

Differentiable Models for Music Signal
Reconstruction, Adaptation, and Synthesis



Differentiable Models for Music Signal Reconstruction, Adaptation, and Synthesis

**Differenzierbare Modelle für die
Rekonstruktion, Adaptierung und Synthese
von Musiksignalen**

Dissertation

Der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

zur Erlangung des Doktorgrades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt von
Simon Johannes Schwär
aus Moers

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung: 08.06.2026

Gutachter: Prof. Dr. rer. nat. Meinard Müller
Prof. Dr.-Ing. Sebastian Stober

Statement on AI-Assisted Editing

The author used ChatGPT (<https://chatgpt.com>), a large language model, to improve spelling, grammar, clarity, and readability during the preparation of this dissertation. The tool was applied exclusively for language editing and was not used to generate, outline, or summarize any portion of the content. The author carefully reviewed and edited the final version of the text and takes full responsibility for the content of this dissertation.

Abstract

Nuanced variations in pitch, loudness, and timbre are key components of musical expression and the perceived naturalness of musical sounds. Recent advances in deep learning have enabled powerful data-driven methods for manipulating these attributes in existing recordings and for synthesizing novel sounds. However, many deep learning approaches remain limited in terms of controllability of the output, their reliance on high-quality training data, and computational cost. In this thesis, we explore differentiable models as one way of addressing these limitations. By explicitly integrating physical, perceptual, or signal-based structure into deep learning pipelines or loss functions, these models can provide gradients with respect to musically interpretable parameters. At the same time, they allow for incorporating inductive biases grounded in audio signal processing, Music Information Retrieval, or psychoacoustics, and support computationally efficient implementations.

The contributions of this thesis are organized around three high-level tasks: reconstruction, adaptation, and synthesis of monophonic music signals. First, reconstruction addresses the problem of recovering corrupted or missing tone attributes. In this context, we introduce a differentiable model inspired by the vocal tract that enables the reconstruction of natural timbre characteristics from low-quality singing voice signals recorded with larynx microphones. The vocal tract model supports learning from limited training data while reliably preserving the original pitch. Second, adaptation focuses on modifying individual tone attributes without affecting others. Within representative case studies, we propose differentiable cost functions for intonation and timbre adaptation that integrate perceptual and musical models. The formulation enables gradient-based optimization that accounts for interactions between voices in multi-track recordings. Finally, synthesis refers to the task of creating an audio signal from a disentangled parametric representation of pitch, loudness, and timbre. Using wind instrument resynthesis as an application scenario, we introduce a physically inspired instrument model based on differentiable pulsetable synthesis that allows for controllable and computationally efficient signal generation.

Across these tasks, we analyze strengths and limitations of differentiable models, with particular emphasis on the interplay between model structure and loss function. Complementing the application-specific contributions, we examine how loss function behavior with respect to individual tone attributes affects the convergence of deep learning systems with and without model-based components. In particular, we investigate the widely used multi-scale spectral loss and show that a favorable configuration leads to a smoother loss function with respect to pitch-related parameters. Together, these insights inform the practical design of model-based deep learning systems and contribute to more controllable, data-efficient, and computationally lightweight methods for music processing and related domains.

Zusammenfassung

Nuancierte Variationen in Tonhöhe, Lautheit und Klangfarbe sind zentral für musikalischen Ausdruck und die Wahrnehmung eines natürlichen Klangs. Fortschritte im Deep Learning ermöglichen leistungsfähige datengetriebene Methoden zur Manipulation dieser Klangattribute in bestehenden Aufnahmen und zur Synthese neuer Klänge. Viele dieser Verfahren sind jedoch nur begrenzt kontrollierbar, benötigen große Mengen hoch-qualitativer Daten und sind rechenintensiv. In dieser Dissertation untersuchen wir differenzierbare Modelle als möglichen Ansatz, diese Einschränkungen zu überwinden. Durch die gezielte Integration physikalischer, perzeptiver oder signalbasierter Strukturen in Deep-Learning-Pipelines oder Kostenfunktionen können diese Modelle Gradienten in Bezug auf musikalisch interpretierbare Parameter bereitstellen. Gleichzeitig ermöglichen sie die Definition induktiver Verzerrungen auf Basis von Methoden aus der Audiosignalverarbeitung, dem Music Information Retrieval oder der Psychoakustik, und unterstützen recheneffiziente Implementierungen.

Die Beiträge dieser Arbeit gliedern sich in drei Aufgabenbereiche: die Rekonstruktion, die Adaptierung und die Synthese von monophonen Musiksignalen. Rekonstruktion bezeichnet die Wiederherstellung beschädigter oder fehlender Klangattribute. Hierzu stellen wir ein differenzierbares Modell vor, das vom Vokaltrakt inspiriert ist und die Rekonstruktion natürlicher Klangfarbenmerkmale aus Gesangssignalen von minderer Qualität, die mit Kehlkopfmikrofonen aufgenommen wurden, ermöglicht. Adaptierung ist die gezielte Veränderung einzelner Klangattribute. Als repräsentative Fallstudien stellen wir differenzierbare Kostenfunktionen für die Intonations- und Klangfarbenadaptierung auf Basis von perzeptiven und musikalischen Modellen vor, die eine gradientenbasierte Optimierung erlauben und Interaktionen zwischen Stimmen in Mehrspuraufnahmen berücksichtigen. Synthese bezeichnet die Aufgabe, aus einer parametrischen Darstellung der Klangattribute ein Audiosignal zu erzeugen. Am Beispiel von Blasinstrumenten zeigen wir, dass ein physikalisch inspiriertes Instrumentenmodell auf Basis einer differenzierbaren Pulsetable-Synthese eine kontrollierbare und effiziente Signalerzeugung ermöglicht.

In diesem Kontext analysieren wir Stärken und Grenzen differenzierbarer Modelle, mit Fokus auf das Zusammenspiel von Modellstruktur und Kostenfunktion. Über die Anwendungsszenarien hinaus untersuchen wir, wie das Verhalten der Kostenfunktion im Bezug auf einzelne Klangeigenschaften die Konvergenz von Deep-Learning-Systemen beeinflusst. Am weit verbreiteten Multi-Scale-Spectral-Loss zeigen wir, dass geeignete Konfigurationen der zugrunde liegenden Spektralanalyse zu einer glatteren Kostenfunktion in Bezug auf tonhöhenbezogene Parameter führen. Zusammen liefern diese Erkenntnisse wertvolle Hinweise für die praktische Gestaltung modellbasierter Deep-Learning-Systeme und tragen zur Entwicklung kontrollierbarer, datenökonomischer und effizienter Methoden für die Musiksignalverarbeitung und verwandte Anwendungsbereiche bei.

Contents

Abstract	v
Zusammenfassung	vii
1 Introduction	1
1.1 Structure and Main Contributions	4
1.2 Underlying Publications	5
1.3 Additional Publications	7
1.4 Acknowledgments	8
2 A Primer on Music Signals	11
2.1 Audio Signals in Time and Frequency Domain	11
2.2 Of Tones and Notes	13
2.2.1 Timing	14
2.2.2 Pitch and Fundamental Frequency	15
2.2.3 Loudness, Dynamics, Amplitudes, and Envelopes	18
2.2.4 Timbre and Harmonic Energy Distribution	19
2.3 The Singing Voice	22
2.4 Music Signal Synthesis	25
2.4.1 Additive Synthesis	25
2.4.2 Source–Filter Models	28
2.4.3 Deep Learning	31
3 Multi-Scale Spectral Loss Revisited	35
3.1 Background	36
3.2 Generalized Multi-Scale Spectral Loss	37
3.3 Sinusoid Frequency Estimation with Spectral Losses	38
3.3.1 Spectral Leakage	39
3.3.2 Loss Configuration Parameters	41
3.4 Experiments	42
3.4.1 Representative MSS Loss Configurations	43

Contents

3.4.2	Extending the Target Signal	44
3.4.3	DDSP Autoencoder	46
3.4.4	Variational Autoencoder	50
3.5	Further Notes	53
4	Singing Voice Reconstruction from Larynx Microphone Recordings	55
4.1	Background	56
4.2	Singing Voice Reconstruction (SVR)	58
4.3	Larynx Microphones (LMs)	58
4.3.1	Compared LM Models	59
4.3.2	Signal Characterization	61
4.4	LM Singer-Songwriter Dataset	64
4.4.1	Recording Setup and Procedure	65
4.4.2	Dataset Content and Structure	66
4.5	Baseline System for LM-SVR	67
4.5.1	Related Work	67
4.5.2	Model Architecture	68
4.5.3	Implementation Details	69
4.6	Experiments	70
4.6.1	Evaluation	70
4.6.2	Downstream Task: Lyrics Transcription	72
4.7	Further Notes	73
5	Intonation Adaptation in Multi-Track Recordings via Gradient Descent	75
5.1	Background	76
5.1.1	Related Work	78
5.2	Intonation Cost Measures	80
5.2.1	Signal Representations	81
5.2.2	Tonal Cost	81
5.2.3	Harmonic Cost	83
5.3	Intonation Adaptation Algorithm	85
5.3.1	Number of Iterations	87
5.3.2	Initial Value	88
5.3.3	Step Size and Momentum	88
5.3.4	Multi-Voice Adaptation	89
5.4	Experiments	90
5.4.1	Datasets	91
5.4.2	12-TET and JI Adaptation	91
5.4.3	Intonation Drift Compensation	92

5.4.4	Global and Local Adaptation	93
5.4.5	Listening Test	94
5.5	Further Notes	96
6	Towards Timbre Adaptation with Sensory Dissonance	99
6.1	Background: Sensory Dissonance	100
6.1.1	Relative Sensory Dissonance	102
6.2	Sensory Dissonance in Multi-Track Music Recordings	104
6.2.1	Tonal Components in Recordings	106
6.2.2	Case Study: ChoraleBricks	107
6.2.3	Discussion	112
6.3	Timbre Adaptation in Additive Synthesis	112
6.3.1	Method	113
6.3.2	Experiments	114
6.3.3	Discussion	117
6.4	Further Notes	118
7	Wind Instrument Synthesis with Differentiable Pulsetables	121
7.1	Background	122
7.2	Pulsetable and Wavetable Synthesis	123
7.2.1	Learnable Pulsetable and Selection	126
7.3	Wind Instrument Synthesis	128
7.3.1	Training Details	129
7.4	Experiments	131
7.4.1	Dataset	131
7.4.2	Qualitative Comparison	132
7.4.3	Quantitative Comparison	137
7.4.4	Timbre Transfer	138
7.5	Further Notes	140
8	Summary and Outlook	143
	Appendix	147
A	Periodicity of Spectral Loss with Rectangular Window	147
B	Differentiability of Dissonance Kernel	149
C	Zero-Interleaved Spectrum of Repeated Signals	150
	Abbreviations	153
	Bibliography	155

1 Introduction

Das Kind—es schwebt!
Es berührt nicht die Erde mit seinen Füßen.
Es ist nicht der Schwere unterworfen.
Es ist fast unkörperlich. Seine Materie ist durchsichtig.
Es ist tönende Luft. Es ist fast die Natur selbst.
Es ist frei.

Ferruccio Busoni

Music, unlike poetry, painting, or architecture, is and has always been a highly abstract art form, created solely by varying pressure in the air over time. Yet, using such pressure variations, artists and composers can express high-level concepts such as love, sorrow, or the acceleration of a steam train¹, in widely recognizable ways [89, 211]. In musical performances, we are often drawn towards sounds that feel *intentional* and *natural*, where nuances in the fundamental attributes of musical tones—pitch, loudness, and timbre—play an important role [88]. For example, a performer may alter the sound of their clarinet from a mellow hum to a sharp blare, intentionally shaping each tone to convey a specific musical idea or emotional quality. At the same time, both the physics of the instrument and the ability of the performer limit the range of producible sounds, so that characteristic constraints and small involuntary fluctuations in the tone attributes are an integral part of what we perceive as natural.

Computers expand the possibilities of musical performance and composition in many ways, including the ability to freely modify pitch, loudness, and timbre in synthesized tones. However, a primary objective in music signal processing is to constrain computer models toward producing signals that sound natural. Increasingly, this problem is addressed by training large, general-purpose generative models on data that implicitly captures the desired constraints and characteristics through its distribution (see, e.g., [64, 70, 96]). If, for example, the training data consists of real-world clarinet recordings, newly generated samples from the model will exhibit similar characteristics and can therefore become perceptually indistinguishable from a recording. While this data-driven deep learning approach can produce highly convincing outputs across many different domains and modalities, it has three main disadvantages for music signals.

1. *Controllability*: Musicians need meaningful controls to creatively and intentionally influence the resulting sound. General deep learning models often use internal representations that do not correspond to concepts such as pitch, loudness, or timbre. This makes it difficult to provide interpretable, musically meaningful control parameters.

¹ Arthur Honegger, *Pacific 231*, 1923

2. *Data Efficiency*: Learning natural constraints and expressive nuances of pitch, loudness, and timbre from data alone typically requires large amounts of high-quality training data. If we care about the subtle, intentional variations used by expert musicians, these must be well represented in the dataset, requiring a labor-intensive and costly data collection.
3. *Computational Efficiency*: Deep learning models are often intentionally overparameterized to improve optimization and performance [127]. As a result, generating or modifying audio is substantially more computationally expensive than with purpose-built, manually optimized systems. This can be a limitation in music production, where real-time processing is often required.

In this thesis, we will explore *differentiable models* as one possible way to address these limitations. We use the term *model* to describe an explicit mapping from inputs (e.g., a parametric representation of tone attributes) to outputs (e.g., an audio signal) based on physical, perceptual, or signal-based knowledge. Depending on the application, an appropriate model choice enables the definition of interpretable and controllable parameters, as well as an efficient implementation. For example, the sound of a clarinet can be modeled at different levels of abstraction, ranging from detailed but computationally costly physical models that mirror controls of a real instrument [52] to efficient signal models that allow for direct manipulation of tone attributes but often require additional constraints to ensure natural-sounding results [186].

A model is called *differentiable* if its output changes smoothly with respect to the input, so that gradients can be computed for individual inputs or control parameters. In combination with a *loss function* measuring the distance between the current model output and a target output, these gradients provide a *learning signal* that indicates how parameters should be adjusted to better satisfy the given objective. We will use this learning signal in two complementary ways. First, it enables the data-driven optimization of neural networks when the model is embedded within a deep learning pipeline, improving computational and data efficiency by introducing an *inductive bias*, i.e., explicit assumptions about the input–output relations that guide the learning process. Second, it allows for the direct optimization of the model’s parameters via gradient descent. In both cases, the properties of the model and its interaction with the loss function are critical to the effectiveness of the optimization process. Related approaches in audio signal processing and Music Information Retrieval (MIR) are studied in the context of differentiable digital signal processing (DDSP) [42, 72] and model-based deep learning [159]. In other research fields, terms like knowledge-guided machine learning [90] or differentiable rendering [92] are used.

In the subsequent chapters, we structure our exploration of differentiable models for music signal processing around three high-level tasks: the *reconstruction*, *adaptation*, and *synthesis* of monophonic music signals, with a focus on controlling and/or preserving the tone attributes pitch, loudness, and timbre. Each task is illustrated using one or more concrete application scenarios, including singing voice reconstruction (Chapter 4), intonation and timbre adaptation (Chapters 5 and 6), and wind instrument synthesis (Chapter 7). In these scenarios, we demonstrate how knowledge from physics, perception, and signal processing can be integrated to process music signals, enabling explicit control over the nuanced variations in the sound, while still leveraging the advantages of gradient-based optimization. At the same time, we critically

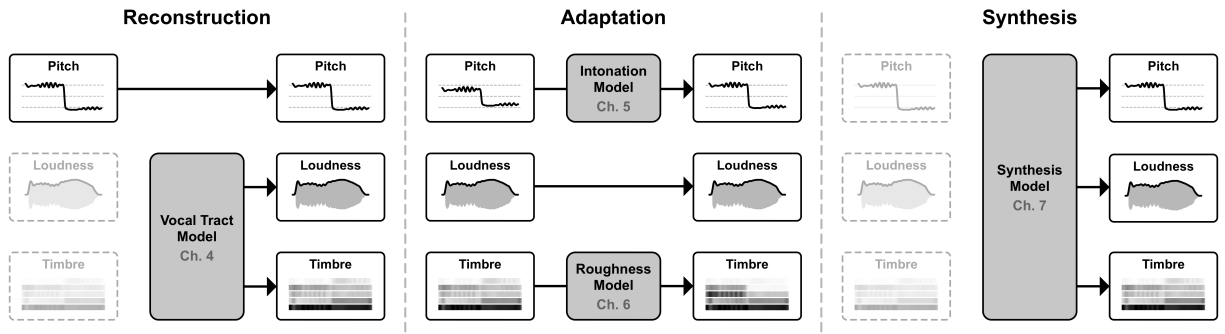


Figure 1.1: Overview of the high-level tasks (reconstruction, adaptation, synthesis) and application scenarios considered in this thesis. Boxes represent tone attributes (pitch, loudness, timbre) manifested in audio signals, which are processed in different ways across scenarios. Semi-transparent dashed elements denote tone attributes that are inferred or synthesized by the model.

analyze limitations of the models used in the respective tasks and identify scenarios in which structural constraints and underlying assumptions restrict performance or applicability. In the following, we define the considered tasks broadly, followed by a more detailed discussion of our specific contributions within each application scenario in Section 1.1. The overview in Figure 1.1 further illustrates how tone attributes present in an audio signal are affected or preserved in each scenario, where grayed-out attributes indicate information that must be inferred or synthesized by the model.

Reconstruction refers to the task of recovering one or multiple corrupted tone attributes in a music signal. Such corruptions may occur, for example, when information was lost during recording, transmission, or storage of the signal. The objective is to restore those degraded attributes faithfully from the available data while preserving attributes that are already present in the signal with all their nuances for further use and analysis. As illustrated on the left side of Figure 1.1, the model-based approach presented in Chapter 4 ensures that tones preserve the original, intended pitch present in a source signal, while data-driven components learn to infer loudness and timbre suitable for a high-quality signal.

Adaptation involves modifying specific tone attributes in a signal while preserving others. This type of processing is ubiquitous in music production, for example with equalization (primarily a timbre modification), dynamic range compression (loudness modification), or intonation correction (pitch modification). In the considered application scenarios, gradients from differentiable perceptual models are used to directly influence pitch (Chapter 5) or timbre (Chapter 6) of tones in a source signal, illustrated in the middle of Figure 1.1.

Synthesis describes the task of converting a parametric or symbolic representation of music (back) into an audio signal. Particularly in conjunction with a suitable *analysis* method, a (re-)synthesis system that disentangles the tone attributes into separate control parameters enables their control in musically meaningful ways. In Chapter 7, we approach this as an *analysis-by-synthesis* problem, where a neural network is trained to estimate parameters for a differentiable synthesis model, with the goal of resynthesizing a signal that is as similar as possible to the input. The right side of Figure 1.1 illustrates the task of synthesizing a music signal from only a parametric representation of tone attributes.

Naturally, these scenarios overlap in methodology and applications. A suitable resynthesis system, for instance, enables the adaptation of tone attributes in a parametric representation before converting it back to an audio signal. Based on the application scenarios considered in this thesis, we examine both the strengths and limitations of differentiable models across a range of music signal processing problems. In doing so, we contribute methodological improvements within the individual tasks, provide curated datasets and reference implementations to support reproducibility and open research practices, and offer a broader perspective on the role of model-based machine learning in music processing and beyond.

1.1 Structure and Main Contributions

This thesis is organized as follows. **Chapter 2** provides a primer on music signals, with a particular focus on the fundamental tone attributes pitch, loudness, and timbre, as well as their technical representations. Furthermore, we give an overview of common music signal synthesis methods, and introduce formal definitions and notation that will be useful throughout this thesis.

In the main part, we then consider several music signal processing scenarios where differentiable formulations of physical, perceptual, or signal models can help constrain a system toward desired musical outcomes, highlighting strengths and weaknesses of such approaches along the way. In particular, the last section in each chapter provides a summary of how the main findings in the context of the respective application scenario relate to the broader objectives of the thesis.



In **Chapter 3**, we describe and analyze a fundamental issue in gradient-based optimization. For such methods to yield meaningful results, the gradients of the loss function with respect to model parameters must consistently “point in a meaningful direction”. When comparing model outputs to a target signal with the widely used multi-scale spectral (MSS) loss, this condition is not necessarily satisfied, even for simple model parameters such as the frequency of a sinusoidal oscillator. We provide an analysis that links this issue to spectral leakage in frequency-domain representations and derive more favorable configurations for the MSS loss that improve the smoothness of the loss function with respect to pitch-related parameters. We further demonstrate that the smoothness with respect to certain signal properties can influence learning beyond explicit parameter gradients in DDSP models, by analyzing the latent space organization of a Variational Autoencoder (VAE) trained with different reconstruction loss functions.



In **Chapter 4**, we introduce a differentiable, physically inspired model for recovering corrupted tone attributes in a data-driven manner while reliably preserving others from the original signal. Specifically, we explore the reconstruction of singing voice signals recorded with a larynx microphone (LM), a device that captures vibrations directly from a singer’s throat. While these signals are largely crosstalk-free, they are of low quality with a limited frequency range and missing speech components like consonants. To reconstruct

a natural-sounding and high-quality singing voice signal from LM recordings, we combine a physical model of the vocal tract with a noise synthesizer that accounts for non-tonal components of singing. This way, our approach allows for a data-driven enhancement of timbre (e.g., to improve intelligibility), but preserves the original fundamental frequency (F0) as intended by the singer.



In **Chapter 5**, we utilize the gradients of a differentiable perceptual model to adapt the pitch of individual voices in a polyphonic musical context. To do this, we define a cost measure based on perceptual and musical models, combining and reformulating existing principles of intonation assessment. By introducing an intonation adaptation algorithm based on gradient descent, we can jointly and iteratively adapt the pitch of multiple voices. We show how our method can be used to mitigate unintended tuning deviations in multi-track recordings in a controllable way while also accounting for the interaction between voices as they naturally occur in ensembles.



In **Chapter 6**, we use a similar perceptual model, but adapt timbre instead of pitch to improve blending between individual voices. Based on the insight that the distribution of signal energy across partials contributes significantly to the measured sensory dissonance in complex sounds (Chapter 6.2), we derive and evaluate strategies to change the timbre of synthesized tones, either by changing the amplitudes of individual partials or by introducing inharmonic frequency shifts (Chapter 6.3).



In **Chapter 7**, we turn to the synthesis of music signals with a physically informed signal model for wind instrument sounds. In particular, we utilize *pulsetable synthesis*, a concatenative audio synthesis method that generates harmonic tones by repeating short waveform prototypes. Formulating pulsetable synthesis in a differentiable way, we can learn both the waveform prototypes themselves, as well as their selection from data through unsupervised training. A complete wind instrument synthesis system based on this method disentangles pitch, loudness, and timbre into individually interpretable and controllable model parameters, while providing an expressive wind instrument synthesizer with as few as 60,000 learnable parameters, demonstrating the efficiency gains that can be achieved with differentiable models.

We conclude the thesis in **Chapter 8** with a summary and outlook on future research directions.

1.2 Underlying Publications

This thesis builds on previously published articles to which I have made the main contributions as first author. The list below includes the publications underlying individual chapters. All listed works have appeared in peer-reviewed journals or international conference proceedings in the fields of MIR and audio signal processing.

Chapter 1. Introduction

- [176] Simon Schwär and Meinard Müller. Multi-scale spectral loss revisited. *IEEE Signal Processing Letters*, 30: 1712–1716, 2023. doi: 10.1109/LSP.2023.3333205

This article forms the basis of Chapter 3. I was the primary contributor to the study design, implementing and conducting the experiments, and preparing the manuscript. Meinard Müller supervised the work and contributed throughout all stages of the research and manuscript preparation.

- [181] Simon Schwär, Michael Krause, Michael Fast, Sebastian Rosenzweig, Frank Scherbaum, and Meinard Müller. A dataset of larynx microphone recordings for singing voice reconstruction. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 7(1):30–43, 2024. doi: 10.5334/tismir.166

This article forms the basis of Chapter 4. I was the primary contributor to creating the dataset, analyzing signals, running experiments, and preparing the manuscript. Michael Krause co-developed the baseline model and contributed to manuscript preparation. Michael Fast helped recording the dataset, running baseline experiments, and conducting the listening test in the context of his Master’s thesis work co-supervised by me. Sebastian Rosenzweig co-developed the baseline model. Frank Scherbaum shared his expertise on larynx microphones and provided the hardware. Meinard Müller supervised the work and contributed throughout all stages of the research and manuscript preparation.

- [177] Simon Schwär and Meinard Müller. Multi-voice intonation adaptation via gradient descent. *IEEE Transactions on Audio, Speech and Language Processing (TASLPRO)*, 34:2491–2503, 2026. doi: 10.1109/TASLPRO.2026.3675812

This article forms the basis of Chapter 5 and builds on a previously published conference paper [178], co-authored by myself, Sebastian Rosenzweig, and Meinard Müller. In this extended journal article, I was the primary contributor to the study design, implemented and conducted all experiments, and prepared the manuscript. Meinard Müller supervised the work and contributed throughout all stages of the research and manuscript preparation.

- [182] Simon Schwär, Stefan Balke, and Meinard Müller. Measuring sensory dissonance in multi-track music recordings: A case study with wind quartets. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 117–124, Daejeon, South Korea, 2025

This article is part of Chapter 6. I was the primary contributor to the study design, conducting the experiments, and preparing the manuscript. Stefan Balke and Meinard Müller contributed throughout all stages of the research and manuscript preparation. Furthermore, Meinard Müller supervised the work.

- [180] Simon Schwär, Meinard Müller, and Sebastian J. Schlecht. Modifying partials for minimum-roughness sound synthesis. In *Proceedings of the 3rd International Conference on Timbre*, pages 130 – 134, Thessaloniki, Greece, 2023

This article is part of Chapter 6. I was the primary contributor to the development of the research concept, the implementation of experiments, and the manuscript preparation. Sebastian Schlecht and Meinard Müller were involved in all stages of the work, contributing to idea development and manuscript writing.

- [183] Simon Schwär, Christian Dittmar, Stefan Balke, and Meinard Müller. Differentiable pulsetable synthesis for wind instrument modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 14792–14796, Barcelona, Spain, 2026. doi: 10.1109/ICASSP55912.2026.11462505

This article forms the basis of Chapter 7. I was the main contributor to developing the research concept, implementing and conducting the experiments, and preparing the manuscript. Christian Dittmar introduced the use of pulsetables for wind instrument synthesis, implemented a non-differentiable version (outlined in [40]), and contributed to the manuscript. Stefan Balke contributed to the development of ideas and manuscript preparation. Meinard Müller supervised the work and contributed throughout all stages of the research and manuscript preparation.

1.3 Additional Publications

In addition to the articles listed above, I have contributed to the following publications in audio and music processing during the research for this thesis.

- [125] Peter Meier, Sebastian Strahl, Simon Schwär, Meinard Müller, and Stefan Balke. Pitch estimation in real time: Revisiting SWIPE with causal windowing. In *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, pages 285–297, London, UK, 2025. doi: 10.5281/zenodo.17496630
- [40] Christian Dittmar, Johannes Zeitler, Stefan Balke, Simon Schwär, and Meinard Müller. PULSE-IT: Lightweight and expressive synthesis of wind instrument playing. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Daejeon, South Korea, 2025
- [12] Hans-Ulrich Berendes, Simon Schwär, and Meinard Müller. Lyrics transcription in Western classical music with Whisper: A case study on Schubert’s Winterreise. In *Workshop on NLP for Music and Audio (NLP4MUSA)*, Oakland, CA, United States, 2024
- [144] Yigitcan Özer, Simon Schwär, and Meinard Müller. Piano concerto accompaniment creation. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, San Francisco, CA, USA, 2024
- [123] Peter Meier, Simon Schwär, and Meinard Müller. A real-time approach for estimating pulse tracking parameters for beat-synchronous audio effects. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 314–321, Guildford, Surrey, UK, 2024
- [143] Yigitcan Özer, Leo Brütting, Simon Schwär, and Meinard Müller. libsoni: A Python toolbox for sonifying music annotations and feature representations. *Journal of Open Source Software (JOSS)*, 9(96):06524:1–6, 2024. doi: 10.21105/joss.06524
- [11] Hans-Ulrich Berendes, Simon Schwär, Maximilian Schäfer, and Meinard Müller. Towards differentiable piano synthesis based on physical modeling. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Milano, Italy, 2023
- [142] Yigitcan Özer, Simon Schwär, Vlora Arifi-Müller, Jeremy Lawrence, Emre Sen, and Meinard Müller. Piano Concerto Dataset (PCD): A multitrack dataset of piano concertos. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 6(1):75–88, 2023. doi: 10.5334/tismir.160
- [121] Peter Meier, Simon Schwär, Gerhard Krump, and Meinard Müller. Evaluating real-time pitch estimation algorithms for creative music game interaction. In *INFORMATIK 2023 – Designing Futures: Zukünfte gestalten*, pages 873–882. Gesellschaft für Informatik e.V., Bonn, Germany, 2023. doi: 10.18420/inf2023_97
- [122] Peter Meier, Simon Schwär, Gerhard Krump, and Meinard Müller. Real-time pitch estimation for creative music game interaction. In *Proceedings of the Deutsche Jahrestagung für Akustik (DAGA)*, pages 1346–1349, Hamburg, Germany, 2023
- [165] Sebastian Rosenzweig, Simon Schwär, and Meinard Müller. libf0: A python library for fundamental frequency estimation. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022
- [179] Simon Schwär, Meinard Müller, and Sebastian J. Schlecht. A variational Y-Autoencoder for disentangling gesture and material of interaction sounds. In *AES 4th International Conference on Audio for Virtual and Augmented Reality (AES AVAR)*, Redmond, WA, USA, 2022

- [120] Peter Meier, Simon Schwär, Sebastian Rosenzweig, and Meinard Müller. Real-time MIR algorithms for music-reactive game world generation. In *Proceedings for the Workshop für Innovative Computerbasierte Musikinterfaces (ICMI)*, Darmstadt, Germany, 2022. doi: 10.18420/muc2022-mci-ws03-225
- [178] Simon Schwär, Sebastian Rosenzweig, and Meinard Müller. A differentiable cost measure for intonation processing in polyphonic music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 626–633, Online, 2021
- [164] Sebastian Rosenzweig, Simon Schwär, Jonathan Driedger, and Meinard Müller. Adaptive pitch-shifting with applications to intonation adjustment in a cappella recordings. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 121–128, Vienna, Austria, 2021

1.4 Acknowledgments

This dissertation is the result of research conducted between 2022 and 2025 at the International Audio Laboratories Erlangen, a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS. My work was funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) through the projects GVM (MU 2686/13-1, SCHE 280/20-1) and GVM+ (MU 2686/13-2) and I am grateful for this support.

But first and foremost, all of this is based on the incredible support from my supervisor and mentor Meinard Müller. Thank you, Meinard, for creating a research environment a PhD student can only dream of. From shielding us from administrative issues to always being there whenever some small advice or a big “Meinard revision” is needed, I truly appreciate what you do and how you do it. Also personally, with your work ethic, your love for the details, and your high regard for the human aspect in everything, you will be a role model for the rest of my career.

The very first step toward this thesis was taken in 2018, in the office of Jan Plogsties at Fraunhofer IIS, when he introduced me to Frank Wefers, who had just started as a professor at the AudioLabs. This was the start of an intense journey into the depths of virtual acoustics, sending me on the path that ultimately led to writing these words. Thank you, Frank, for sharing your passion for many things, including the 138 at the Indian restaurant! In this context, I also thank Jürgen Herre for creating and leading the MPEG-I team at Fraunhofer IIS while I was part of it. I thoroughly enjoyed this formative experience.

I would also like to thank three additional academic role models who have influenced my work in various ways: Sebastian Schlecht for hosting me for my Master’s thesis at Aalto University in the winter of 2021/22 and for offering a thoughtful perspective in any discussion, Frank Scherbaum for sharing his expertise on larynx microphones and for his inspiring way of approaching research, and Sebastian Stober for taking the time to review this thesis and for always having some good advice wherever we met.

I am grateful for the institutional support and great research conditions at the AudioLabs, made possible by the professors, executives, and coordinators (in alphabetical order) Christian Dittmar, Bernd Edler, Bernhard Grill, Emmanuël Habets, Jürgen Herre, Meinard Müller, Frederik Nagel, Nils Peters, Stefan Turowski,

and Frank Wefers. Furthermore, things would not run nearly as smoothly without the administrative staff. Thank you very much to Carmen Craciun, Tracy Harris, Petra Neubarth, Day-See Riechmann, and Elke Weiland.

It is hard to overstate how enjoyable it is to work in a team of good people. A special thank you goes to the members of Meinard's extended research group over the years (in alphabetical order): Jakob Abeßer, Vlora Arifi-Müller, Stefan Balke, Judith Bauer, Hans-Ulrich Berendes, Ching-Yu "Sunny" Chiu, Jueun Kang, Michael Krause, Ben Maman, Peter Meier, Yiğitcan Özer, Manuel Peters, Sebastian Rosenzweig, Abhirup Saha, Sebastian Strahl, Christof Weiß, Philipp Weyers, and Johannes Zeitler. Thank you all for sharing the journey and for letting me build on your work.

In addition to the aforementioned people, the AudioLabs (as any other institution) are made up of and by the community members. During my time, these were (you can guess the order): Alexander Adami, Ahmad Aloradi, Carlotta Anemüller, Mohamed Elminshawy, Richard Füg, Ünal Ege Gaznepoğlu, Philipp Götz, Ning Guo, Alexander Heinmüller, Adrian Herzog, Yaqiong Hou, Weilong Huang, Guanxin Jiang, Srikanth Korse, Kishor Kayyar Lakshminarayana, Jeremy Lawrence, Anna Leschanowsky, Wolfgang Mack, Daniele Mirabilii, Thomas Robotham, Dipanjan Datta Roy, Lorenz Schmidt, Martin Strauss, Matteo Torcoli, Anika Treffehn, Julian Wechsler, Nils Werner, Niklas Winter, and Zeyu Xu.

To everyone at AudioLabs, specifically also including *GroupMM*, thanks for sharing knowledge, perspectives, coffee breaks, music recommendations, bike rides, hotel rooms, ski lifts, dairy facts, recording sessions, shuttlecocks, Kaiserschmarrn, stages, huts, and camper vans. It was a blast!

Conferences and workshops were a formative part of my time as a PhD student and I am grateful to have had the opportunity to travel to so many different places and meet so many great people. I particularly acknowledge Dagstuhl Seminar 24302 which was eye-opening in showing how amazing academia can be. Also, I would like to thank everyone who has organized the fantastic ISMIR conferences that I attended: Online 2021, Milano 2023, San Francisco 2024, and—especially—Daejeon 2025. I hope to be able to give back in this regard at some point.

I thank Stefan Balke, Hans-Ulrich Berendes, Patricia Hu, Manuel Peters, Veronika Schwarm, Sebastian Strahl, and Johannes Zeitler—and of course Meinard Müller—for proof-reading parts of this thesis.

Finally, I thank my family, and especially my parents Ursula and Michael, for creating the low-key warm environment that made me who I am. This thesis is for you!

Nürnberg, January 2026

2 A Primer on Music Signals

Before we turn to the individual application scenarios where music signals are reconstructed, adapted, or synthesized, we first outline and formalize fundamental concepts from Music Information Retrieval (MIR) and audio signal processing that are used throughout this thesis. In particular, we review audio signals (Section 2.1), describe how the tone attributes pitch, loudness, and timbre can be measured in audio signals and expressed in symbolic representations (Section 2.2), take a closer look at the singing voice (Section 2.3), and discuss common techniques for synthesizing music signals (Section 2.4).

2.1 Audio Signals in Time and Frequency Domain

A *sound* is a variation of air pressure over time. At a particular point in space, e.g., the position of a microphone capsule, we can describe this change in air pressure as a function $\psi : \mathbb{R} \rightarrow \mathbb{R}$, mapping a time t to a deviation from ambient pressure $\psi(t)$. This function can be regarded as a *continuous* audio signal in time domain. Throughout this thesis, we mainly work with *discrete* audio signals, which we will typically represent by a function $x : \mathbb{Z} \rightarrow \mathbb{R}$, where

$$x[n] := \psi(nT) \quad (2.1)$$

is the n th *sample* of an underlying signal ψ .² The sampling period $T = 1/F_s$ in seconds is defined as the reciprocal of the sampling rate F_s , which, measured in Hz, defines the number of samples per second used for x . Alternatively, we can view an excerpt of this discrete signal as a vector $\mathbf{x} \in \mathbb{R}^L$ that contains L consecutive samples. As an example, consider the audio signal of a violin playing two successive notes shown in Figure 2.1b. By plotting this time-domain signal of roughly 3.5 seconds length ($L \approx 150\,000$ samples at $F_s = 44\,100$ Hz), we can see some fluctuations in the amplitude of the two tones over time, which strongly correlates with the perceived loudness of the sound (see Section 2.2.3). However, other properties of the tones, such as pitch and timbre, are not immediately visible in this time-domain representation often referred to as a *waveform*.

This is a main reason for the widespread use of *time–frequency* representations to analyze and visualize the time-varying presence of different frequency components in a music signals. An extensively used time–frequency representation is based on the Short-Time Fourier Transform (STFT), which combines multiple Discrete Fourier Transforms (DFTs) of short segments or *frames* of the audio signal x , each

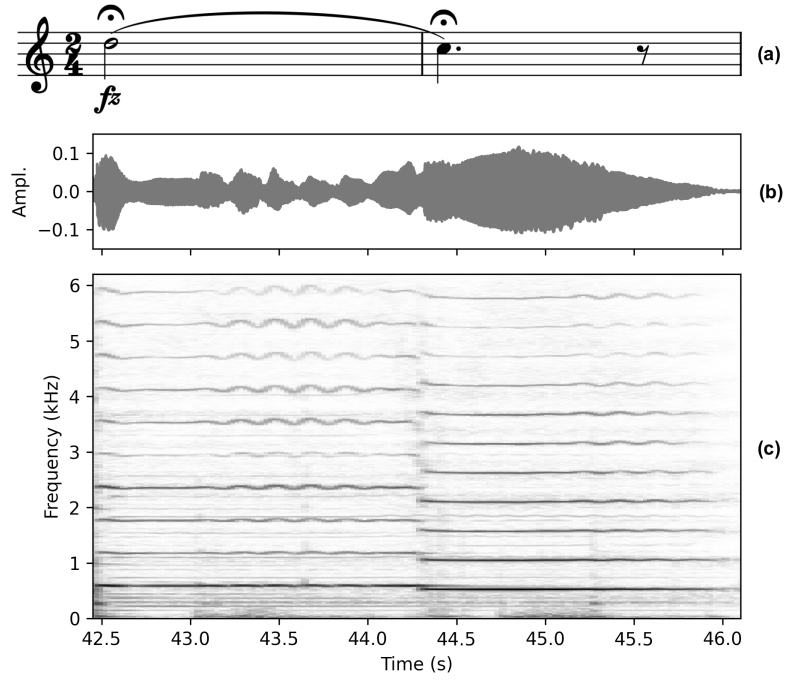
² For a clear distinction between functions with discrete or continuous independent variables, we use square brackets $[\cdot]$ for the former and round brackets $(\cdot)$ for the latter, following the convention from [140].

Figure 2.1: An audio recording of a violin playing two successive notes (from Joseph Haydn String Quartet op. 74 no. 1, first violin, bars 48-49, recording Q-JH-VN1-NR-10 from the *Virtuoso Strings* dataset [213]).

(a) Sheet music representation.

(b) Time-domain representation of the recording.

(c) Magnitude spectrogram $\mathcal{Y}_{w,N,p}$ derived from the Short-Time Fourier Transform (STFT) of the recording (w is a Hann window of length $N = 4096$, $p = p_\gamma$ with $\gamma = 50$).



with length $N \in \mathbb{Z}$. These segments are potentially overlapping and advance in the signal with a *hop size* $H \in \mathbb{Z}$. Because the time-domain signals we consider are real-valued, the complex DFT is symmetric around 0 Hz, and only the $K = \lceil (N+1)/2 \rceil$ positive-frequency bins are required to uniquely represent the signal in frequency domain. The resulting complex STFT matrix $\mathcal{X}_{w,N} \in \mathbb{C}^{M \times K}$ (where $w \in \mathbb{R}^N$ is a *window function*) can be computed for time index $m = 0, \dots, M-1$ and frequency index $k = 0, \dots, K-1$ by

$$\mathcal{X}_{w,N}[m, k] := \sum_{n=0}^{N-1} x[n + mH] w[n] \exp\left(\frac{-i2\pi kn}{N}\right). \quad (2.2)$$

While the hop size can be an important parameter to control the desired amount of redundancy in the STFT representation, we use $H = \lceil N/2 \rceil$ throughout this thesis, so that we omit H as a subscript of $\mathcal{X}$ for brevity of notation. The multiplication of the signal segment with a window function w ensures that boundary effects at the edges of the signal segments do not have adverse effects on the DFT analysis, often by tapering the segment towards zero at its start and end. Unless otherwise specified, we use the Hann window of length N for w , which is a common choice in music signal processing [151]. We will discuss the influence of the window function (and other STFT configuration parameters) in detail in Chapter 3.

Both the time index m (often called *time step* or *frame number*) and frequency index k (or *frequency bin number*) each have a physical interpretation, where

$$T_{\text{coef}}(m) := \frac{mH}{F_s} \quad (2.3)$$

is the start point of the segment in seconds and

$$F_{\text{coef}}(k) := \frac{kF_s}{N} \quad (2.4)$$

is the center frequency of the *frequency bin* in Hertz. Each frequency bin refers to a finite interval on the frequency axis resulting from the discretization of the spectrum in the DFT.

To analyze the presence of certain frequency components in the signal at a given time, it is often sufficient and more convenient to consider the *spectrogram* $\mathcal{Y} \in \mathbb{R}^{M \times K}$, which disregards the phase component of $\mathcal{X}$ and only considers its magnitude, so that

$$\mathcal{Y}_{w,N,p}[m, k] := p\left(|\mathcal{X}_{w,N}[m, k]|\right), \quad (2.5)$$

where $p : \mathbb{R}_+ \rightarrow \mathbb{R}$ is a *compression function*. This compression of the magnitudes can be used to account for the fact that loudness perception largely follows a logarithmic scale [51]. For example, we can convert a value $v \in \mathbb{R}_+$ to *Decibel* (dB) with $p_{\text{dB}}(v) = 20 \log_{10}(v + \varepsilon)$, where ε is a small value (e.g., $\varepsilon = 10^{-6}$) to avoid numerical issues when $v \rightarrow 0$. An alternative compression function is $p_\gamma(v) = \log(1 + \gamma v)$ [131], where $\gamma > 0$ is a parameter for the amount of compression. This compression function has two main advantages over p_{dB} . First, the hyperparameter γ makes it easy to control the amount of compression depending on the desired amplification of frequency components that are less prominent in the signal. Second, since $p_\gamma(0) = 0$ and the function being monotonically increasing, $\mathcal{Y}$ becomes non-negative, which can be more convenient for some downstream tasks.

Figure 2.1c shows a spectrogram of our violin signal, using logarithmically compressed magnitudes to map the color contrast better to perceptual differences in loudness. This representation makes several properties of the two tones more apparent: the F0 of the first tone is slightly higher than that of the second (around 500 Hz), and each tone exhibits a harmonic structure with varying amplitudes. We will discuss these properties in detail in Section 2.2. For a more comprehensive account of these fundamental signal representations, we refer to textbooks on MIR [131] or audio signal processing [139, 140].

2.2 Of Tones and Notes

In addition to the general concept of a *sound*, we have already used the term *tone*. Throughout this thesis, we use tone for a particular kind of sound that is characterized by (nearly) periodic pressure variations, which leads to the perception of a clear pitch, so that tones can be ordered from “low” to “high.” Beyond pitch, a tone can be described by its timing (including onset and duration), loudness, and timbre. In the subsections below, we will define and examine these attributes in detail.

Closely related is the term *note*, which we use to refer to the underlying musical event corresponding to a tone. Notes can be used as an abstraction that discretizes pitch and time as musical quantities, either

explicitly for notation or implicitly, e.g., during improvisation. In Western classical music, the notes are often specified to a performer through sheet music, an example of which is shown in Figure 2.1a. This *symbolic representation* gives a performer some indication of timing (e.g., whether to play a quarter or half note), pitch (e.g., prescribing a D_5), loudness (often referred to as *dynamics* in the context of music performance, e.g., that a note should be played *fz*, or *forzato*, with an emphasized beginning), and timbre (e.g., by specifying an instrumentation or instrument-specific playing technique such as *sul tasto*, bowing near the fingerboard to create a “warmer” sound). However, sheet music often leaves many performance aspects implicit. For example, the excerpt in Figure 2.1a does not specify any dynamics for the second note, so that the performer has to decide on a suitable continuation of the dynamics after the *forzato* of the first note.

Other symbolic representations like the Musical Instrument Digital Interface (MIDI) [81], designed originally for the control of digital synthesizers, allow for more precise definitions of note qualities. For example, dynamics (or *velocity* in MIDI language) are specified for each individual note on a discrete scale from 0 to 127 in MIDI. This is more detailed than sheet music but in its basic form still does not allow for a specification of dynamic progression *within* individual notes.³ In any case, when we view a tone as a realization of a note, performers have considerable freedom in shaping the details of the tone attributes, not only by “filling the gaps” of an incomplete symbolic representation, but also by suitably varying timing, pitch, loudness, and timbre over time, within and across tones.

In the following, we will describe the tone attributes separately, but it is important to keep in mind that they are not independent, neither physically nor perceptually. For example, varying pitch or loudness of a tone on an acoustic musical instrument will almost inevitably also change its timbre as a consequence of amplitude- and frequency-dependent instrument properties. Similarly, to imitate this natural behavior of musical instruments, a synthesizer controlled via MIDI may map parameters like velocity to multiple tone attributes, often affecting both loudness and timbre.

2.2.1 Timing

Timing describes how musical events are positioned and organized in time, with onset and duration determining the timing of each tone. While these two parameters leave considerable freedom for musical expression, timing is not a primary focus of this thesis, since we consider the reconstruction, adaptation, and synthesis of attributes *within* individual tones with a given, fixed onset time and duration. Nevertheless, the temporal organization of tones strongly influences the other tone attributes in performance and perception, and is therefore briefly outlined here.

On a larger temporal scale, musical timing is determined on the symbolic note level, where a score may prescribe a global tempo (e.g., in beats per minute, BPM), a meter (e.g., two beats per measure as in

³ There are methods in MIDI to achieve time-varying velocity, e.g., using *Aftertouch* or the *MIDI Polyphonic Expression (MPE)* extension, but we consider the most basic variant of the protocol here.

Figure 2.1a), and the succession of individual notes and rests with a quantized duration (e.g., a half note, followed by a dotted quarter note, followed by an eighth rest, as in Figure 2.1a). By interpreting this symbolic information about *musical time* given in measures and beats, a performer creates a musical performance where the onset and duration of each tone is mapped to *physical time* in seconds. However, individual performances often vary considerably in physical time (see, e.g., [237]), reflecting interpretative choices, performance conventions, and acoustic conditions [108]. In ensemble contexts, timing further includes synchronization between performers, where precise coordination is required to establish and maintain an impression of cohesion or “togetherness” between individual musical voices over time [59].

Beyond these larger-scale aspects, the so-called *microtiming* refers to small but systematic deviations from nominal timing positions, which play a crucial role in musical expression. Such deviations are the basis of widely studied phenomena such as swing [39], groove [1], and expressive timing in Western art music performance [156, 212]. Although variations in physical time can be as low as few milliseconds, these microtiming variations are important for nuanced expression and the naturalness of a performance.

2.2.2 Pitch and Fundamental Frequency

When we look at symbolic representations of music, each note is typically assigned a single, discrete pitch value. The predominant system in Western music divides the octave (i.e., the interval between each doubling of frequency) into twelve equally spaced pitch classes or semitones on a logarithmic axis, a system that is intuitively illustrated by the piano keyboard layout. The pitch classes corresponding to white keys are labeled A-G, with the accidental $\sharp$ (*sharp*) raising the pitch class by one step and $\flat$ (*flat*) lowering it. These pitch classes are repeated every octave, so that the pitch of a note is uniquely defined by a pitch class plus an octave number, which is called the *scientific pitch notation*. For example, the two notes in Figure 2.1a have pitch D_5 and C_5 , respectively.

In the MIDI specification [81], each pitch is associated with an integer value, where A_4 corresponds to MIDI pitch 69, D_5 , being a fourth (five semitones) higher, to MIDI pitch 74, and C_5 to 72. MIDI further specifies a corresponding frequency for each pitch, adopting the widely used 12-tone equal temperament (12-TET) tuning system and the standard tuning for the A_4 of 440 Hz [86].⁴ This way, following [131] in notation, for any MIDI pitch p , the corresponding frequency $F_{\text{pitch}}(p)$ in Hz is given by

$$F_{\text{pitch}}(p) := 440 \cdot 2^{(p-69)/12}. \quad (2.6)$$

Consequently, in our violin example, the D_5 is associated with a frequency of $F_{\text{pitch}}(74) \approx 587$ Hz and the C_5 with $F_{\text{pitch}}(72) \approx 523$ Hz.

⁴ Note that the choice $A_4 = 440$ Hz is somewhat arbitrary, since reference tuning frequencies vary in practice. Symphony orchestras in Europe, for instance, often use a higher tuning of up to 444 Hz for A_4 , whereas in historically informed performance of Baroque music substantially lower tunings are common, with A_4 reaching values around 415 Hz, so that 440 Hz is closer to Bb_4 in this case. Nevertheless, the MIDI pitch convention provides a convenient and widely adopted reference frame for the computational processing of pitch-related data.

In a performance, a musician would seldom produce a tone with exactly this frequency. Instead, most musical instruments leave considerable freedom in forming the pitch for a given tone. Notably, the term pitch is used differently in literature for notes and tones. Where for notes it is used for a concrete quantity, for tones it refers to a continuous and subjective perceptual property that allows for ordering tones from “low” to “high” [149]. However, this perceptual property closely corresponds to the fundamental frequency (F0), which is the reciprocal of the longest *predominant* period in the pressure variations, reflecting that real recorded signals are only approximately periodic, e.g., due to background noise.

Because of this presence of noise as well as additional periodic components in real-world signals, reliably estimating F0 is a challenging problem. Many different algorithms for measuring F0 in monophonic signals have been proposed. Approaches based on signal processing models either measure periodicity in time domain with methods based on auto-correlation [15, 35, 115] or try to identify peaks and harmonic patterns in frequency domain [19, 167]. Data-driven approaches can be trained on previously annotated [97, 224] or synthetic [43] training data, or use *self-supervised objectives* to learn a relative pitch representation from comparison of signals with a known F0 *difference* [57, 160]. Both model-based and data-driven approaches often yield good results for monophonic and (nearly) harmonic signals. Yet, when working with or evaluating the quality of F0 estimates, we have to account for three main caveats. First, the time–frequency uncertainty principle in spectral analysis⁵ also constrains the precision of F0 estimates. This limitation aligns with perceptual requirements: reliable pitch perceptions requires the tone and its F0 to be stable for at least a few periods, where pitch perception accuracy increases with longer duration of the tone [130]. Consequently, statements about small differences between the F0 of two tones (e.g., when comparing intonation accuracy), should be interpreted with caution. Second, interferences such as crosstalk or other background noise can substantially reduce the reliability of F0 estimates [124], making clean signals preferable whenever possible. Third, most F0 estimation methods include model assumptions such as continuity between successive frames to improve robustness [3]. As a result, the estimates especially in the transition regions between successive tones can be less reliable.

F0 is typically measured on a linear frequency scale in Hz. If we want to map it to a logarithmic scale, we can treat Equation 2.6 as a continuous mapping and calculate the inverse

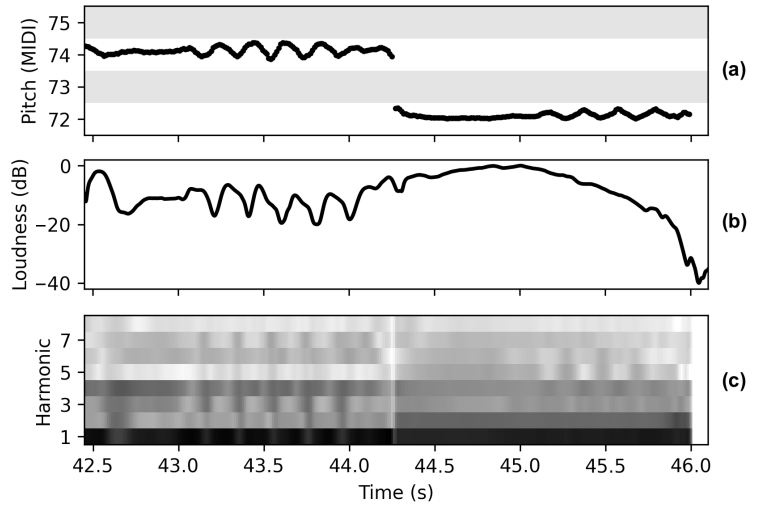
$$F_{\text{pitch}}^{-1}(f) = 12 \cdot \log_2 \left(\frac{f}{440} \right) + 69 \quad (2.7)$$

to obtain a “fractional MIDI pitch” value for a given frequency f . To avoid the terminology clash with different definitions of pitch here and in the following, we will refer to pitch as an objective quantity on a logarithmic scale (e.g., MIDI pitch), and to F0 when talking about the fundamental frequency in Hz. Finally, an important unit for measuring frequency differences or intervals is the *cent*. 100 cents are defined as one semitone in 12-TET, so that for two MIDI pitch values p_1 and p_2 , their distance in

⁵ Within each individual time frame, the DFT discards time information. If we choose a smaller window size N to get a better time resolution, we reduce the frequency resolution of the DFT. This way, we cannot achieve a high time and frequency resolution at the same time in a spectrogram.

Figure 2.2: Tone attributes extracted from the recording shown in Figure 2.1.

- (a) Pitch trajectory estimated with a score-informed *SWIPE* implementation [19].
- (b) Signal envelope using a windowed root mean square (RMS) algorithm with $N = 2048$.
- (c) Frame-wise relative energy of the first eight harmonics extracted using an F0-informed peak picking algorithm [182].



cents can be simply calculated with $100 \cdot |p_1 - p_2|$. By inserting Equation 2.7, we can also calculate the distance d_{cents} between two frequencies f_1 and f_2 with

$$d_{\text{cents}}(f_1, f_2) = 1200 \cdot \log_2 \left(\frac{f_1}{f_2} \right). \quad (2.8)$$

Equipped with these prerequisites of methods, terminology, and notation we can now analyze the *F0 trajectory*, which is based on frame-wise F0 estimates and describes the progression of fundamental frequency over the duration of a signal such as our violin example recording.⁶ As we can see in Figure 2.2a, the most prominent deviation from the nominal MIDI pitch of 74 and 72, respectively, is the periodic variation in pitch in the second half of each tone. This *vibrato* is a common expressive feature [184] and especially characteristic for string instruments [6] and singers [152]. In our example, the shape of the vibrato is not quite sinusoidal, but clearly periodic. The *vibrato rate* (i.e., the frequency of the modulation) is similar in both tones at around 5 Hz, where values between 4 and 6 Hz are common for string instruments and singers alike [61]. The *vibrato extent* (i.e., the amplitude of the modulation, measured from peak to valley) is 51 cents for the first tone and 34 cents for the second tone. This is consistent with a reported typical extent of 50 cents for string instruments [61], while the modulation can be much stronger in some singing styles (see, e.g., [234], but also Figure 2.6).

Furthermore, also the stable part of the pitch trajectory is not exactly at the prescribed MIDI pitch. The median deviation of the first tone from MIDI pitch 74 is +13 cents, whereas the second tone deviates from MIDI pitch 72 by +12 cents in the median. Using the median here is robust against measurement outliers or other short-term deviations, e.g., at the onset or end of the tone. While these deviations are relatively small compared to the vibrato extent and the pitch distance between the tones, they are systematic and can play a relevant role in the interplay with other voices. Possible sources for this offset can be differences in the reference tuning for A₄, deviations from 12-TET by the other voices, or particularities of the tuning of individual intervals within chords. We will discuss these effects in detail in Chapter 5.

⁶ Analogously, we will use the term *pitch trajectory* when the F0 estimates are converted to MIDI pitch using Equation 2.7.

2.2.3 Loudness, Dynamics, Amplitudes, and Envelopes

Loudness is a subjective perceptual property that allows tones, or sounds more generally, to be ordered from “quiet” to “loud” [53]. While aspects such as the duration of a tone and, more importantly, its pitch influence how loud a tone is perceived, this perception is mainly correlated with the *sound pressure level (SPL)*, a relative logarithmic measure of the time-averaged sound pressure expressed in Decibels (dB SPL). Sound pressure describes the deviation of the local air pressure from static or ambient pressure due to sound waves, which can be measured by a microphone and, as indicated in Section 2.1, represented by an audio signal x . Time averaging is typically done by calculating the frame-wise geometric mean of the local pressure. In signal processing, this is often called the *root mean square (RMS) envelope*⁷ of a signal. With a window function w , window size N , and hop size H , the RMS envelope for time frame m is given by

$$\text{RMS}_x[m] := \sqrt{\sum_{n=0}^{N-1} (x[n + mH]w[n])^2}. \quad (2.9)$$

Here, we assume that w has unit energy, so that we can omit normalization with the RMS value of the window function itself. To obtain an envelope e_x in dB, we now have to relate RMS_x to a reference value v_0 by calculating

$$e_x[m] := 20 \cdot \log_{10} \left(\frac{\text{RMS}_x[m]}{v_0} \right). \quad (2.10)$$

For dB SPL, v_0 is defined to be the value corresponding to a standard threshold of hearing for a 1 kHz tone (at $2 \cdot 10^{-5}$ Pascal). However, when we consider general signals, this reference is somewhat arbitrary, since recording, transmission, and reproduction can all influence the scale of the samples in x . Without specially calibrated recording and reproduction setups, it is often more convenient to normalize to a certain value range of $x[n]$. For example, $x[n] \in [-1, 1]$ is commonly used when working with audio signals in floating point representations, and the dB values calculated with $v_0 = 1$ are referred to as dB FS (full-scale). Notably, the highest value of e_x is 0 dB in this case, whereas all lower signal levels result in negative dB values. In this thesis, we use dB without any additional qualifier when the absolute scale of the signal is arbitrary or not important.

The time-domain RMS envelope is related to the magnitude variations in frequency domain by Parseval’s theorem. It states that the sum of squared magnitudes over all frequency bands is the same as the *energy* of the signal (the square of RMS_x) in the considered signal segment. Practically, we can therefore interpret the magnitude changes over time in each frequency band as the envelope for this specific frequency, where all individual “per-frequency” envelopes sum up to the overall signal envelope. We will use this property in the following Section 2.2.4 to characterize differences in timbre between tones independent of loudness.

In sheet music, a target loudness is specified only implicitly based on the related but distinct concept of dynamics. Dynamic markings are typically given for musical passages or phrases, using indications

⁷ The RMS values of windowed segments are technically not an “envelope,” since they do not outline the extremal points of the signal. However, following common engineering terminology, we use this term for the concept of a *loudness trajectory*.

such as *p* (*piano*) for playing more softly, *f* (*forte*) for playing more strongly, as well as gradual increases (*crescendo*) and decreases (*decrescendo*). Their detailed realization is largely left to the performer, allowing considerable freedom in shaping how strongly individual notes within a phrase are played. Moreover, dynamic markings are primarily relative and often do not directly correspond to absolute values of the RMS envelope. Beyond global indications, the dynamic progression within an individual note, i.e., the resulting tone envelope, can also be influenced by articulation markings such as accents or staccato. In the sheet music example in Figure 2.1a, the first note carries a *forzato* marking, and the notes are connected by a *slur*, indicating that they should be played *legato*, i.e., without clear separation. While the exact realization is left to the performer, the *forzato* should typically lead to a loudness increase at the onset of the tone, whereas the slur encourages a more continuous envelope in the later phase of the tone.

The envelope of each tone is also often characteristic for the respective instrument. For example, a piano tone is largely determined by the brief hammer impact at the onset followed by a quasi-exponential decay until the tone subsides. The RMS envelope computed from the violin example is shown in Figure 2.2b. In this visualization, 0 dB corresponds to the highest *amplitude* in the excerpt, i.e., the largest absolute value in x . The first tone starts with a short increase in the envelope, likely the realization of the *fz* prescribed in the score. Loudness then fluctuates with a similar rate as the vibrato, an expressive effect often called *tremolo*. As it is characteristic for string instruments, there is no clearly visible onset for the second tone. This second tone envelope then first increases and then decreases until the tone ends at around 45.8 seconds. Note how the RMS envelope does not drop to zero, but bottoms out at around -40 dB due to background noise present in the signal. As opposed to the first tone, the vibrato in the second tone is not accompanied by a tremolo.

2.2.4 Timbre and Harmonic Energy Distribution

Timbre is a central perceptual property of tones that is somewhat difficult to define precisely. When defined by exclusion, timbre comprises all aspects of a tone that make it distinguishable from another tone with equal duration, pitch, and loudness [194]. These aspects include, for example, the character of the onset (e.g., sharp and percussive, or soft and gradual), the distribution of energy across the frequency spectrum, and the presence of non-tonal signal components such as air flow or bowing noise.

When musicians communicate about timbre, they often rely on vague, associative terms based on analogies in other modalities, such as “dark,” “bright,” “warm,” or “cold.” While “dark” as opposed to “bright,” for example, is often used to describe tones with comparatively more energy at lower frequencies, these terms generally lack a one-to-one correspondence with specific physical or acoustic properties and are strongly context-dependent. In Western music notation, timbre is primarily specified implicitly through instrumentation, whereas most instrument-specific timbral nuances are left to the performer. Some specialized performance instructions, especially in contemporary music notation, can further prescribe certain desired timbre characteristics. For example, string players are sometimes instructed by a score to

play *sul ponticello* (bowing near the bridge) or *sul tasto* (bowing near the finger board), which is associated with “brighter” or “darker” tones, respectively.

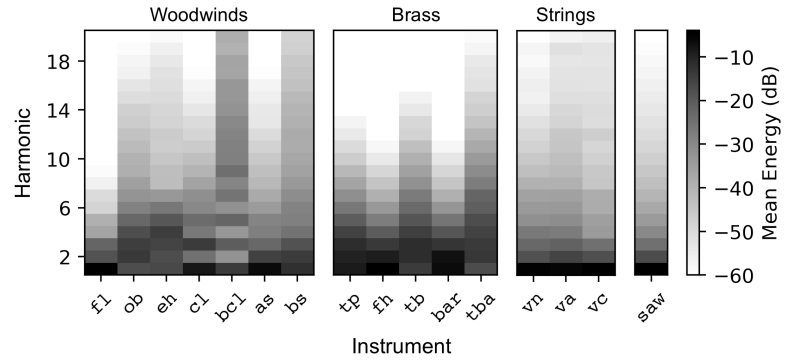
In this thesis, we adopt a narrower, signal-based perspective on timbre, focusing on the distribution of energy in the spectrum and across *partials*, an aspect that is particularly important for distinguishing musical instruments in sustained tones and contributes to perceptual phenomena such as sensory dissonance (see Chapters 5 and 6). Partials or *tonal components* of a sound can be described as individual sinusoidal oscillations, each uniquely defined by a frequency, amplitude, and phase. Together with *non-tonal* or noise-like components, these partials form the overall spectrum of a sound and determine the waveform shape. By our working definition above, tones are characterized by an (approximately) periodic waveform, in which case the *partials* are necessarily (near) integer multiples of the F_0 (see Appendix C for a proof). Throughout the following chapters, we refer to these integer-related partials as *harmonics*, distinguishing them from more general tonal components. The harmonics can be ordered by their relation to F_0 , where the *first* harmonic corresponds to the F_0 itself, the *second* harmonic to two times F_0 , and so on.

Most musical instruments and many vibrating objects produce nearly harmonic sounds. Physical constraints, such as the stiffness of a string, can introduce slight deviations from exact integer frequency ratios, resulting in *inharmonic* partials and waveforms that are no longer perfectly periodic. Such small deviations from harmonicity typically do not affect tonal perception [130], whereas sounds with strongly inharmonic spectra—e.g., those produced by certain percussion instruments such as cymbals or tam-tams—fall outside our narrow definition of tones. In this case, the mixture of partials is less likely to be perceived as a single tonal entity and the notions of pitch and related perceptual attributes become more ambiguous. These sounds therefore require separate consideration, as they differ from harmonic tones in several respects, including the perception of consonant or pleasant intervals [190]. Accordingly, throughout this work we assume tones to be harmonic unless otherwise specified, and consider deviations from harmonicity only in special cases in Chapters 5 and 6.

Using the STFT, we can analyze how the signal energy is distributed across harmonics over time. To do this, we leverage the previously obtained F_0 trajectory and, for each frame, look for spectral peaks near the integer multiples of the corresponding F_0 estimate. The exact procedure described in Section 6.2.1 requires a few additional steps to account for unvoiced frames (i.e., when no clear F_0 is detected), missing or very weak harmonics, as well as the limited resolution of the frequency bins within each frame. However, broadly speaking, the squared amplitude of the spectral peak at the n th integer multiple of F_0 provides an estimate of the energy of the n th harmonic. Normalizing these energies within each frame to sum to one yields a relative harmonic energy distribution that is independent of the signal’s overall RMS envelope e_X .

Figure 2.2c illustrates this timbre representation for the violin excerpt, showing the relative harmonic energy of the first eight harmonics over time, where darker color indicate more energy. The visualization reveals timbral differences between the two tones, where, for instance, the fifth harmonic becomes more prominent in the second tone. In addition, fluctuations within each tone are visible that appear to be modulated by the vibrato. For example, in the second tone between 45 and 46 seconds, the relative

Figure 2.3: “Harmonic fingerprint” or mean relative harmonic energy across the first 20 harmonics in a representative sample for each instrument (using data from [8] for wind instruments and from [213] for strings, averaging over the entire dataset). The sawtooth timbre *saw* is shown for comparison.



strengths of the fourth and fifth harmonics alternate, with the fourth becoming stronger when the vibrato raises the pitch and the fifth being more prominent when the pitch decreases. This observation shows that vibrato cannot be regarded as an isolated pitch effect, as it is accompanied by fluctuations in the energy of certain harmonics, a behavior that has also been observed in wind instruments [157]. We will examine this effect of the instrument’s resonances in more detail in Chapter 7.

Figure 2.3 shows the time-averaged relative harmonic energy across the first 20 harmonics for different brass, woodwind and string instruments, as well as the synthetic sawtooth waveform. To calculate this “harmonic fingerprint”, we used data from the *ChoraleBricks* dataset [8] for wind instruments and *Virtuoso Strings* [213] for string instruments and calculate the mean over the time axis in a representation similar to Figure 2.2c.⁸ These fingerprints reveal characteristic patterns across instrument families. For example, the flute (fl) shows the highest concentration of energy in the first harmonic. Instruments of the clarinet family (B_♭ clarinet, cl, and bass clarinet, bcl) exhibit strongly attenuated even harmonics, a consequence of the vibration characteristics of the air column in the cylindrical instrument body [52]. Among brass instruments, those with a *conical* bore (flugelhorn, fh, and baritone, bar) can be distinguished from those with a *cylindrical* bore (trumpet, tp, and trombone, tb), with the latter exhibiting a “brighter” timbre characterized by an increased concentration of energy in higher harmonics. More generally, instruments in the bass register (e.g., bass clarinet, bcl, baritone saxophone, bs, or tuba, tba) tend to have stronger higher-harmonic components, as the absolute frequencies of these harmonics are much lower than those of treble instruments.

However, as already evident from Figure 2.2c, the variation in relative harmonic energy *within* an individual instrument can be substantial. Consequently, the harmonic fingerprint can capture large-scale trends but may obscure perceptually relevant nuances in an instrument’s timbre. To illustrate this variability, Figure 2.4 shows the distribution of relative harmonic energy across all frames used to compute the

⁸ *ChoraleBricks* is musically very homogeneous, with a limited pitch range, few dynamic variations, and simple melodies, whereas *Virtuoso Strings* is musically more varied, with more virtuosic playing. This may also affect the average timbre, so that the differences between the string and wind instrument groups (also in Figure 2.4) should be interpreted with caution.

We furthermore verified the accuracy of this analysis through resynthesis (see Section 2.4.1) by perceptually comparing the original and resynthesized signals by listening to a subset of the data. Although some inaccuracies occur particularly at higher harmonics, e.g., due to noise in the original signal, the synthesized signals largely preserve the characteristic timbre of the instruments (see, e.g., Figure 2.7).

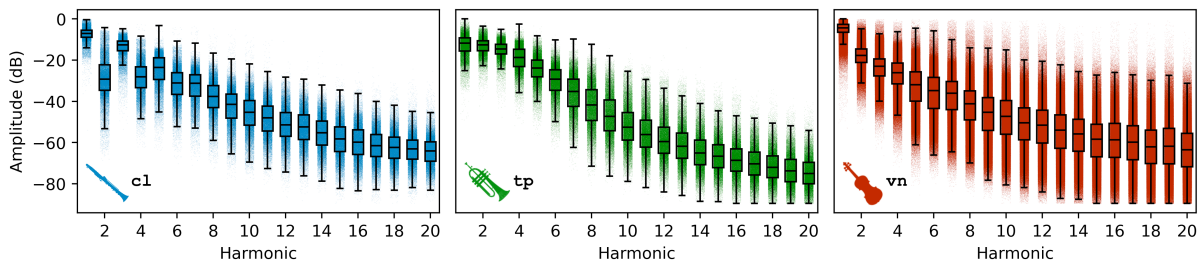


Figure 2.4: Distribution of the relative harmonic energy in the first 20 harmonics of clarinet (c1, blue), trumpet (tp, green), and violin (vn, red). Each colored dot is a single measurement. The black box plots indicate median, first and third quartile, as well as the overall range excluding outliers. The statistics for c1 and tp are based on 34764 frames, for vn on 348876 frames.

fingerprints in Figure 2.3, for clarinet (c1, blue), trumpet (tp, green), and violin (vn, red). For the clarinet, a clear attenuation of the second and fourth harmonics remains visible, but these harmonics also exhibit higher variance than the first and third, indicating that the degree of attenuation varies across tones with different pitch and dynamic levels. In general, lower harmonics that have been shown to be especially important for timbre discrimination [79], appear comparatively less variable in Figure 2.4, particularly for the trumpet. On the other hand, higher harmonics are often more than 30 dB below the strongest components, making them perceptually less salient. Finally, it should be noted that this visualization does not capture co-occurrence patterns between harmonics (e.g., whether a stronger third harmonic implies a stronger fifth in the clarinet).

Overall, these results suggest that a perceptually accurate reconstruction, adaptation, or synthesis of music signals requires modeling the highly variable relative energy distribution across harmonics, which is important both for distinguishing instruments and for capturing the timbral nuances in musical performances. Together with noise-like components—which we will consider in more detail in Section 2.4.2, as well as Chapters 4 and 7—these aspects cover a large part of a tone’s technical characteristics beyond timing, pitch, and loudness.

2.3 The Singing Voice

So far, we have not considered the arguably most important musical instrument: the singing voice. Widely used around the world for centuries, it is special in three main aspects. First, lyrics add an entirely new layer of musical expression. Through words, a singer can complement, explain, or contradict the musical ideas that are abstractly conveyed through the tones. Second, even when we disregard lyrics, the voice is the most versatile instrument in terms of expression. For example, it has a wider timbral range than any other instrument, where tones can be described as sweet, harsh, strong, weak, light, dark, and so on. Vocal techniques such as *lip trills*, *vocal fry*, or *growling*, further extend this repertoire. Finally, the voice is intimately familiar to listeners. From lullabies [233] to funeral dirges [171], singing is ubiquitous and

shapes our relationship to others, so that not only experts and musically trained listeners have a clear expectation of how a voice should or should not sound like.

The study of the singing voice (not to speak of its non-musical counterpart speech) is a large and active research field, and we refer to relevant textbooks [34, 206] and overview articles [82, 98] for a comprehensive treatment. For the purpose of this thesis, we focus on selected aspects that influence the tone attributes discussed in Section 2.2, and illustrate some distinctive features of the singing voice with two examples shown in Figures 2.5 and 2.6. The examples are taken from the *VocalSet* dataset, a collection of isolated recordings of professional singers using different vocal techniques. Figure 2.5 shows a C major scale from C₄ to D₅ and back sung by a female soprano singer (f3 in the dataset), using the vowel /a/ throughout and being instructed to sing with a “straight tone.” This is contrasted in Figure 2.6 with the same scale sung by the same singer, but on the vowel /i/ and with the instruction to sing with “vibrato.”

Pitch. A key characteristic of the singing voice is its ability to vary pitch continuously, while little to no absolute pitch reference is available to the performer. A violin, for example, also enables continuous pitch variation, but provides tactile feedback through the hand position on the fingerboard. Trained singers may know how a certain pitch range “feels” but this perception is generally much less accurate. In Figure 2.5, the singer starts the first tone with a median F₀ of 263 Hz, 10 cents higher than the nominal frequency of the C₄ (MIDI pitch 60). In the second tone, the median F₀ deviates from MIDI pitch 62 by 50 cents. While some notes in between are closer to the respective MIDI pitch, the C₅ (MIDI pitch 72) around 4.5 seconds is almost 80 cents higher and therefore closer to a C_♯₅. Yet, when listening to the recording, it can still be clearly identified as a major scale. Furthermore, even in the “straight tone” condition, the pitch variability within tones is also larger than for many other musical instruments. For example, we can observe a short pitch rise at the onset of many tones, portamenti (i.e., pitch glides between notes) especially around 8 seconds, and some vibrato. By comparison, the vibrato in Figure 2.6 is much more pronounced with a range of more than two semitones (200 cents) in the last tone.

Loudness. The overall dynamic range of the human voice is in the range of 70 dB [210], depending on pitch register and other physiological factors. While this range is comparable to that of a grand piano, trained singers can also control loudness *within* individual tones to produce effects like *messa di voce* [18], a crescendo and decrescendo without changes in pitch or timbre. In the examples shown in Figures 2.5 and 2.6, no dynamics instructions were given to the singer. In Figure 2.5, the RMS envelope is relatively stable within individual tones but varies across tones, with the highest pitch being sung the loudest. In Figure 2.6, the envelope exhibits a more gradual rise and fall that broadly follows the pitch contour. While the vibrato introduces some fluctuations in signal amplitude, these variations are not regular or periodic like deliberate tremolo.

Timbre. The timbre of the singing voice enables the perception of lyrics, primarily through vowels and the noise-like components added by consonants. Beyond this linguistic function, vocal timbre is highly variable and expressive, with the physiological mechanisms underlying vowel production playing a central role. Vowels are shaped by formants, i.e., resonances of the vocal tract that can be controlled through the

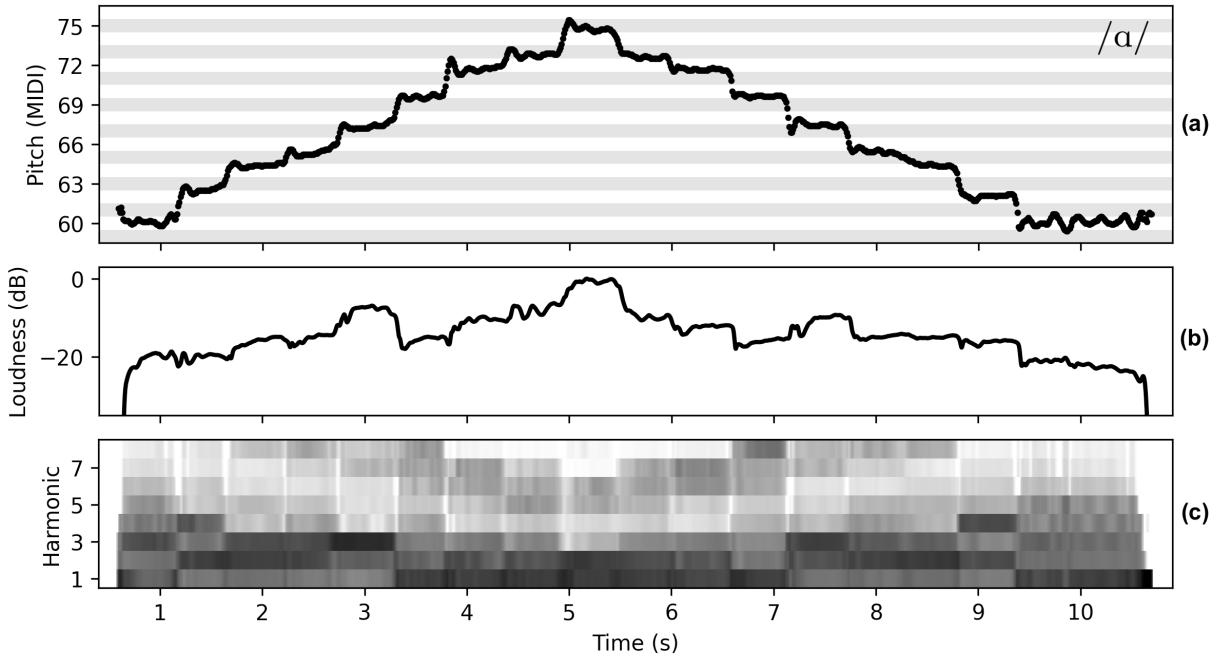


Figure 2.5: Tone attributes extracted from a recording of a female singer (f3 from [229]) singing an ascending and descending C major scale on the vowel /a/, instructed to sing with a “straight tone.” (a) F0 trajectory from [46] (estimated with pYIN [115] and manually corrected). (b) Signal envelope using a windowed RMS algorithm with $N = 2048$. (c) Frame-wise relative energy of the first eight harmonics extracted using an F0-informed peak picking algorithm [182].

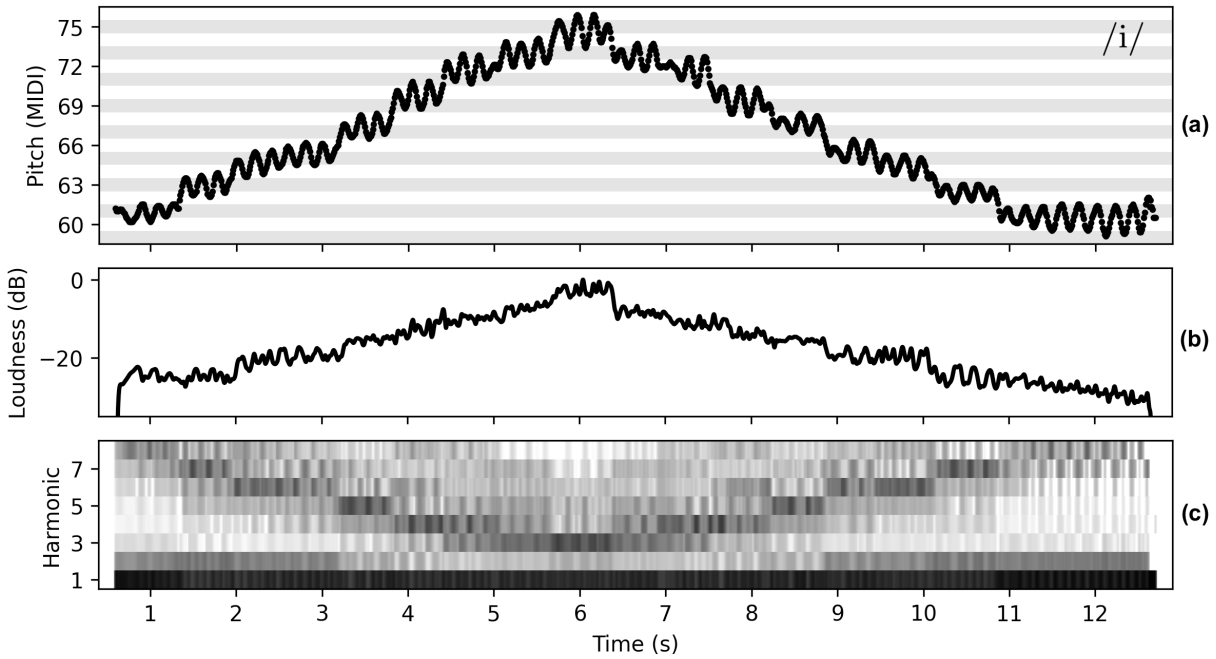


Figure 2.6: Tone attributes extracted from a recording of a female singer (f3 from [229]) singing an ascending and descending C major scale on the vowel /i/, instructed to sing with “vibrato.” (a) F0 trajectory from [46] (estimated with pYIN [115] and manually corrected). (b) Signal envelope using a windowed RMS algorithm with $N = 2048$. (c) Frame-wise relative energy of the first eight harmonics extracted using an F0-informed peak picking algorithm [182].

configuration of the tongue and palate. These resonances selectively amplify certain frequency regions while attenuating others and are largely independent of the pitch of the vocalization. Formants are typically characterized by the center frequency of this resonance. For an average speaker, the first three formants of the vowel /a/ are located around 700, 1100, and 2500 Hz, whereas for /i/ they lie around 300, 2200, and 2900 Hz [206]. Trained singers, however, continuously adapt formant positions to achieve musically appropriate tone quality, facilitate voice blending in ensemble contexts, and enhance vocal projection, for example through the use of the so-called singer’s formant [205]. To illustrate the interaction between our measure of relative harmonic energy and vocal tract resonances, consider the first tone in Figure 2.5. The nominal pitch is C₄, corresponding to a fundamental frequency of 262 Hz, such that the harmonics occur at approximately 262, 524, 786 Hz, and so on. In this tone, the third harmonic is the strongest component, being close to the first formant of /a/. As the pitch increases in subsequent tones, lower harmonics move in and out of this resonance. Furthermore, the third formant of /a/ around 2500 Hz becomes relevant between approximately 2 and 9 seconds. In the example shown in Figure 2.6, on the other hand, the eighth harmonic in the first tone is close to the second formant of /i/ at around 2200 Hz, then moving out of the resonance as F₀ increases. Notably, the pronounced vibrato in this example also leads to rapid changes in the harmonic energy distribution, since the vocal tract resonances remain approximately fixed while the harmonic frequencies vary. This timbre adds to the perceptually rich effect created by the vibrato, similar to the violin example in Figure 2.2.

2.4 Music Signal Synthesis

In this section, we provide an overview of methods for synthesizing an audio signal from a parametric description of tones as described in Section 2.2. In particular, we focus on two approaches that play a central role in the following chapters and are widely used in DDSP [72]: additive synthesis (Section 2.4.1) and the source–filter model (Section 2.4.2). Section 2.4.3 then discusses the advantages and limitations of common deep learning approaches for synthesizing tonal signals. For a broader overview of musical sound synthesis techniques based on signal processing, we refer to [30].

2.4.1 Additive Synthesis

The goal of additive synthesis is to reproduce tonal components in a signal through the superposition of sinusoids. A sinusoid is a simple periodic function that depends on three parameters: an amplitude $a \in \mathbb{R}_+$, frequency $f \in \mathbb{R}_+$ in Hz, and an initial phase $\varphi \in [0, 2\pi]$. A discrete audio signal x_{sin} with sampling rate F_s based on this sinusoid can therefore be synthesized by

$$x_{\text{sin}}[n] = a \sin(2\pi f n / F_s + \varphi). \quad (2.11)$$

From a perceptual perspective, a mainly controls the loudness of the sinusoid (also called a *pure tone*), while f determines its pitch. In this isolated case, the phase parameter φ is perceptually irrelevant, as it only shifts the relative positions of peaks and troughs in the waveform.

Since we typically have more than one partial in a tone, we have to combine multiple sinusoidal oscillations in a single signal, which can be done additively (hence the name *additive synthesis*). For R sinusoids with amplitude a_r , frequency f_r and phase φ_r , the resulting signal x_{add} is given by

$$x_{\text{add}}[n] = \sum_{r=0}^{R-1} a_r \sin(2\pi f_r n / F_s + \varphi_r). \quad (2.12)$$

Here, the phase parameters φ_r can, in principle, become perceptually relevant. In particular, when sinusoids with similar or equal frequencies are superimposed, their relative phases determine the alternation between constructive and destructive interference, resulting in the perceptual phenomenon known as beating (see Chapter 6 for a detailed discussion of this effect). For strictly harmonic tones, however, the relative phase relationships between partials have little perceptual impact. As a result, the phase parameters are often neglected in practice, for example in parameter estimation or analysis-by-synthesis applications. With strictly harmonic signals, the model in (2.12) can be further simplified by constraining the sinusoidal frequencies to integer multiples of a common fundamental frequency f_0 , i.e., setting $f_r = r f_0$. This further reduces the number of free parameters and reflects the underlying structure of harmonic sounds. In the discrete-time formulation, the condition $f_r < F_s/2$ for all r must be satisfied to prevent aliasing.

We have to account for one more complication in our additive synthesis model. In any nontrivial music signal, amplitudes and frequencies of all partials are varying over time, producing different tones with vibrato, crescendo, timbre variations, etc. Let $\mathbf{a}_r \in \mathbb{R}_+^M$ be the amplitudes of the r th partial for M frames⁹. Then we can calculate an *instantaneous amplitude*

$$\tilde{a}_r[n] := \mathcal{I}[n; \mathbf{a}_r], \quad (2.13)$$

where $\mathcal{I}$ denotes an *interpolation* operator that maps the sample index n to an amplitude value that is calculated based on frame-wise amplitude values in $\mathbf{a}_r$, using the frames in the vicinity of n as support points. A simple choice for $\mathcal{I}$ is nearest neighbor interpolation, in which the amplitude of the temporally closest frame is used. While computationally inexpensive, this approach could lead to large changes in amplitude from one sample to the other, resulting in audible artifacts because of discontinuities in the waveform. When the frame spacing is sufficiently small, linear interpolation typically provides a satisfactory compromise between smoothness and computational complexity. For larger frame spacings, corresponding to fewer support points per unit time, higher-order interpolation schemes such as spline-

⁹ Representing time-varying parameters on a frame-wise basis instead of per sample dramatically reduces the amount of data required, while changes in the parameter trajectory at the sample level are typically not perceptible anyway. A frame-wise representation is also advantageous for real-time processing and deep learning, as computations can be pooled efficiently on the frame level.

based methods may be more appropriate. For the purposes of this thesis, we use linear interpolation throughout. Analogously, we calculate an *instantaneous frequency*

$$\tilde{f}_r[n] := \mathcal{I}[n; \mathbf{f}_r] \quad (2.14)$$

for frame-wise frequencies $\mathbf{f}_r \in \mathbb{R}_+^M$.

The concept of an instantaneous frequency, however, is somewhat limited in practice, since a frequency cannot be meaningfully defined over a time interval shorter than its corresponding wavelength. To properly account for time-varying frequency trajectories, it is therefore necessary to formulate sinusoidal synthesis in terms of the *instantaneous phase* $\tilde{\phi}_r$, which ensures a continuously evolving oscillation. $\tilde{\phi}_r$ can be obtained by integrating over all previous instantaneous frequencies, i.e., by accumulating the frequency values $\tilde{f}_r$ up to the current sample. For the r th partial, this yields

$$\tilde{\phi}_r[n] = \frac{2\pi}{F_s} \sum_{s=0}^n \tilde{f}_r[s] + \varphi_r, \quad (2.15)$$

where φ_r denotes the initial phase. The factor $2\pi/F_s$ normalizes $\tilde{\phi}_r$ to the proper period length in samples, depending on the sampling rate F_s .¹⁰ Together with $\tilde{a}_r$, we can now synthesize a signal x_{tv} consisting of R sinusoids whose amplitudes and frequencies are time-varying with

$$x_{\text{tv}}[n] = \sum_{r=0}^{R-1} \tilde{a}_r[n] \sin(\tilde{\phi}_r[n]). \quad (2.16)$$

It is important to note that both the instantaneous amplitude and instantaneous frequency are assumed to vary slowly relative to the current value of $\tilde{f}_r$. Faster variations result in effects commonly described as Amplitude Modulation (AM) or Frequency Modulation (FM), which can be viewed as separate synthesis methods. Especially with FM [26], where the instantaneous frequency of a *carrier* oscillation is modulated by other periodic functions, complex harmonic and inharmonic sounds can be generated with very few computational operations. However, controlling the timbre of tonal sounds can be challenging because small parameter changes can lead to large spectral differences. While this synthesis method has also been explored in the context of DDSP [21], we do not consider it further in this thesis.

An example for additive synthesis is shown in Figure 2.7. In particular, we used the frame-wise F0, RMS envelope, and relative harmonic energy from Figure 2.2 to set values for $\mathbf{a}_r$ and $\mathbf{f}_r$. We further set $\varphi_r = 0$. It can be seen that the synthesized signal faithfully reproduces most features of the harmonic components in the recording. Artifacts from discontinuities in the amplitude and frequency trajectory can be seen between the tones around 44.25 seconds and at the end of the second tone around 46 seconds. To avoid

¹⁰ Note that $\tilde{\phi}$ can become very large when n increases, since we are summing up the phase evolution since the beginning of the signal. If this is a numerical concern, e.g., due to limited floating point precision, $\tilde{\phi}$ can be restricted to the range $[0, 2\pi]$ with a modulo operation.

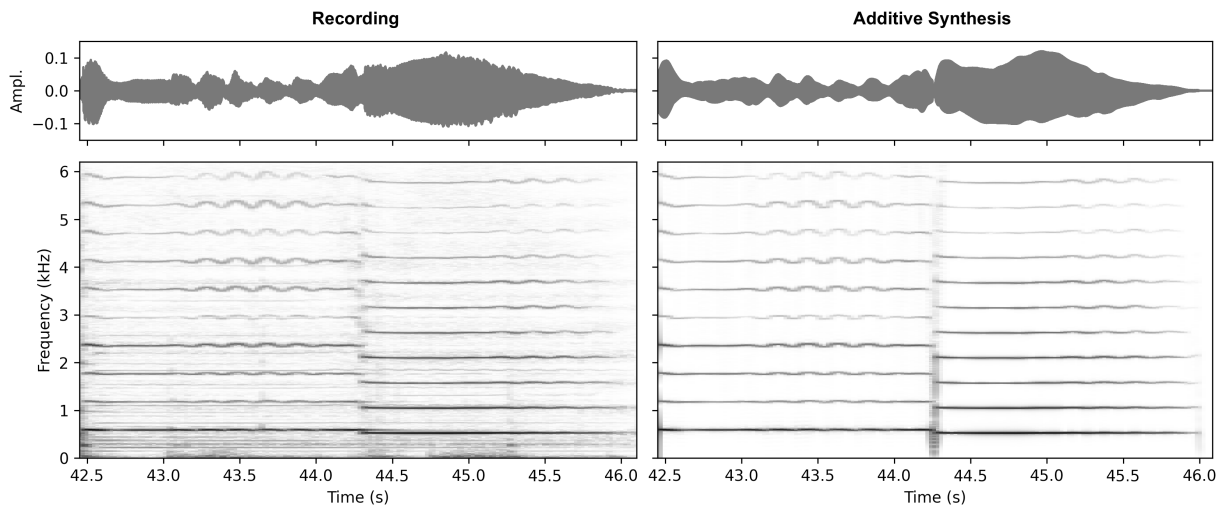


Figure 2.7: Comparison of recorded violin tones (left) with a signal created via additive synthesis (right). Analogously to Figure 2.1, we show the time-domain waveform (top) and a spectrogram with logarithmic magnitude compression (bottom).

this, further pre-processing of $\mathbf{a}_r$ and $\mathbf{f}_r$ is required, for example by smoothing the trajectories with an averaging filter applied over time. Furthermore, the synthesized version is missing all non-harmonic components of the recording, which, in this particular case, consists of the noisy sounds of the violin itself, but also the crosstalk from the other instruments which is visible in the spectrogram as faint horizontal lines between the harmonics.

2.4.2 Source–Filter Models

An alternative to additive synthesis is to do the opposite: starting from a signal with a broad spectrum and selectively attenuating or amplifying certain frequency components to create the desired sound. This approach is sometimes referred to as *subtractive synthesis* in contrast to its additive counterpart. In the following, however, we adopt the more general perspective of *source–filter models*, emphasizing the separation of sound generation and spectral shaping as successive steps in a signal processing pipeline. Source–filter models are typically introduced in the context of speech production, where the vibrations of the vocal folds act as a source signal and the resonant properties of the vocal tract (i.e., the throat and mouth) are described as a filter. Beyond that, the source–filter model is a powerful method for describing a wide range of sound-producing systems, including musical instruments. In wind instruments, the source is usually a vibration created by the lips or a reed, and the filter corresponds to the resonant air column of the instrument body. In string instruments, the source may be the string displacement caused by a bowing motion, while the filter captures the resonances of the string and the instrument body. In percussion instruments, the source can be the impact of a mallet or other object, while the drumhead and drum body together act as the filter.

More generally, the source signal (in physical modeling contexts also referred to as *excitation signal*) can be a simple periodic waveform such as a sawtooth wave, square wave, or pulse train, but also a non-tonal signal like noise or a single impulse. More detailed source models are also possible, such as explicit physical models of the vocal folds [47], the interaction between bow and string in bowed instruments [119], or the hammer–string interaction in the piano [16]. Note that in different scenarios, the physical accuracy of the source as well as the scope of the filter can differ substantially. In a more physically inspired model, the entire characteristics of a vibrating string, including the mechanisms that lead to periodic oscillations, can be part of the filter [11]. In more signal-oriented models, on the other hand, we can already start with a harmonic source signal and only model the coloration through instrument body resonances (see, e.g., Chapter 7). For representing the filter part in a signal processing system, there are generally two types of filters available, finite impulse response (FIR) and infinite impulse response (IIR) filters, which we will describe separately in the following.

An **FIR filter** is a linear time invariant (LTI) system whose output depends only on a finite number of past input samples. As a result, an FIR filter can be fully described by a finite set of coefficients, either in the time domain or equivalently in the frequency domain. A filter of *order* L is applied to an input signal x by convolution with $L + 1$ time-domain coefficients b_i , such that the n th sample of the output signal y is given by

$$y[n] = \sum_{i=0}^L b_i x[n-i]. \quad (2.17)$$

This operation is equivalent to an element-wise multiplication of the input signal and the filter in the frequency domain.¹¹ The order of the filter determines the frequency resolution of the impulse response. In other words, if we want fine-granular control over which frequencies are attenuated or amplified, we need a longer filter. A common way to design an FIR filter is to specify its desired characteristics in the frequency domain, for example by defining a target magnitude response. In practice, this is often done at a limited number of logarithmically spaced frequency bands (such as octave or one-third–octave bands), and the full, linearly spaced magnitude spectrum of the filter is then obtained by interpolation.¹² After designing this magnitude spectrum, a suitable phase response must be added. A *zero-phase* filter (i.e., one that is real-valued in frequency domain) is not *causal*, i.e., it requires future input samples to calculate the current output sample. Simply shifting this zero-phase filter in time domain, so that it only requires past input samples, leads to a *linear-phase* filter, where all frequencies are delayed by a constant time, corresponding to a larger phase shift towards higher frequencies. A common alternative is to create a *minimum-phase* filter, where different frequency components experience different delays, but the overall delay introduced by the filter is minimized [197].

¹¹ To be precise, frequency-domain multiplication is equivalent to *circular* convolution, so that we have to account for wrap-around effects through zero-padding of the input signal. For a comprehensive discussion of the issues that arise when applying FIR filters in the frequency domain, we refer to [223].

¹² Although simple linear interpolation can be used for this purpose, smoother magnitude responses generally lead to more well-behaved filters. For that reason, higher-order interpolation is often preferred. In particular, *piecewise cubic Hermite interpolating polynomials* (PCHIP) [54] are often a good choice for this task because they produce a smooth overall shape without introducing overshoot between the specified frequency points.

An **IIR filter**, in contrast, makes use of previous output samples and therefore has an impulse response of theoretically infinite length. The feedback structure allows for achieving fine-grained spectral characteristics with fewer coefficients than a comparable FIR filter. At the same time, this efficiency comes with trade-offs, including a more involved design process, the possibility of stability issues, and generally stronger phase distortions. IIR filters are often represented through the coefficients a_i and b_i of the time-domain equation

$$y[n] = \sum_{i=0}^L b_i x[n-i] + \sum_{i=1}^Q a_i y[n-i], \quad (2.18)$$

where x is the input signal to the filter and y is the filtered output signal. L is the *feed-forward* filter order (corresponding to the filter order of an FIR filter that has only a feed-forward part) and Q is the *feed-back* filter order. Determining these coefficients based on a desired spectral response is not straightforward, although closed-form solutions are available for some standard cases. Examples include low-pass, high-pass, and band-pass filters based on the *Butterworth* design [140], as well as simple peak filters with a single dominant resonance. We refer to relevant textbooks [140, 197] for a comprehensive account of filters and their characteristics.

While both FIR and IIR filters are LTI systems in this basic form, time-varying characteristics can be modeled by changing the filter coefficients b_i and a_i over time. As in additive synthesis where amplitudes and frequencies are time-dependent, such coefficient changes can introduce discontinuities in the output signal. These can be mitigated either by ensuring that the coefficient updates from one frame to the next are sufficiently small, or by cross-fading between the output signal produced by the previous filter and that produced by the updated filter.

As an example, Figure 2.8 shows two different source signals filtered with the same time-invariant FIR filter. The source signal on the left in Figure 2.8a was obtained by synthesizing a band-limited sawtooth waveform using the F0 contour from the original violin example in Figure 2.1, and then scaling it by the original RMS envelope. For this illustrative example, the filter was constructed by computing the average spectrum of the violin recording and smoothing it by convolution with a Hann window of length 51 (resulting in a window bandwidth of around 550 Hz). From this smoothed magnitude response, we derived a minimum-phase FIR filter, which was then applied to the source signal using frequency-domain multiplication. Although this simple filter does not accurately capture the resonant characteristics of a violin, the filtered output (shown on the right in Figure 2.8a) already exhibits some perceptual similarity to a violin sound.

A key advantage of the source–filter model is its flexibility, for example, allowing to replace the source signals. As a simple example, Figure 2.8b shows how the same filter as above is applied to white noise that has been scaled according to the RMS envelope of the violin recording. The spectrogram of the output signal shows more clearly that certain frequencies are attenuated while others are amplified. In practice, these two types of sources could also be combined to account for both the harmonic and noise-like components of a violin signal. A convincing synthesis, however, would likely require two

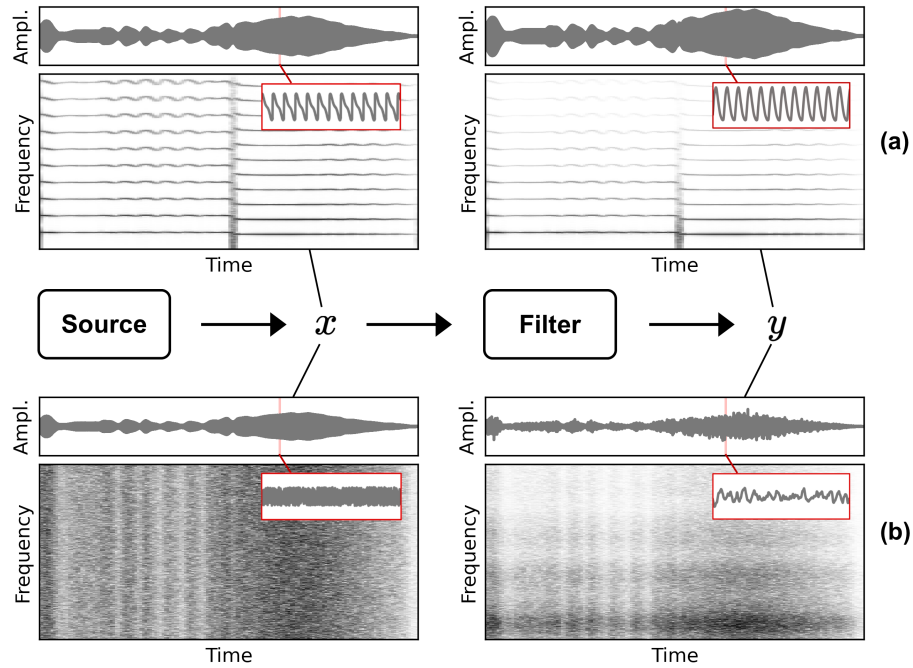


Figure 2.8: Process of source-filter synthesis illustrated with the violin example from Figure 2.1. Each signal view shows the waveform (top) and the spectrogram (bottom), plus a zoomed in section of the waveform (outlined in red). **(a)** Using a sawtooth waveform as the source signal x . **(b)** Using white noise as the source signal x . Both sawtooth and white noise are filtered with the same time-invariant filter.

separate, and potentially time-varying, filters to capture the instrument characteristics. Estimating such filters is considerably more challenging than the simple approach used in this example. Although a prior decomposition of the signal, for example using harmonic-percussive separation [49, 50], could enable a separate analysis of noise-like and tonal components, these approaches are not robust across different input signals and require careful tuning of the analysis parameters. In this context, additive and subtractive synthesis can also be combined, using each method to its respective strengths. This *Spectral Modeling Synthesis (SMS)*, proposed by Serra and Smith [186], models partials using additive sinusoids, while residual or transient components are captured by a filtered noise model. This combination of synthesis methods also forms the basis of the original DDSP synthesis model proposed by Engel et al. [42].

2.4.3 Deep Learning

Advances in deep learning have substantially expanded the range of methods available for audio and music signal synthesis. Recent generative models can produce audio signals conditioned on a wide variety of inputs, including symbolic representations, continuous control parameters, and text descriptions of the desired output. In the specific context of monophonic music signals, prominent examples include autoencoder-based models that learn compact latent representations of a variety of instrumental sounds [44, 45], approaches for expressive instrument synthesis from enriched symbolic inputs such as MIDI

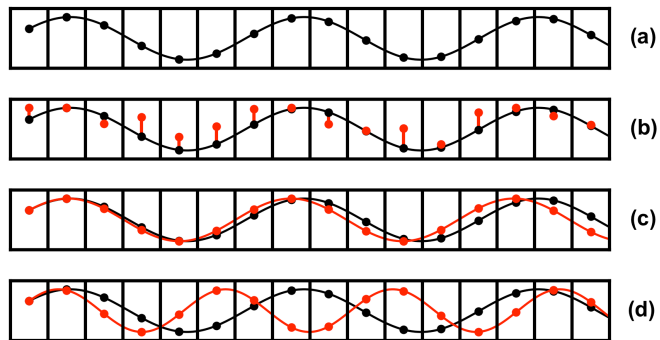
Figure 2.9: Illustration of different approaches of learning to resynthesize a given sinusoid signal.

(a) Target signal of length $N = 16$ at $F_s = 500$ Hz. The sinusoid has a frequency of 80 Hz.

(b) An unconstrained synthesis model can approach the target sinusoid with each sample individually. Output samples are shown in red compared to the target signal in black.

(c) A constrained synthesis model can only output sinusoids. The output signal (in red) has a frequency of 84 Hz.

(d) Like (c), but the output signal has a frequency of 112 Hz.



[95, 96], as well as the active field of singing voice synthesis (SVS) [84, 238] (see [145] for a recent review). In principle, there is no technical reason to restrict such models to monophonic signals. Consequently, similar systems are also used to generate mixtures of multiple overlapping sounds and complex music signals, ranging from commercially successful text-conditioned popular music generation (see, e.g., *Suno*¹³, *Udio*¹⁴ or *Producer.ai*¹⁵) to research-oriented applications in orchestral and multi-instrument ensemble music synthesis [70, 112].

In generative music synthesis models, the actual waveform is typically generated using one of two architectural principles. In the first approach, time domain audio signals are synthesized directly from a latent or control input, e.g., *auto-regressively* in the time domain on a sample-by-sample basis, as originally introduced with *WaveNet* [215]. In the second approach, the task is split between two separate neural networks: an *acoustic model* predicting a non-invertible time–frequency representation, and a *neural vocoder* converting this representation to a time domain signal. With this separation, the task of the acoustic model is simplified. By omitting the phase component and/or by using a coarse frequency resolution such as a mel spectrogram, intricate aspects of the time domain signal, including phase coherence across frames, do not need to be modeled at this stage. The vocoder, on the other hand, can be trained separately to either reconstruct the signal auto-regressively or by restoring an invertible time–frequency representation from the acoustic model output (see, e.g., [70, 100, 132]). For the purposes of this discussion, we view the acoustic model and vocoder jointly as a single synthesis system.

As discussed in Chapter 1, generative models for music synthesis offer substantial flexibility with respect to both control (or *conditioning*) inputs and target sound characteristics. At the same time, they are typically computationally demanding, require large amounts of training data to consistently produce convincing outputs, and are difficult to constrain effectively using nuanced control parameters. One underlying conceptual reason for these limitations can be illustrated with the simplified toy example shown

¹³ <https://suno.com/>

¹⁴ <https://www.udio.com/>

¹⁵ <https://www.producer.ai/>

in Figure 2.9. Assume the task is to synthesize a pure sinusoid signal with unit amplitude and a fixed but unknown frequency. The example shown in Figure 2.9a corresponds to a sinusoid with a frequency of 80 Hz at a sampling rate of $F_s = 500$ Hz. If we consider a general generative model, there are typically no explicit constraints on the relationship between individual output samples. Such an *unconstrained model* has a clear advantage: If the training objective is to minimize the distance between the target and output signals, the model can smoothly reduce this distance by approaching the target with each output sample individually (illustrated in Figure 2.9b). When trained on a sufficiently large set of examples and number of trainable parameters, the model may therefore learn to approximate the statistical properties of sinusoids. However, the output is not guaranteed (and not likely) to form a perfectly sinusoidal waveform. For audio signals, even small deviations may be perceived as noise or other artifacts. Furthermore, for more complex signals, both the amount of training data and the model capacity necessary to learn the statistical properties increase substantially, which directly adds to the data and computational burden mentioned above.

To address these limitations of unconstrained models, it can be beneficial to explicitly constrain the output signal space of a model. In the context of the toy example considered here, this corresponds to constraining the model output to perfect sinusoids. Imposing such structure is a central idea behind DDSP [42, 72]. By using a sinusoid model, the output signal is guaranteed to be periodic, even if the parameter estimates (e.g., the sinusoid frequency), are not exactly correct, as illustrated in Figure 2.9c for a frequency mismatch of 4 Hz. This way, a small model output error likely leads to a perceptually similar signal rather than introducing noise or other artifacts. More generally, embedding prior knowledge about the expected signal in the model simplifies the learning problem, reducing both the amount of training data and the number of trainable parameters required, since the model no longer needs to infer the overall waveform shape from data alone. On the other hand, this increased structure also introduces new challenges. In particular, the distance between individual samples changes non-linearly with varying frequency. Small changes in frequency can cause some samples to move closer to the target while others move further away, as illustrated in the comparison between Figure 2.9c and d. This behavior may complicate optimization with standard stochastic gradient descent-based methods and can make training less stable, leading to less smooth loss landscapes overall. This particular issue in the context of sinusoids and general periodic signals will be the main subject of the upcoming Chapter 3.

3 Multi-Scale Spectral Loss Revisited

This chapter is based on [176]. I was the primary contributor to the study design, implementing and conducting the experiments, and preparing the manuscript. Meinard Müller supervised the work and contributed throughout all stages of the research and manuscript preparation.



The multi-scale spectral (MSS) loss is commonly used to compare audio signals for analysis-by-synthesis approaches and training generative models, as it provides a good trade-off between temporal and spectral resolution. However, some configuration choices, including window type and size, magnitude compression, as well as the distance between spectrograms, are often set without explicit consideration, even though they can significantly impact the loss properties and convergence behavior during training. Particularly in the context of differentiable digital signal processing (DDSP), where learned parameters may explicitly control the frequency of synthesis components, the MSS loss often fails to provide smooth loss landscapes. In this chapter, we want to gain a better understanding of how different configurations of the MSS loss affect this problem. As an illustrative example, we consider the task of sinusoid frequency estimation via gradient descent to compare different configurations and their effect on the loss properties. Based on these insights, we show that favorable configurations can facilitate unsupervised training of more complex DDSP additive synthesis autoencoders and have a positive effect on the latent space structure in a Variational Autoencoder (VAE). Our results indicate that a careful choice of configuration and the resulting smoother loss landscape may benefit many applications where the MSS loss is utilized.

This chapter is structured as follows. After introducing the problem setting in Section 3.1, we define a generalized version of the MSS loss and its associated parameters in Section 3.2. Section 3.3 examines the effects of window type and length, magnitude compression, and the choice of distance function in the context of sinusoidal frequency estimation via gradient-based optimization. In Section 3.4, we describe experiments where a set of representative MSS loss configurations is compared in different machine learning scenarios. Finally, Section 3.5 summarizes our main findings and discusses their broader implications for DDSP and model-based machine learning.

3.1 Background

A multitude of machine learning tasks require a loss function to compare audio signals, including many end-to-end approaches for sound, music and speech synthesis. *Spectral* loss functions are among the most commonly used distances between audio signals and rely on an element-wise comparison of magnitude spectrograms, which can be computed from time-domain signals using the STFT. This way, signals are compared in terms of the temporal and spectral distribution of signal energy, which better correlates with human perception than, for example, the numerical similarity of waveforms [133]. The spectrogram, however, is limited by the fundamental trade-off between time and frequency resolution of the STFT and thus—without phase information—cannot achieve high temporal and spectral accuracy at the same time. This trade-off can be mitigated by comparing multiple spectrograms with different time–frequency resolutions in a combined loss function [222, 232], so that signals must conform at all these resolutions simultaneously to minimize the loss.

This concept of a multi-scale spectral (MSS) loss is used extensively in the context of differentiable digital signal processing (DDSP) [42, 72]. DDSP was introduced as an umbrella term for the concept of back-propagating gradients through fixed DSP components, which allows for including domain knowledge (e.g., about the physics of a generative process) in model architectures to restrict them in a meaningful way (*inductive bias*). The DDSP paradigm has recently proven useful for various tasks in audio signal processing and MIR, including F0 estimation [43], musical source separation [175], as well as estimating parameters for piano synthesis [155] and singing voice synthesis [230] or artificial reverberation [105].

All these applications use a variant of the MSS loss, but in most approaches, certain DSP parameters cannot be estimated simply by comparing the target and output audio signals. In particular, the MSS loss has been shown to be highly irregular and non-convex for parameters that directly or indirectly control the frequency of tonal synthesis components [21, 71, 214]. As a result, optimization methods like stochastic gradient descent are unlikely to converge reliably without additional information during training or inference. This information can for example be provided through *self-supervision*, a term which in the DDSP context is used for training on synthetic data with known parameters [43, 114]. Such a training setup makes it possible to train using parameter-level loss functions, but can introduce domain shift issues when real audio signals are used at inference time. Alternatively, many DDSP approaches incorporate external information sources, such as a separate pitch estimation [42, 175], increasing the overall complexity of the system. While randomizing configurations of the STFT has been proposed to improve training robustness [199], to our knowledge, no systematic analysis of the influence of different configurations on the loss behavior has been presented.

In this chapter, we use the specific DDSP scenario of sinusoidal frequency estimation via gradient descent as a simple scenario for explicitly analyzing the MSS loss and its behavior in detail. We show that a suitable configuration leads to a smoother *loss landscape* with gradients that allow for a more robust convergence to the true frequency parameter of a sinusoid (we call those *informative* gradients in the

following), where a significant part of both the favorable and the unfavorable loss characteristics can be ascribed to the effects of spectral leakage. Some loss configurations may amplify unfavorable aspects, so that, for example, the choice of window type and size, magnitude compression or spectrogram distance can strongly influence convergence behavior.

Based on this analysis, we identify three representative loss configurations and evaluate their behavior in a series of experiments, highlighting different aspects of the loss function’s utility. In Section 3.4.3, we use a DDSP additive synthesis autoencoder and replicate earlier experiments [43], training the model on both synthetic input signals and singing voice recordings. We demonstrate that an appropriately chosen MSS loss configuration enables robust model convergence without the need for an additional parameter-based loss term that was previously required. However, the results also reveal remaining limitations of the unsupervised approach when harmonic structure is not explicitly enforced in the additive synthesis. In Section 3.4.4, we train a Variational Autoencoder (VAE) to reconstruct harmonic waveforms using different loss functions. While our loss landscape analysis is based on a highly simplified setting with respect to a single sinusoid frequency parameter, we show that the observed effects extend to the more general case of jointly trained neural encoders and decoders, where smoother loss behavior is reflected in a more interpretable latent space structure.

In summary, our experiments provide evidence for spectral leakage as a possible underlying cause of the MSS loss’ failure to provide informative gradients for frequency parameters. Yet, an appropriately chosen configuration can partially mitigate this issue, and can be associated with improved convergence behavior in DDSP model training as well as a more structured latent space in VAEs.

3.2 Generalized Multi-Scale Spectral Loss

An autoencoder consists of an encoder $\mathcal{E} : \mathcal{X} \rightarrow \mathcal{Z}$ and a decoder $\mathcal{D} : \mathcal{Z} \rightarrow \mathcal{X}$ which are chosen so that

$$\mathcal{D}(\mathcal{E}(x)) \approx x \quad (3.1)$$

for all $x \in \mathcal{X}$. Often, $\mathcal{E}$ and $\mathcal{D}$ are NNs that jointly learn a suitable encoding and decoding by minimizing a loss function $\mathcal{L} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ between x and $\hat{x} = \mathcal{D}(\mathcal{E}(x))$ over a training dataset. This minimization is typically achieved by a form of stochastic gradient descent on $\mathcal{L}(x, \hat{x})$.

If $\mathcal{X} = \mathbb{R}^L$ is the space of real discrete audio signals of length L , the MSS loss is a popular choice for $\mathcal{L}(x, \hat{x})$. This loss function aggregates the distance between multiple spectrograms with specified window types, window sizes, and magnitude compressions to achieve a high temporal and spectral accuracy without enforcing phase coherence of the compared signals. Let $\mathcal{Y}_{w,N,p}[m, k]$ be the spectrogram of x as defined in Eq. 2.5, where $w \in \mathbb{R}^N$ is a discrete window function, $N \in \mathbb{N}$ is the window size in samples, and $p : \mathbb{R}_+ \rightarrow \mathbb{R}$ is an (optional) magnitude compression function. As a reminder, each time–frequency coefficient can be accessed with the time index $m \in [0 : M - 1]$ with $M = L/H$, assuming for simplicity

Table 3.1: Considered MSS Loss configuration choices.

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x + \varepsilon)\}, \varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1 + \gamma x)\}, \gamma = 1$
	C3	$\mathcal{P} = \{20 \log_{10}(x + \varepsilon)\}, \varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x, \log(x + \varepsilon)\}, \varepsilon = 10^{-7}$
Matrix	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \ \mathcal{Y} - \hat{\mathcal{Y}}\ _1$
Distance	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \ \mathcal{Y} - \hat{\mathcal{Y}}\ _2^2$

that L is a multiple of H , and frequency index $k \in [0 : K - 1]$ with $K = \lceil (N + 1)/2 \rceil$, discarding negative frequencies since spectra of real signals are symmetric. To further simplify notation, we consider $\mathcal{Y}_{w,N,p} \in \mathbb{R}_+^{(K \times M)}$ to be a matrix. Analogously, we denote the spectrogram matrix of $\hat{x}$ by $\hat{\mathcal{Y}}_{w,N,p}$.

With this, a generalized MSS loss can be defined as

$$\mathcal{L}_{\text{MSS}}(x, \hat{x}) := \sum_{N \in \mathcal{N}} \sum_{p \in \mathcal{P}} d(\mathcal{Y}_{w,N,p}, \hat{\mathcal{Y}}_{w,N,p}), \quad (3.2)$$

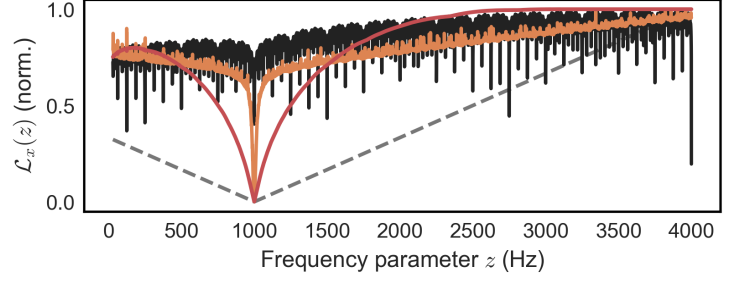
where $\mathcal{N}$ is a set of suitable window sizes, $\mathcal{P}$ is a set of suitable compression functions, and $d(\cdot, \cdot)$ is a distance between two matrices. This formulation allows for many different configurations of w , $\mathcal{N}$, $\mathcal{P}$, and d . In Table 3.1, we introduce a coding scheme for the configurations used in our experiments. As an example, the “original” MSS loss proposed in [42] is attained by the configuration (WH, S4, C4, D1). The small value ε in C1, C3, and C4 is used to avoid taking the logarithm of zero. The parameter $\gamma \in \mathbb{R}_+$ in C2 can be used to control the strength of compression [131]. We set $\gamma = 1$ for comparability with C1.

3.3 Sinusoid Frequency Estimation with Spectral Losses

As a defining property of a DDSF autoencoder, $\mathcal{D}$ in Eq. 3.1 becomes a fixed mapping from a latent parameter space $\mathcal{Z}$ to $\mathcal{X}$, while only the encoder $\mathcal{E}$ is a Neural Network (NN) with learnable parameters. The training objective for $\mathcal{E}$ given a fixed target signal x can thus be rephrased as a loss function $\mathcal{L}_x : \mathcal{Z} \rightarrow \mathbb{R}$ defined by

$$\mathcal{L}_x(z) = \mathcal{L}(x, \mathcal{D}(z)) \quad (3.3)$$

Figure 3.1: Loss landscapes $\mathcal{L}_x(z)$ (normalized) with different MSS configurations. Black: Original MSS (WH, S4, C4, D1). Orange: Modified Hann MSS (WH, S5, C4, D2). Red: Smooth MSS (WF, S5, C2, D2). The grey dashed line shows an ideal convex loss, the absolute difference between z and f_{tgt} .



for an encoder output $z = \mathcal{E}(x)$. As opposed to a general autoencoder, the fixed properties of $\mathcal{D}$ for a given z considerably influence the convergence of $\mathcal{E}$ to a minimizer of $\mathcal{L}$.

Typically (see, e.g., Section 3.4), $\mathcal{D}$ is a non-trivial mapping with many control parameters which all have to be learned jointly. In the following, we consider a simpler but illustrative scenario where $\mathcal{D}$ is a single sinusoidal oscillator with fixed amplitude $A = 1$ that maps a single frequency parameter $z \in \mathbb{R}$ to an output signal $\hat{x}$ with

$$\hat{x}[n] = A \sin(2\pi zn/F_s), \quad (3.4)$$

for all $n \in [0 : L - 1]$, where $F_s = 16000$ Hz is the sampling rate used in our experiments. We further assume that the target signal x is also generated according to Eq. 3.4, with frequency $f_{\text{tgt}} = 1000$ Hz. In this setting, we can visualize the *loss landscape* $\mathcal{L}_x(z)$ for different $\mathcal{L}_{\text{MSS}}$ as shown in Figure 3.1. Particularly the original MSS loss (in black) appears to be very noisy and thus provide uninformative gradients $d\mathcal{L}_x/dz$ to find the optimal value $z = f_{\text{tgt}}$. Understanding the causes for this loss behavior and the differences between configurations in Figure 3.1 is a main goal within this chapter.

3.3.1 Spectral Leakage

The windowing of x and $\hat{x}$ in the spectrogram defined in Eq. 2.5 leads to a “blurred” spectral representation of the sinusoids, since multiplication with the window function w in time domain is equivalent to convolution in frequency domain. Spectra of finite-length windows have multiple local maxima separated by zeros [151] that can be differentiated into a *mainlobe* (the central maximum around the sinusoid frequency up to the first zero on both sides) and *sidelobes* (all other local maxima). This effect of *spectral leakage* is illustrated in Figure 3.2, showing the influence of different configuration choices on the spectrum of a windowed excerpt of x (in black) and $\hat{x}$ with an arbitrarily chosen z (in light blue). Each plot depicts the DFT bin frequencies as vertical lines and the DFT coefficient values as circles. It further shows the (approximate) continuous spectra of the sinusoids, illustrating that they are shifted versions of the symmetric window spectra centered around f_{tgt} and z . The DFT bin frequencies form a “rigid sampling grid” on the frequency axis, so that all coefficient values change when the window spectrum is shifted relative to the grid. In our example, f_{tgt} is equal to a DFT bin frequency, so that most DFT coefficients coincide with a zero of the window spectrum, while z lies between two bins and the coefficients are

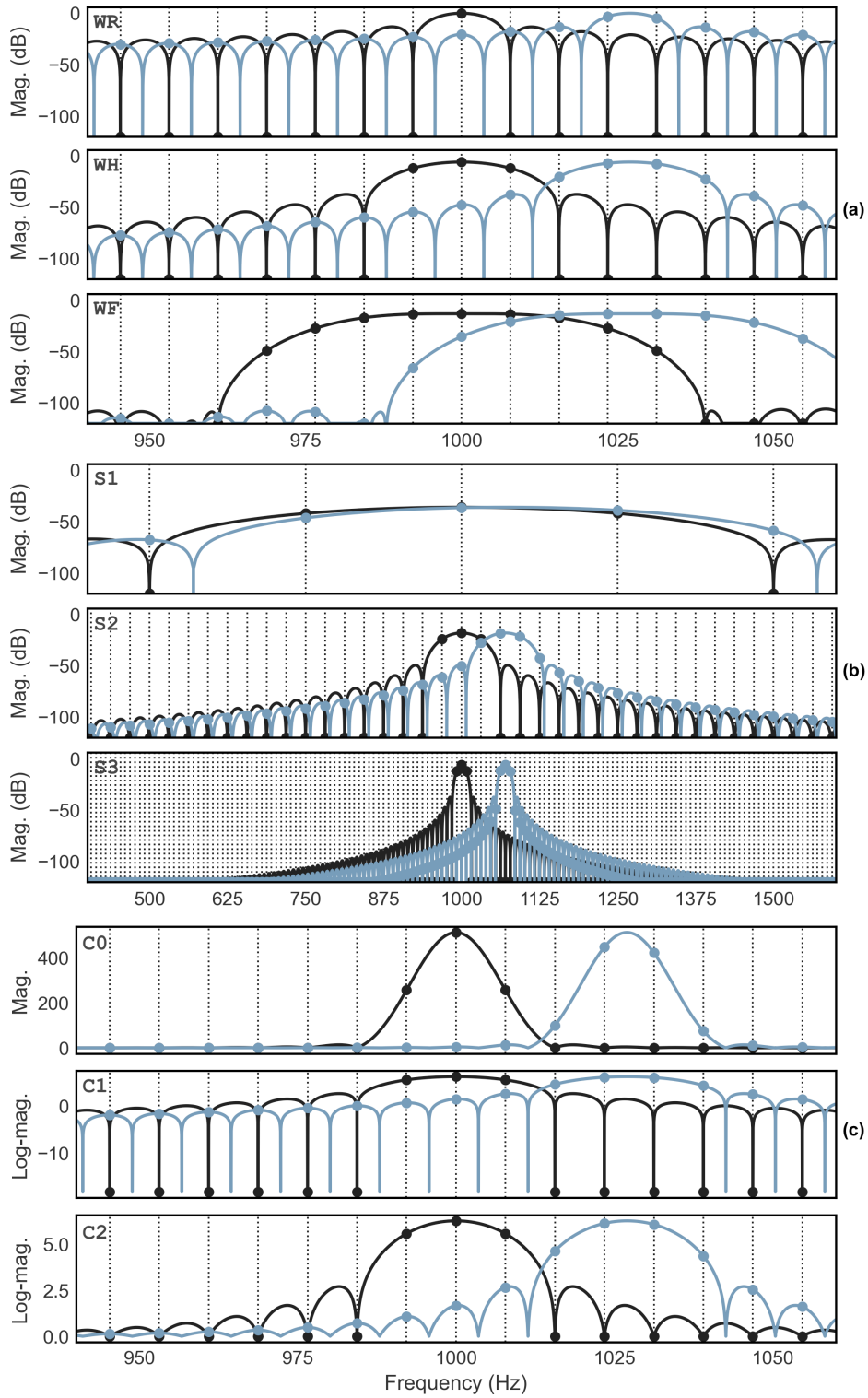


Figure 3.2: Influence of (a) window type, (b) window size, and (c) magnitude compression on the spectrum of x (black) and $\hat{x}$ (light blue). Circles denote DFT coefficients and lines the approximate continuous spectra. In (a) and (b), some coefficient values are below the y-axis range, indicated by half circles. The default configuration is (WH, S3, C3) unless otherwise specified.

non-zero. This can lead to large numerical differences between the two DFT spectra, especially when sidelobes are prominent.

Without any blurring of the sinusoid spectra, $\mathcal{L}_x(z)$ would not provide informative gradients at all. In our example, $d\mathcal{L}_x/dz$ would be zero when the spectral peaks do not overlap, since $\mathcal{L}_{\text{MSS}}$ depends on the element-wise difference between spectra (i.e., the vertical distance between circles in Figure 3.2). However, an ideal “kernel” for blurring has one unique maximum, since in this case $\mathcal{L}_x(z)$ could only be reduced when z moves closer to f_{tgt} and not by other changes of z relative to the DFT bins. In other words, when choosing a suitable $\mathcal{L}_{\text{MSS}}$ configuration for frequency estimation via gradient descent, we aim for a wide mainlobe and numerically negligible sidelobes. Figure 3.3 illustrates $\mathcal{L}_x(z)$ with different configurations for $\mathcal{L}_{\text{MSS}}$, which we will discuss in the following.

3.3.2 Loss Configuration Parameters

In the following, we will qualitatively analyze how different configurations for the window type w , window size N , magnitude compression p , and spectrum norm d influence the spectral representation itself (visualized in Figure 3.2) and the resulting loss landscape $\mathcal{L}_x(z)$ (shown in Figure 3.3).

Window Type. The window spectra of three different window types—Rectangular (WR), Hann (WH), and Flat Top (WF)—are compared in Fig. 3.2a with an otherwise fixed configuration (S3, C2, D2). Many considerations influence window choices in practice [151, Ch. 5.3.3], since for example narrower windows have a higher *sensitivity* (i.e., better ability to detect sinusoids in noise), while wider windows have a better *dynamic range* (i.e., sidelobes of a loud sinusoid are less likely to mask a weaker sinusoid). For frequency estimation, the sidelobe level is a central property that influences the behavior of $\mathcal{L}_x(z)$. Using WR (and to a lesser extent WH) leads to strong periodic fluctuations of $\mathcal{L}_x(z)$ with a period of F_s/N as shown in Fig. 3.3a (see Appendix A for a closer examination of the periodic behavior), due to the changes in spectral leakage depending on the relative value of z compared to the DFT bin frequencies. The low sidelobe levels of WF result in a smoother loss landscape with a unique local minimum at $z = f_{\text{tgt}}$.

Window Size. The width of mainlobe and sidelobes is also influenced by the window size, as illustrated in Fig. 3.2b, using fixed (WH, C0, D2). The choice of a suitable set of window sizes was originally motivated by the resolution of amplitude comparisons, where a small window size leads to a better time resolution and a worse frequency resolution. For frequency estimation, the width of the locally convex sections depend on the maximal difference between f_{tgt} and z where the mainlobes still overlap, so that the convex section is wider for smaller window sizes. Larger windows, conversely, increase the loss’ ability to discriminate sinusoids with similar frequencies.

Magnitude Compression. Compression enables a comparison of magnitudes over a wide dynamic range and is often perceptually motivated. However, it may also exacerbate the problem of periodically changing spectral leakage by decreasing the relative difference between mainlobe and sidelobe levels (see Fig. 3.2c).

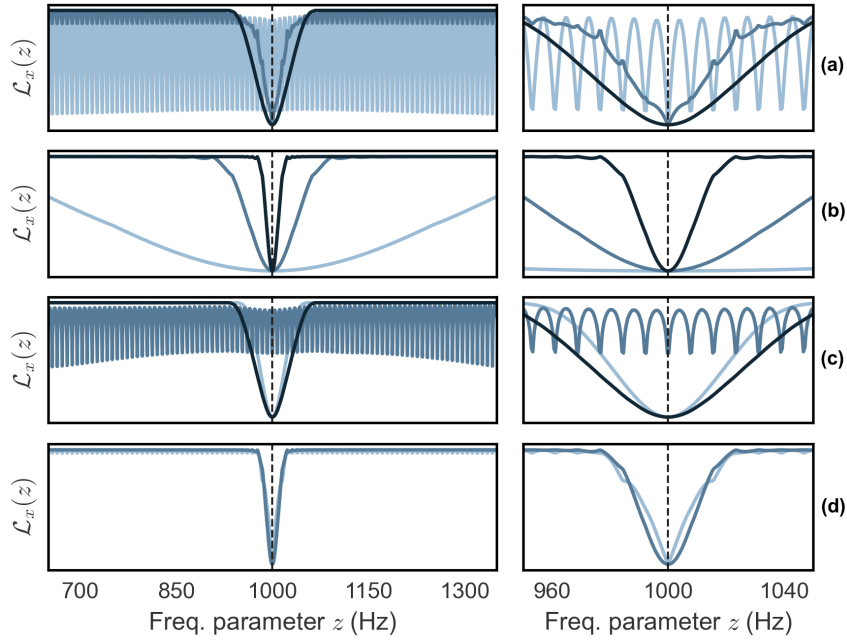


Figure 3.3: Influence on $\mathcal{L}_x(z)$ of (a) window type (light: WR, medium: WH, dark: WF) compared using (S3, C2, D2), (b) window size (light: S1, medium: S2, dark: S3) compared using (WH, C0, D2), (c) magnitude compression (light: C0, medium: C1, dark: C2) compared using (WF, S3, D2), and (d) matrix distance (light: D1, medium: D2) compared using (WH, S3, C0).

The simple logarithmic compression (C1) as used in [42] leads to large negative values at the zeroes of the window function bounded below by $\log(\varepsilon)$. This amplifies the periodic behavior of $\mathcal{L}_x(z)$ as shown in Fig. 3.3c with fixed (WF, S3, D2). To mitigate this issue, we propose to replace ε with 1 (C2), so that the compressed value is always larger or equal to 0. A factor $\gamma \geq 0$ can further be chosen to adjust the compression strength.

Spectrum Norm. We compare two distances between the spectrogram matrices, the ℓ^1 norm (D1) and the squared ℓ^2 norm (D2) in Fig. 3.3d with fixed (WH, S3, C0). Other considerations like outlier sensitivity are often relevant for choosing a distance, but D1 slightly amplifies the periodic fluctuations in $\mathcal{L}_x(z)$.

3.4 Experiments

Based on the qualitative analysis in Section 3.3, we can derive representative MSS loss configurations and explore their behavior in different experimental settings. Note that we choose the configurations not to represent a “best” setting that is universally applicable, but to illustrate how different choices affect the loss landscape.

Table 3.2: GRA for selected configurations.

Configuration	GRA			
	0.3 ct	3 ct	30 ct	300 ct
original (WH, S4, C4, D1)	0.523	0.529	0.573	0.755
mod. Hann (WH, S5, C4, D2)	0.613	0.635	0.708	0.923
smooth (WF, S5, C2, D2)	0.999	0.993	0.952	0.860

3.4.1 Representative MSS Loss Configurations

In the following experiments, we compare three different configurations for $\mathcal{L}_{\text{MSS}}$.

1. **“Original” MSS loss (WH, S4, C4, D1):** This is the configuration proposed in [42], which is widely adopted in DDSP literature.
2. **“Modified Hann” MSS loss (WH, S5, C4, D2):** This configuration uses a set of co-prime window sizes (S5) and the squared ℓ^2 norm between the spectra (D2).
3. **“Smooth” MSS loss (WF, S5, C2, D2):** This configuration also uses S5 and D2, but combined with a window function that minimizes the numerical influence of sidelobes (WF) and the C2 compression.

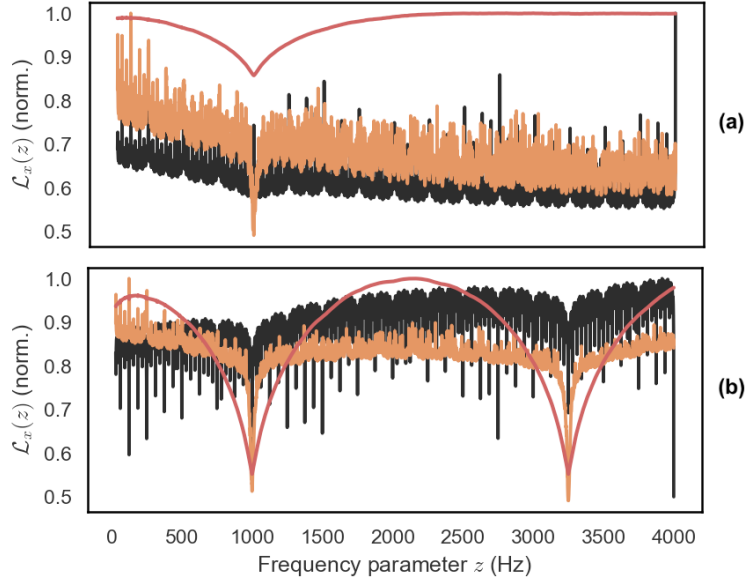
We have already seen the resulting loss landscapes $\mathcal{L}_x(z)$ for these configurations in Figure 3.1. In addition to this qualitative comparison, Table 3.2 shows the Gradient-Sign Ranking Accuracy (GRA) [214], which quantifies how often, on average, $\mathcal{L}_x(z)$ decreases when moving towards a random f_{tgt} by c cents from a random initial frequency z_0 . While it is not an analytic evaluation of the loss, a larger GRA indicates that $\mathcal{L}_x(z)$ tends to provide informative gradients. A value of 0.5 suggests that changes in $\mathcal{L}_x(z)$ are random. We calculate the GRA for step sizes c of 0.3, 3, 30, and 300 cents with all other settings as in [214].

The original MSS loss results in a GRA near 0.5 for smaller step sizes (e.g., 0.523 for $c = 0.3$ cents), so that gradient descent algorithms are unlikely to converge. In fact, for noise-free signals, this loss behavior is entirely a result of spectral leakage at different window sizes, amplified by the logarithmic compression C1. The modified Hann MSS loss uses D2 and S5, where all window sizes are prime instead of powers of two. This way, z does not coincide with the DFT bin frequencies of multiple window sizes at the same time, which reduces the amplitude of the fluctuations in $\mathcal{L}_x(z)$. The modified Hann MSS loss achieves a high GRA of 0.923 for $c = 300$ cents while some artifacts still impact the GRA for smaller step sizes. The smooth MSS loss results in the fewest spectral leakage artifacts by using WF and C2, while also having the widest mainlobe overlaps. The high GRAs of 0.999 for $c = 0.3$ cents and 0.993 for $c = 3$ cents indicate that this configuration also approximates local convexity with a unique minimum in these synthetic conditions.

The better performance of the original and modified Hann MSS loss configurations at a step size of 300 cents can be attributed to the relative contribution of sidelobes to the loss. For WR and WH, the sidelobes decay in amplitude with increasing distance from the mainlobe, as shown in Figure 3.2a. In the absence of other interfering signal components, larger sidelobes overlapping with the mainlobe of the target signal

Figure 3.4: Loss landscapes $\mathcal{L}_x(z)$ (normalized) for different MSS configurations as in Figure 3.1 (black: original, orange: modified Hann, red: smooth) with two different out-of-domain target signals x .

- (a) Using $M = 1$ ($f_1 = 1000$ Hz) and $\sigma_\eta^2 = 3.16$ (-5 dB SNR).
 (b) Using $M = 2$ ($f_1 = 1000$ Hz, $f_2 = 3250$ Hz) and $\sigma_\eta^2 = 0$.



reduce the loss value more than smaller sidelobes. As a result, the loss function continues to exhibit a trend toward the true frequency over a wider range, even when the mainlobes themselves do not overlap. This effect is particularly pronounced for the modified Hann MSS configuration, illustrated by the orange curve in Figure 3.1. Here, the co-prime selection of window sizes ensures that the zeros between sidelobes do not occur at the same frequency across different scales. Conversely, in the smooth MSS loss configuration using the flat top window (WF), the sidelobes are numerically negligible, causing the gradient to vanish (i.e., approach zero) as soon as the mainlobes no longer overlap.

3.4.2 Extending the Target Signal

The simple scenario with a single sinusoid does not account for the influence of other effects on the loss that are often observed in practice, including noise or multiple tonal components with varying amplitudes in the target signal x . As a first step towards more practically relevant applications of the MSS loss, we consider an extended signal model

$$x[n] = \sum_{m=1}^M A_m \sin(2\pi f_m n / F_s) + \eta[n], \quad (3.5)$$

where M is the number of sinusoids, each with amplitude A_m and frequency f_m . For simplicity, we set $A_m = 1$ throughout the following experiment. Furthermore, $\eta[n] \sim \mathcal{N}(0, \sigma_\eta^2)$ is i.i.d. Gaussian white noise whose energy relative to the sinusoid signal can be controlled by changing the variance σ_η^2 . By still using Eq. 3.4 to generate $\hat{x}$ (i.e., keeping $\mathcal{D}$ unchanged, producing a single sinusoid), we can again visualize a loss landscape $\mathcal{L}_x(z)$ as in Figure 3.1 and analyze the influence of an “out-of-distribution” target signal.

Figure 3.5: Illustration of the effect of window size when two sinusoids with similar frequency ($f_1 = 1000$ Hz and $f_2 = 1300$ Hz) are present in the target signal. All plots show a loss landscape using the configuration (WF, C2, D2) with variable window size.

- (a) Window size $N = 64$ (S1).
- (b) Window size $N = 512$ (S2).
- (c) Window size $N = 2048$ (S3).

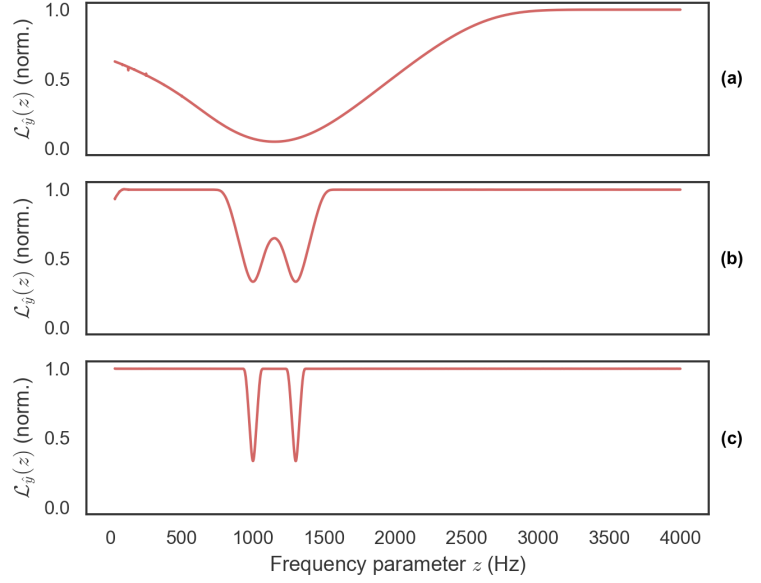


Figure 3.4 shows results for the three representative configurations of the MSS loss. When white Gaussian noise is added to the target signal with a signal-to-noise ratio (SNR) of -5 dB (i.e., with $M = 1$, $A_1 = 1$, and $\sigma_\eta^2 = 10^{5/10} \approx 3.16$) as shown in Figure 3.4a, a local minimum at the target frequency of $f_1 = 1000$ Hz remains visible for all loss configurations. However, only the smooth MSS loss configuration (shown in red) exhibits a well-behaved loss surface over a wider frequency range, whereas both the original (black) and modified Hann (orange) MSS loss curves exhibit many local minima and maxima and no large-scale trends that are still present in the loss landscape shown in Figure 3.1. This highly erratic behavior prevents optimization methods based on gradient descent from converging to the local minimum. Note that, as expected, the lowest possible loss value for the smooth MSS loss is much larger than 0, since the noise component is not modeled in $\hat{x}$ and the distance between the signals cannot be reduced further.

When two sinusoids are present in the target signal (specifically, $M = 2$, $A_1 = A_2 = 1$, $\sigma_\eta^2 = 0$, $f_1 = 1000$ Hz, and $f_2 = 3250$ Hz) as shown in Figure 3.4b, $\mathcal{L}_x(z)$ exhibits two clear local minima at these frequencies with the smooth MSS loss configuration. For the other two configurations, these minima are again obscured by the overall noise in the loss landscape arising from spectral leakage. This way, given a suitable initialization, only the smooth MSS loss may allow for simultaneously learning the frequencies of multiple sinusoids present in the target signal via gradient descent.

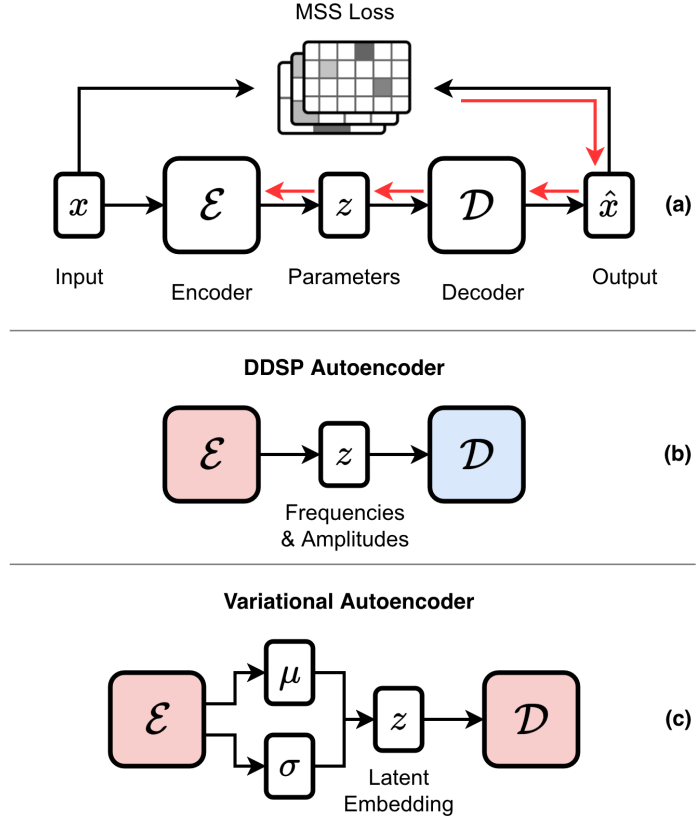
Furthermore, when the two target frequencies are close to each other, small window sizes may cause a single shared local minimum to appear, whereas larger window sizes improve the frequency selectivity of $\mathcal{L}_x(z)$, at the cost of reduced time resolution. This effect is illustrated in Figure 3.5, which compares different window sizes while keeping the remaining loss configuration fixed to (WF, C2, D2). For the smallest window size $N = 64$ (S1, Figure 3.5a), the loss landscape fails to resolve the two sinusoids at $f_1 = 1000$ Hz and $f_2 = 1300$ Hz present in the target signal, resulting in a single merged minimum. In contrast, a window size of $N = 2048$ (S3, Figure 3.5c) yields two clearly separated local minima

Figure 3.6: Schematic overview of the experimental architectures in Sections 3.4.3 and 3.4.4.

(a) General structure of a signal autoencoder with input/target signal x , encoder $\mathcal{E}$, latent parameters z , decoder $\mathcal{D}$, and output signal $\hat{x}$. x and $\hat{x}$ are compared with an MSS loss, whose gradients are back-propagated as a learning signal for $\mathcal{E}$ (and potentially $\mathcal{D}$) as illustrated by the red arrows.

(b) In a DDSP autoencoder, $\mathcal{E}$ is a NN (indicated by red background), and $\mathcal{D}$ consists of fixed, differentiable signal processing components (indicated by blue background), so that z can be directly interpreted as frequency and amplitude parameters.

(c) In a VAE, both $\mathcal{E}$ and $\mathcal{D}$ are NNs. $\mathcal{E}$ outputs parameters μ (mean) and σ (variance) of a latent distribution from which the latent embedding z is sampled.



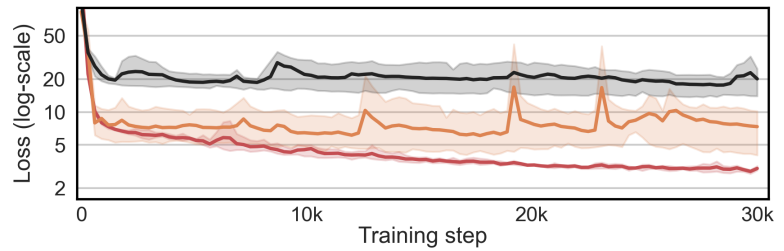
corresponding to the individual frequencies. Given a suitable initialization, this effect may be utilized to learn the parameters of multiple sinusoids by progressively refining the loss landscape.

3.4.3 DDSP Autoencoder

To investigate differences between the configurations in an actual machine learning scenario, we conduct experiments where the gradients of the MSS loss are used to optimize a neural network encoder $\mathcal{E}$ of a DDSP autoencoder, as defined in the beginning of Section 3.3 and illustrated in Figure 3.6a. For these DDSP experiments, we use the architecture of the encoder (F_{sin}^{θ}) and sinusoidal synthesizer (S_{sin}) from [43] for $\mathcal{E}$ and $\mathcal{D}$, respectively. The task of $\mathcal{E}$ is to estimate time-varying parameters (frequency and amplitude) for 100 sinusoids that are then synthesized by $\mathcal{D}$, resulting in $z \in \mathbb{R}^{200}$ for the setup shown in Figure 3.6b. The additive synthesis model of S_{sin} is also described in more detail in Section 2.4.1.

In addition to a reconstruction loss, the training method in [43] relies on a method that the authors call *self-supervision*, where $\mathcal{E}$ is trained at least partially with a parameter loss. To do this, known parameters $\tilde{z}$ are sampled, from which a synthetic target signal $\tilde{x} = \mathcal{D}(\tilde{z})$ can be generated. $\tilde{x}$ then serves as the input to $\mathcal{E}$ with $\tilde{z}$ as the target for a supervised training for $z = \mathcal{E}(\tilde{x})$. In our experiments, we instead use only a reconstruction loss ($\mathcal{L}_{\text{MSS}}$ with the respective configuration), evaluating how informative the gradients of

Figure 3.7: Training loss for $F_{\sin}^{\theta}$ with original MSS loss (black), mod. Hann MSS loss (orange), and smooth MSS loss (red), with mean and variance of three runs each.



this loss function are for learning the sinusoidal parameters. We train this DDSP autoencoder with two different datasets, one synthetic and one consisting of recorded signals.

3.4.3.1 Synthetic Signals

First, we train our model on the *NSynth* dataset [44], following the training procedure of the original DDSP work [42]. The dataset consists of individual, monophonic tones generated with a variety of commercial synthesizers and a wide range of MIDI pitches, velocities, and playing styles. The signals have a length of four seconds, corresponding to 64 000 samples at $F_s = 16\,000$ Hz. Because each example contains a single tone, the pitch content of the training data is very controlled, while still including background noise and transient components inherent in the synthesizer sound textures. Note that the synthesizer in $\mathcal{D}$ can only use sinusoids to minimize the loss between x and $\hat{x}$, so that those non-tonal components in the training signals may be detrimental to the loss landscape, similar to the added noise in Section 3.4.2.

The training loss with the different $\mathcal{L}_{\text{MSS}}$ configurations is shown in Figure 3.7. Since the different configurations result in substantially different loss magnitudes, each $\mathcal{L}_{\text{MSS}}$ was multiplied by an empirically determined scaling constant to ensure comparability in the visualization while also enabling a constant learning rate across all trainings. With all other settings kept consistent with [43], only the smooth MSS loss configuration leads to robust convergence of $F_{\sin}^{\theta}$. Across three training runs, this configuration not only reaches smaller loss values than the alternatives but also exhibits a much lower variance.

From a practical perspective, the most important indicator for training success is whether the model learns to estimate semantically meaningful synthesis parameters, i.e., in our case, reproduces only the partials present in a tone. Since there are typically much fewer than 100 perceptually salient partials in a tone (especially for higher-pitched tones), only a subset of the 100 sinusoids in $\mathcal{D}$ is expected to have a perceptually relevant amplitude. Furthermore, in the given unsupervised setup, it is not enforced that individual harmonics are represented by a single sinusoid. As a result, evaluating the accuracy of the estimates directly in the parameter domain is not trivial. To get a first upper bound of the estimation accuracy, we can measure how closely the estimated frequencies with non-zero amplitude match the ground truth. Notably, the *NSynth* test set is not suitable for this experiment, as only the MIDI pitch of each tone is annotated, while the exact frequencies of the individual harmonics are unknown and can vary over the course of the four-second signals. Therefore, we generate 1000 synthetic signals of one second

duration, each with a randomly chosen constant F0 between 30 and 800 Hz and ten integer harmonics with randomly assigned amplitudes. With the exactly known frequencies in this synthetic test set, we compute the mean distance between each harmonic and the nearest estimated sinusoid frequency that has a non-zero amplitude. In this metric, $F_{\sin}^{\theta}$ trained with the smooth MSS loss achieves a mean distance of 27 ± 35 cents, compared to 838 ± 647 cents for the original and 1467 ± 932 cents for the modified Hann configurations. For reference, linearly distributing the estimated frequencies across the frequency spectrum leads to a mean distance of 47 ± 115 cents, indicating that the models trained with the original or modified Hann configuration perform significantly worse than this trivial baseline, mainly because most of the sinusoids are estimated to have zero amplitude. Although a mean distance of 27 cents for the model trained with the smooth MSS loss does not indicate particularly good estimates, the result clearly indicates the benefit of a smoother loss landscape with respect to frequency parameters in this DDSP scenario.

In addition to an evaluation in the parameter domain, the combined effect of estimated frequencies and amplitudes can be assessed through the quality of the resynthesized output signal $\hat{x}$. To quantify this in a way that goes beyond a spectral distance (which is already the training loss function), we compare F0 estimates of the input signal x with those obtained from the resynthesized signal $\hat{x}$. The underlying rationale is that F0 estimators such as CREPE [97] rely on perceptually salient, approximately harmonic partials. Consequently, discrepancies in the estimated F0 indicate the presence of prominent sinusoidal components in $\hat{x}$ that are not harmonically related to the target signal. For this experiment, we use input signals from the *MDB-melody-synth* dataset [169] and compute the raw pitch accuracy (RPA) [154] between the F0 estimates of x and $\hat{x}$ obtained with CREPE, using a tolerance of 50 cents for correct estimates. Unlike the isolated tones used during training on NSynth, the *MDB-melody-synth* signals contain complete melodies and are therefore substantially more complex, requiring the model to generalize beyond single-note inputs. When trained with the smooth MSS loss, the model achieves an RPA of 0.81 ± 0.09 on the *MDB-melody-synth* test set. In contrast, training with the original and modified Hann configurations results in substantially lower RPA values (0.01 ± 0.03 and 0.27 ± 0.15 , respectively), indicating that only training with the smooth MSS loss leads to an analysis–synthesis model that exhibits a reasonable degree of robustness when estimating parameters for these more complex input signals.

3.4.3.2 Singing Voice Recordings

The training objective in our DDSP experiments can be paraphrased as follows: *“Produce a signal consisting of 100 sinusoids with unrestricted time-varying frequencies and amplitudes, so that the spectrum is as close as possible to the spectrum of the target signal.”* For the synthetic signals above, we have shown that this leads to somewhat meaningful parameter estimates only with the smooth MSS loss. In the following experiment, we examine whether such a DDSP-based unsupervised learning framework is suitable for an analysis-by-synthesis setup, where the aim is to identify and track the harmonics of a singing voice signal. With this unsupervised learning method, the model could be trained or fine-tuned on unlabeled data that closely matches (or even is identical to) the signals it will later be applied to.

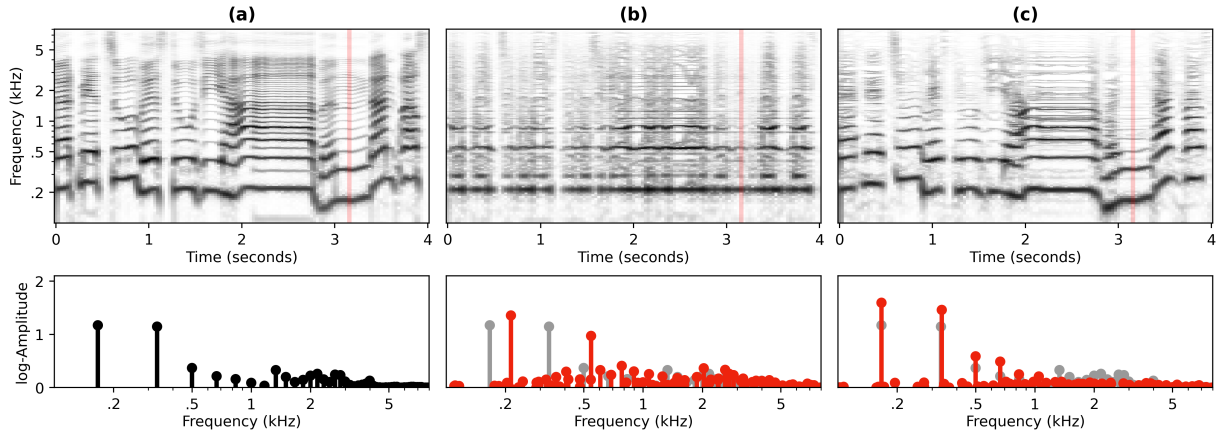
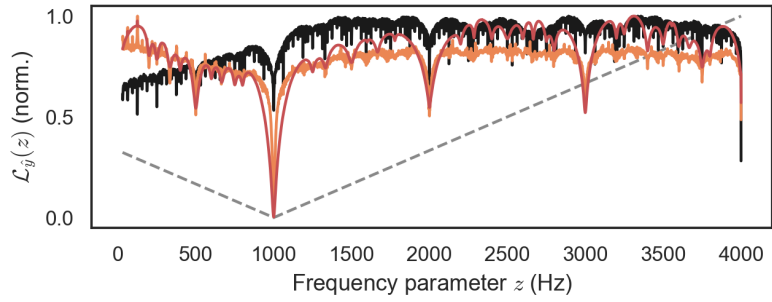


Figure 3.8: Qualitative comparison of results from unsupervised training with LM-SSD singing voice signals. Each plot shows the spectrogram of the respective signal (top) and a visualization of parameters for frame 101 at around 3.1 seconds (bottom, frame is highlighted in the spectrogram). (a) Original signal from the test set. The parameters shown in black are the pseudo-reference. (b) Output of model trained with original MSS loss. The parameters shown in red are the estimates, the pseudo-reference is shown in gray. (c) Output of model trained with smooth MSS loss. The parameters shown in red are the estimates, the pseudo-reference is shown in gray.

To simulate this scenario and conduct a qualitative experiment, we train the sinusoid parameter estimation model ($F_{\sin}^{\theta}$) on a dataset of clean high-quality singing voice recordings. Specifically, we use the crosstalk-free vocal microphone recordings from the *Larynx Microphone Singer-Songwriter Dataset* (LM-SSD), which is described in detail in Chapter 4. The dataset comprises 48 crosstalk-free vocal microphone recordings from four different singers, approximately 2.5 hours of audio. One singer (2M) is exclusively reserved for the test set. From each recording of the remaining three singers, we randomly extract 21 000 four-second excerpts (each having $L = 64\,000$ samples at a sampling rate of $F_s = 16\,000$ Hz). From these excerpts, 20 000 excerpts are used for training, while 1000 excerpts each are allocated to a validation set. For the test set, we further extract 1000 excerpts from the recordings of singer 2M. To obtain a *pseudo-reference* for the harmonic parameters, we first estimate the F0 trajectory from the audio signal with CREPE [97]. Following the strategy described in Section 6.2.1, harmonic frequencies and amplitudes are then identified by finding spectral peaks near integer multiples of the F0. An example of the resulting pseudo-reference for a single frame is illustrated in black at the bottom of Figure 3.8a.

As in the previous experiments, we now learn frame-wise frequency and amplitude estimates for 100 sinusoids to minimize the distance between x and $\hat{x}$. Figure 3.8 shows the results for two models trained with the original and the smooth MSS loss, respectively. In Figure 3.8b, the spectrogram of $\hat{x}$ shows that the model trained with the original MSS loss is unable to track the harmonics of the target signal, and keeps the sinusoid frequencies approximately constant over the course of the excerpt, while only varying amplitudes. Also in the detailed visualization of the frequencies and amplitudes for a single frame below the spectrogram there is no clear correspondence between the pseudo-reference (in gray) and estimated parameters (in red). In the output of the model trained with the smooth MSS loss shown in Figure 3.8c, the harmonic contours are mainly reproduced correctly. However, the visualization of the individual

Figure 3.9: Loss landscape as in Figure 3.1, but with target and output signals that have five harmonics each, parameterized by an F0 z .



sinusoid parameters in the bottom of Figure 3.8c reveals an issue. Since the decoder parameterization does not enforce the sinusoids to be harmonic (i.e., near-integer multiples) of each other, many of the 100 available sinusoids do not reproduce individual harmonics of the target signal, but are used by the model to reconstruct noisy components of the target signal. As an example, the sinusoid at around 200 Hz in the visualized frame has non-zero amplitude but does not correspond to a harmonic of the input signal. This effect results in many audible non-harmonic tonal components in the output signal and a less interpretable parametric representation in the analysis-by-synthesis setup.

In a way, this is a limitation of the model assumptions, since the non-harmonic components contribute to the training objective to minimize the MSS loss. One possible way to mitigate this issue is to model the non-tonal components separately using a noise synthesizer, similar to the approach taken in Spectral Modeling Synthesis (SMS) [186]. This strategy is discussed in Chapters 4 and 7. Further constraining the model by enforcing strictly harmonic sinusoids whose frequencies are parameterized by a single F0, as is common in DDSP additive synthesis (see, e.g., [42, 231]), leads to a loss landscape that inherently contains multiple (false) local minima for the F0 parameter whenever harmonics of the target and generated signals overlap. This is illustrated in Figure 3.9, showing the same loss landscape as in Figure 3.1, but where both the target and output signal have five harmonics with decreasing amplitudes. We can observe that the smooth MSS loss now has additional local minima whenever harmonics of the target and output coincide (prominently, for example, at 500 Hz, 2000 Hz, 3000 Hz, etc.). The unsupervised training of DDSP models parameterized by an F0 using spectral reconstruction losses therefore remains a fundamental challenge that would require novel and specifically designed loss functions.

3.4.4 Variational Autoencoder

In a final experiment, we investigate how the smoothness of the loss landscape affects the organization of the latent space in a less constrained autoencoder architecture. Generally, intuitions derived from low-dimensional problems often fail to translate directly to higher-dimensional scenarios. It therefore remains unclear how the loss properties analyzed in our simplified setting with a single sinusoid extend to a model that generates the output waveform on a sample-by-sample basis. In particular, we consider a Variational Autoencoder (VAE) [99] that is trained to reconstruct short target signals $x \in \mathbb{R}^L$ with $L = 1024$. The sample-by-sample generation is generally not restricted to sinusoids, so that we can extend

this experiment to any periodic waveform. In the following experiments, we use sawtooth periodic signals with an F0 between 100 and 1000 Hz, as well as a peak amplitude between 0.1 and 1. Both parameters, F0 and amplitude, are kept constant over the duration of each target signal. At first, we set the initial phase to a fixed value of 0, before considering the influence of phase in a second stage.

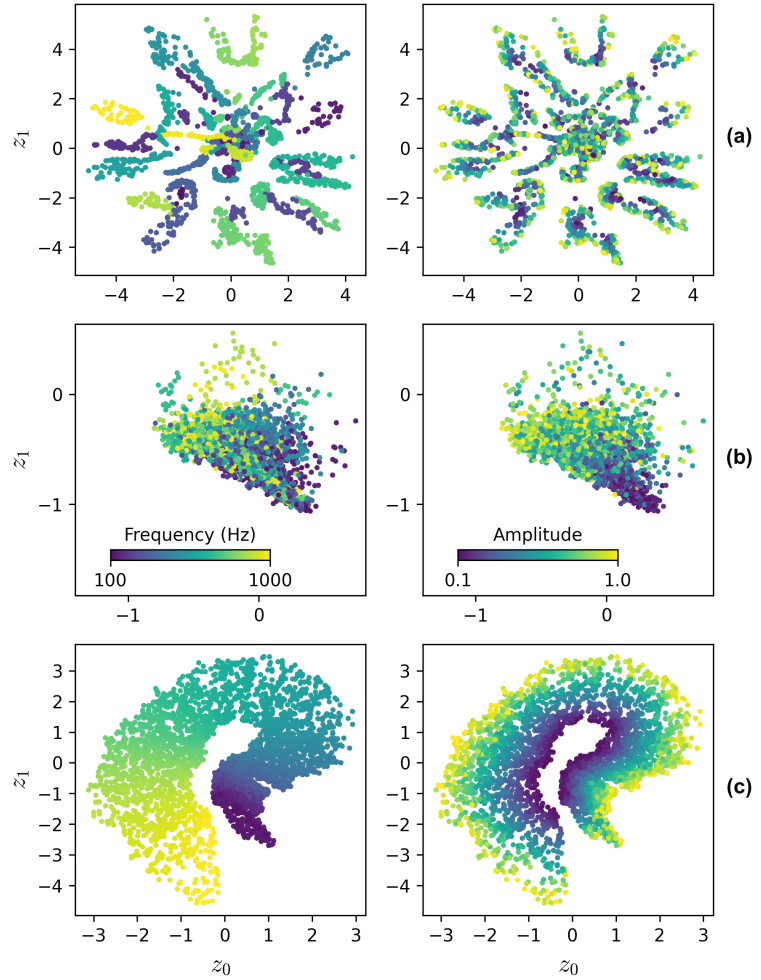
A VAE is a generative model that learns a probabilistic latent representation of data in an unsupervised manner. The encoder neural network $\mathcal{E}$ maps an input to the parameters of a Gaussian latent distribution (mean and variance). The decoder neural network $\mathcal{D}$ takes a latent sample drawn from this distribution as input and learns to output a reconstruction of the original encoder input. At inference time, it is therefore in principle possible to sample from the latent space and generate new outputs with similar properties to the training data. The general structure in comparison to the DDSP autoencoder considered before is illustrated in Figure 3.6c. Training optimizes a trade-off between reconstruction accuracy and regularization of the encoded distribution toward a known prior (typically also a Gaussian with zero mean and unit variance), which facilitates a smooth, continuous latent space suitable for generation and interpolation.

Generally, there are no restrictions on the network architecture and size of $\mathcal{E}$ and $\mathcal{D}$. In this illustrative example, we implement both neural networks with four fully connected layers with layer sizes (256, 64, 32, 2) for $\mathcal{E}$ and the inverse ordering used for $\mathcal{D}$. Furthermore, we apply *tanh* nonlinearities between layers. The bottleneck latent representation $z = \mathcal{E}(x)$ therefore is two-dimensional, so that the network has just sufficient degrees of freedom to encode F0 and amplitude of the target signals. While we keep the regularization loss term, i.e., the Kullback-Leibler (KL) divergence between the output distribution and the unit Gaussian prior, and its weight constant, we vary the reconstruction loss, comparing three different options: a time-domain mean squared error (MSE) loss, the original MSS loss, and the smooth MSS loss, while ensuring that their magnitude is comparable by multiplying them with an empirically determined constant. We train each model on a training dataset consisting of 10 000 signals. For the following visualizations, we use a separate test set of 4000 signals, with parameters drawn from the same distribution as the training data.

Figure 3.10 illustrates how the choice of reconstruction loss function fundamentally affects the organization and interpretability of the learned latent space. When using a time-domain MSE loss, the latent representation shown in Figure 3.10a exhibits some local structure. Encoded signals with similar F0 tend to be mapped to nearby regions. However, this organization is fragmented, with multiple disconnected clusters of similar signals and no coherent global arrangement across the latent space. Training with the original MSS loss does not lead to stable or consistent convergence. The resulting latent space shown in Figure 3.10b lacks a meaningful structure and only encoded signals with similar amplitudes are somewhat clustered, with lower-amplitude signals appearing toward the lower-right region of the visualization. While we do not consider reconstruction quality in this comparison, this indicates that the original MSS loss provides no meaningful guidance for encoding information about the reconstructed signals. Furthermore, the fact that the values of the latent variable z shown in Figure 3.10b are closer to zero than those of the

Figure 3.10: Visualization of the 2-dimensional VAE latent space of models trained with different reconstruction losses. On the left side, points are colored based on the frequency of the encoded signal. On the right side, color indicates the amplitude of the encoded signal.

- (a) Using time-domain MSE loss.
 (b) Using original MSS loss.
 (c) Using smooth MSS loss.



other models indicates that the regularization term is reduced more easily than the reconstruction loss during optimization, pushing the latent distribution more strongly towards a unit Gaussian. Finally, the smooth MSS loss, as shown in Figure 3.10c, yields a well-organized and interpretable latent space that reflects both frequency and amplitude of the input signal in a clear and globally interpretable structure. This result is an indication for the importance of using reconstruction losses for VAEs that are smooth with respect to perceptually relevant parameters, and suggests that such loss functions can improve both training stability and the interpretability of the learned latent space.

We repeat this experiment by introducing an additional phase parameter for the input signals, varying the normalized initial phase of the sawtooth waveform between 0 and 1. The results are shown in Figure 3.11. Using the time-domain MSE loss (Figure 3.11a), the latent space remains locally organized with respect to frequency. However, the additional degree of freedom introduced by the phase offset leads to a degradation of the organization with respect to amplitude in the two-dimensional latent space. The original MSS loss (Figure 3.11b) still fails to produce a meaningful organization with respect to frequency, amplitude, or phase. Finally, the smooth MSS loss (Figure 3.11c) is unaffected by the phase parameter, since the compared spectra are insensitive to this perceptually irrelevant variation in the signal. As a result, the latent

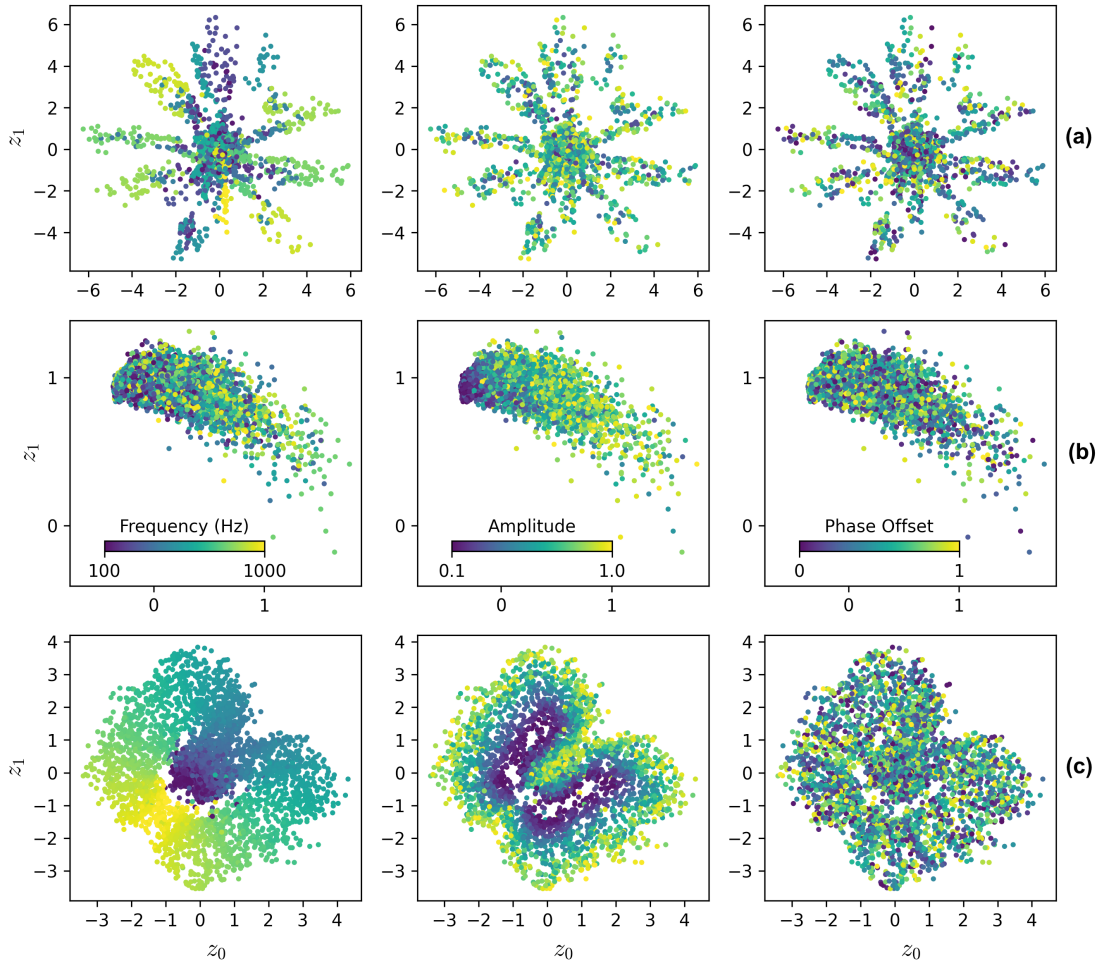


Figure 3.11: Visualization of the 2-dimensional VAE latent space analogously to Figure 3.10. On the left side, points are colored based on the frequency of the encoded signal. In the middle, color indicates the amplitude of the encoded signal. On the right side, color indicates the normalized phase offset of the encoded signal. **(a)** Using time-domain MSE loss. **(b)** Using original MSS loss. **(c)** Using smooth MSS loss.

space organization with respect to frequency and amplitude is not affected by this additional variation when using the smooth MSS loss.

3.5 Further Notes

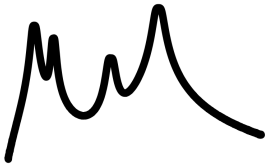
We have shown that the behavior of the loss function between audio signals with respect to individual parameters can have a significant impact on the convergence of an optimization. In particular, we identified spectral leakage, specifically the relative strength of mainlobes vs. sidelobes, as a major contributing factor to the irregularity of the MSS loss with respect to frequency parameters. The numerical influence of sidelobes can be controlled by choosing appropriate configuration parameters for the loss, including window type, window size, and magnitude compression for each spectrum, as well as the distance function

between spectra. We demonstrated that the “smooth” MSS configuration leads to a smoother loss landscape both in a simple scenarios with synthetic signals and when training an additive synthesis DDSP model on more complex recorded signals. Yet, when the DDSP model parameterization leads to local minima in a comparison of spectra (e.g., due to overlapping harmonics), learning these parameters with a spectral reconstruction loss is inherently limited. Moreover, initial experiments indicate that this smoothness property analyzed in the simplified setting may generalize to more complex optimization problems, leading to a more interpretable latent space organization in VAEs.

Taking a broader perspective, this chapter highlights a fundamental challenge in model-based deep learning. It is important to consider the interaction between the loss function and the model itself. In autoencoder-based DDSP setups, for example, the effective loss seen by the neural network encoder can be understood as the composition of the decoder model and the reconstruction loss, so that any nonlinearity in the model can lead to unfavorable behavior of the gradient with respect to individual input parameters. At the same time, our analysis shows that an appropriately designed loss or intermediate transform (such as the DFT), can mitigate these effects and result in more favorable loss landscapes even when for nonlinear model parameters. However, this does not constitute a general solution, and the impact of such nonlinearities must be assessed on a case-by-case basis. Finally, the observations with the VAE suggest that considering the smoothness of loss functions with respect to perceptually or semantically relevant signal properties can be beneficial beyond explicitly model-based approaches, potentially leading to more interpretable and better-structured learned representations.

4 Singing Voice Reconstruction from Larynx Microphone Recordings

This chapter is based on [181]. I was the primary contributor to creating the dataset, analyzing signals, running experiments, and preparing the manuscript. Michael Krause co-developed the baseline model and contributed to manuscript preparation. Michael Fast helped recording the dataset, running baseline experiments, and conducting the listening test in the context of his Master’s thesis work co-supervised by me. Sebastian Rosenzweig co-developed the baseline model. Frank Scherbaum shared his expertise on larynx microphones and provided the hardware. Meinard Müller supervised the work and contributed throughout all stages of the research and manuscript preparation.



Larynx microphones (LMs) make it possible to obtain practically crosstalk-free recordings of the human voice by picking up vibrations directly from the throat. This can be useful in a multitude of MIR scenarios related to singing, e.g., the analysis of individual voices recorded in environments with lots of interfering noise. However, LMs have a limited frequency range and barely capture the effects of the vocal tract,

which makes the recorded signal unsuitable for downstream tasks that require high-quality recordings. In this chapter, we introduce the task of reconstructing a natural sounding, high-quality singing voice recording from an LM signal. With an explicit focus on the singing voice, the problem lies at the intersection of speech enhancement and singing voice synthesis with the additional requirement of faithful reproduction of expressive parameters like intonation. In this context, we make three main contributions. First, we publish a dataset with over four hours of popular music we recorded with four amateur singers accompanied by a guitar, where both LM and clean close-up microphone signals are available. Second, we propose a data-driven baseline approach for singing voice reconstruction from LM signals using DDSP, inspired by a source–filter model that emulates the missing vocal tract effects. Third, we evaluate the baseline with a listening test and further show that it can improve the accuracy of lyrics transcription as an exemplary downstream tasks.

The chapter is structured as follows. After giving an overview of the problem context in Section 4.1, we define the task of singing voice reconstruction (SVR) in Section 4.2. In Section 4.3, we give an overview

of related work on LMs and describe the signal characteristics of two LM used in this study, including best practices for their handling during a recording session. We then introduce our dataset, the *Larynx Microphone Singer-Songwriter Dataset* (LM-SSD) in Section 4.4. Section 4.5 describes our baseline system for SVR from LM recordings. The system is evaluated in Section 4.6 using both a listening test and an objective evaluation based on a lyrics transcription downstream task. We conclude with a discussion of the broader implications for the topic of this thesis and an outlook on future work in Section 4.7.

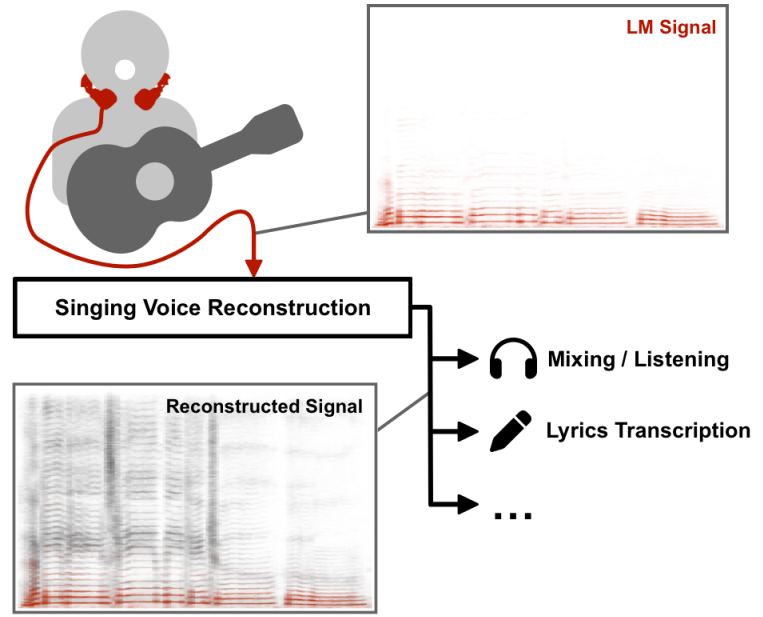
4.1 Background

Many tasks in the field of MIR, like automatic music transcription or F0 analysis, become significantly easier when separated, monophonic audio signals of individual instruments and voices are available. Similarly, music post-production and mixing rely on multi-track recordings to selectively apply audio effects and balance levels. The singing voice is often particularly challenging in those scenarios, due to its large dynamic range and possibility for nuanced expression. While in popular music the separation into monophonic multi-track signals can often be accounted for in the recording process, e.g., by consecutively recording individual singers with a close-up microphone (CM), this is not always possible or desirable. In order to facilitate the natural interaction (in terms of timing, expression, intonation, etc.) between musicians, voices, and/or room acoustics, it may instead be preferable or even necessary to record multiple musicians or instruments at the same time and in the same space. This way, a choir performing in a church, an ethnomusicological field recording of a vocal ensemble, or a singer-songwriter accompanying themselves on a guitar can become challenging for computational analysis and post-production.

Larynx microphones (LMs) provide a practical way to obtain almost crosstalk-free signals of the voice by picking up vibrations at the throat. By design, such a sensor is insensitive to sound waves transmitted through the air and therefore to other interfering sound sources in the environment. However, the LM signal quality is typically degraded by a limited frequency response as well as missing effects of the vocal tract, i.e., the contribution of the oral and nasal cavity responsible for vowel and consonant formation, which are only indirectly picked up at the throat. This limits applications of LMs to cases where audio quality is not a primary concern, like radio communication in noisy environments.

In this chapter, we explore the task of singing voice reconstruction (SVR), which aims at obtaining high-quality recordings of singing voice using impaired signals as the input. By using the new term SVR, we want to emphasize important differences to related and established tasks in the research fields of signal processing and MIR, like speech enhancement or singing voice synthesis. In particular, the focus on the singing voice requires careful treatment and preservation of expressive parameters, while at the same time music recordings in general often include many highly correlated interfering sources. With a focus on SVR from larynx microphone signals (referred to as LM-SVR in the following), we consider a particular scenario in this chapter: Reconstructing a high-quality singing voice recording—as it could be recorded with a typical CM in ideal conditions—from a monophonic LM signal, thus circumventing the problem of

Figure 4.1: Conceptual overview of SVR from LM signals. Time-frequency representations of an exemplary LM signal and the corresponding reconstructed signal are depicted in red and gray, respectively.



crosstalk during recording, while faithfully retaining the nuanced vocal expression captured in the original signal. Such a reconstructed recording could then be used for downstream applications like mixing or computational analysis, as outlined in Figure 4.1. For the latter, especially analysis tools based on large pre-trained models, including speech recognition systems [153] and multi-modal large language models [101], often benefit from high-quality inputs and tend to be difficult to fine-tune for degraded signals such as LM recordings.

Beyond introducing this novel MIR task, we make three main contributions. First, we introduce the *Larynx Microphone Singer-Songwriter Dataset* (LM-SSD), a collection of twelve pop songs performed by four amateur singers accompanied by an acoustic guitar, comprising over 4 hours of unique recordings of LM and corresponding CM signals. This dataset may facilitate research on LM-SVR, but also other tasks like source separation or singing voice analysis. Both LM and CM signals of the singing voice are available for each recording, and each song is recorded multiple times with and without guitar crosstalk on the CM. We further include mixes of each song, both with and without audio effects. Second, we describe a baseline LM-SVR system which is inspired by source-filter models of the vocal tract, using the LM signal as a source signal and learning the control parameters of a time-variant filter with a neural network. By choosing a model-based approach, the LM input signal is not resynthesized but rather enhanced by adding missing signal components and successive filtering, so that the degraded tone attributes timbre and loudness are reconstructed, while the pitch is faithfully reproduced. Third, we evaluate the baseline LM-SVR system with a listening test and show that our approach improves the subjective quality of the LM recordings, while also improving objective speech intelligibility in the exemplary downstream task of lyrics transcription with a state-of-the-art large-scale speech recognition model.

4.2 Singing Voice Reconstruction (SVR)

We define singing voice reconstruction (SVR) as the task of obtaining a natural sounding, artifact-free, and broadband signal without crosstalk from an impaired recording of the singing voice, without changing the original expression in the recording. Typical impairments include adverse recording conditions or limitations of the microphone. With that, SVR is related to several established research areas in signal processing and MIR.

Speech Enhancement (SE) focuses on the suppression of artifacts, noise, and other interfering sources from an otherwise high-quality speech signal [219]. While classical SE methods often rely on additional (e.g., spatial, see [10]) information about the recorded signals, state-of-the-art SE systems use data-driven approaches to encode the impaired signal in a typically lower-dimensional latent representation and then decode a new, clean speech signal from that (e.g., [187]). Similar methods have been employed for SVS, where an interpretable latent representation may enable general-purpose synthesis models [24, 25]. Naturally, such systems depend critically on the expressivity of the latent representation and the distribution of training examples, as the decoder may not learn to generate a signal with characteristics that are not observable in the training set. This lack of control is particularly problematic for singing voice signals [23], as important expressive parameters like F_0 are highly variable within individual notes [32] and may not be accurately reproduced by SE or SVS systems, even when F_0 is explicitly given in the latent representation [25].

Furthermore, in music, desired and interfering sources may be highly correlated, for example when multiple musicians are singing in unison. This problem is subject of the MIR task of musical source separation (MSS, [20]). State-of-the-art systems often introduce characteristic artifacts and reach signal-to-distortion ratios of around 8 dB for vocals on popular music recordings [128], so that separately recording individual instruments and voices is still preferable for many use cases. This recording-time separation can be achieved, for example, with LMs. Since these produce band-limited sensor signals, LM-SVR is also related to Blind Bandwidth Extension (BBWE). BBWE aims for the reconstruction of high-frequency content from a clean but band-limited audio signal without additional side information. Systems often focus on a specific signal domain like speech, where successful approaches model aspects of speech production [173]. Apart from subjective quality, BBWE can also improve performance of downstream tasks like speech recognition [106]. Adapting such an approach to the singing voice and larynx microphones is one objective of SVR as introduced in this chapter.

4.3 Larynx Microphones (LMs)

Larynx microphones (LMs), also called throat microphones, are a type of contact microphone. In general, contact microphones are designed to record vibrations of the surface they are attached to, while being insensitive to sound waves transmitted through air. For LMs, this is typically achieved using a

piezo-electric sensor placed on the skin of the neck. This way, one can obtain well-separated signals of individual speakers or singers in all kinds of acoustic conditions. The quality of the recorded signal depends on two factors: The properties of the sensor itself and the way how vibrations are propagated from the source through tissue and/or bones to the receiver. Notably, while the vocal fold vibrations are predominant at the neck, some influences of the vocal tract producing formants and consonants are also present as vibrations in neck tissue [141], so that for example speech recorded with an LM can become intelligible. A detailed account of the signal characteristics of two particular LM models used for our dataset is given in Section 4.3.2. To get a subjective impression of the signal qualities, we also refer to the online examples accompanying this chapter¹⁶.

Another sensor type for LMs has been subject of research on Non-Audible Murmur (NAM) [134]. Here, the goal is to pick up whispered speech, which is achieved with a condenser microphone embedded in a soft material that is attached to the skin [192]. Similarly, bone conduction microphones are contact microphones that are adapted to the specific impedance of the skull [75]. While the recording quality for both NAM and bone conduction microphones is generally comparable to LMs and especially bone conduction microphones offer more flexibility in their positioning [117], they lack the robustness and widespread availability of LMs.

A different technique for picking up the vocal fold vibrations is electroglottography [76], where the periodic opening and closing of the vocal folds is measured by picking up the resulting impedance change through electrodes placed near the larynx. Recordings with this method are particularly insensitive to any influences of the vocal tract and do not contain enough information to reconstruct a speech or singing signal, but can for example be used to measure F0 [7].

LMs have shown their utility in a number of applications in speech and music processing. [62] showed that the information from LMs can improve speech recognition in noisy environments. [7] used LM signals for F0 estimation, which recently received renewed attention in the context of pitch and intonation analysis of Western choir recordings [163]. [170] showed how LM recordings can aid musicological research, e.g. to computationally determine the tonal organization of traditional Georgian music [166, 172].

4.3.1 Compared LM Models

In order to analyze differences between sensors, their handling, and their utility for singing voice recordings, we used two LM models for recording the dataset: The commercially available *Albrecht AE-38-S2a* (LM-A) and a self-made microphone based on *TE Connectivity CM-01B* piezo-electric vibration sensors (LM-B). A direct comparison of their properties can be found in Section 4.3.2.

LM-A: Albrecht AE-38-S2a. LM-A is a commercially available device built for radio communication in the security sector. It has two contact microphones at either end of a size-adjustable neck brace (see

¹⁶ <https://audiolabs-erlangen.de/resources/MIR/LM-SVR/>

Figure 4.2: Photograph of the recording setup (top) and detailed depiction of the used LMs (bottom). LM-A: Albrecht AE-38-S2a larynx microphone, LM-B: Self-made larynx microphone with TE Connectivity CM-01B sensor, CM: Close-up microphone (Neumann U87), GP: Guitar Pickup (AMG Electronics C-Ducer), GL/GR: Guitar Stereo Left/Right (AKG C414).



Fig. 4.2, bottom left). With the adjustable brace it suits a variety of neck sizes, even though it tends to have a looser fit on smaller necks, resulting in a higher possibility for movement-induced noise. For all recordings, we aimed to position the sensors as close to the larynx as possible while allowing the musicians to be comfortable while singing. The two analog sensor signals are electrically summed by a connection in series, but the manufacturer provides no details on frequency range or other properties of the sensor. We adapt the 3.5mm TRS connector of the device to XLR using a *Røde VXL+*, which also converts 48V phantom power to the required supply voltage of 3.8V.

LM-B: CM-01B. LM-B is a self-made larynx microphone using two piezo-electric vibration sensors (*TE Connectivity CM-01B*) and a 3D-printed neck brace. The used sensor is optimized for detecting body sounds, whose vibrations are transmitted via a small rubber pad on the device. It is marketed with a frequency range of 8 to 2200 Hz (± 3 dB). To achieve comparability with LM-A, we use two sensors on either end of the brace and digitally sum the signals after recording. The sensors are attached to the neck brace with a screw that is glued onto the backside of the sensor and a spring that pushes the sensor lightly towards the skin of the neck (see Fig. 4.2, bottom middle). The sensor contact point can be adjusted and is further away from the larynx than with LM-A, which singers reported to be more comfortable. The neck brace is not adjustable to different neck sizes, but it can be printed in different dimensions, which also makes it possible to specifically fit the brace for individual persons.

4.3.2 Signal Characterization

Obtaining objective measurements of the LM voice signal characteristics is challenging, as the properties of the transmitting medium and the vibration source(s), as well as possible losses at the contact point have to be taken into account. A direct transfer function between source and receiver cannot be measured, because the ground truth “source signal” is not available. Instead, we consider the relative transfer function (RTF) between CM and LM as a first indication of the similarity between the LM signals and traditional microphone recordings. Using an unbiased estimator, we calculate RTF estimates for the two LM models using the crosstalk-free recordings from our introduced dataset (cf. Section 4.4.1).

Let $\mathcal{X}_{\text{CM}}, \mathcal{X}_{\text{LM}} \in \mathbb{C}^{M \times K}$ be the STFT time-frequency representations of two simultaneously recorded CM and LM signals of equal length, calculated using a Hann window with window size N , resulting in K frequency bins, and hop size H , resulting in M time frames. Furthermore, with $\circ$ denoting element-wise multiplication, we can define

$$\begin{aligned} S_{\text{CC}} &= |\mathcal{X}_{\text{CM}}| \circ |\mathcal{X}_{\text{CM}}|, \\ S_{\text{LL}} &= |\mathcal{X}_{\text{LM}}| \circ |\mathcal{X}_{\text{LM}}|, \\ S_{\text{CL}} &= |\mathcal{X}_{\text{CM}}| \circ |\mathcal{X}_{\text{LM}}|, \end{aligned} \quad (4.1)$$

as the frame-wise power spectrum of the CM signal and the LM signal, and the frame-wise cross-power spectrum between CM and LM signal, respectively. Without access to the true RTF $H_{\text{CL}} \in \mathbb{R}^K$ an unbiased estimate $\hat{H}_{\text{CL}} \in \mathbb{R}^K$ between CM and LM [55, Eq. 31] can be calculated with

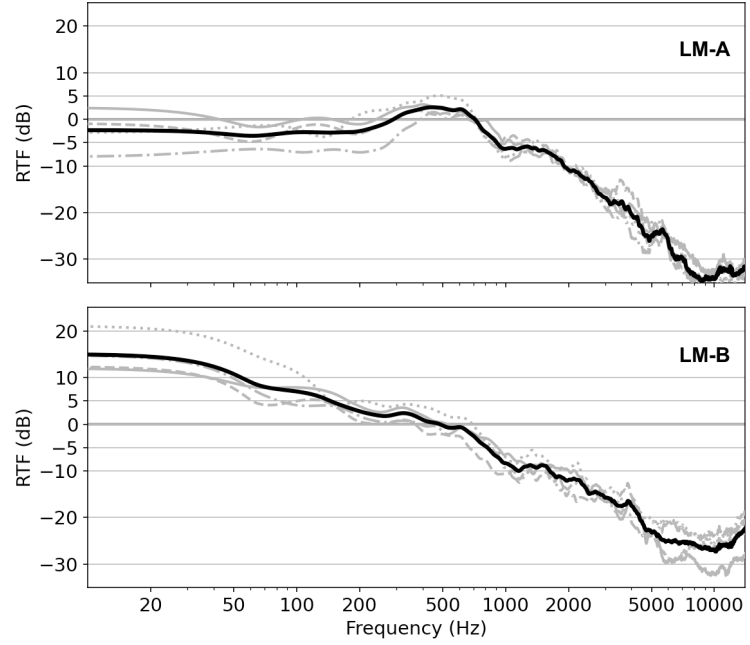
$$\hat{H}_{\text{CL}} = \frac{\{S_{\text{CC}} \circ S_{\text{CL}}\}_M - \{S_{\text{CC}}\}_M \circ \{S_{\text{CL}}\}_M}{\{S_{\text{CC}} \circ S_{\text{CC}}\}_M - \{S_{\text{CC}}\}_M \circ \{S_{\text{CC}}\}_M}, \quad (4.2)$$

where $\{\cdot\}_M$ denotes taking the arithmetic mean over the M time frames. For the following analysis, we use $N = 32768$ and $H = 8192$, with the signals at a sampling rate of 44.1 kHz. The RTFs are smoothed along the frequency axis with a Hann window of size 47 (a bandwidth of approx. 63 Hz).

Results are presented in Figure 4.3 for signals from individual singers (in gray) and the average RTF over all singers (in black). An RTF value around 0 dB signifies that CM and LM signals tend to have similar energy at a given frequency. The RTF for LM-A remains around 0 dB below 700 Hz and drops off at higher frequencies with -9 dB per octave on average. LM-B boosts low frequencies with up to $+20$ dB below 20 Hz and the RTF continuously drops towards high frequencies with around -6 dB per octave. Between 80 to 700 Hz, the approximate range of the F0 of male singing voices, energy levels of LM-A and LM-B are fairly similar to the CM signal. The estimates for different singers are similar within ± 5 dB in the relevant frequency range, which is an indication that both LM models are fairly robust w.r.t. fit and exact positioning.

The RTF does, however, not allow conclusions about noise levels and distortion of the LM relative to the CM signal. For that, we can additionally measure the coherence between the two signals, showing whether

Figure 4.3: Relative transfer function (RTF) estimates w.r.t. CM for LM-A (top) and LM-B (bottom). RTF estimates for individual singers are shown in gray (1M: solid, 2M: dashed, 3F: dotted, 4F: dash-dotted), the black line indicates the mean RTF between singers for each LM model.



they are linearly related at a given frequency. The frequency-dependent coherence estimate $\hat{C}_{CL} \in [0, 1]^K$ is given by

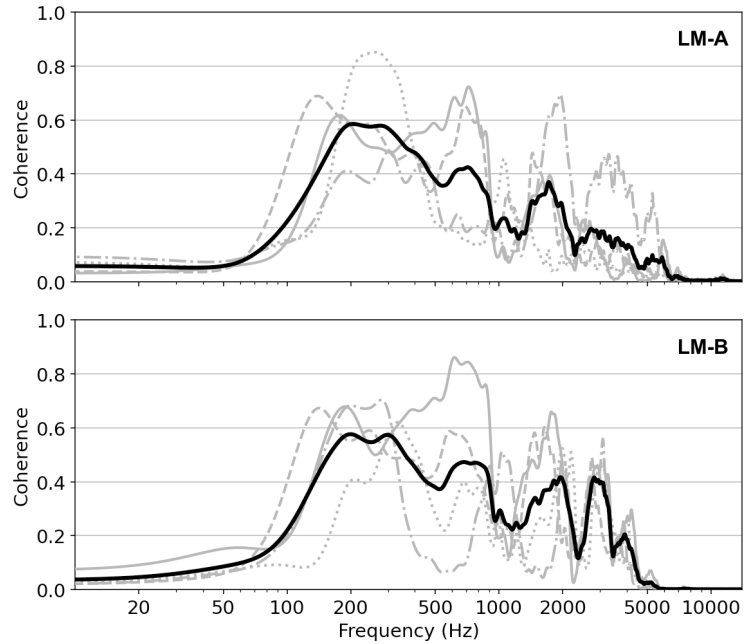
$$\hat{C}_{CL} = \frac{\left| \{ \overline{\mathcal{X}_{CM}} \circ \mathcal{X}_{LM} \}_M \right|^2}{\{ S_{CC} \}_M \circ \{ S_{LL} \}_M}, \quad (4.3)$$

where $\overline{\mathcal{X}}$ is the element-wise complex conjugate of $\mathcal{X}$ [91]. We use the same STFT parameters and smoothing as for the RTF. Choosing a large N and smoothing along the frequency axis contributes to a reduction of bias and variance, respectively, in the coherence estimate. Note that the naturally occurring time delay between CM and LM signals is small enough to have no relevant influence on the estimates. When the maximum coherence of 1 is achieved, a linear filter exists to calculate the corresponding CM signal from an LM recording and vice versa. Conversely, minimum coherence of 0 is reached when no linear relationship exists, e.g., when one or both signals are uncorrelated noise or one signal contains non-linear distortions.

Figure 4.4 shows coherence estimates calculated according to Equation 4.3 for the signals from individual singers (in gray) and their average (in black). Between 80 Hz and 3.5 kHz (the range of F0 and first few harmonics of the recorded voices), CM and LM signals are somewhat linearly related, but the coherence rarely exceeds 0.6, hinting at the presence of noise and distortion in the LM recordings. The low coherence at low frequencies is due to the singing voice not being present on the CM below F0, while the LMs, particularly LM-B, might still record relevant signal. The mean coherence for LM-B is slightly larger towards high frequencies, which can at least partially be attributed to a stronger distortion in the LM-A signal when singing with higher intensity¹⁷.

¹⁷ cf. for example Song BB on the accompanying website for a qualitative comparison of the different distortions in T1 (LM-A) and T4 (LM-B).

Figure 4.4: Coherence estimates w.r.t. CM for LM-A (top) and LM-B (bottom). Coherence estimates for individual singers are shown in gray (1M: solid, 2M: dashed, 3F: dotted, 4F: dash-dotted), the black line indicates the mean coherence between singers for each LM model.



Finally, we also measure the sensitivity of each LM w.r.t. interfering sound transmitted through the air. We consecutively record the LM signals while an external noise source is playing and while the wearer is singing, both reaching the same sound level at a fixed measurement position in the room. The level difference between these two recordings gives a relative measure for the “crosstalk sensitivity”, indicating the dampening of surrounding sounds compared to the voice of the wearer. We estimate -60 dB for LM-A and -55 dB for LM-B.

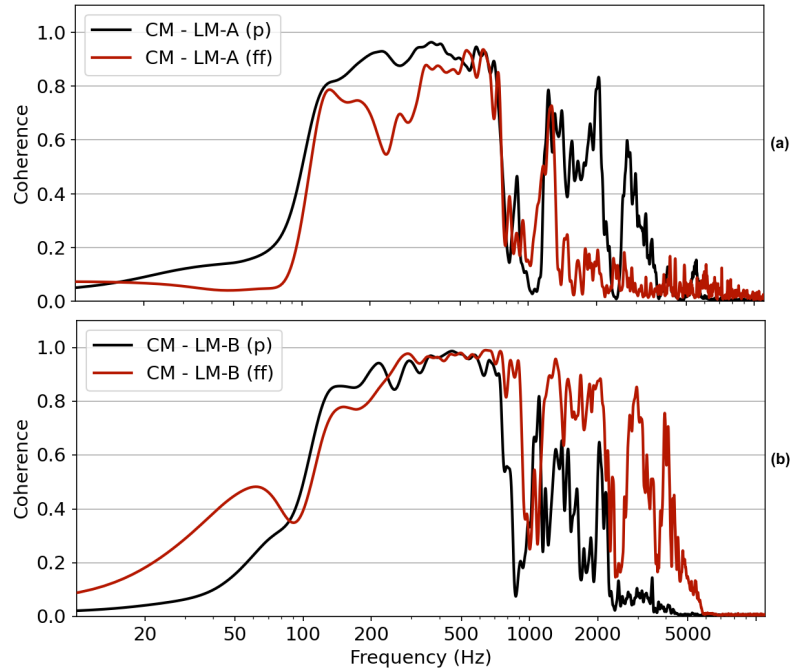
While the objective properties are similar for both LM models, the high-frequency distortion of LM-A at higher volumes also subjectively reduces the signal quality. Furthermore, singers reported higher wearing comfort of LM-B, as the contact point of the sensors is less close to the throat, while the self-made construction offers higher flexibility due to the possibility of customizing the neck brace and individually recording left and right sensor signals. On the other hand, LM-A requires only one recorder channel and is more robust in handling (as it is constructed as a single piece without removable parts) and fit (due to the adjustable brace), which can be an advantage, e.g., for field recordings.

4.3.2.1 Singer Dynamics

The two LM models exhibit different behavior across dynamic levels of the singer’s voice. Particularly at higher dynamic intensities, audible distortion can be observed, most notably with LM-A. For example, in the song BB¹⁸, a direct qualitative comparison of T1 (LM-A) and T4 (LM-B) highlights these differences between the LM models. To obtain a quantitative perspective on this effect, we separately recorded a male singer performing ascending and descending major scales at the two dynamic levels *piano* (*p*) and

¹⁸ <https://www.audiolabs-erlangen.de/resources/MIR/LM-SVR/song/BB>

Figure 4.5: Comparing coherence estimates between CM and LM-A (a) / LM-B (b) with two different singing intensities: *piano* (black) and *fortissimo* (red).



fortissimo (*ff*). This experiment was repeated for both LM-A and LM-B with a simultaneously recorded CM reference signal, resulting in four paired LM/CM recordings of approximately one minute each.

Coherence estimates between these recording pairs are shown in Figure 4.5. The comparison indicates that, for LM-A, coherence generally decreases toward higher frequencies at *ff* intensity, whereas for LM-B, coherence increases. In the case of LM-A, this behavior may be attributed to additional non-linear distortion at increased signal levels. For LM-B, in contrast, the higher level appears to improve the signal-to-noise ratio of higher-frequency throat vibrations.

4.4 LM Singer-Songwriter Dataset

The Larynx Microphone Singer-Songwriter Dataset (LM-SSD) is a collection of twelve pop songs that we recorded with four different amateur singers accompanied by a guitar, featuring a solo singer for nine songs and a duet for three songs. In total, the dataset consists of 72 takes with a total playback duration of 250 minutes, as detailed in Table 4.1 below. While similar datasets exist for speech (e.g., [37, 204]), this is, to our knowledge, the first systematic dataset with singing voice recordings using LMs and a synchronous, high-quality, and crosstalk free CM signal. Beyond research on LM-SVR and related tasks, this makes LM-SSD suitable for research on analyzing LM signals directly (e.g., for F0 estimation). Additional mix tracks for each song, as well as annotated lyrics furthermore enable use cases like experiments with source separation (possibly informed by LM signals) or lyrics transcription. In the following, we will describe

the used sensors (Section 4.3.1), the recording process (Section 4.4.1), and the content of the dataset (Section 4.4.2) in detail.

4.4.1 Recording Setup and Procedure

LM-SSD comprises recordings of two male (denoted 1M and 2M) and two female (3F and 4F) university students. All musicians are amateur singers without formal music training and only limited stage or recording experience and consented in writing to the processing and publication of the recordings. Each singer selected three pop songs they felt comfortable with in terms of vocal range and techniques. While 1M and 2M accompanied themselves on the guitar, 3F and 4F were accompanied by a second musician, while 3F also sang in duet with 1M. Two of the songs have lyrics in German language, while the others are in English. An overview of the songs in the dataset is given in Table 4.1.

The musicians were recorded in a studio room with little reverberation and optimized acoustics for pop music recording. As depicted in Figure 4.2 (top), the recording comprises a traditional microphone setup, including a close-up vocal microphone (CM; Neumann U87, set to cardioid) with a pop filter, a stereo close-up microphone pair for the guitar (GL and GR for left and right channel, respectively; AKG C414, set to cardioid) and a guitar pickup microphone (GP; AMG Electronics C-Ducer). In addition, all singers were wearing one of the two LM models (see Section 4.3.1) at a time to record their voice. Preliminary experiments showed that the optimal positioning of the LM on the neck would be compromised if both LMs were worn at the same time, as both record the cleanest signals when positioned high on the neck and as close as possible to the larynx without becoming uncomfortable for the singer. Note that, depending on the singer’s distance to the CM, there is a small time-varying delay between CM and LM signal¹⁹, which we do not compensate for in the dataset.

In total, six takes were recorded for each song, three with LM-A (T1–T3) and three with LM-B (T4–T6). Of those takes, the first of each group (T1 and T4) was played with live guitar accompaniment, i.e. the guitar playing in the same room as the singer (singers 1M and 2M accompanied themselves, whereas 3F and 4F were accompanied by a second musician). In takes T2, T3, T5, and T6, singers performed the same song again but with the guitar pickup signal from takes T1 and T4, respectively, played back to them over headphones. Thus, only takes T1 and T4 have crosstalk from the guitar on CM. The presence of crosstalk is denoted by C1 in the naming scheme (see Table 4.2), while takes without crosstalk are marked with C0. By including these different recording conditions in the dataset, it contains both a real-world recording scenario (C1) which can for example be used for validation of an LM-SVR system, as well as a crosstalk free reference (C0) which can serve as training data. Similarly, the songs SG, OC, and PL are performed by two singers at the same time, which provides a scenario for computational analysis of unison singing.

¹⁹ A distance of 0.3m to the CM would result in a delay of around 1ms due to the additional propagation time through the air.

ID	Song name	Original Artist	Singer	Takes LM-A		Takes LM-B		Duration (mm:ss)
				C1	C0	C1	C0	
AA	All Alone	Michael Fast	1M	1	2	1	2	27:03
TS	The Scientist	Coldplay	1M	1	2	1	2	21:37
YF	Your Fires	All The Luck In The World	1M	1	2	1	2	24:21
DL	Dezemberluft*	Heisskalt	2M	1	2	1	2	14:47
BB	Books From Boxes	Maximo Park	2M	1	2	1	2	17:39
NB	Narben*	Alligatoah	2M	1	2	1	2	11:47
SG	Supergirl	Reamonn	3F, 1M	1	2	1	2	26:34
OC	One Call Away	Charlie Puth	3F, 1M	1	2	1	2	19:32
PL	Past Life	Trevor Daniel & Selena Gomez	3F, 1M	1	2	1	2	17:45
CC	Chasing Cars	Snow Patrol	4F	1	2	1	2	28:10
BT	Breakfast At Tiffany's	Deep Blue Something	4F	1	2	1	2	22:16
LL	Little Lion Man	Mumford & Sons	4F	1	2	1	2	19:06
Total				12	24	12	24	250:37

Table 4.1: Overview of the songs and takes in LM-SSD (Songs marked with * have German lyrics).

4.4.2 Dataset Content and Structure

LM-SSD can be accessed and explored in two ways: By using our accompanying website²⁰ or by downloading the complete dataset²¹. The website allows to play back and compare all provided signals using a multi-track player [228] directly in the browser. The complete dataset contains 348 audio files in the WAV format with single-channel audio at a sampling rate of 44.1 kHz. The files follow the naming convention `SSD[UID]_[SongID]_[Type]_[Crosstalk]_[Singer]_[Take].wav`, where each placeholder in square bracket is filled with corresponding values as summarized in Table 4.2. Furthermore, lyrics for each song, as sung in the recordings, are provided in the dataset with filenames `[SongID].txt`.

In addition to the raw microphone signals, we provide two stereo mixes for takes T1 and T4. For the first mix setting (**MixA**), we use no effects and just combine CM, GL and GR with appropriate levels and panning. In the second mix setting (**MixB**), we apply additional compression, reverb, equalization, and a saturation effect to CM, GL and GR, creating a genre-typical mix. While mixes use an additional limiter, raw microphone signals are normalized to 0 dB true peak. Apart from mixing, all takes remain unedited, so that occasional inconsistencies or mistakes are preserved in the performances.

²⁰ <https://audiolabs-erlangen.de/resources/MIR/LM-SVR/>

²¹ <https://audiolabs-erlangen.de/resources/MIR/LM-SVR/lm-ssd.zip>

Field	Description	Values
UID	Unique numerical identifier for a take across songs	001 - 072
SongID	Two-letter abbreviation of the song	cf. Table 4.1
Type	Microphone type or mix setting	LM-A, LM-B, CM, GP, GL, GR, MixA, MixB
Crosstalk	Whether guitar crosstalk is present on CM (C1) or not (C0)	C1, C0
Singer	Singer identifier (with gender)	1M, 2M, 3F, 4F
Take	Take number for the given song (T1-T3 use LM-A, T4-T6 use LM-B)	T1 - T6

Table 4.2: Dataset dimensions and naming scheme.

4.5 Baseline System for LM-SVR

In this section, we illustrate how our dataset could be used for training a data-driven baseline LM-SVR system with its LM signals and corresponding high-quality CM singing voice recordings. As shown for the tasks speech enhancement [187] and singing voice synthesis [25], large generative models require careful conditioning of the decoder to preserve the desired characteristics of an encoded example. To avoid this issue for the fundamental frequency in particular, we base our approach on the direct processing of an LM input signal with a system that is inspired by source-filter models for speech and singing production. Using DDSP [42], we train a Neural Network (NN) that controls several digital signal processing (DSP) building blocks to transform the LM input into an output as close as possible to our reference CM signal. This approach has several advantages. First, we can keep the number of trainable weights of the NN relatively low so that good results can be achieved with a limited amount of training data. For example, in our experiments below, we train singer-specific models with as little as 20 minutes of training data. Second, the musically motivated model architecture is inherently interpretable, so that the contributions of individual model components can be inspected individually to analyze certain reconstructed characteristics, like vowel formants or unvoiced consonants. Finally, signals with different lengths or a changing frame size can easily be accounted for without retraining the model.

4.5.1 Related Work

The term DDSP was introduced by [42] for the concept of using fixed DSP building blocks in NN architectures. These building blocks, while having no trainable weights themselves, are ensured to be differentiable which allows for training an NN (utilizing standard back-propagation) that in turn controls the DSP blocks. The original DDSP system [42] uses Spectral Modeling Synthesis (SMS) [186] for the generation of musical instrument sounds using a superposition of sinusoidal oscillators and filtered white

noise. The NN outputs the time-varying frequency and amplitude parameters of the oscillators as well as band-wise magnitudes from which the noise filter is designed.

With its interpretability and natural way of including domain knowledge in the model architecture, DDSP has recently been adapted for various audio synthesis task, e.g., to generate piano sounds [155]. However, sinusoidal synthesis is not particularly suitable for SVS, as it does not sufficiently constrain the output to produce consistent phonemes [5]. A source–filter model can help with singing-specific constraints as demonstrated for the task of musical source separation by [175], where they modeled individual singers in a mixture with a synthesis module followed by a time-varying “vocal tract” filter. Similarly, [230] generate time-domain singing voice signals from a time-frequency representation using a DDSP source–filter model, where a synthetic sawtooth waveform is filtered and enriched with subtractive noise synthesis.

4.5.2 Model Architecture

Let x_{LM} and x_{CM} denote the LM input signal and the corresponding reference CM signal, respectively. Furthermore, let y be the output signal of our model. The main idea of our approach is to apply a filter to x_{LM} and add noise-like signal components, so that y becomes similar to x_{CM} with respect to a metric $\mathcal{L}$. Our model consists of three main DSP building blocks inspired by a source–filter model, similar to [230]. However, instead of using a synthetic source, we employ x_{LM} as our source signal for the subsequent “vocal tract” filter. Figure 4.6 shows an overview of the signal flow and a visualization of intermediate signals within the model. The function of the main DSP building blocks can be summarized as follows:

1. **TI-FIR Filter:** The time-invariant FIR (TI-FIR) filter equalizes x_{LM} to match the overall frequency characteristics of y as closely as possible. Subsequently, the signal is normalized for further processing.
2. **Noise Synth:** The task of the noise synthesizer is to add noise-like signal components that are missing in x_{LM} . The main task of the noise synth is the reconstruction of fricatives, i.e., consonant sounds like [s] or [f] that are produced with the mouth and are therefore more or less missing in an LM recording.
3. **TV-FIR Filter:** The time-variant FIR (TV-FIR) filter resembles the vocal tract filter in the source–filter model. Its input signal is the superposition of the normalized TI-FIR filter and noise synthesizer outputs. With the time-dependency of the filter, it is possible to augment or add missing vocal formants in the input signal.

The parameters for these building blocks are provided by a control NN consisting of a Convolutional Neural Network (CNN) with residual connections (ResNet, [73]) that receives a mel-spectrogram of the complete x_{LM} as input. As the actual audio processing is done frame-wise in time domain, a subsequent recurrent neural network directly outputs the parameters for the DSP building blocks at each frame, while accounting for long-term dependencies between frames. The TI-FIR filter uses only the last recurrent

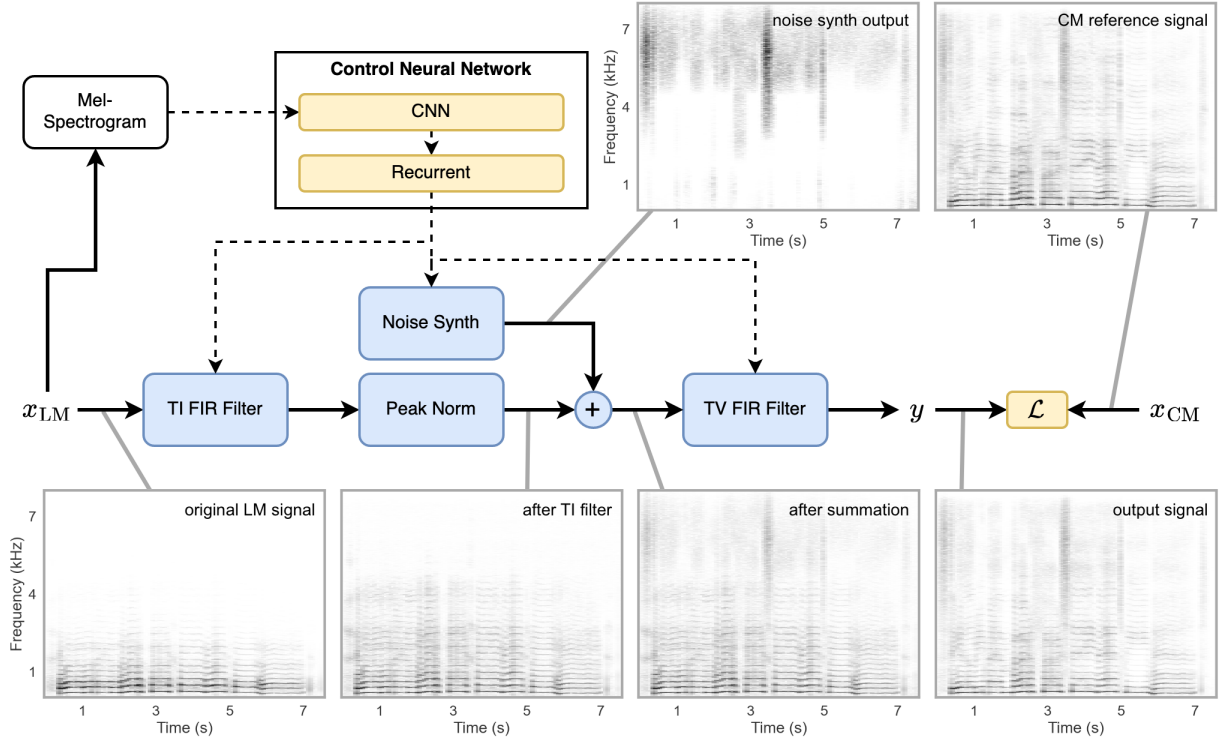


Figure 4.6: Architecture of the DDSP-based baseline system. Blue color is used for differentiable DSP building blocks, yellow color for NN building blocks with learnable parameters, and white color for fixed pre-processing steps. Control parameter flow is denoted with dashed line arrows, while solid lines indicate flow of audio signals. The spectrograms show signal content at the indicated position in the signal flow diagram. The shown example uses an excerpt from the LM-B signal of song AA T5 as the input signal x_{LM} and a corresponding model trained with the OF scenario.

network output. For the TI-FIR and the TV-FIR filters, the control parameters are given as magnitudes in 64 linearly-spaced frequency bands, from which a linear-phase FIR filter is designed with an interpolation strategy, as proposed by [42]. Similarly, the Noise Synth uses randomly generated white noise, equalized by a filter whose magnitude response is prescribed by only 10 bands. Here, the reduced number of bands keeps the Noise Synth from producing narrow-band, quasi-harmonic noise sounds, which may improve the performance in terms of $\mathcal{L}$, but can lead to audible artifacts in the output audio. With this DSP processing pipeline, the model is guaranteed to retain the F0 that is present in x_{LM} .

4.5.3 Implementation Details

We process audio with a sampling rate of 16 kHz and a frame size of 320 samples (resulting in a frame rate of 50 Hz). The input mel-spectrogram to the control NN with 64 frequency bands is calculated via an STFT using the Hann window of length 2048 samples and 320 samples hop size, resulting in the same number of frames as the time-domain processing. The time-varying control parameters of the TV filter and the Noise Synth are also updated at this rate. Therefore, this hyper-parameter influences the temporal resolution of the synthesis and thus the reconstruction quality. In initial experiments, we did not observe

improvements from increasing the frame rate. Using lower rates may reduce training and inference times, while producing lower quality output signals. The TI, TV and Noise Synth filters are designed using linear interpolation of the magnitudes to obtain a linear-phase filter of length 320 samples. Filters are applied using FFT convolution and overlap-add between successive frames. The specific architectures for CNN and recurrent part are adapted from the original DDSP code base and we also use the multi-scale spectral (MSS) loss function for $\mathcal{L}$ between y and x_{CM} with their proposed configuration. During training, we use random excerpts from the respective training set (cf. Section 4.6) of four seconds length and train the model for 40,000 steps.

4.6 Experiments

In this section, we evaluate our baseline model with regard to its suitability for LM-SVR, using training data from LM-SSD. In the experiments, we use only takes without guitar crosstalk on the CM microphone (C0 condition, see Section 4.4.1) and train separate models for LM-A and LM-B inputs, respectively. To illustrate how the dataset can be split in different ways, we define three training scenarios, corresponding to increasing difficulty of the LM-SVR task:

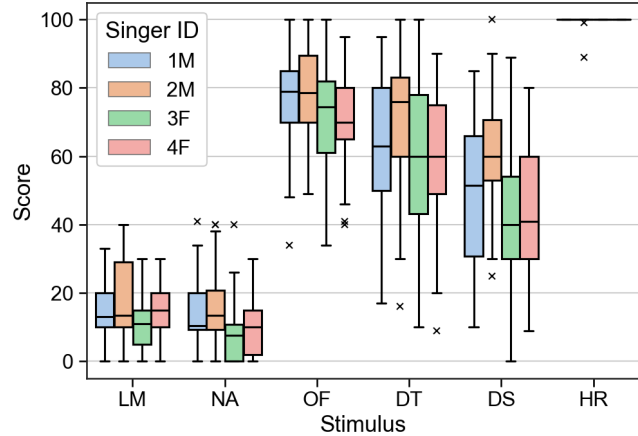
1. **Overfitting (OF):** The model is trained on all six C0 takes of a single singer (from three songs) and evaluated on a take that is part of the training set. While this is not a realistic use case for the system, these results demonstrate an “upper bound” on the reconstruction quality that our baseline can achieve.
2. **Different Take (DT):** Here, the evaluation take is excluded from the training set, resulting in five training takes for each singer. Since LM-SSD contains multiple takes per song, the model has seen a different take of the evaluation song during training.
3. **Different Song (DS):** All takes of the evaluation song are excluded from the training set, resulting in four training takes. This is the most challenging scenario we consider, as the notes and words sung in the evaluation recording have not been seen during training.

A complete training is done for each of these scenarios as described in Section 4.5.3. To keep the model in our illustrative example small and maintainable, we restrict the training scenarios to a separate model for each singer and LM type.

4.6.1 Evaluation

As an objective measure for the reconstruction quality of our baseline model, we use the Fréchet Audio Distance (FAD, [94]) between the CM signal and the reconstruction results. The FAD is based on the distance between learned general-purpose embeddings of the audio signals and we use the originally proposed *VGGish* [77] embeddings for our evaluation. Since this embedding distance is independent of

Figure 4.7: Listening test results according to stimulus and singer ID. LM: Larynx Microphone; NA: Naive Approach (linear filtering); OF, DT, DS: Overfitting, Different Take, and Different Song training scenarios; HR: Hidden Reference (CM signal)



the exact waveform shape and has been shown to correlate with human perception, it is more suitable for our experimental setup than for example the signal-to-distortion ratio (SDR, [218]), which requires a direct linear relationship between the compared signals. Beyond OF, DT, and DS, we include the conditions LM (for the unaltered LM signal) and NA (a naive reconstruction approach where the LM signal is filtered with the inverse of the RTF between the LM and CM microphones) in the comparisons. Using LM-A as the input signal, we obtain an average FAD of 0.76 for OF, 1.29 for DT, 1.51 for DS, 9.83 for LM, and 6.17 for NA (with LM-B: 0.77 for OF, 0.90 for DT, 1.17 for DS, 8.56 for LM, and 9.28 for NA). These results indicate that the baseline model generally is able to reconstruct some perceptually relevant characteristics that are present in the CM signal but not in the LM inputs.

In order to assess the subjective quality of the reconstructions, also incorporating perceptual issues like lyrics intelligibility, which the FAD does not account for, we conducted a listening test comparing the six different conditions from above using a multi-stimulus test with hidden reference (denoted HR). In an online experiment using the WebMUSHRA interface [174], participants were presented the reference (the original CM signals without crosstalk from LM-SSD) and asked to rate the other six stimuli (including HR) relative to that with respect to the overall perceived quality. All signals were played back with 16 kHz sampling rate and presented in mono via headphones. Rating was done on a continuous scale from 0 – 100, where 100 should be used for a signal that is indistinguishable from the reference and 0 should be used for a signal with subjectively bad quality. Participants were encouraged to make use of the whole scale in the test instructions. The test comprised 24 trials in random order using signals with a length between 6 and 10 seconds from all four singers and the different LM types. In total, 11 participants completed the test, two of which were excluded from our analysis as they did not consistently (in more than 85% of trials) rate HR with at least 90 points.

The results are presented in Figure 4.7, where the responses are split by singer identity. Generally, the training scenarios were rated in order of their difficulty level, where OF reached a mean score of 76, DT of 65, and DS of 49. This is in line with the occurrence of more artifacts like slurred consonants, particularly

Table 4.3: Word error rate (WER) of lyrics transcription with the Whisper [153] *medium* model for a selection of songs from LM-SSD. Song DL uses the dedicated German Whisper model.

Song	Singer	WER (%)				
		CM	LM	OF	DT	DS
AA	1M	1.83	72.56	1.83	22.56	20.73
TS	1M	2.82	31.69	2.82	24.65	33.10
DL	2M	2.16	10.81	2.70	5.95	7.03
BB	2M	7.40	11.25	9.65	11.90	21.22
SG	3F	3.70	11.11	5.76	11.11	57.61
OC	3F	3.31	84.30	4.96	12.40	58.68
CC	4F	0.49	92.65	0.49	29.41	91.67
LL	4F	1.98	85.71	1.98	15.87	69.44
Average		3.27	49.05	4.25	15.89	46.13

with the DS scenario²², which, according to comments from some participants, impaired text intelligibility in the excerpts. All training scenarios reach better scores on average than the LM signal (mean score of 17). The NA condition (mean score of 14) was not rated higher than the LM signal, possibly due to amplified noise. The stimuli with female singers (3F and 4F) tend to be rated slightly lower than male voice signals (mean score 50 vs. 57 over all stimuli). This discrepancy could follow from a lower F0 for males, which results in more partials in the sensitive frequency range of the LMs. However, these differences do not seem to be ameliorated by our baseline model.

4.6.2 Downstream Task: Lyrics Transcription

We consider lyrics transcription as an exemplary downstream task where LM-SVR may improve the performance of approaches that are optimized for high-quality singing recordings. In particular, we use *Whisper* [153], a state-of-the-art data-driven speech transcription system with a publicly available model that is pre-trained on over one million hours of data. Even though it is not explicitly trained for the transcription of lyrics, it has been shown to generally perform well for this out-of-domain task [12, 240]. Consequently, it also yields good results transcribing the singing voice from the CM signals in LM-SSD with a Word Error Rate (WER) of 3.27% on average. For the LM signals, however, it yields a worse WER on average (49.05%), possibly because the model relies on features from full-band, high-quality speech recordings, which are largely lacking in the LM signals. As retraining such a black box system with LM signals is not feasible, we explore how our LM-SVR baseline can improve transcription results, where a better performance would indicate that our model is able to reconstruct relevant characteristics of the singing voice signal. Detailed results for a selection of songs are presented in Table 4.3. The OF scenario (4.25% WER on average) reaches a similar performance to CM. Notably, the results with the DS scenario are sometimes inferior to the LM transcription (songs TS, BB, and SG in particular). While some errors are due to inconsistent spelling in the transcriptions (e.g., “supergirl” vs. “super girl”), these results indicate some unnatural distortion of formants and consonants in the model output when trained with

²² c.f. the example stimuli on the accompanying website

the DS scenario. This may inspire future work on LM-SVR which can benefit from our dataset and the introduction of objective evaluation tasks like lyrics transcription.

4.7 Further Notes

In this chapter, we have introduced LM-SVR as a novel MIR task. We proposed a DDSP baseline for this task, showing that a model trained with LM-SSD has the potential to improve subjective quality and machine intelligibility of LM singing voice recordings, while preserving expressive intonation from the original performance. The LM-SSD dataset, comprising LM and CM recordings of pop music with singer-songwriter instrumentation, was published alongside this work²³ to facilitate further research on related tasks where LM signals may be utilized. Furthermore, we characterized two commercially available LM models, supporting informed hardware selection for singing voice applications involving contact microphones.

There are several promising directions for future work. Since the DDSP baseline in this chapter primarily serves an illustrative purpose, we intentionally did not employ additional techniques to improve reconstruction quality, such as data augmentation or pre-training on larger, potentially lower-quality or synthetic datasets. With such methods, we expect that the quality of results, as well as the generalization to unseen speakers can be improved considerably. Another promising direction is the integration of controlled or learned distortion or saturation effects to selectively enrich the harmonic structure of the signal (see, e.g., [136]). By selectively increasing the harmonic content of the LM signal, strong high-frequency amplification in the TI-FIR filter could be avoided, potentially leading to more natural reconstructions and improved robustness to high-frequency noise, e.g., due to movement of the singer. More broadly, the reconstruction paradigm considered in this chapter may be extended to other scenarios involving degraded or corrupted tone attributes, for example instrument or singing voice recordings obtained with low-cost or poorly positioned microphones. However, it should be noted that, particularly for speech enhancement, where large amounts of high-quality training data are available, as well as for related tasks such as singing voice source separation, recent approaches based on large generative models achieve remarkable performance (see, e.g., [187] or commercial products like *AudioShake*²⁴, *Auphonic*²⁵, or *Music AI*²⁶). With sufficient and varied training data, as well as careful design of a pre- and post-processing pipeline, these methods can produce outputs without noticeable signal artifacts or unintended alterations of the tone attributes present in the input signal.

Nevertheless, model-based approaches can continue to fill an important niche in this context, particularly where controllability and efficiency (both in terms of data and computation) are critical. For example, the explicit modeling of time-invariant and time-varying FIR filters, as well as the noise synthesis component,

²³ <https://audiolabs-erlangen.de/resources/MIR/LM-SVR/>

²⁴ <https://audioshake.ai/>

²⁵ <https://auphonic.com/>

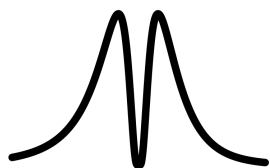
²⁶ <https://music.ai/>

Chapter 4. Singing Voice Reconstruction from Larynx Microphone Recordings

enables detailed control over individual elements that contribute to the reconstructed voice signal. This allows for directly modifying specific attributes, such as formant frequencies, in order to achieve desired tone qualities or stylistic effects, which we similarly explore in Chapter 6.

5 Intonation Adaptation in Multi-Track Recordings via Gradient Descent

This chapter is based on [177], which builds on a previously published conference paper [178], co-authored by myself, Sebastian Rosenzweig, and Meinard Müller. For both articles, I was the primary contributor to the study design, implemented and conducted all experiments, and prepared the manuscript. In [178], Sebastian Rosenzweig contributed to the development of ideas and manuscript preparation. Meinard Müller supervised the work and contributed throughout all stages of the research and manuscript preparation for both articles.



Intonation in ensemble performances on instruments with flexible tuning involves a complex interaction between musicians shaped by musical context, acoustic conditions, and each performer’s perception and preferences. In post-production, it is often desirable to compensate for unintended deviations while preserving expressive fluctuations and musically meaningful interaction between individual tracks and voices.

In this chapter, we formulate multi-voice intonation adaptation as a cost minimization problem, making three main contributions. First, we introduce a differentiable cost function that explicitly balances the adherence of each voice to an equal-tempered pitch grid and the harmonic fit between voices using sensory dissonance. Second, based on this differentiable cost function, we derive a gradient descent adaptation algorithm that produces smooth, time-varying pitch-shift curves without requiring score or note-level information. We show how a small set of interpretable hyperparameters, including initialization, stopping criterion, step size, and momentum, allows for a controlled trade-off between the compensation of unintended intonation deviations and the preservation of expressive fluctuations. Third, we evaluate our method on string, wind, and vocal quartet multi-track recordings through objective and subjective experiments, demonstrating quality comparable to a commercial pitch-correction baseline while offering particular advantages in handling intonation drift and modeling interactions between voices. Beyond these results, we focus on the conceptual aspect of making a typically heuristic post-production task transparent and controllable through an explicit cost-based optimization framework.

The remainder of this chapter is structured as follows. After providing context and a review of related work in Section 5.1, we introduce the intonation cost measure in Section 5.2. In Section 5.3, we present

our intonation adaptation algorithm and discuss the most important parameters and their effects. In Section 5.4 we analyze the cost gradients, present three real-world application scenarios, and evaluate listener preferences for different loss term combinations. Finally, in Section 5.5, we discuss the broader implications in the context of this thesis.

5.1 Background

The widely-used 12-tone equal temperament (12-TET) tuning system divides the octave into twelve semitones that are equally spaced on a logarithmic frequency axis. This allows musical instruments with fixed pitch, such as the piano, to play in any key. While 12-TET has become the dominant tuning system in Western music since the 18th century [9], the irrational ratio of $2^{1/12} \approx 1.0595$ between semitones implies that most intervals deviate slightly from the natural overtone relationships of harmonic sounds. In contrast, just intonation (JI) scales are explicitly constructed from intervals with small integer ratios, yielding tunings that align more closely with harmonic spectra. However, the absolute pitch values in JI scales depend on the chosen root note, requiring adaptation across keys and musical contexts.

Many musical instruments allow for continuous pitch variation or at least offer some flexibility in intonation²⁷. This enables musicians to dynamically adjust their pitch to achieve a context-dependent intonation beyond equal-tempered scales. In choir singing [69] or string ensembles [31], for example, an adaptation in response to concurrently sounding voices is often referred to as *vertical intonation* and JI intervals with minimal beating between overtones are considered desirable [4]. However, performers constantly have to balance this aspect with other intonation constraints like melodic intervals (*horizontal intonation*) and expressive pitch deviations such as vibrato [6, 56, 152] in their own and other voices. Together with practical limitations in pitch control and perception, this interaction can lead to subtle but perceptible unintended intonation deviations, including intonation *drift*, i.e., the gradual, systematic deviation from an original reference pitch.

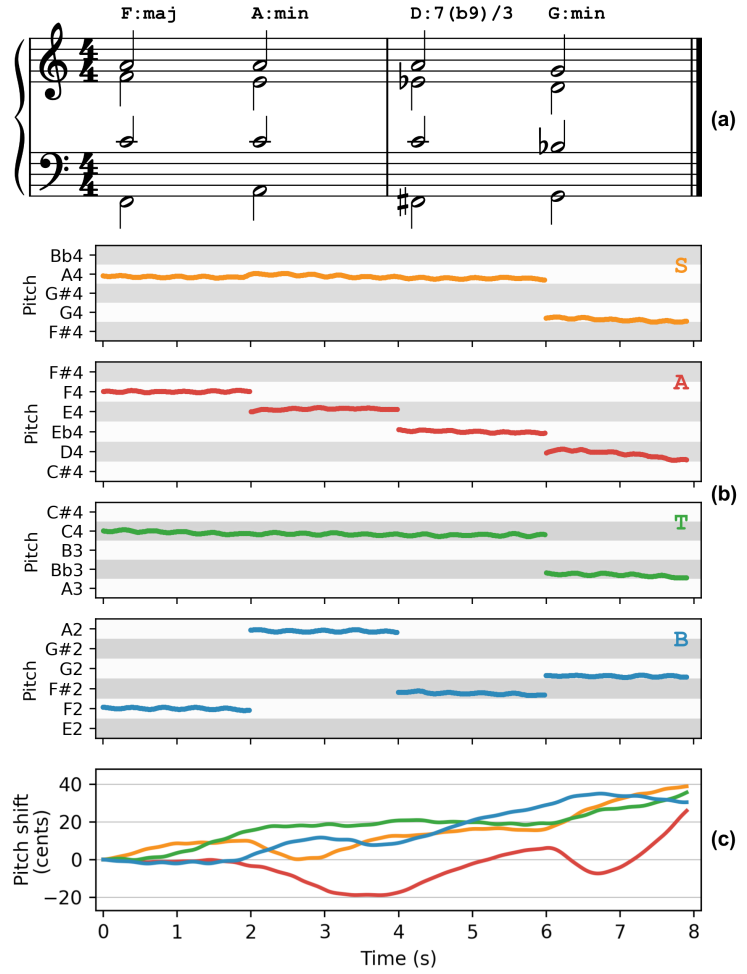
Figure 5.1 illustrates these phenomena in a synthesized four-part cadence. The score in Figure 5.1a specifies soprano (S), alto (A), tenor (T), and bass (B) voices with chord labels following [68]. The rendered performance in Figure 5.1b reveals several characteristic effects. All voices exhibit a global downward drift of approximately 50 cents relative to the 12-TET grid while largely preserving their relative intervals. In the S voice, the repeated A₄ notes are each played with different intonation, demonstrating a vertical intonation depending on the harmonic context. In addition, all voices show local expressive variations such as vibrato. Despite all these deviations, the harmonic structure remains perceptually stable.²⁸

²⁷ Here and in the following, we refer to a set of pitches that defines a musical scale as *tuning* and the realization of tuning by a performer as *intonation*.

²⁸ Audio examples and code are available online:
<https://audiolabs-erlangen.de/resources/MIR/2025-TASLP-Intonation>

Figure 5.1: Synthesized example cadence to illustrate aspects of intonation.

- (a) Sheet music with chord labels.
 (b) Fundamental frequencies (F0) of soprano (S, orange), alto (A, red), tenor (T, green), and bass (B, blue).
 (c) Pitch shift curve $p_v(n)$ for all voices $v \in \{S, A, T, B\}$ obtained with multi-voice gradient descent (hyperparameters $w = 0.33$, $L = 1$, $\mu = 50$, $\beta = 0.9$).



In the post-processing of multi-track recordings, a sound engineer may attempt to correct unintended intonation deviations via pitch shifting. However, this requires defining a target pitch that, analogous to live performance, depends on multiple interacting criteria. For example, one may wish to compensate for global intonation drift, often regarded as a flaw in a cappella singing [80, 185], while preserving expressive fluctuations and context-dependent vertical alignment. Figure 5.1c illustrates pitch-shift curves that achieve such drift compensation while retaining vibrato and chord-dependent pitch deviations that improve harmonic fit. Designing algorithms that automatically estimate time-varying pitch-shift trajectories without manual intervention, while still offering musically meaningful control, remains a challenging task.

In this chapter, we build upon our previous work [178] and formulate intonation adaptation with multiple voices as a cost minimization problem. This can be seen as a generalization of commonly used grid-based methods, for which the target pitches have to be predefined, whereas a cost-based formulation allows to flexibly and dynamically account for different influences on intonation, such as the interaction between voices. In this context, we make three main contributions. First, we propose a differentiable cost measure that combines two terms which are designed to have local minima at pitch shifts where a voice may reach desirable intonation. The first term captures a *tonal* aspect by encouraging proximity to an

equal-tempered pitch grid, reflecting the widespread use of equal temperament as an approximation of tonal organization in Western music [13]. The second term models a *harmonic* aspect by measuring the sensory dissonance [67, 190] between simultaneously sounding voices, thereby encouraging JI-like vertical alignment. With a weighted combination of these terms, our method enables flexible intonation adaptation between equal-tempered anchoring and harmonic adaptation. The tonal term ensures stable local minima even in harmonically complex or unstable chords, while the harmonic term allows musically meaningful deviations from the grid when supported by the ensemble context.

As a second main contribution and extension of [178], we introduce a gradient-based adaptation algorithm that minimizes the intonation cost to compute smooth, time-varying pitch-shift curves for each voice. By utilizing established methods from gradient-based optimization in this novel context, we show how hyperparameters such as the initial value, step size, stopping criterion and momentum provide direct control over the adaptation, ranging from a global drift compensation that preserves vibrato to more fine-grained local adaptation. The differentiability of the proposed cost function is central to this framework. Beyond gradient-based optimization, it is compatible with automatic differentiation tools and gradient-based learning frameworks. This opens perspectives for integrating perceptually motivated intonation objectives into hybrid signal-processing and deep learning systems.

As the third main contribution, we evaluate our approach using case studies on diverse multi-track recordings of string, wind, and a cappella vocal quartets. In comparison to existing commercial methods for intonation adaptation, our approach shows advantages in counteracting intonation drift and enables a flexible weighting of intonation aspects depending on user preference. In addition to our methodological contributions, a small-scale listening test suggests that combining both cost terms can yield improved results compared to using each term individually, with performance comparable to a commercial baseline. Given the inherently subtle and often subjective nature of intonation, we do not claim that our method is perceptually superior to these baselines. Rather, the primary strength of the approach lies in the flexible optimization framework that replaces a heuristic post-production workflow with an explicit cost-based formulation.

5.1.1 Related Work

The intonation adaptation strategies implemented in many commercial products like *Melodyne*²⁹, *AutoTune*³⁰, or *Flex Pitch*³¹ are typically based on an estimate of the fundamental frequency (F0) trajectory in a monophonic recording, with *Melodyne* also supporting F0 estimation for polyphonic instruments such as the piano. After (manually or semi-automatically) segmenting the F0 trajectory into individual notes, the signal is pitch-shifted so that the average F0 for each note approaches a predefined target value. This target can be chosen manually by the user or determined automatically from a predefined scale or musical

²⁹ *Celemony Melodyne*: <https://www.celemony.com/en/melodyne/what-is-melodyne> (accessed Nov 11, 2025)

³⁰ *Antares AutoTune*: <https://www.antarestech.com/> (accessed Nov 11, 2025)

³¹ *Apple Logic Pro X*: <https://www.apple.com/logic-pro/> (accessed Nov 11, 2025)

score. Additional heuristics are often used to preserve expressive pitch variations such as portamenti or vibrato. However, these approaches generally do not account for the interaction between multiple voices, which would require manual intervention by the user.

Dynamically tuning individual tones of simultaneously played intervals and chords has been explored in the context of digital synthesizers. Rule-based algorithms like *Groven.Max* [29] or *Hermodé Tuning* [129] analyze the current chord and adjust the F0 for each voice to approximate JI, compromising when just intervals between all pairs of notes are impossible. This problem can also be addressed by solving a quadratic program [198], distributing the deviation from JI evenly across the voices, with auxiliary constraints for temporal continuity within voices. These approaches operate primarily on symbolic representations or synthesized tones and assume explicit chord knowledge, making them less directly applicable to multi-track audio recordings with existing intonation deviations.

Data-driven approaches aim to infer favorable tuning from training examples and have mainly been explored for singing voice adaptation. A desired F0 trajectory can be directly copied from a time-aligned reference recording [111, 235], or predicted from a score input using neural networks [64, 239]. A key challenge is the limited availability of suitable training data, ideally consisting of matched recordings with varying intonation quality. In practice, heuristic strategies are often used for dataset construction [220] or for designing training objectives, for example by learning to synthesize natural pitch variations while minimizing deviations from a target pitch input [239]. Moreover, few data-driven methods explicitly consider harmonic context. Wager et al. [221] condition pitch correction on a backing track rather than a fixed score. Hyodo et al. [85] propose a polyphonic vocal synthesizer trained on the relative intervals between voices rather than absolute F0 values. However, these approaches implicitly encode adaptation behavior in model weights, offering limited controllability to accommodate individual artistic intent.

Sethares [188] established a relationship between tuning and timbre of simultaneous sounds using the concept of sensory dissonance [67, 190], which is largely influenced by the perception of *roughness* due to beating between tonal components (e.g., partials). By evaluating pairwise interactions between partials, a *dissonance landscape* can be obtained, in which, if the tones have only harmonic partials, local minima occur at intervals with small integer ratios. This principle has been explored for adaptive tuning using gradient descent [189], updating the F0 of tones to minimize sensory dissonance, achieving a tuning similar to JI without requiring explicit musical analysis of the chords. However, depending on the number and amplitudes of the partials, not all intervals that are commonly used in Western music have clearly defined local minima in the dissonance landscape (see, e.g., MIDI pitches 59–61 in Figure 5.2c), limiting robustness for more musically complex chords. To mitigate this, Villegas and Cohen [217] introduce a proximity constraint, allowing the F0 to only deviate by a few cents from the initial tuning. While this stabilizes the optimization locally, it does not allow for effectively addressing larger deviations such as global intonation drift. To our knowledge, no previous work formulates multi-voice intonation adaptation for recorded ensemble signals as a fully differentiable optimization problem that explicitly trades off adherence of each voice to an equal-tempered pitch grid and the harmonic fit between multiple

voices combined with controllable temporal behavior through interpretable optimization parameters. The approach proposed in the following addresses this gap.

5.2 Intonation Cost Measures

By formulating intonation assessment as a cost measure, we obtain a flexible way to combine and balance multiple intonation criteria within a single mathematical framework. In particular, we define an *intonation cost* C for a monophonic *target voice* in the context of a fixed, potentially polyphonic *accompaniment*. For concise notation, we first consider a simplified setting with a single target voice and a single time frame. This is then extended in Section 5.3, where we will take into account the temporal relations over consecutive time frames as well as the interaction between multiple voices that can be adapted simultaneously.

To model how the target voice adapts and how this affects the intonation cost, we introduce a parameter p representing a pitch shift of the target voice in cents. Since we later want to compute the gradient $\partial C / \partial p$, we treat p as a direct parameter of the cost function, yielding a function $C : \mathbb{R} \rightarrow \mathbb{R}_+$, where $C(p)$ quantifies the intonation cost when the target voice is pitch-shifted by p cents. In the context of our adaptation algorithm, a suitable cost function C should satisfy two requirements: it should be differentiable with respect to p to enable gradient-based optimization, and it should attain local minima at pitch shifts p that are favorable under the chosen intonation criteria. In this work, we consider two such criteria, each represented by a separate term in the overall intonation cost. The *tonal* cost C_t measures the distance to an equal temperament (ET) grid, while the *harmonic* cost C_h measures sensory dissonance in the overall sound. To account for different artistic intents, both terms can be balanced via a weighting factor $w \in [0, 1]$, favoring dissonance minimization with $w \approx 0$ and adherence to ET with $w \approx 1$. This results in a combined cost function

$$C(p) := wC_t(p) + (1 - w)C_h(p). \quad (5.1)$$

The individual terms C_t and C_h reflect two key criteria of intonation [147], but obviously do not exhaustively capture all aspects that influence the perception of “good” intonation. For example, the combined cost does not consider horizontal intonation, which could be included by adding a third cost term that analyzes the intervals between consecutive time frames within the target voice. Our cost-based framework and adaptation method (see Section 5.3) allows for such extensions, as long as they preserve differentiability and yield local minima at favorable pitch shifts.

5.2.1 Signal Representations

In addition to the pitch shift p , the cost terms depend on the *tonal components* (or *partials*) of the target and/or accompaniment signals. For a given time frame, we represent an audio signal by the set

$$\mathcal{P} := \{(f_0, a_0), \dots, (f_{M-1}, a_{M-1})\} \quad (5.2)$$

of M tonal components with frequency $f_m \in \mathbb{R}_+$ in Hz and amplitude $a_m \in \mathbb{R}_+$ for each component $m = 0, \dots, M - 1$.

In general, $\mathcal{P}$ can be obtained from an audio signal by detecting prominent peaks in the signal's magnitude spectrum, where the components are not necessarily related by integer multiples. For monophonic and (approximately) harmonic sounds, such as the individual monophonic tracks in our running example, f_0 corresponds to the F0 of the tone, and $f_m \approx m f_0$ represent the integer-multiple partials. In this case, a more robust alternative to general peak finding is to first estimate the F0 and then identify spectral peaks at (or close to) its integer multiples [182], which reduces sensitivity to background noise and other spurious spectral peaks.

For the following definitions of the cost terms, we denote the salient tonal components of the monophonic target signal by $\mathcal{P}^{\text{tgt}}$ and those of the accompaniment by $\mathcal{P}^{\text{acc}}$. In our case, $\mathcal{P}^{\text{acc}}$ is the union of the peaks identified from the individual accompaniment voices. We further denote the F0 from $\mathcal{P}^{\text{tgt}}$ by f_0^{tgt} to simplify notation. An example of $\mathcal{P}^{\text{tgt}}$ and $\mathcal{P}^{\text{acc}}$ is illustrated in Figure 5.2a, using S from the example cadence (cf. Figure 5.1) as the target voice and the mixture of all other voices (ATB) as the accompaniment.

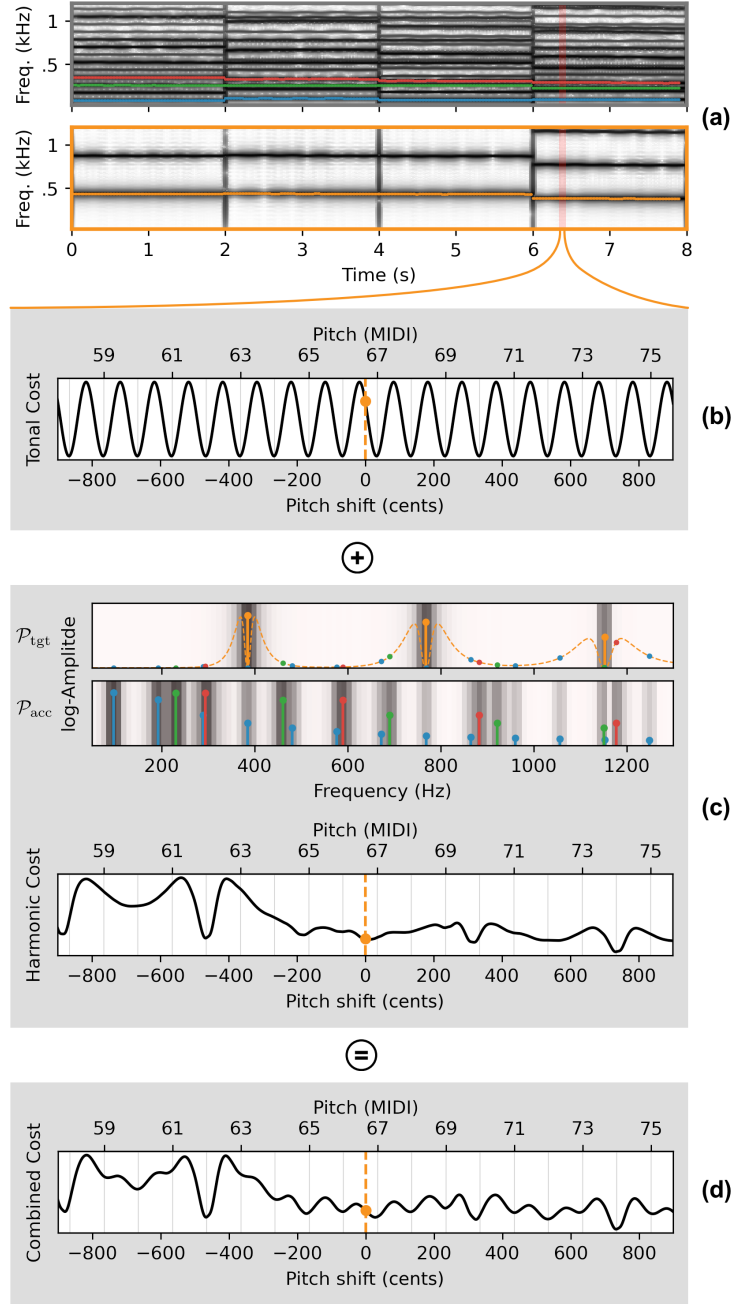
5.2.2 Tonal Cost

The tonal cost C_t aims to quantify how far the target voice, represented by f_0^{tgt} , deviates from an equal temperament (ET) pitch grid. In general, we denote a grid with K divisions of the octave as K -TET. In Western music, $K = 12$ (the 12-TET grid with twelve pitches per octave) is widely used to assess intonation quality, both in literature [116, 135, 226] and in practice (e.g., using chromatic tuners). While ET is sometimes seen as a compromise for instruments with fixed tuning [9], it provides a reasonable approximation of the tonal organization principle underlying many musical traditions [13].

To measure the deviation of f_0^{tgt} from an ET grid, we first introduce a function that takes two frequencies f and f' and measures how closely the interval between them matches an ET interval. Since the ET grid is evenly spaced on a logarithmic frequency axis, a suitable distance function should have periodic local minima and increase smoothly as the interval deviates from the grid. A cosine over the log-interval satisfies this requirement and is continuously differentiable, making it a suitable choice for modeling the

Figure 5.2: Intonation cost terms for a single frame in the synthesized running example from Figure 5.1.

- (a) Spectrogram of the accompaniment (ATB) and target voice (S) input signals with estimated F0 trajectories.
- (b) Tonal cost C_t using f_0^{tgt} .
- (c) $\mathcal{P}^{\text{tgt}}$, $\mathcal{P}^{\text{acc}}$ and harmonic cost C_h for the S voice in the highlighted frame.
- (d) Combined cost C with $w = 0.33$.



tonal cost. We thus define the distance of the interval between f and f' to a K -TET interval by

$$d_t^K(f, f') := \frac{1}{2} \left(1 - \cos(2\pi K \log_2(f/f')) \right). \quad (5.3)$$

This distance d_t^K is zero if the interval between f and f' is exactly a K -TET interval, i.e., $f/f' = 2^{k/K}$ with $k \in \mathbb{Z}$. With this, we define C_t as

$$C_t(p) := d_t^K(f_0^{\text{tgt}} \cdot 2^{p/1200}, f_{\text{ref}}), \quad (5.4)$$

where p is the pitch shift parameter defined above. The hyperparameters K and f_{ref} define the ET grid and can either be estimated from the accompaniment (see, e.g., [226]) or set to predefined values. In our experiments, we set $K = 12$ and $f_{\text{ref}} = 440$ Hz (corresponding to standard A₄ tuning).

The resulting tonal cost C_t is illustrated in Figure 5.2b. In the depicted frame, a pitch shift of $p = 0$ for the S voice leads to a non-zero cost, meaning that the original intonation does not lie exactly on the 12-TET grid. The nearest local minimum is exactly at the center frequency of MIDI pitch 67 which matches the target pitch (G₄) defined by the score shown in Figure 5.1a. Since C_t has a minimum at each integer MIDI pitch, the nearest local minimum would be different from the score target if the voice deviates by more than 50 cents. Note that C_t depends only on f_0^{tgt} and therefore neither considers differences in timbre (in particular, the energy distribution across partials) nor any interaction between target and accompaniment.

5.2.3 Harmonic Cost

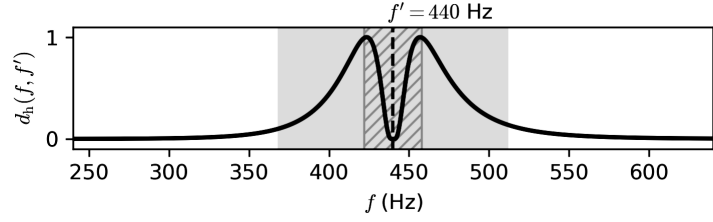
The harmonic cost C_h aims to quantify the dissonance between simultaneously sounding tones. Following [190], we consider sensory dissonance as a differentiable surrogate for this perceptual property. In contrast to C_t , which evaluates the target voice in isolation against a fixed pitch grid, C_h explicitly models the interaction between the target voice and the accompaniment. It is important to note that sensory dissonance is not inherently a negative property, since its variation across different chords and timbres is an important component of musical expression [208]. However, when tones have only harmonic partials, locally minimizing sensory dissonance can lead to intonation similar to JI [189], which may be desirable in some ensemble performance. For chords consisting of more complex intervals, as well as inharmonic tones (such as many metallophones), the minima can differ considerably from typical intervals in Western harmony [190], which is a key limitation of this approach compared to the tonal stability offered by C_t .

The sensory dissonance between two complex simultaneous sounds can be expressed as the sum of the *pure-tone* dissonance between all combinations of tonal components in each sound [188]. This pure-tone dissonance was initially determined from perceptual experiments with simultaneously played sinusoids [150] and has since been parameterized in a multitude of different models (see [67] for an overview). We adopt the model from [13] for two reasons. First, it yields a smooth, continuously differentiable function (cf. Appendix B), which is a prerequisite for our cost terms as outlined above. Second, it is based on a logarithmic ratio $\log_2(f/f')$ between frequencies f and f' instead of a linear distance as many other models, simplifying the computation of the gradient $\partial C_h / \partial p$. The pure-tone dissonance between two sinusoids is given by

$$d_h(f, f') := \begin{cases} \exp\left(-\ln^2\left(\frac{|\log_2(f/f')|}{d_{\max}(\bar{f})}\right)\right) & f \neq f' \\ 0 & f = f', \end{cases} \quad (5.5)$$

where $d_{\max}(\bar{f})$ is a frequency-dependent parameter controlling the interval of maximal dissonance and the decay of the dissonance curve, which in [13] is defined based on a heuristic that increases $d_{\max}$ towards

Figure 5.3: Pure-tone dissonance $d_h(f, f')$ for $f = 440$ Hz. The shaded area indicates the ERB around f (0.25 ERB marked with hatched rectangle).



higher frequencies. To make this parameter more interpretable, we instead express $d_{\max}(\bar{f})$ explicitly in terms of the *equivalent rectangular bandwidth* (ERB) [58], a standard measure for auditory frequency resolution [241], placing the dissonance maximum precisely at 0.25 critical bands as established by perceptual studies [113, 150]:

$$d_{\max}(\bar{f}) := \log_2 \left(1 + 24.7(0.00437\bar{f} + 1)/\bar{f} \right). \quad (5.6)$$

Here, the ERB is evaluated in octaves at the mean frequency $\bar{f} := (f + f')/2$, ensuring symmetry (i.e., $d_h(f, f') = d_h(f', f)$). As illustrated in Figure 5.3, d_h approaches zero when the two frequencies are very similar or very far apart, and reaches its maximum of one at $|\log_2(f/f')| = d_{\max}(\bar{f})$. In the derivative of d_h , we assume $d_{\max}(\bar{f})$ to be constant, since it varies only minimally for small changes in $\bar{f}$.

The harmonic cost is then defined as the amplitude-weighted sum of pure-tone dissonances between all pairs of tonal components in $\mathcal{P}^{\text{tgt}}$ and $\mathcal{P}^{\text{acc}}$:

$$C_h(p) := \frac{1}{A} \sum_{\substack{(f, a) \in \mathcal{P}^{\text{tgt}} \\ (f', a') \in \mathcal{P}^{\text{acc}}}} \min(a, a') d_h(f \cdot 2^{p/1200}, f'), \quad (5.7)$$

where $\min(a, a')$ weights each pairing proportionally to the amplitude of the beating that can occur between the two components [190]. The normalization factor A ensures that C_h and C_t are in a comparable range. Normalizing by the sum of all weighting factors would guarantee $C_h \in [0, 1]$, but since most pairs of tonal components are further apart than one critical band, their contribution to the sum is negligible ($d_h \approx 0$), resulting in a very small expected value for C_h . We therefore normalize only by the sum of weighting factors for pairs where $|\log_2(f/f')| \leq 1$, i.e., within one octave, which yields magnitudes comparable to those of C_t .

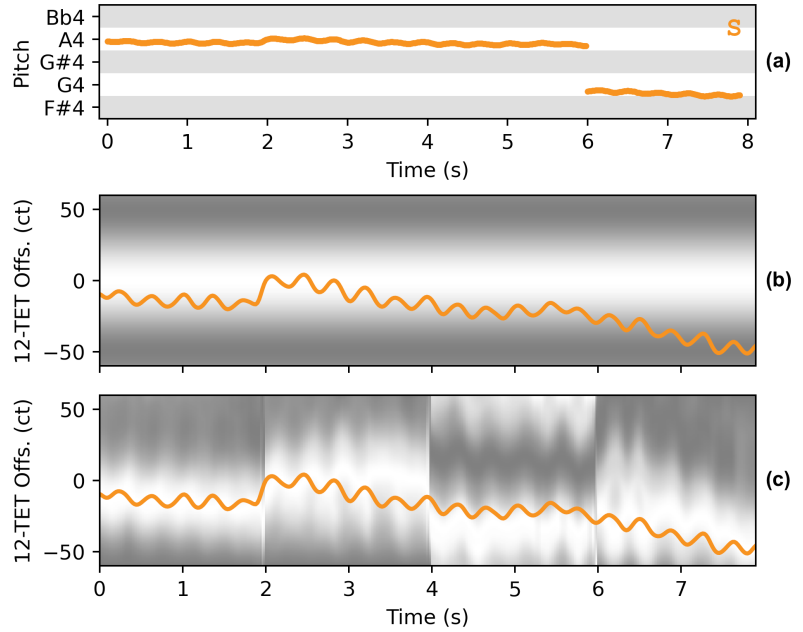
In Figure 5.2c, the harmonic cost C_h is illustrated for a single frame from our example cadence. The upper plot shows $\mathcal{P}^{\text{tgt}}$ and $\mathcal{P}^{\text{acc}}$ in this frame, along with the pure-tone dissonance between each pairing. As an example, consider the third harmonic of the S voice at around 1200 Hz: d_h is non-negligible for its pairing with the fourth harmonic of the A voice, as well as the eleventh and thirteenth harmonics of the B voice. Summing all these interactions yields the harmonic cost shown in the lower plot for pitch shifts p between -1100 and 800 cents. In contrast to C_t in Figure 5.2b, C_h exhibits a local minimum very close to the actual F0 of the target voice, reflecting the advantage of modeling voice interactions explicitly. On the other hand, C_h does not necessarily have a clear local minimum at every musically relevant target pitch.

Figure 5.4: Illustration of the intonation cost over time for the S voice of the example cadence from Figure 5.1.

(a) F0 of the S voice, reproduced from Figure 5.1b for reference.

(b) Visualization of cost term C_t . The y-axis represents the 12-TET offset, i.e., how far the F0 of the S voice deviates from the nominal 12-TET pitch of the played note. The actual deviation of the S voice is shown with an orange line. A darker background color denotes a higher cost.

(c) Like (b), but visualizing cost term C_h .



For example, near MIDI pitches 59 and 61 in Figure 5.2c, the nearest local minimum is close to MIDI pitch 60, illustrating the complementary role of C_t in ensuring tonal stability.

Combining both cost terms as in Figure 5.2d allows for a flexible weighting of the two modeled intonation aspects. With $w = 0.33$ in this example, there is a clear minimum near most MIDI pitches, and this minimum is shifted towards reduced sensory dissonance where C_h has a clear local minimum on its own.

5.3 Intonation Adaptation Algorithm

The differentiability of C with respect to p enables a gradient-based adaptation algorithm that computes a pitch shift curve $p(n)$, specifying for each time frame n by how many cents the target voice should be pitch-shifted to reduce the overall intonation cost. By controlling the amount of adaptation applied in each frame, the algorithm can be tuned to correct global intonation deviations while preserving local ones, or vice versa, resulting in a more natural-sounding adaptation than a purely frame-wise correction would yield. This behavior can be controlled by several hyperparameters described below, which can be adjusted to match the properties of the input signal (e.g., the strength of intended and unintended pitch variations) as well as the desired output (e.g., a natural-sounding or strongly modified pitch trajectory). The resulting time-varying pitch shift $p(n)$ can then be applied to the target signal using a suitable pitch shifting algorithm (see, e.g., [164]). Extending the cost function to time-varying signals, we further account for the frame-wise changing properties of the input signals, denoted by $f_0^{\text{tgt}}(n)$, $\mathcal{P}^{\text{tgt}}(n)$, and $\mathcal{P}^{\text{acc}}(n)$, and write the corresponding intonation cost as $C(p, n)$.

To analyze the behavior of the cost terms and the adaptation algorithm over time, we express the pitch shift p relative to the nominal 12-TET pitch of the currently played note, referring to this quantity as the *12-TET offset*. Figure 5.4 shows this offset for the S voice of our example cadence, overlaid on the *cost landscape*, which depicts the intonation cost as a function of the 12-TET offset over time, with darker colors indicating higher cost. As expected, the landscape of the tonal cost C_t (Figure 5.4b) is constant over time and identical across tones, since it depends only on the deviation from the ET grid and not on the musical context. The landscape of the harmonic cost C_h (Figure 5.4c), on the other hand, is strongly influenced by the musical context, as reflected in three observable phenomena. First, the chord function of the target voice affects the position of the local minimum. Since S is playing the major third in the first chord, its local minimum lies lower than in the second chord, consistent with JI tuning where the major third is 14 cents below 12-TET, whereas the same pitch A_4 serves as the root in the second chord and therefore has a local minimum around the 12-TET offset of 0 cents. Second, a gradual downward drift of the local minima across later notes reflects a global intonation drift present throughout the entire example. Third, the vibrato of the accompanying voices manifests as small periodic fluctuations in the position of the local minimum.

In [189], gradient descent is used to update the frequency of an individual salient tonal component directly. Inspired by this approach, we formulate an iterative update rule for the pitch shift p with L total iterations ($\ell = 1, \dots, L$)

$$p^{(\ell)}(n) = p^{(\ell-1)}(n) - \mu g^{(\ell)}(n), \quad (5.8)$$

where

$$g^{(\ell)}(n) = \left. \frac{\partial C(p, n)}{\partial p} \right|_{p=p^{(\ell-1)}(n)} \quad (5.9)$$

is the gradient of the cost function $C(p, n)$ evaluated at $p^{(\ell-1)}(n)$ and μ is a step size parameter. The unit of $g^{(\ell)}(n)$ is cost/cent, or *cost change per cent shifted*. Therefore, to obtain an update amount in cents for Eq. 5.8, the step size μ has the unit $\text{cent}^2/\text{cost}$ and can be interpreted as the pitch shift applied per frame when the cost gradient equals one. For example, with $\mu = 50$ and a frame length of 20 ms (a frame rate of 50 Hz, as in all examples presented here), a sustained unit gradient would result in a pitch shift rate of $50 \times 50 = 2500$ cents per second. However, the actual maximum gradient of C_t is $\pi K/1200 = \pi/100 \approx 0.031$, so that even in the most extreme case, $\mu = 50$ yields a maximum shift rate of approximately 78.5 cents per second.³²

In the following Sections 5.3.1–5.3.3, we introduce the hyperparameters of the algorithm and discuss their effects, illustrated also in Figure 5.5 using the S voice of our example cadence. For each varied hyperparameter, Figure 5.5 shows the resulting pitch shift curve $p(n)$ as well as the 12-TET offset of the target voice overlaid on the cost landscape, before (orange) and after (shades of gray) applying $p(n)$. To keep these illustrations simple, we set $w = 1$ in Eq. 5.1, using the cost C_t only. The harmonic cost C_h and the extension to a multi-voice setting are then introduced in Section 5.3.4.

³² For C_h , the maximum gradient is less straightforward to derive analytically, but is expected to be smaller than that of C_t .

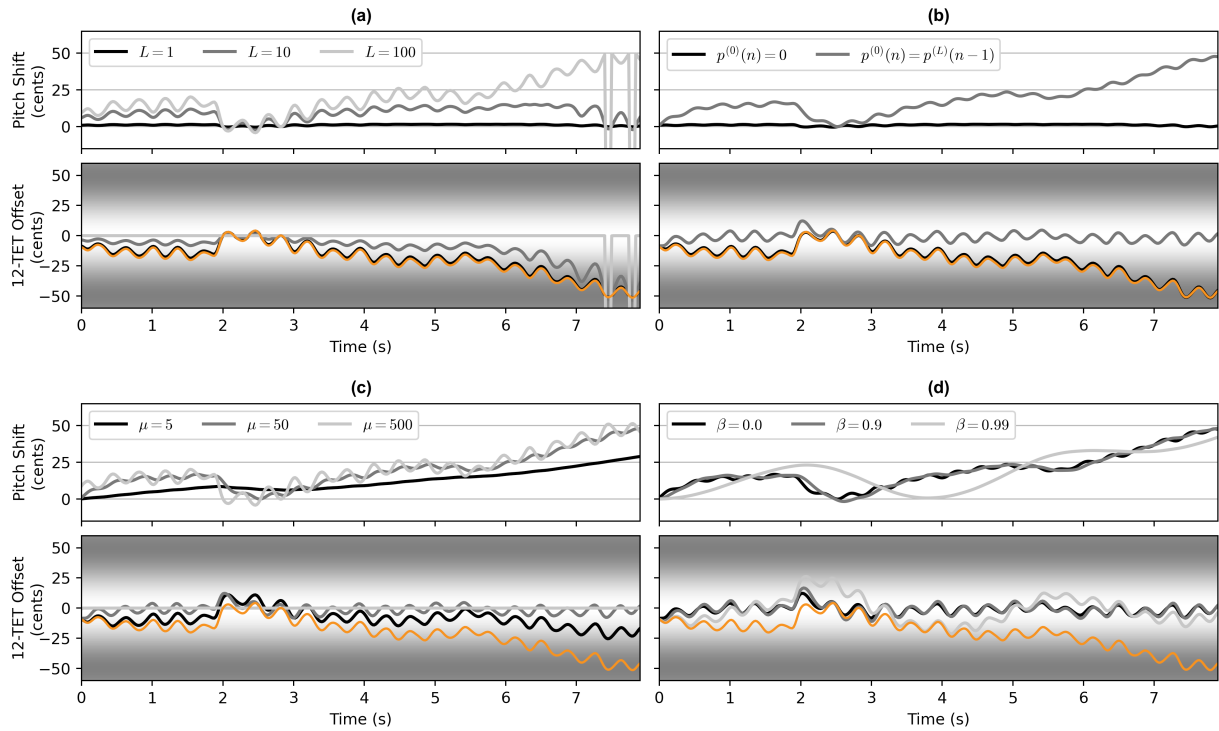


Figure 5.5: Examples for intonation adaptation with different hyperparameter settings using only the tonal cost ($w = 1$) and the S voice from the example cadence. Each panel (a-d) shows the resulting pitch shift curves $p(n)$ (top) and the original and adapted 12-TET offsets overlaid on the cost landscape (bottom). **(a)** Varying the number of iterations $L = 1$ (dark gray), $L = 10$ (medium gray), and $L = 100$ (light gray), with $p^{(0)}(n) = 0$ and $\mu = 50$. **(b)** Varying the initial conditions $p^{(0)}(n) = 0$ (dark gray), and $p^{(0)}(n) = p^{(L)}(n-1)$ (medium gray), with $L = 1$ and $\mu = 50$. **(c)** Varying the step size $\mu = 5$ (dark gray), $\mu = 50$ (medium gray), and $\mu = 500$ (light gray), with $L = 1$ and $p^{(0)}(n) = p^{(L)}(n-1)$. **(d)** Varying the momentum parameter $\beta = 0.0$ (dark gray), $\beta = 0.9$ (medium gray), $\beta = 0.99$ (light gray), with $L = 1$, $p^{(0)}(n) = p^{(L)}(n-1)$, and $\mu = 50$.

5.3.1 Number of Iterations

As a first hyperparameter for controlling the amount of adaptation, we vary the number of gradient descent iterations L , using an initial value of $p^{(0)}(n) = 0$ and a fixed step size of $\mu = 50$. The three shades of gray in Figure 5.5a show the resulting $p(n)$ for $L = 1$, $L = 10$, and $L = 100$, from dark to light. With $L = 1$, the adaptation has virtually no effect, as a single gradient step is insufficient to meaningfully reduce the cost with the given step size. At the other extreme, $L = 100$ successfully counteracts the global intonation drift, but also removes local pitch variations such as vibrato, and may push the adapted pitch towards a neighboring note whenever the original 12-TET offset exceeds 50 cents (cf. the two jumps in $p(n)$ after 7 seconds), independently of the temporal context. An intermediate value of $L = 10$ offers a compromise, but reveals that varying L alone does not allow for independent control over local and global adaptation behavior.

5.3.2 Initial Value

A key modification for incorporating temporal context is to initialize each frame with the previous result, setting

$$p^{(0)}(n) = p^{(L)}(n-1). \quad (5.10)$$

Figure 5.5b compares both initial conditions with $L = 1$ and all other hyperparameters unchanged. Initializing from the previous frame (lighter gray) yields a pitch shift curve that effectively compensates for the global drift while preserving most of the local vibrato. It also allows for 12-TET offsets above 50 cents without snapping to a neighboring note, since the temporal context anchors the adaptation to the preceding frame. This reflects the temporal consistency of intonation decisions made by performers, who naturally only slowly update their frame of reference instead of reassessing independently at each moment in time.

5.3.3 Step Size and Momentum

The step size μ directly controls the strength of the adaptation per frame. As shown in Figure 5.5c, a small value of $\mu = 5$ (darkest gray) yields a gentle adaptation that preserves most local fluctuations, whereas a large value of $\mu = 500$ (lightest gray) counteracts even fine-grained variations such as vibrato, similar to the effect of a large L .

While μ offers intuitive control over the overall adaptation strength, it does not allow for precise and independent control over the reaction to global versus local pitch variations, motivating the introduction of an additional mechanism to address this trade-off. To do this, we introduce a *momentum* parameter $\beta \in [0, 1]$, as commonly used in gradient-based optimization [60]. To do this, we modify the update rule from Eq. 5.8 to

$$p^{(\ell)}(n) = p^{(\ell-1)}(n) - \mu b^{(\ell)}(n), \quad (5.11)$$

where

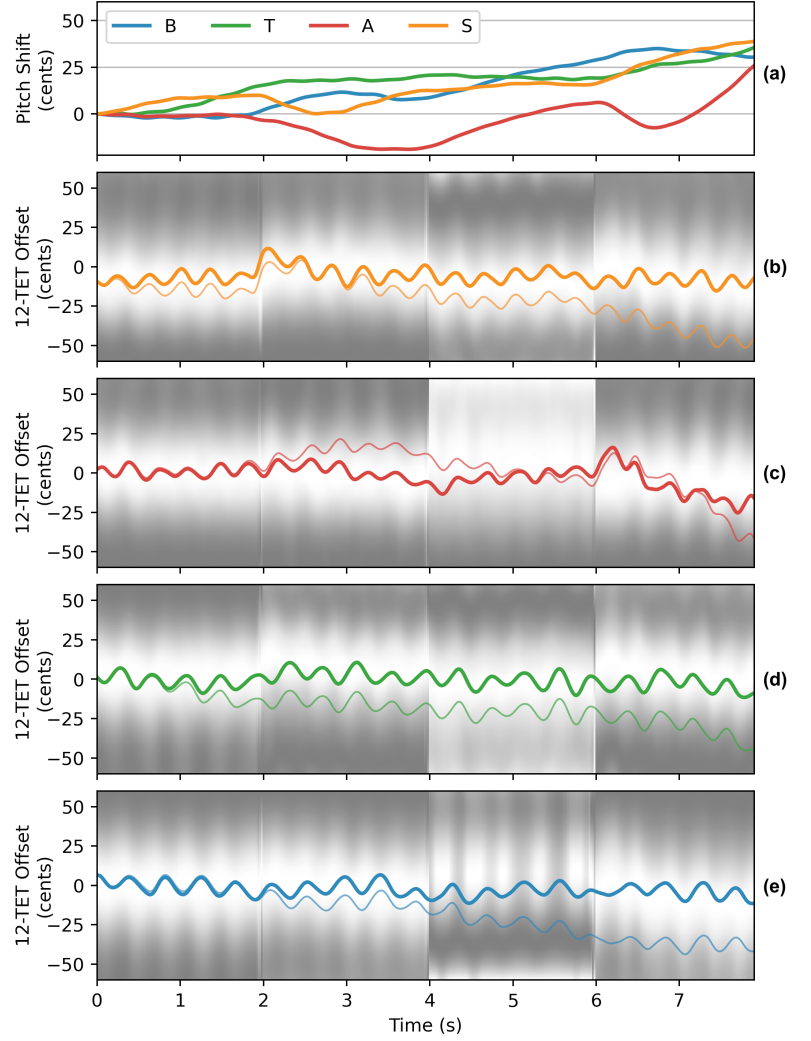
$$b^{(\ell)}(n) = \beta b^{(\ell-1)}(n) + (1 - \beta)g^{(\ell)}(n) \quad (5.12)$$

is an exponentially weighted moving average of the gradients across iterations, with $b^{(0)}(n) = b^{(L)}(n-1)$, analogous to Section 5.3.2. Setting $\beta = 0$ reduces to the original update rule Eq. 5.8, while $\beta = 1$ freezes the update at $b^{(0)}(n)$, preventing any adaptation.

The results for $\mu = 50$ and varying β are shown in Figure 5.5d. The momentum acts as a low-pass filter on the gradient sequence, effectively suppressing the influence of rapid local fluctuations while allowing the algorithm to react to slower global trends. However, since momentum introduces a group delay proportional to $1/\log(\beta)$, analogous to a low-pass filter, choosing β too large can slow the reaction to pitch changes and an oscillating “filter resonance”, as visible for $\beta = 0.99$ (light gray pitch shift curve in Figure 5.5d).

Figure 5.6: Example for multi-voice adaptation in the example cadence from Figure 5.1, using hyperparameters $w = 0.33$, $L = 1$, $\mu = 50$, and $\beta = 0.9$.

- (a) Pitch shift curves $p_v(n)$ for each voice (same as Figure 5.1c).
- (b) 12-TET offset of the S voice before (thin line) and after adaptation (thick line) overlaid on the cost landscape.
- (c) Like (b), but for the A voice.
- (d) Like (b), but for the T voice.
- (e) Like (b), but for the B voice.



5.3.4 Multi-Voice Adaptation

Up to this point, we have considered a single target voice adapted against a fixed accompaniment, with iterations over time frames n and optimization steps ℓ . With a set of monophonic voices, the gradient-based framework naturally extends to the joint adaptation of multiple voices, where each voice alternates between the role of target voice and accompaniment. This is particularly relevant for modeling the dynamic tuning interactions in ensembles such as string quartets, wind ensembles, or choirs. Since this kind of multi-voice adaptation is only meaningful when the cost reflects the interaction between voices, we set $w = 0.33$ in the following example, combining both cost terms to encourage JI-like intonation while maintaining tonal stability.

For a formal definition, let $\mathcal{V}$ denote the set of monophonic voices in the performance, e.g., $\mathcal{V} = \{S, A, T, B\}$ for a quartet. Each voice $v \in \mathcal{V}$ can take the role of the target voice, while the accompaniment is formed

by the combination of the remaining voices. We denote the individual pitch shift curves by $p_v(n)$ and introduce a third iteration over $\mathcal{V}$ for each time frame n . Crucially, the cost function for each voice is evaluated with the pitch shifts $p_v(n-1)$ of all voices from the previous frame applied to $\mathcal{P}^{\text{tgt}}$ and $\mathcal{P}^{\text{acc}}$. This mirrors real ensemble performance, where intonation decisions in the current moment are informed by what happened in the preceding one. Since all adaptations for a given frame n are solely based on $p_v(n-1)$, the order in which individual voices are processed is irrelevant.

The effect of multi-voice adaptation is illustrated in Figure 5.6. Figure 5.6a shows the resulting pitch shift curves $p_v(n)$, and Figure 5.6b to e show the 12-TET offsets of each voice overlaid on the cost landscape combining C_t and C_h . The algorithm successfully counteracts the global drift in all voices while preserving vibrato and retaining chord-specific intonation tendencies, such as the lowered major third in S during the first chord. An interesting edge case is visible in the B voice (Figure 5.6e) during the third chord, where the cost landscape shows a tonally unstable region, yet the algorithm maintains tonal stability as opposing gradients across voices tend to cancel out over time.

5.4 Experiments

The proposed method offers three main advantages over existing approaches. First, the combination of a tonal and harmonic cost term within a unified cost-based framework yields better stability in realistic musical contexts compared to methods based solely on sensory dissonance minimization [189], while allowing for flexible control between targeting 12-TET or JI-like intonation. Second, the gradient-based multi-voice adaptation with temporal consistency can handle large intonation drifts that grid-based methods cannot compensate for without score information or manual intervention. Third, the hyperparameters L , μ , and β provide effective control over the adaptation of global and/or local intonation deviations.

To demonstrate these advantages, we conduct three experiments with real-world music signals, using quartet recordings of three different ensemble types. We evaluate the properties of the multi-voice adaptation quantitatively, analyzing the resulting pitch shift curves and cost landscapes in terms of the suitability for 12-TET and JI-like adaptation (Section 5.4.2), the compensation of intonation drift without manual intervention (Section 5.4.3), and the trade-offs between global and local intonation adaptation (Section 5.4.4).

Additionally, we report on a small-scale listening test (Section 5.4.5) that evaluates listener preferences for the weighting of the cost terms, in comparison to a commercial baseline. It is important to note that we do not aim to establish the perceptual superiority of our method over existing ones, as listener preferences are influenced by many factors beyond the adaptation algorithm itself, including the quality of the pitch shifting method, the musical context, and the discrimination ability of individual listeners. Rather, the subjective evaluation should be understood as a verification that the proposed method can produce results

comparable to industry-standard approaches, while offering the additional flexibility of the cost-based formulation and gradient-based adaptation.

5.4.1 Datasets

Our method requires monophonic input tracks, from which the signal representations f_0^{tgt} and $\mathcal{P}$ can be calculated. While this is a common way to record popular music (e.g., through *studio overdubbing*), advances in musical source separation (see, e.g., [158]) or specialized recording techniques [181] could also extend the applicability to music that was recorded in a shared acoustic space. For the experiments below, we create a test set of recordings from three datasets which already provide individual tracks with little or no crosstalk between voices. All signals are resampled to a sampling rate of 44100 Hz. F0 annotations are interpolated to a common frame rate of 882 samples (20 ms) before calculating $\mathcal{P}$ for each track and frame as described in Section 5.2.1.

ChoraleBricks (CB) [8] provides recordings of Baroque four-part chorales, where each voice is played on various wind instruments and recorded in isolation using a conductor video for temporal synchronization. Aligned chord and F0 annotations are available in the dataset. We use the chorale *Befiehl Du Deine Wege*, played on brass instruments trumpet (S), fluegelhorn (A), and baritone horn (T and B).

Dagstuhl ChoirSet (DC) [163] is a multi-track dataset of a cappella choral music, performed by amateur singers. In addition to spot microphone recordings of each individual voice with little crosstalk, the dataset provides aligned score and F0 annotations. We use a recording of the four-voice motet *Locus Iste* by Anton Bruckner performed with one singer per voice (take `QuartetB T4` from the dataset).

Virtuoso Strings (VS) [213] contains multi-track recordings of string quartet performances, using spot microphones with little crosstalk between the simultaneously playing instruments. We use a single recording of an excerpt from the String Quartet Op. 74 No. 1 by Joseph Haydn (take `Q-JH-VN1-NR 10` from the dataset) and create F0 annotations for each track using the SWIPE implementation from [165].

5.4.2 12-TET and JI Adaptation

The mathematical framework of cost-based adaptation enables the combined application of grid-based approaches (like our tonal cost C_t) and multi-voice interactions (like our harmonic cost C_h). Using the CB performance as an example scenario, we demonstrate how the combination of the two terms stabilizes the optimization while still enabling JI-like intonation without requiring score information. We run the proposed algorithm with hyperparameter settings $\mu = 50$, $\beta = 0.0$, $L = 1$, and $p^{(0)}(n) = p^{(L)}(n - 1)$, varying the cost weighting w . To evaluate the adaptation, we calculate nominal target pitches for 12-TET and JI according to the time-aligned score and respective chord degree, which allows us to measure the deviation of adapted tones from this hypothetical target pitch. Results are shown in Table 5.1. For each note (with a segmentation based on the aligned annotations), we compute the median F0 of all original and

Table 5.1: Adaptation results for the CB performance with different cost weightings. All values given in cents. For explanation, see Section 5.4.2.

Configuration	Median Dev.		Range
	12-TET	JI	
Original	9.7 ± 7.0	10.6 ± 7.8	24.1 ± 14.9
$w = 1$ (C_t only)	4.5 ± 6.0	6.9 ± 6.6	23.8 ± 15.9
$w = 0.33$	5.7 ± 5.0	5.6 ± 5.1	23.6 ± 14.9
$w = 0$ (C_h only)	12.5 ± 12.9	11.8 ± 12.8	25.6 ± 16.3

adapted voices and report its mean absolute deviation from the nominal 12-TET (*Median Dev. 12-TET*) and JI (*Median Dev. JI*) frequencies, as well as the range (the difference between the maximum and minimum) of the F0 within each note. All values are given in cents.

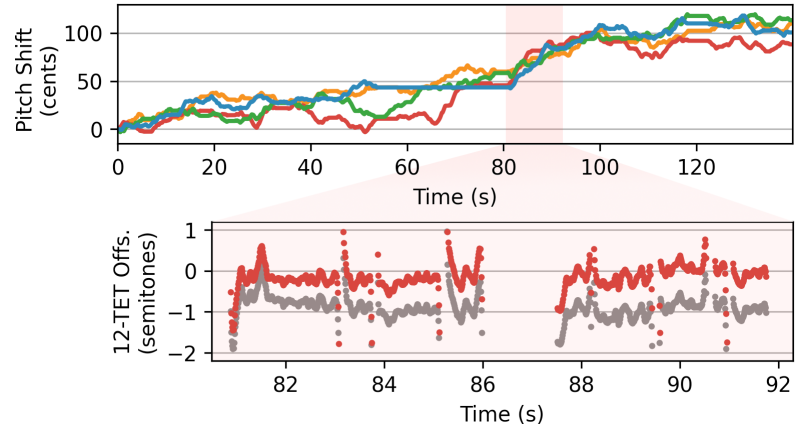
The closest adaptation to 12-TET is achieved with $w = 1$, reducing the mean absolute deviation of the note-wise median F0 from 9.7 to 4.5 cents. Using both cost terms ($w = 0.33$) moves the median F0 closer to JI, reducing the deviation from 10.6 to 5.6 cents. Using only C_h ($w = 0.00$), which corresponds to the approach from [189], results in median F0 deviations exceeding those of the original. This confirms that the tonal cost term is essential for a stable adaptation. The remaining deviations in the adapted voices can partly be attributed to the delayed response in $p_v(n)$ to large 12-TET offsets in very short notes. Across all settings, the F0 range within notes is largely maintained, indicating that local F0 variations are preserved with the chosen hyperparameters. The trade-off between absolute accuracy and local variation is discussed further in Section 5.4.4.

5.4.3 Intonation Drift Compensation

Especially in vocal ensemble recordings, compensating for intonation drift can be an important post-production step, for example when combining multiple takes into a coherent performance. While the drift in our synthetic cadence occurred over a short time span, drift in real performances is often more subtle and non-linear, with physiological [4] or musical [80] origins. The DC performance used in this application example exhibits a global drift of over 100 cents, with the ensemble collectively singing around a semitone lower by the end of the recording. Compensating for this drift while preserving local intonation is not possible with grid-based methods without score information, since they would typically adapt towards the nearest semitone. With suitable hyperparameter settings, our method can account for the drift without manual intervention.

To compensate for drift while preserving local fluctuations, we choose a small step size $\mu = 5$ and use momentum ($\beta = 0.9$) to further smooth the pitch shift curves $p_v(n)$. The resulting $p_v(n)$ for the entire performance are visualized in Figure 5.7. It can be seen that while $p_v(n)$ behaves similarly across individual voices, the drift is not linear over the course of the performance. As indicated by the excerpt from the A voice in the lower plot of Figure 5.7, most local intonation phenomena (such as vibrato and the relative tuning of individual notes) are retained in the adapted version (shown in red).

Figure 5.7: Adapting global intonation drift in DC with $w = 1$, $\mu = 5$, $\beta = 0.9$, $L = 1$. The upper plot shows the pitch shift curves $p_v(n)$ over the entire piece (S: orange, A: red, T: green, B: blue). The lower plot shows the 12-TET offset of the A voice for an excerpt (gray: original, red: adapted).



5.4.4 Global and Local Adaptation

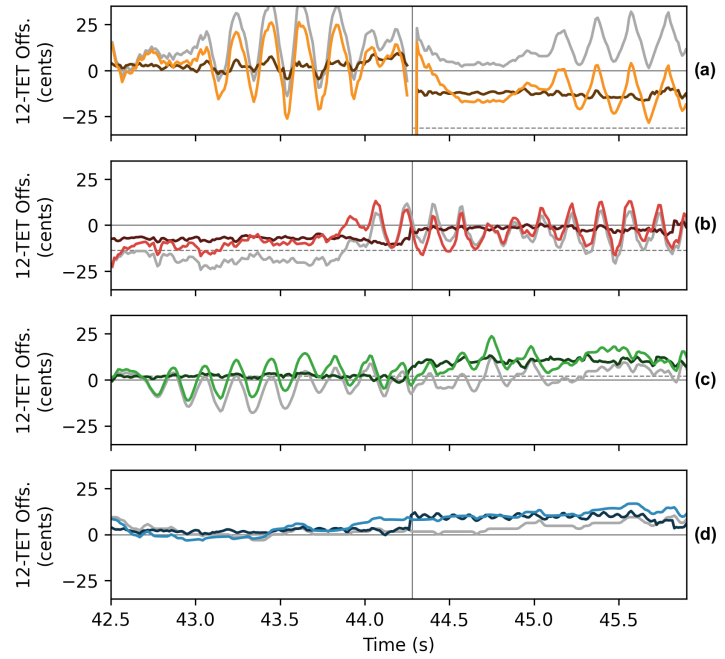
The hyperparameters L , μ , and β provide an interpretable way to balance between global and local intonation adaptation. To illustrate their effect in a real-world example, we fix $w = 0.25$ to target JI-like intonation and compare two contrasting configurations in an excerpt of the VS performance. The *global setting* ($\mu = 50$, $\beta = 0.9$, $L = 1$) shifts the median F0 of each note toward cost-minimizing intervals while preserving vibrato and other local fluctuations. The *local setting* ($\mu = 50$, $\beta = 0.0$, $L = 100$) applies multiple gradient steps per frame to approach a local minimum, effectively suppressing local pitch fluctuations such as vibrato.

Results are shown in Figure 5.8. For each voice, we visualize the 12-TET offset of the original (in gray), after adaptation with the global setting (in lighter color corresponding to the respective voice) and with the local setting (darker color). Both settings yield similar per-note median F0 values, while the local setting does not preserve any of the vibrato. In the first chord, between 42.5 and 44.3 seconds, the median F0 lies close to the respective JI target pitch (shown as horizontal dashed lines and clearly visible for the A voice playing a major third) due to the strong weighting of the harmonic cost. In the second chord of this example, the S voice plays a minor seventh in a D : 7 chord. In this case, the deviation from 12-TET is distributed among the voices, with the adapted A, T and B voices considerably above their nominal JI target pitches.

This example also exposes two inherent limitations of the proposed approach. First, the harmonic cost C_h does not allow the user to select which JI ratio is targeted. Depending on the musical context, there are several options for tuning the minor seventh interval in the S voice relative to the root, including the harmonic seventh with an interval ratio of $7/4$ (or -31 cents relative to 12-TET), as well as ratios of $16/9$ (-4 cents) or $9/5$ ($+8$ cents). The harmonic cost attains a minimum near -31 cents, consistent with both the downward shift of the S voice and the upward shifts of the remaining voices. This results in a strong deviation from 12-TET intervals that may not be musically desirable and is an inherent consequence of using sensory dissonance minimization as a proxy for JI-like intonation, as the location of the local

Figure 5.8: Global vs. local intonation adaptation in an excerpt from VS using $w = 0.25$ (larger emphasis on C_h). Each plot shows the original F0 offset (gray dots), the F0 offset of the voices adapted with $\mu = 50$, $\beta = 0.9$, $L = 1$ (lighter color) and $\mu = 50$, $\beta = 0.0$, $L = 100$ (darker color). The vertical line indicates a change of the chord from D:maj to D:7. The horizontal dashed line indicates JI intonation (with a $7/4$ minor sevenths in the S voice).

- (a) S voice,
- (b) A voice,
- (c) T voice,
- (d) B voice.



minimum depends on the voice timbres and may not coincide with the artistic intent. Second, preserving local intonation variations comes at the cost of slower adaptation at chord transitions, a trade-off that is clearly visible at the chord change at 44.3 seconds, where the local setting responds more rapidly.

5.4.5 Listening Test

To complement the methodological discussion above, we conducted a small-scale listening test with expert listeners to examine whether combining the two cost terms is advantageous compared to optimizing only tonal or only harmonic aspects, and to anchor the perceptual quality in comparison to a strong commercial baseline. Specifically, we performed a paired comparison test with five different conditions.

OR: The original, unaltered excerpt.

ME: Each voice was separately processed with the commercial pitch correction software *Melodyne Essential* v5.4.2 [22], using automatic F0 estimation and segmentation with the “Melodic” algorithm and subsequently running the “Correct Pitch” algorithm with both “Pitch Center” and “Pitch Drift” set to 100%. By using only automatic adaptations, we ensure comparability with our method.

WH: Our method, using only the harmonic cost ($w = 0$). This can be seen as a version of the method proposed in [189].

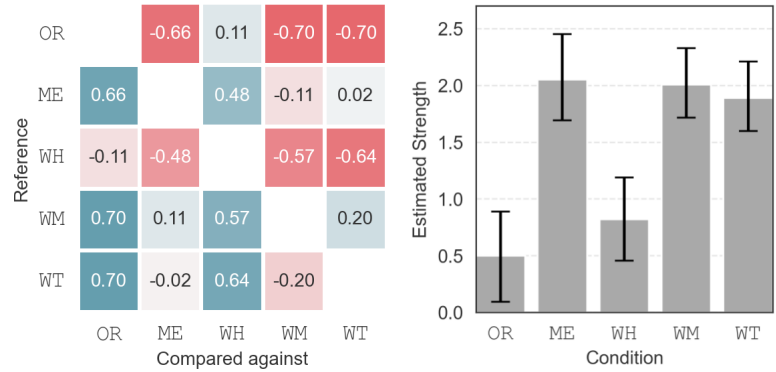
WM: Our method, mixing both cost terms ($w = 0.33$).

WT: Our method, using only the tonal cost ($w = 1$).

Figure 5.9: Results of listening test with 11 participants. OR: Original, ME: Melodyne, WH ours with $w = 0$, WM: ours with $w = 0.33$, WT: ours with $w = 1$.

(left) Average rating for each paired comparison.

(right) Relative strengths for each condition estimated using a CLMM model [207], showing mean and 95% confidence interval obtained by 1000× bootstrapping.



For easier comparisons, we kept all other hyperparameters fixed for our approach with $\mu = 50$, $\beta = 0.9$, and $L = 1$. No condition requires manual intervention by a sound engineer. As test items, we used excerpts of four to eight seconds length from the application scenarios DC, VS, and CB, as well as the synthetic example cadence. For applying the pitch shift curves $p_v(n)$ to the audio signals, we used the time-varying pitch shifting from [164] for WH, WM, and WT, and the proprietary pitch shifting of Melodyne for ME.

In an online experiment using *WebMUSHRA* [174], participants were presented with paired comparisons in random order, with four items and five conditions leading to $4 \times ((4 \times 5)/2) = 40$ comparisons in total. In each comparison, the two stimuli should be rated in terms of the perceived intonation quality and naturalness on a five-point Likert scale with values ‘2’ (stimulus A is much better), ‘1’ (A is a bit better), ‘0’ (no preference), ‘-1’ (B is a bit better), and ‘-2’ (B is much better). The signals were presented in mono over headphones. Of 11 participants in total (median age 33 years), 9 reported “a lot” of or “expert”-level experience in judging musical performances in a post-test questionnaire and all of them reported “a lot” of or “expert”-level experience as musicians. All listeners participated voluntarily and consented to the processing of their responses in an anonymized way.

The left side of Figure 5.9 shows the mean rating for each paired comparison, averaged over participants and test items. It can be observed that WM, WT, and ME achieve the largest subjective improvement over the OR condition, with an average rating of 0.70, 0.70, and 0.66, respectively. WH, on the other hand, does not seem to lead to favorable intonation and is rated worse than all other conditions on average. In direct comparison, WM is slightly preferred over WT and ME with an average rating of 0.20 and 0.11, respectively, while there is almost no difference between WT and ME (average rating of -0.02 in favor of ME). We additionally analyzed the occurrence of circular judgments (such as preferring A over B, B over C, and C over A) which were observed in 16 of 44 possible cases (11 listeners and 4 test items). This may indicate limited perceptual discriminability between the test items [146] and is consistent with post-test reports from some listeners, who indicated difficulty in forming clear preferences for some comparisons.

For further analysis, we employ a Cumulative Link Mixed Model (CLMM) [27, 207] to estimate a *relative strength* of each condition, indicating the likelihood that a certain condition is preferred over the others.

In the CLMM, we take individual differences between listeners and test items into account by modeling both as random effects. We did not observe issues with model convergence due to circular or otherwise inconsistent ratings. By applying bootstrapping with 1000 resamplings of the listener responses, we estimate mean and 95% confidence intervals of the estimated strengths which are displayed as a bar plot with error bars on the right side of Figure 5.9. While OR and WH are clearly separated from the other conditions, ME, WM, and WT have overlapping confidence intervals and a mean strength of 2.06, 2.01, and 1.89, respectively.

ME and WT share the same general objective (aligning the pitch with the 12-TET grid), so that they should theoretically yield similar results. However, several other aspects could influence the observed difference in perceived intonation quality and naturalness, including the behavior of the pitch shift curves at note onsets (due to the previous segmentation step, ME allows for faster adaptation) and differences in the quality of the pitch shift algorithms. Notably, WM uses the same hyperparameters (except w) and pitch shift algorithm as WT, yet is closer to ME in estimated strength and is even slightly preferred in the mean ratings, hinting at an overall positive effect of combining the two cost terms. WH (using only harmonic cost, as proposed in [189]) does not yield favorable results, possibly due to limitations in the accuracy of gradients. While this small-scale listening experiment only provides tentative insights, the results suggest that combining cost terms that model different intonation aspects can yield results that are preferred over tonal or harmonic adaptation individually.

5.5 Further Notes

In this chapter, we introduced a method for intonation adaptation via gradient descent. Based on a differentiable cost measure that models a perceptual and a musical aspect of favorable intonation in a multi-voice context, our method can have an advantage over industry-standard methods by automatically handling intonation drift and allowing for a flexible configuration of hyperparameters to accommodate different artistic preferences.

Future work may extend the proposed approach by incorporating additional aspects of intonation into the cost measure. One possibly relevant aspect is horizontal or melodic intonation [227], which can be quantified via the intervals between successive frames or between time-averaged windows spanning multiple frames. Incorporating such measures would enable the adaptation of intonation in melodic lines, such as the singing voice examples discussed in Chapter 2, allowing interval sizes in a melody to be modified without enforcing an absolute tuning grid like 12-TET. In addition, alternative models for dissonance, like *harmonicity* [66, 126, 202], could be explored in contrast or in addition to sensory dissonance, potentially covering additional perceptual aspects of intonation [126].

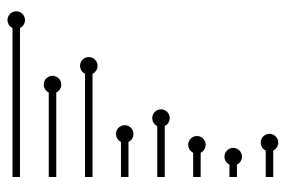
Furthermore, the small-scale listening test presented above, together with additional informal evaluations, suggests that both listener preferences and sensitivity to intonation differences vary substantially across

individuals. While the development of tools for musical expression may reasonably prioritize what performers perceive or intend to convey, the evaluation of intonation adaptation methods ultimately needs to be grounded in the perceptual experience of listeners. Addressing this requires large-scale subjective studies using real-world musical material, in order to better understand under which conditions intonation differences become more or less perceptually salient.

Finally, the differentiable formulation and the context of the chapter within this thesis naturally suggests an integration of the intonation cost measure into machine learning pipelines. For example, when learning intonation between different synthesis models as suggested by Hyodo et al. [85], it may be beneficial to encode previous knowledge about perception (e.g., the sensory dissonance model), or music theory (e.g., the 12-TET cost measure) in the loss function. Directly “learning to synthesize in tune” or performing gradient-based optimization with respect to model inputs during inference (a process called *inference-time optimization*, e.g., [200]) can potentially make a subsequent pitch shifting step obsolete.

6 Towards Timbre Adaptation with Sensory Dissonance

This chapter is based on [182] and [180]. For [182], I was the primary contributor to the study design, conducting the experiments, and preparing the manuscript. Stefan Balke and Meinard Müller contributed throughout all stages of the research and manuscript preparation. Furthermore, Meinard Müller supervised the work. For [180], I was the primary contributor to the development of the research concept, the implementation of experiments, and the manuscript preparation. Meinard Müller and Sebastian Schlecht were involved in all stages of the work, contributing to idea development and manuscript writing.



Sensory dissonance (SD) quantifies the interference between partials in a mixture of simultaneously sounding tones and correlates with the perceived dissonance or unpleasantness of this mixture. In Chapter 5, we used SD to model an aspect of intonation, where intervals closer to just intonation (JI) generally result in lower SD values. However, our analysis did not address how SD behaves for real-world audio signals

beyond pitch modifications. More generally, existing work in music perception and related fields has primarily applied SD to analyze musical dissonance in symbolic representations, such as chords or pitch-class sets, often assuming a fixed, synthetic timbre. As a result, it remains unclear how SD behaves in the context of real-world recorded music. In this chapter, we pursue two objectives. First, we investigate SD as an analysis tool for studying interactions between voices in multi-track music recordings. Using visualizations and statistical analyses on an existing dataset of four-part chorales recorded with various wind instruments, we examine how score, tuning, and instrument timbre influence observed SD values, using the application of this measure in music production workflows as a motivating example. A key finding of this study is that instrument timbre is a dominant factor in determining the relative SD contribution of individual tracks. Second, motivated by this observation, we shift our focus to timbre adaptation. Inspired by the way choir singers adapt their timbre to improve voice blending, we introduce methods for reducing SD in polyphonic additive synthesis by dynamically modifying individual partials. Through a case study and a controlled, small-scale listening experiment, we demonstrate that this approach can perceptually mitigate dissonance in out-of-tune chords and offers creative opportunities to influence the sound of synthesized and recorded tones.

This chapter is organized as follows. First, we revisit the concept of SD in more detail and formalize the notion of *relative SD* in Section 6.1. For the first objective, we then analyze the behavior of SD in multi-track music recordings in Section 6.2. This includes a consideration of practical aspects when analyzing tonal components in recordings (Section 6.2.1), a case study on wind quartets (Section 6.2.2), and a discussion of the key observations (Section 6.2.3). For the second objective, we investigate timbre adaptation in Section 6.3, with an introduction to the methods used (Section 6.3.1), a description of the experiments (Section 6.3.2), and a discussion of the results (Section 6.3.3). Finally, Section 6.4 concludes the chapter with a reflection and an outlook on future work.

6.1 Background: Sensory Dissonance

The musical concept of dissonance is a multi-faceted issue [41] and strongly shaped by cultural influences [118]. While acoustically measurable effects [188, 202] have been shown to correlate with subjective dissonance ratings for isolated intervals and chords [67, 113], they can only serve as indirect proxies for musical dissonance. Sensory dissonance (SD) [190] is such an indirect measure and is based on quantifying the interactions between tonal components in complex sounds [74, 150]. This way, it enables the direct estimation of dissonance in recorded music signals. We previously introduced SD for the *harmonic cost term* in Section 5.2.3, where it was used to adapt intonation toward JI intervals. In the following, we examine the concept of SD in more detail, including its properties and previous applications.

SD plays a central role in music research. It has been employed in perceptual models [14, 83, 113, 216], in bottom-up approaches to scale construction and music-theoretical modeling [13, 190], and in the analysis of intonation and tuning systems [178, 189]. Across these contexts, a variety of models have been proposed to quantify SD [13, 14, 33, 83, 113, 216]. All of these models are based on summing a (weighted) measure of *pure-tone dissonance* over all pairs of tonal components that comprise a complex sound, that is, over all harmonic and non-harmonic partials of all simultaneously sounding tones.

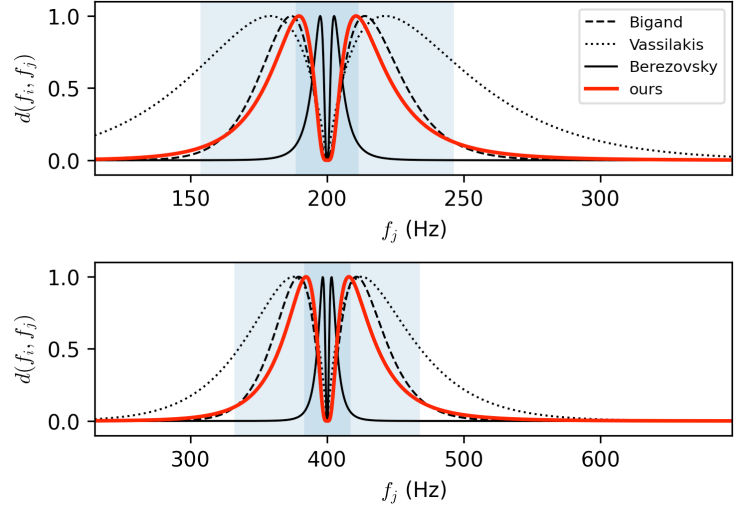
Formally, given a set $\mathcal{P} = \{(f_1, a_1), \dots, (f_K, a_K)\}$ of K pairs of tonal components with frequency f in Hz and amplitude a , SD can be calculated with

$$D(\mathcal{P}) := \sum_{\substack{(f_i, a_i) \in \mathcal{P} \\ (f_j, a_j) \in \mathcal{P}}} w(a_i, a_j) d(f_i, f_j), \quad (6.1)$$

where $w : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is an amplitude-dependent weighting function and $d : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow \mathbb{R}$ is a model for the perceived dissonance between two pure tones. This general formulation forms the basis of all SD models we consider in the following, which differ mainly in their design of the *weighting function* w and *dissonance kernel* d .

The **dissonance kernel** d is generally based on the seminal perceptual experiments by Plomp and Levelt [150], in which listeners were presented with pairs of simultaneously sounding sinusoids and asked to

Figure 6.1: Dissonance kernels $d(f_i, f_j)$ for $f_i = 200$ Hz (top) and $f_i = 400$ Hz (bottom). The blue shaded areas indicate 0.25 ERB (darker color) and 1 ERB (lighter color) around f_i . The kernels shown are from models by Bigand et al. [14] (dashed black line), Vassilakis and Kendall [216] (dotted black line), Berezovsky [13] (solid black line), and our adaptation (red line).



rate the pleasantness of the resulting sound. These experiments indicated that pure-tone dissonance is highest when the frequency difference between two components is small but not too small, reflecting the range of intervals where a strong *beating* effect is present that is associated with the unpleasant perception of *roughness* [209]. Figure 6.1 illustrates several realizations of such dissonance kernels as used in different SD models [13, 14, 216] (black lines), shown for reference frequencies of $f_i = 200$ Hz (top) and $f_i = 400$ Hz (bottom). Across models, the kernels share a qualitatively similar overall shape³³, but differ both in the location of the dissonance maximum and in the rate at which dissonance decays for larger distances between f_i and f_j . For example, with $f_i = 200$ Hz, the kernel proposed by Vassilakis and Kendall [216] (dotted black line) assigns high SD values over a comparatively wide range of intervals, whereas the kernel used by Berezovsky [13] (solid black line) exhibits a much narrower peak.

For our experiments, we use the formulation that we already introduced in Chapter 5, and repeat it here for added context. In Figure 6.1, our resulting dissonance kernel is shown with a red line. It is based on the model by Berezovsky³⁴ [13] and has the advantage of being particularly smooth, also around the point where $f_i = f_j$ (see Appendix B for details on the first derivative). In particular, we set

$$d(f_i, f_j) := \begin{cases} \exp\left(-\ln^2\left(\frac{|\log_2(f_i/f_j)|}{d_{\max}(\bar{f})}\right)\right) & f_i \neq f_j \\ 0 & f_i = f_j, \end{cases} \quad (6.2)$$

where

$$d_{\max}(\bar{f}) := \frac{1}{4} \log_2\left(1 + 24.7(0.00437\bar{f} + 1)/\bar{f}\right), \quad (6.3)$$

³³ A notable exception not shown in Figure 6.1 is the kernel proposed by Marjeh et al. [113]. To better fit the subjective results they obtained in a large-scale perceptual study, they propose to add a region of *negative* dissonance when f_i is very close to f_j , accounting for the particularly pleasant effect of slow beating.

³⁴ The general form $d(x) = \exp(-\log^2(\beta x))$ is mentioned in a footnote in [190, Appendix E], but we were not able to trace the original source cited there.

is a frequency-dependent parameter that controls the interval of maximal dissonance and the decay of the dissonance curve. With this definition of the interval of maximal dissonance, we differ from [13] by explicitly parameterizing it in terms of the *equivalent rectangular bandwidth* (ERB) at the mean frequency $\bar{f} = (f_i + f_j)/2$. ERB is an approximate psychoacoustic measure of the frequency-dependent auditory resolution based on the concept of *critical bands*. The experiments reported by Plomp and Levelt [150] suggest that perceived dissonance reaches a maximum at a frequency separation of approximately 0.25 critical bands. Accordingly, we explicitly set the position of the maximum of d to 0.25 ERB, which we compute at the mean frequency $\bar{f}$ using the formula given in [58] converted to the unit of octaves. It is important to note that this choice is not intended to improve perceptual accuracy but to allow for a more transparent and interpretable parameterization. The methods described in Chapters 5 and 6 remain applicable when using alternative parameterizations of the dissonance kernel d , as long as they are differentiable. As illustrated in Figure 6.1, Equation 6.2 results in a dissonance kernel (shown in red) whose maximum is located exactly at 0.25 ERB (indicated by the blue shaded area).

The **weighting function** w ensures that pairings with high amplitudes contribute more to $D(\mathcal{P})$ since they are more salient. While multiplicative weightings (i.e., $w(a_i, a_j) = a_i a_j$) are often used [113, 216], we instead take the minimum of the two amplitudes following [13], which is proportional to the amplitude fluctuation of the beating that occurs between the two sinusoids [190]. Additionally, many models include an exponent < 1 to account for the non-linearity of loudness perception [13, 113, 190]. However, since the behavior of exponential compression changes when all amplitudes are scaled by a constant factor, we use logarithmic compression, resulting in the weighting function

$$w(a_i, a_j) := \log(1 + \gamma \min(a_i, a_j)), \quad (6.4)$$

where adding the constant 1 ensures positive values for w and $\gamma \in \mathbb{R}_+$ is a parameter to control the amount of compression [131]. Throughout our experiments, we fix $\gamma = 10$. Preliminary tests indicated that $D(\mathcal{P})$ is sensitive to the degree of compression, as stronger compression increases the relative weight of less salient harmonics. Just as for the dissonance kernel d , we are not aiming for the best possible alignment with perception with our parameterization, but provide an interpretable compression parameter γ that is robust to the absolute scale of the amplitudes.³⁵ Finally, it is common to also include some kind of normalization that makes $D(\mathcal{P})$ independent of the overall loudness of the sound. We omit this step here and consider loudness normalization an optional preprocessing step, for example in Section 6.2.1.

6.1.1 Relative Sensory Dissonance

Extending ideas from [190] and [178], we can decompose $D(\mathcal{P})$ into contributions from different components of a sound. In a multi-track polyphonic music scenario, this may give additional insights into

³⁵ For perceptual accuracy, γ could be fitted to observed subjective results, similar to the exponent r being empirically set to a value of $r = 1.359$ in [113].

Figure 6.2: Illustration of considered subsets for relative SD.

$D_{v,v}$	$D_{v,\bar{v}}$	$D_{S,S}$	$D_{S,A}$	$D_{S,T}$	$D_{S,B}$
		$D_{A,S}$	$D_{A,A}$	$D_{A,T}$	$D_{A,B}$
$D_{v,\bar{v}}$	$D_{\bar{v},\bar{v}}$	$D_{T,S}$	$D_{T,A}$	$D_{T,T}$	$D_{T,B}$
		$D_{B,S}$	$D_{B,A}$	$D_{B,T}$	$D_{B,B}$

how individual sources or voices contribute to dissonance. To formalize the decomposition, we define subsets of $\mathcal{P}$ corresponding to different voices in a sound, so that for an index set $\mathcal{V}$,

$$\mathcal{P} = \bigcup_{v \in \mathcal{V}} \mathcal{P}_v. \quad (6.5)$$

Furthermore, for $v \in \mathcal{V}$, we define the complementary set

$$\mathcal{P}_{\bar{v}} = \bigcup_{v' \in \mathcal{V} \setminus \{v\}} \mathcal{P}_{v'} \quad (6.6)$$

containing the tonal components of all other voices. Extending Equation 6.1, we compute the SD between the tonal components from two separate sets $\mathcal{P}_1$ and $\mathcal{P}_2$ with

$$D(\mathcal{P}_1, \mathcal{P}_2) := \sum_{\substack{(f_i, a_i) \in \mathcal{P}_1 \\ (f_j, a_j) \in \mathcal{P}_2}} w(a_i, a_j) d(f_i, f_j), \quad (6.7)$$

which we also denote by $D_{1,2}$ in the following for brevity, along with D without subscripts for the overall dissonance $D(\mathcal{P}, \mathcal{P})$. Equation 6.7 allows us to decompose D for any voice $v \in \mathcal{V}$ as shown in Figure 6.2 on the left. Here, $D_{v,v}$ represents the *intrinsic* SD within a single voice, $D_{\bar{v},\bar{v}}$ is the SD independent of v , and $D_{v,\bar{v}}$ is the *relative* SD between a voice and its accompaniment. This results in a final decomposition $D = D_{v,v} + 2D_{v,\bar{v}} + D_{\bar{v},\bar{v}}$.

As a concrete example, consider the four-part chorale case, where $\mathcal{V} = \{S, A, T, B\}$. The right side of Figure 6.2 illustrates the possible relative SD pairings in this scenario. Particularly relevant are the intrinsic dissonances $D_{S,S}$, $D_{A,A}$, $D_{T,T}$, and $D_{B,B}$, as well as the relative dissonances $D_{S,\bar{S}}$ (depicted in orange), $D_{A,\bar{A}}$ (red), $D_{T,\bar{T}}$ (green), and $D_{B,\bar{B}}$ (blue). It becomes evident from the right side of Figure 6.2 that the individual $D_{v,\bar{v}}$ are not independent, since a change in any $\mathcal{P}_v$ influences the relative SD measurements for all other voices due to the symmetry of the matrix. Yet, Figure 6.3d shows how $D_{v,\bar{v}}$ can still be used to clearly identify the different contributions to overall SD.

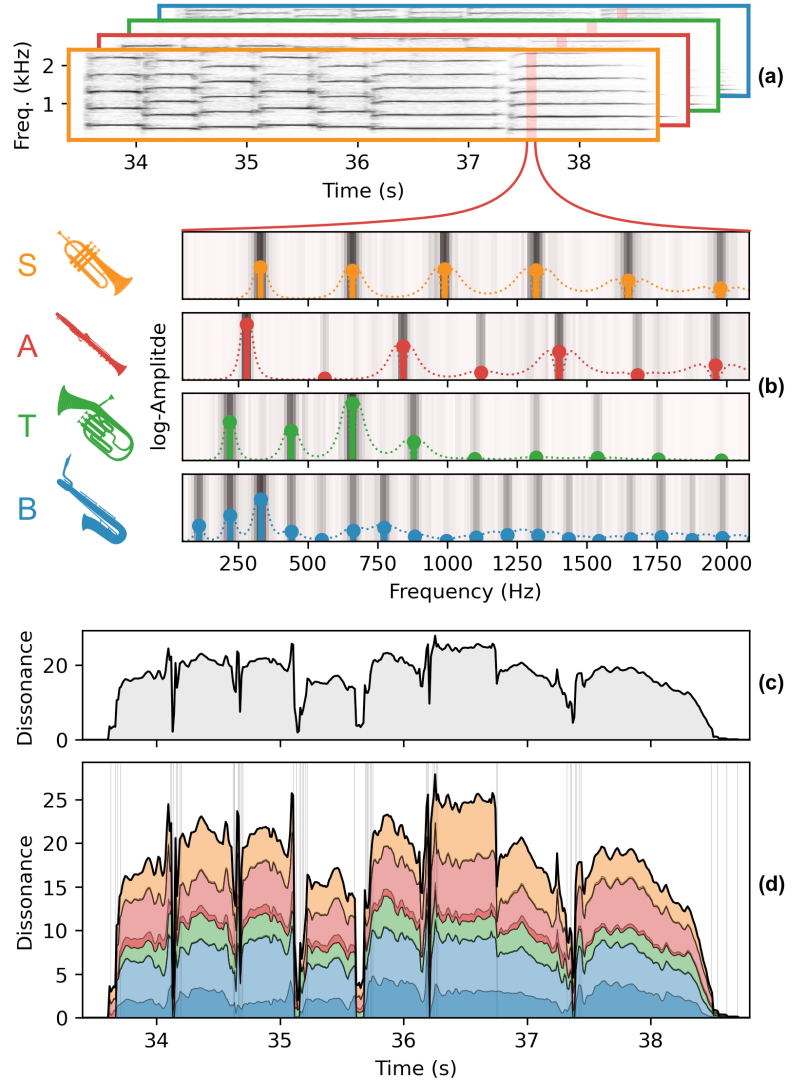
6.2 Sensory Dissonance in Multi-Track Music Recordings

In music production, the creative process of editing and mixing can often be aided by objective measures of sound properties, such as displaying loudness differences with a *level meter* or visualizing phase differences between stereo channels with a *goniometer*. To our knowledge, a property that has not yet been considered in this context is the dissonance in a track or recording. With the interplay between consonance and dissonance being considered a core component of musical expression [208], such a measure could give insights into musical properties of a mix, both for relative comparisons (e.g., to evaluate intonation and voice blending between different tracks) and as an absolute quantity (e.g., for retrieving sections with high dissonance). For the first objective in this chapter, we explore SD as a way to quantify dissonance in recorded music performances directly from audio. Similar to the goniometer that does not measure the subjective impression of the stereo image, the goal of the following case study is not to analyze perceptual properties of SD, but to better understand its behavior in a realistic musical scenario, exploring what it can reveal about the relations between individual voices in a multi-track recording.

We approach this question with an exploratory case study using the ChoraleBricks dataset [8]. It comprises performances of ten Baroque chorales, each with four voices—soprano (*S*), alto (*A*), tenor (*T*), and bass (*B*)—that are recorded in isolation and played on several different wind instruments. This way, the dataset provides a controlled scenario for discerning the influence of different musical aspects, including timbre (instrument characteristics), tuning (pitch deviations), and score (chords and voicings). As an initial example, we consider an excerpt from the chorale “Befiehl Du Deine Wege” (GE1 in ChoraleBricks) in Figure 6.3, played with trumpet (*S*, orange), clarinet (*A*, red), baritone horn (*T*, green), and baritone saxophone (*B*, blue). After estimating tonal components (in this case, the harmonics) over time from each of the four tracks (Figure 6.3a and b), we can calculate the overall SD (shown in Figure 6.3c) and split it into the relative contributions of each voice (shown in Figure 6.3d).

This visualization—inspired by Sethares’ *dissonance score* [190]—reveals three main aspects of SD. First, it allows us to explore differences between voices. With Figure 6.3d showing the relative SD for the four voices, we can clearly see, for example, that the *T* voice (green) tends to contribute less to the overall SD than the *B* voice (blue). Second, it allows us to observe differences between chords. With Figure 6.3d showing the note onsets of each voice as vertical lines for orientation, it becomes clear that the overall and relative SD of individual voices change substantially from one chord to another. As an example, consider the transition from the chord at around 36.5 seconds (D minor with suspended second) to the following chord at around 37 seconds (D minor). In this case, the relative SD of the *A* voice (red) drops sharply, since the musical dissonance of a suspended second in this voice is resolved to a musical consonance. Third, the visualization allows us to see and analyze local fluctuations of SD within chords. For example, around 38 seconds, the overall SD goes down, in particular driven by a reduction in relative SD in the *A* voice (red) and the *B* voice (blue).

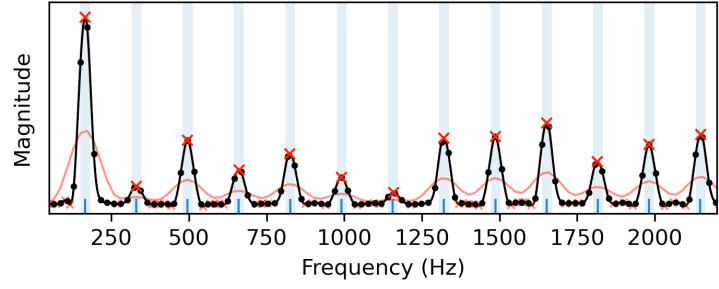
Figure 6.3: Excerpt from the chorale GE1 in ChoraleBricks [8]. **(a)** Spectrograms of individual voices played with trumpet (*S*), clarinet (*A*), baritone (*T*) and baritone sax (*B*). **(b)** Peak representation $\mathcal{P}_v(m)$ for frame m and each voice v . Dotted lines illustrate the amplitude-weighted dissonance kernels. **(c)** Overall dissonance D of the excerpt. **(d)** Relative SD by voice (light: $D_{v,\bar{v}}$, dark: $D_{v,v}$).



Exploring the effects illustrated in Figure 6.3 in detail is a main objective of this case study. In this context, we consider practical aspects of calculating SD from real signals (Section 6.2.1), we examine the influence of timbre, tuning, and score on relative SD by introducing new visualizations and conducting systematic experiments with ChoraleBricks (Section 6.2.2). Finally, we discuss our main observations (Section 6.2.3). A Python library with all tools used in the experiments and our new chord annotations for ChoraleBricks are available online.³⁶

³⁶ <https://audiolabs-erlangen.de/resources/MIR/2025-ISMIR-SD>

Figure 6.4: Illustration of harmonic peak picking for a single frame of a bass clarinet signal, showing the original DFT spectrum (black dots), the cubic spline interpolation (black line), extremal points of the interpolation function (red crosses), search range (light blue areas) around F0 multiples (blue ticks), and the local threshold (red line).



6.2.1 Tonal Components in Recordings

In Section 6.1, we have only considered the case where $\mathcal{P}$ represents a sound with constant characteristics. To account for time-varying music signals, let $\mathcal{P}(m)$ represent the tonal components (equivalent to harmonics in the monophonic case) at a time index $m \in \mathbb{Z}$. In a multi-track scenario, we further require a method to obtain robust estimates of $\mathcal{P}_v(m)$ from a time-domain signal x_v for $v \in \mathcal{V}$. The frequency resolution of a DFT with any reasonable window size would not be sufficient for this purpose. Prior methods have used a peak picking algorithm on the magnitude spectrum, refined with parabolic interpolation [195] or by zero-padding [190], which requires intricate fine-tuning of parameters for robust detection of harmonics. Instead, we propose a targeted peak picking approach that leverages F0 estimates (denoted by f_0 in the following), assuming a quasi-harmonic overtone structures. f_0 estimates can be obtained robustly using monophonic algorithms (e.g., [19]), or with predominant [168] or polyphonic [225] f_0 estimation algorithms for more complex input signals. Since many musical instruments produce near-harmonic spectra, this method is applicable to a wide variety of recordings.

The frequency and amplitude of each harmonic is obtained in three steps, as illustrated in Figure 6.4. First, we construct a cubic interpolating spline representation (black curve) of the magnitude spectrum (black dots) and compute its derivative, a piecewise quadratic function. Second, we find the roots of the spline derivative, identifying the extremal points of the original cubic splines (red crosses). The largest extremum in the vicinity of nf_0 (blue rectangles) is then assumed to be the spectral peak corresponding to the n th harmonic. To refine these estimates (e.g., when the even harmonics of a clarinet are below background noise level), we finally apply local thresholding, removing any peaks whose total magnitude is below a certain value. The local threshold (red curve) is calculated using a sliding Hann window with an adaptive width depending on f_0 , so that the averaging spans $N_w = \lceil 1.2 \cdot f_0 N / F_s \rceil$ frequency bins. This way, a single harmonic with large magnitude does not influence the threshold at the neighboring harmonics. With this method, it is possible to accurately track harmonics across multiple frames of an STFT without explicitly modeling continuity between frames. For STFT frames where the f_0 estimate indicates an unvoiced frame, we set $\mathcal{P}_v(m) = \emptyset$.

When using logarithmic compression for w in Equation 6.1, the scale of the amplitudes a_i in $\mathcal{P}_v(m)$ can be arbitrary, as long as it is consistent across all voices. It can further be desirable to remove the influence of loudness (and also relative level differences between the individual voices), which can be achieved by

normalizing all amplitudes in $\mathcal{P}_v(m)$ to sum to one. In the particular case of ChoraleBricks, voices are recorded in isolation and not in a musically meaningful loudness balance anyway. Therefore, we divide each a_i by a constant

$$C = \max \left(c, \sum_{(f_i, a_i) \in \mathcal{P}_v} a_i \right), \quad (6.8)$$

where limiting C to not become smaller than $c = 0.1$ avoids artificially inflating the influence of frames with very low overall level.

6.2.2 Case Study: ChoraleBricks

The first objective in this chapter is to study the behavior of SD in a realistic musical scenario. Specifically, we want to disentangle the influence of different musical properties—timbre, tuning, and score—of individual voices in a multi-track recording on the SD values, and to which extent this measure enables comparisons between different instruments, takes, or compositions.

For this exploratory analysis, we use the ChoraleBricks dataset [8], which provides recordings of ten Baroque four-part chorales. Their composition style is *homophonic*, i.e., voices follow a synchronized rhythm while forming chords with a main melody (usually in the soprano), using relatively simple chords and voicings. Furthermore, the dataset contains isolated recordings of each voice played on several different wind instruments, allowing for a comparison of various four-instrument combinations (*ensembles*) playing the same chorale. To simplify the notation of ensembles, we introduce shorthands. The ensemble used in most experiments and visualizations (e.g., Figure 6.3) is denoted by $E = (\text{tp}, \text{cl}, \text{bar}, \text{bs})$, using a tuple of instrument IDs as shown in Table 6.1 in the order of S , A , T , and B . To denote a single instrument being replaced in E , we use a subscript, for example, $E_{S=\text{fl}} = (\text{fl}, \text{cl}, \text{bar}, \text{bs})$. In addition to E (with two woodwinds and two brass instruments), we assemble a pure woodwinds ensemble $E_{\text{wood}} = (\text{ob}, \text{cl}, \text{bs}, \text{bcl})$ and a pure brass ensemble $E_{\text{brass}} = (\text{tp}, \text{fh}, \text{bar}, \text{bar})$, always choosing the instrument with the highest number of available recordings for the respective voice. Finally, as a synthetic baseline for comparison, we create an ensemble E_{saw} , where each voice is synthesized using a sawtooth waveform with 20 harmonics and the respective 12-tone equal temperament (12-TET) frequency for each note as F0. Using these ensembles and the excerpt from Figure 6.3 as a running example, we can try to disentangle the influences of timbre, tuning and score on SD.

6.2.2.1 The Influence of Timbre

In Figure 6.3d, large differences between instruments in relative (lighter color) and intrinsic SD (darker color) can be observed, most prominently for the T and B voice, played on bar and bs . To reveal the cause of these differences, we first characterize the timbre of the different instruments. While the full phenomenon of timbre encompasses many properties of a sound [194], we focus on the most relevant aspect for SD, namely the distribution of amplitude across harmonics independent of the overall loudness

Figure 6.5: Amplitude distribution across the first 20 harmonics in $\mathcal{P}_v$ for each instrument (averaged over the entire dataset, not showing values below -60 dB). The sawtooth timbre *saw* is shown for comparison. This is a reproduction of Figure 2.3.

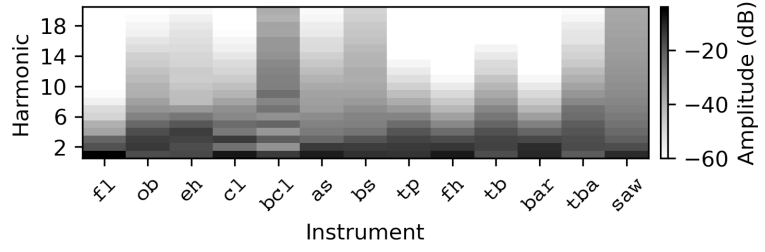


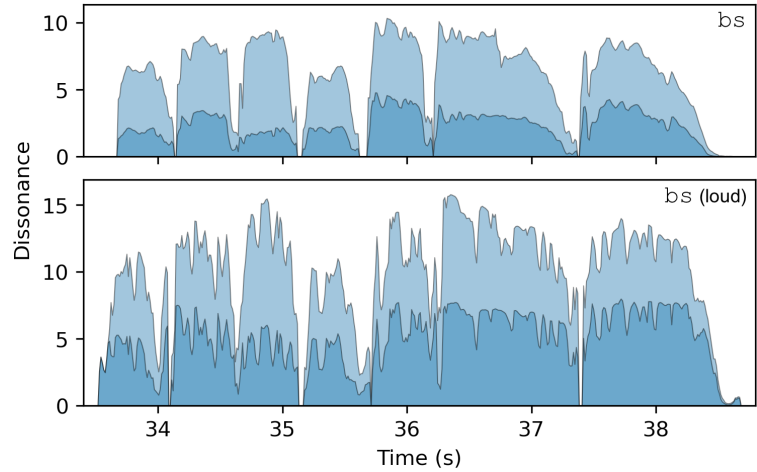
Table 6.1: SD statistics by instrument, showing intrinsic SD $D_{v,v}$, relative SD $D_{v,\bar{v}}$, and the harmonic centroid (HC, mean $\pm$ standard deviation). Values for the synthetic sawtooth ensemble E_{saw} shown for comparison.

Instrument	ID	# Notes	$D_{v,v}$	$D_{v,\bar{v}}$	HC
Flute	fl	449	0.03	1.85	1.55 ± 0.3
Oboe	ob	449	0.60	3.65	3.67 ± 0.7
English Horn	eh	460	0.74	3.70	3.78 ± 0.8
Clarinet	cl	909	0.53	3.43	2.94 ± 0.7
Bass Clarinet	bc1	545	4.60	6.08	6.98 ± 2.0
Alto Sax	as	103	0.94	4.38	3.06 ± 0.9
Baritone Sax	bs	816	1.92	4.68	3.99 ± 1.4
All Woodwinds		3731	1.34	3.97	3.71
Trumpet	tp	909	0.22	3.35	3.00 ± 0.8
Fluegelhorn	fh	909	0.12	2.75	2.25 ± 0.6
Trombone	tb	372	0.45	2.83	3.69 ± 1.3
Baritone	bar	963	0.16	1.90	2.43 ± 0.8
Tuba	tba	464	1.12	2.59	4.50 ± 1.3
All Brass		3668	0.41	2.68	3.17
Sawtooth S		449	1.37	5.76	5.56 ± 0.0
Sawtooth A		460	1.43	7.01	5.56 ± 0.0
Sawtooth T		456	1.50	7.02	5.56 ± 0.0
Sawtooth B		464	1.71	4.92	5.56 ± 0.0
All Sawtooth		1829	1.50	6.18	5.56

of the sound, as expressed by the normalized $\mathcal{P}_v$. Figure 6.5 (a reproduction of Figure 2.3) shows the average amplitude distribution in the first 20 harmonics for each instrument in ChoraleBricks. Woodwinds (except for the flute) tend to have higher values in the upper harmonics (a *brighter* timbre), while flute, fluegelhorn and baritone horn on average have most sound energy concentrated in the low harmonics. We can quantify this difference with the harmonic centroid (HC), describing the mean harmonic index weighted by average amplitude, similar to the spectral centroid, which is often used as a timbre descriptor [148]. The HC for each instrument is given in Table 6.1, and we can observe that for example the bass clarinet (bc1) has its energy centered around the seventh harmonic (HC of 6.98), while for the flute (fl), the HC indicates that the most energy is in the fundamental (HC of 1.55).

These differences in timbre naturally also affect both intrinsic SD $D_{v,v}$ and relative SD $D_{v,\bar{v}}$. Table 6.1 also lists the average $D_{v,v}$ and $D_{v,\bar{v}}$ for each instrument, which are calculated relative to E , i.e., by replacing the respective voice in E with this specific instrument. For example, when the trombone played both T and B , we consider $E_{T=\text{tb}}$ and $E_{B=\text{tb}}$ for each chorale where the respective recording is available. We can observe that HC and the SD values are highly correlated. Furthermore, the instrument choice accounts for 49% of the variation in $D_{v,\bar{v}}$ over the entire dataset and 78% of the variation in $D_{v,v}$, as

Figure 6.6: Two different takes for *B* played with bs (performed normally and loudly) for the excerpt from Figure 6.3. Note that both $\mathcal{P}_v$ are loudness-normalized, so that differences in D only stem from changes in timbre and tuning.



indicated by the effect size (η^2) for the Kruskal-Wallis non-parametric statistical test [102]. This makes the instrument’s harmonic amplitude distribution the largest predictor of a voice’s contribution to SD.

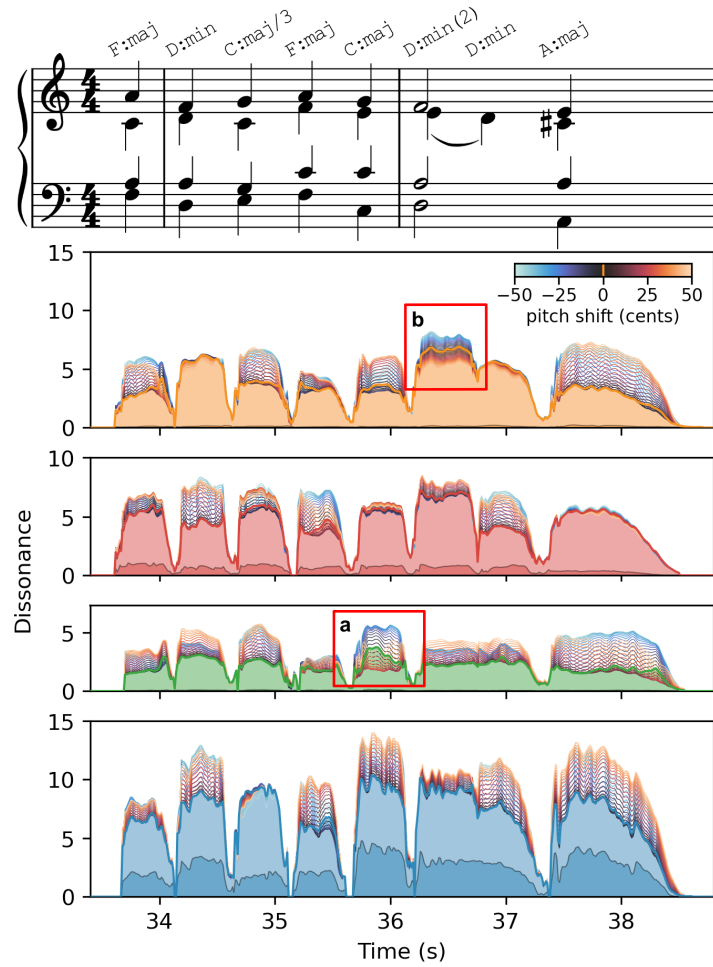
In addition, there are also considerable timbre variations *within* each instrument, reflected for example in the standard deviation of HC reported in Table 6.1. As an illustrative example, Figure 6.6 shows two different takes of the *B* voice played on bs. In the second take (“loud”), the player was instructed to play as loud as possible. Notably, despite the loudness normalization of $\mathcal{P}_v$, both $D_{v,v}$ and $D_{v,\bar{v}}$ are larger by almost a factor of two for the loud take, which also shows stronger fluctuations of relative SD within the individual chords. The possibility to influence SD through variations in an instrument’s timbral qualities may also explain the reduction in SD towards the end of the example in Figure 6.3, an effect that is not present for the loud take.

6.2.2.2 The Influence of Tuning

In Chapter 5, we used SD as a context-sensitive measure for tuning related to just intonation (JI). In this section, we aim to quantify how large the effect of tuning is on SD compared to other influences like timbre. To visualize this in our running example, we *virtually pitch-shift* each voice by a certain amount $p \in [-50, 50]$ in cents by multiplying all frequencies in $\mathcal{P}_v(m)$ with a factor $2^{p/1200}$. We then calculate for each value of p the new relative dissonance $D_{v,\bar{v}}$ against the unmodified other voices of the ensemble ($D_{v,v}$ remains mostly unaffected by a small pitch shift). In other words, we measure how much relative SD changes when a performer would change their intonation while the rest of the ensemble plays normally.

The result is shown in Figure 6.7. $D_{v,v}$ and $D_{v,\bar{v}}$ is plotted for each voice separately, and the lines from light blue ($p = -50$ cents) to light red ($p = 50$ cents) indicate the change in $D_{v,\bar{v}}$ with variable tuning. Notably, the *tuning sensitivity* of $D_{v,\bar{v}}$ (i.e., the range between the maximum and minimum $D_{v,\bar{v}}$ within a note when p is varied by ± 50 cents) changes between notes and chords. In particular, for most major and minor thirds in the excerpt, SD increases only slightly even for the largest pitch shifts. Considering

Figure 6.7: Influence of detuning each voice in *E* by ± 50 cents (excerpt from GE1, as in Figure 6.3). Chord labels and sheet music are shown time-aligned for reference.



the statistics over the entire dataset, we find that for the ensemble *E*, the root note has an average tuning sensitivity of 2.71, the minor third of 1.09, the major third of 1.01, and the fifth of 1.70 (only considering chord degrees with frequent occurrences). We must however account for the fact that the root note of each chord is often doubled (in the same or a different octave) in the four-part voicings of chorales. If we consider only cases where the respective note is not doubled, we record an average tuning sensitivity of 1.13 (root), 0.75 (minor third), 0.87 (major third), and 1.62 (fifth). This suggests that SD is more sensitive to the tuning of the root and fifth than to that of the thirds. Furthermore, in comparison to the mean relative SD by instrument in Table 6.1, tuning sensitivity in general is relatively small, so that measuring tuning (e.g., 12-TET and JI, which often differ only by a few cents) with SD is only meaningful in conditions where timbre remains unchanged.

There are two cases in the excerpt where the visualization shows that a change in tuning would significantly reduce relative SD. First, the root note of the C:maj chord in the *T* voice (red box **a** in Figure 6.7) would contribute less to *D* if it was played around 25 cents higher. In fact, according to the annotated F0, the *T* voice is on average 17 cents flat relative to the octave formed with *B*. Interestingly, this does not affect the “optimal tuning” (w.r.t. SD) of *B*, indicating that the intervals formed with *S* and *A* are stable. Second, for

Table 6.2: SD statistics by different chord types. Chord symbols follow the notation scheme from [68].

Chord Type	#	D			
		E	E_{wood}	E_{brass}	E_{saw}
maj	229	15.21	31.74	7.90	29.73
min	119	15.99	34.14	9.02	31.96
maj/3	55	17.65	36.16	10.50	31.65
sus4	16	16.64	34.75	8.72	30.02
min/3	12	19.17	36.52	7.83	27.78
others	32	17.00	33.11	10.25	31.20

the minor third of the $D:\text{min}(2)$ chord in S (red box **b** in Figure 6.7), SD would be reduced if the note was played up to 50 cents lower. Given that this note forms the musically dissonant interval of a minor second with A , this is an indication for a case where SD does not exhibit a local minimum near the interval prescribed by the score.

6.2.2.3 The Influence of Score

Finally, we consider the influence of score on SD, in particular by analyzing differences between chords. In Figure 6.7, a prominent example for the score influence is the $D:\text{min}(2)$ chord that has the highest overall SD in the excerpt. The relative SD of S , A , and B , as well as the overall SD significantly drops when the musically dissonant suspended second is resolved in voice A .

Since the chord vocabulary of the compositions is relatively limited (major and minor chords in root position account for 75% of all chords), we can statistically analyze differences between chord types by grouping them as shown in Table 6.2, using the chord labels without root. In particular, we compare how the chord type influences the overall SD for E , E_{wood} , E_{brass} , and E_{saw} by computing the mean D over the entire dataset. In fact, the variations in mean SD by chord type are mostly consistent across ensembles. For example, the mean SD of maj chords is lower than that of min chords. /3 chords with the third in B are more dissonant, except for min/3 in E_{brass} . These trends partly resemble a ranking based on subjective dissonance ratings of triads and tetrads [17], replicating a previous result with synthetic data [67]. However, even within one ensemble, we find that the chord type only accounts for between 6% (E_{wood}) and 11% (E_{brass}) of the variation in D (according to the Kruskal-Wallis test as above).

Another score-related property of each chord is its voicing, i.e., the assignment of notes to the individual voices. Two effects can be observed in the statistics for individual voices of E_{saw} in Table 6.1. First, the average $D_{v,v}$ increases towards lower voices. This follows from more harmonics falling into the critical bands around neighboring harmonics, as can be observed in Figure 6.3b. Second, the average $D_{v,\bar{v}}$ is higher for middle voices, indicating that these voices tend to contribute more to the overall SD, independent of timbre and tuning, which is fixed in E_{saw} .

6.2.3 Discussion

From this analysis, we can draw three main conclusions for using SD as an informative measure in music production. First, the relative SD of individual tracks provides insights into their musical interaction, e.g., in terms of tuning and the musical dissonance of intervals. However, as an absolute measure, SD is mainly determined by instrument timbre, prohibiting direct comparisons across instrument classes. Second, SD may offer an advantage over F0-based tuning analysis because it is context-sensitive, accounting for intervals between all voices. The sensitivity to tuning differences depends on the chord degree and voicing, where even strongly detuning a third by ± 50 cents may in some chords only slightly affect the SD measure. As an example, if we consider the *A* voice in the last chord in Figure 6.7 and recapitulate or use of SD as a harmonic cost measure, the gradients of this cost measure would not have a particularly large magnitude, even though nuances of a few cents can have musical relevance in ensemble performance practice (see, e.g., [69]). Third, while systematic score influences are present, they are comparatively small, which makes the use of SD in applications like musicological analysis as outlined in [190] only feasible within controlled scenarios.

At this point, it should be emphasized again that SD does not measure perceived or musical dissonance directly, and that our used model was not parameterized to achieve maximal agreement with perceptual data. As a result, different modeling choices for SD, as well as alternative acoustic measures such as those based on harmonicity (e.g., [66, 126, 202]), could lead to substantially different, but equally valid, outcomes. Nevertheless, the patterns observed when SD is used as a relative measure or visualization tool suggest several practical applications, some of which we will outline in Section 6.4. In particular, the observation that SD is strongly influenced by timbre motivates approaches that adapt this tone attribute directly, rather than focusing exclusively on pitch, as in the intonation adaptation algorithm introduced in Chapter 5. This is a primary motivation for the following experiment.

6.3 Timbre Adaptation in Additive Synthesis

For the second objective of this chapter, we build on the insight that the relative SD of a voice is strongly influenced by the instrument’s timbre and explore methods for timbre adaptation in the context of additive synthesis. Our goal is to better understand how changes in timbre, achieved by adapting the frequency and amplitude of individual harmonics, affect SD and, in turn, the perception of dissonance in complex chords. Our approach is inspired by choir singers, who not only adjust their pitch dynamically but also control their vocal formants, i.e., the prominence of certain partials, to support intonation and voice blending [69].

In this experiment, we do not explicitly make use of the differentiability of SD. Instead, we introduce and compare two heuristic methods that suppress contributions from pairs of partials that would otherwise have a strong influence on SD due to their close spacing in frequency. Although it would be straightforward to use gradients of SD with respect to frequencies and amplitudes to directly update the parameters

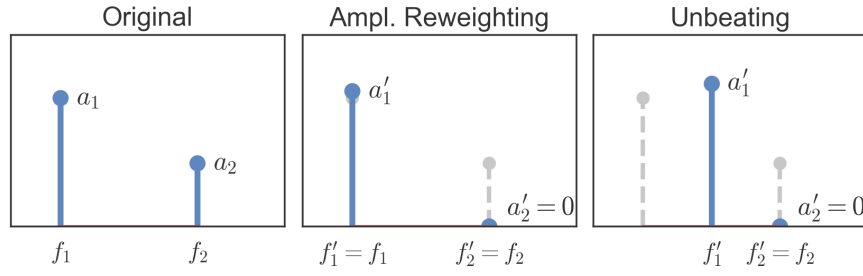


Figure 6.8: Visualization of the adaptation of a clashing pair of partials with amplitude reweighting and unbeating.

for the additive synthesis, our heuristic methods are designed to illustrate boundary cases of maximal timbre adaptation. Overall, with this form of timbre adaptation, we introduce an additional degree of freedom in sound synthesis, allowing the roughness of a sound to be modified without altering other tonal attributes such as F0 or loudness. In the following, we will describe these heuristic adaptation methods (Section 6.3.1), present experiments with synthetic signals (Section 6.3.2), and discuss the objective and subjective results (Section 6.3.3).

6.3.1 Method

Recall that the overall SD $D(\mathcal{P})$ increases when there are individual pairs of tonal components with non-zero amplitude in $\mathcal{P}$ whose frequency is similar but not equal. We call these “clashing pairs” in the following. In a polyphonic sound, $D(\mathcal{P})$ is governed by the number of clashing pairs, since all other combinations do not contribute meaningfully to the sum in Equation 6.1, either because the two frequencies are too far away from each other, or because one of the amplitudes is (near) zero. Considering additive synthesis (see Section 2.4.1), where the frequencies and amplitudes in $\mathcal{P}$ can be directly used as parameters for individual sinusoids, we can freely modify individual tonal components instead of just pitch-shifting an entire tone with all its harmonics, as we did in Chapter 5.

This leaves us with two options to reduce the impact of a clashing pair with parameters (f_i, a_i) and (f_j, a_j) on $D(\mathcal{P})$: lower the amplitude a_i and/or a_j to affect the weighting w in Equation 6.1, or change the frequencies f_i and/or f_j to affect the dissonance kernel d . While both adaptations, changing amplitude or individual frequencies, naturally alter the timbre as they change the relation between partials, different adaptation strategies may have different implications on timbral attributes and perception. We introduce two possible methods for SD minimization by adapting a given clashing pair of partials (f_i, a_i) and (f_j, a_j) to their new frequency and amplitude values (f'_i, a'_i) and (f'_j, a'_j) , respectively. The effects of the methods are illustrated in Figure 6.8.

Method 1: Amplitude Reweighting – Given $a_i > a_j$, set $a'_i = (a_i + a_j)/\sqrt{2}$ and $a'_j = 0$. By setting the smaller of the two amplitudes to zero, it is effectively removed from the synthesis, while the overall energy remains constant due to the reweighting of a'_i . The frequencies of the partials remain unchanged ($f'_i = f_i$ and $f'_j = f_j$).

Method 2: Unbeating – High SD coincides with the occurrence of beating between two partials [209]. This amplitude modulation of a “carrier” frequency can be shown by the trigonometric sum-to-product identity [87]. For the two corresponding sinusoidals with time t in seconds, we can write

$$a_i \sin(2\pi f_i t) + a_j \sin(2\pi f_j t) = c_t \sin\left(2\pi((f_i + f_j)/2)t + \xi_t\right), \quad (6.9)$$

where

$$c_t = (a_i^2 + a_j^2 + 2a_i a_j \cos(4\pi(f_i - f_j)/2t))^{(1/2)} \quad (6.10)$$

is the time-varying amplitude modulation and

$$\xi_t = \arctan\left((a_i - a_j)/(a_i + a_j) \tan(2\pi(f_i - f_j)/2t)\right) \quad (6.11)$$

is a time-varying carrier phase modulation. To reduce beating, we can set

$$c_t = c_0 = \sqrt{a_i^2 + a_j^2} \quad (6.12)$$

to be constant over time, removing the amplitude modulation (i.e., “unbeating” the sound). As a result, we set $a'_i = c_0$, $a'_j = 0$, $f'_i = (f_i + f_j)/2 + \xi_t/(2\pi t)$ and $f'_j = f_j$. Optionally, we can additionally remove the phase modulation, setting $\xi_t = 0$.

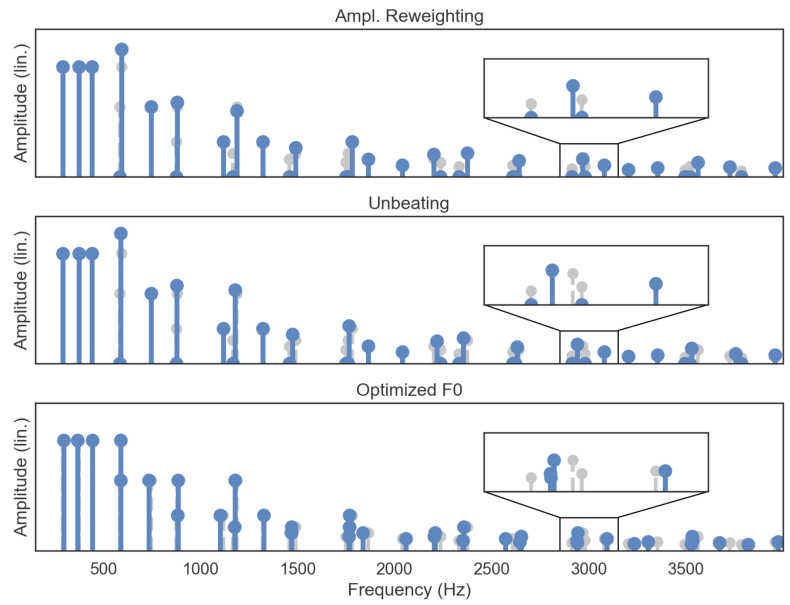
Note that these methods allow the timbre of complex, polyphonic sounds to be adapted dynamically by adjusting partial amplitudes and/or frequencies over time, for example on a per-chord basis.

6.3.2 Experiments

For an initial exploration of the introduced methods, we consider the use case of reducing SD that is induced by inaccurate intonation. To simulate this situation, we use additive synthesis as introduced in Section 2.4.1, where we have full control over all the partials of each voice. We synthesize example sounds that consist of three different chords with four voices: D (D₅, A₄, F₄[#], D₄), Am (A₄, E₄, C₄, A₃), and E7 (G₄[#], D₄, B₃, E₃) and introduce slight random detuning (± 20 cents) of the F0 of each voice to introduce additional dissonance in the original sounds. For each chord, we use two different timbres with $K = 12$ partials: A sawtooth waveform and a “clarinet-like” timbre, where partial amplitudes and frequencies are estimated from a recorded clarinet tone. All sounds are synthesized with constant parameters over the four-second duration of the signal. Finally, we compare six experimental conditions:

- *Original* — detuned without adaptation.
- *Amplitude Reweighting* — as described above.
- *Unbeating I* — as described above, without phase modulation ($\xi_t = 0$).
- *Unbeating II* — as described above, including phase modulation ξ_t .
- *12-TET* — original partials but without detuning of the F0, i.e., using 12-tone equal temperament

Figure 6.9: Comparison of partials for the D chord in different experimental conditions (gray: original partials, blue: adapted partials).



- *Optimized F0* — original partials, but the F0 is adapted to minimize SD, as proposed in Chapter 5, yielding results comparable to a JI tuning.

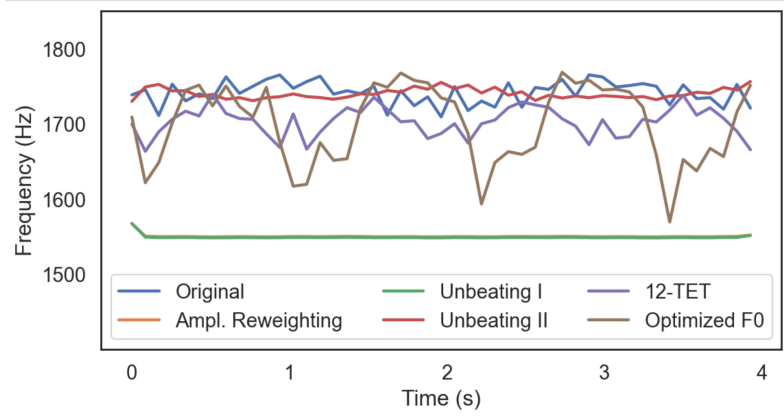
In total, this results in 36 unique sound items (two timbres, three chords, six conditions), which we can compare in terms of their objective and subjective properties. The discussed sound examples are available on a supplemental website.³⁷

Throughout the experiments, we treat a pair of partials as clashing when the distance between frequencies is larger than one cent and smaller than 50 cents. While different definitions of clashing partials are possible, for example based on critical bands or a threshold of the dissonance kernel d , this simple strategy can easily be used to adjust the effect strength and it is musically interpretable. To remove clashing pairs in $\mathcal{P}$, we iterate over all partials in the set and apply the respective adaptation method when another partial is closer than the threshold.

Figure 6.9 illustrates how the different experimental conditions and the chosen distance threshold affect the partials of the D chord. The zoomed-in view for each method focuses on the frequency range between approximately 2900 and 3100 Hz. In this range, a cluster of partials can be observed whose frequency distances fall below the threshold. In the visualization, the original partials are shown in gray and the adapted partials are shown in blue. In the *Amplitude Reweighting* condition, the amplitudes of the weaker partials are set to zero, and the strongest partial is amplified to compensate for the resulting reduction in overall loudness. In the *Unbeating* condition, the cluster is represented by a single tonal component placed between the original partials of the cluster. In contrast, in the *Optimized F0* condition, the harmonics of each tone are shifted by a constant factor to minimize SD. For the cluster shown here, this causes the

³⁷ <https://audiolabs-erlangen.de/resources/2023-TIMBRE-AdaptiveTimbre>

Figure 6.10: Spectral centroid fluctuations over time for the D chord with sawtooth timbre.



partials to move closer together. Note how the F0 shift also affects the separate higher harmonic that is visible in the zoomed-in section.

To compare the different conditions objectively, we compute two metrics for each sound item: the overall SD $D(\mathcal{P})$ as a proxy measure for dissonance, and the frame-wise spectral centroid of the synthesized audio to characterize how the methods affect the spectral envelope of the sound. The spectral centroid is defined as the amplitude-weighted mean frequency of a spectrum and provides a simple measure of perceived brightness [63].

Furthermore, we conduct a small-scale listening test, asking participants to judge the perceived “pleasantness” of the sound items using a best-worst scaling method [110]. The participants were presented with four sound items in each trial and asked to select the most and least pleasant item. This format simplifies the task for the listener, since no absolute rating of the conditions is required. The comparisons within a single trial were limited to one type of timbre and chord, so that no rankings of the pleasantness of chords or timbres relative to each other can be obtained. Each item was presented twice, resulting in three trials per timbre/chord combination. The listening test was completed by eight participants with varying degrees of musical experience. From the participants’ answers, we calculate a relative ranking between the conditions using the `bwsample` Python package [65].

6.3.2.1 Results

The mean $D(\mathcal{P})$ over all chords and timbres is reduced by the proposed methods, and it is also lower compared to the *12-TET* and *Optimized F0* conditions (*Original*: 3.38, *12-TET*: 3.23, *Optimized F0*: 3.16, *Amplitude Reweighting*: 2.28, *Unbeating I & II*: 2.30). While the absolute $D(\mathcal{P})$ naturally depends strongly on timbre and chord (as shown in Section 6.2), all three proposed methods yield similar relative reductions in SD for the different chord and timbre combinations. Note that the reduction is stronger for the timbre modification methods than for the *Optimized F0* approach. This is because modifying individual partials provides additional degrees of freedom that are not available when only adjusting F0.

Regarding spectral envelope, *Amplitude Reweighting* and *Unbeating I* result, on average, in a higher deviation from the spectral centroid compared to the *Original* condition (*12-TET*: 26.4 Hz, *Optimized F0*: 27.0 Hz, *Amplitude Reweighting*: 91.5 Hz, *Unbeating I*: 85.8 Hz, *Unbeating II*: 13.9 Hz). Figure 6.10 shows how *Amplitude Reweighting* and *Unbeating I* yield the same spectral centroid that is constant over the entire duration of the D chord, which is lower than for all other conditions and indicates an effect of the adaptation on the brightness impression of the sound. Retaining the phase modulation in *Unbeating II* leads to a spectral centroid much closer to the *Original* condition, while still reducing fluctuations that are a result of the amplitude modulations introduced by beating between partials.

In the listening test, participants ranked the *12-TET* condition as most pleasant overall, followed by *Unbeating I*, *Optimized F0*, *Unbeating II*, *Amplitude Reweighting*, and *Original* as the least preferred condition. We did not observe large differences in the rankings between different chords/timbres, except that *Optimized F0* was ranked worse for the Am chord. Between listeners, rankings of *12-TET* as the most and *Original* as the least pleasant condition was mostly consistent, while the order of the proposed methods and *Optimized F0* varied. After the test, some participants reported that the *12-TET* condition (particularly with the sawtooth timbre) sounded “harsher” to them than the proposed methods, but this did not influence their pleasantness rating.

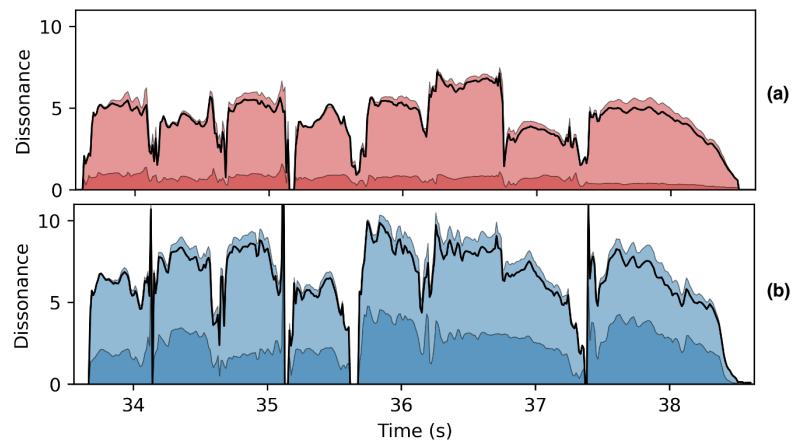
6.3.3 Discussion

In our simple experimental setting, the proposed methods consistently reduce measured SD, while the spectral centroid is least affected by *Unbeating II*. In the listening test, the proposed methods were consistently ranked more pleasant than the *Original* condition. *Amplitude Reweighting* was overall ranked less pleasant than the *Unbeating* methods. This is surprising, since *Unbeating* explicitly introduces inharmonicity into previously harmonic sounds, while *Amplitude Reweighting* only changes the relative weighting of harmonics, and both methods yield a similar reduction in the measured SD. The phase modulation in *Unbeating II* seems to reduce subjective pleasantness compared to *Unbeating I*. However, the varied order in rankings of individual participants hints at relatively small differences between the compared conditions. Somewhat surprisingly, most participants rated the *12-TET* condition as most pleasant, which does not correlate with the measured SD. We suspect an influence of the familiarity of listeners with the 12-TET tuning that influences pleasantness ratings in this context (see, e.g. [109]). Controlled experiments with a larger number of participants, combined with more sophisticated test methods, as for example proposed by Marjeh et al. [113], may help to isolate such effects in future work.

In summary, our results provide early evidence that, in additive synthesis, both modifying the amplitudes and shifting the frequencies of individual partials can increase perceived pleasantness of complex polyphonic chords with detuned F0s. From a practical perspective, this case study also raises the perceptual question of how much timbre can be modified before a tone loses the characteristic quality or natural

Figure 6.11: Illustration of SD reduction with dynamic equalization using the example from Figure 6.3. Colored areas show the relative SD for the respective voice, exactly reproducing the values shown in Figure 6.3d. The black line indicates the relative SD after applying a dynamic equalization with a reduction at 2 kHz between 0 and 6 dB depending on the absolute value of SD.

- (a) *A* voice (clarinet).
 (b) *B* voice (baritone saxophone).



impression of a given instrument (cf. Section 2.2.4). In any case, the observations presented here motivate further investigation of timbre adaptation based on SD as a creative musical tool.

6.4 Further Notes

In this chapter, we have examined two separate but connected research objectives. First, we investigated SD as an informative measure and visualization tool for multi-track music productions. Through a case study with wind instrument quartets, we identified instrument timbre as a main contributing factor to the absolute SD value. Second, we explored how timbre can be adapted in additive synthesis to reduce (or deliberately increase) SD. In this context, we compared methods that modify amplitudes and/or frequencies of individual partials, potentially introducing inharmonicity. We found that both adaptation methods can increase the perceived pleasantness of chords with intonation deviations.

Based on these insights, we now outline three possible applications of timbre adaptation and use these scenarios to frame directions for future work. In this context, we also discuss how differentiable dissonance models can be utilized effectively.

Dynamic Equalization. Equalization, i.e., the frequency-selective attenuation or amplification of a signal through filtering, is commonly used in music production to shape the timbre of recorded sounds. As a first proof of concept for timbre adaptation based on SD, we aim to reduce SD by attenuating higher harmonics in a way that preserves the characteristic sound of the instrument. This timbre adaptation can be made dynamic by attenuating the harmonics only when they contribute strongly to SD, or in other words, when other interfering voices are present in the mix. This method can be seen as similar to side-chain compression, where the level of one voice is controlled by the presence of other signal components. As an example, Figure 6.11 shows the relative SD of the *A* and *B* voices from our running example from Section 6.2. For each voice, the black line indicates the reduction in relative SD achieved

Figure 6.12: Screenshot of *SWARM*. The user interface shows three adaptation modes: *Global Pitch Shift*: F0 adaptation as described in Chapter 5, *Harmonic Pitch Shift*: Using gradients of SD w.r.t. the frequency of individual partials, and *Harmonic Reweighting*: Using gradients of SD w.r.t. the amplitude of individual partials (while also ensuring that overall loudness is preserved). The bottom plot shows all harmonics that are active in the current frame in real time. As an example, the harmonics in the red circle coincide almost exactly in frequency due to the *Harmonic Pitch Shift*.



through dynamic equalization, implemented by attenuating the signal from 0 to 6 dB in the frequency range from approximately 1 to 5 kHz using a single bell filter. In this example, the amount of attenuation is linearly controlled by the current relative SD value. The results confirm that this approach reduces relative SD more strongly in segments with higher overall SD, while leaving less dissonant chords (e.g., the fourth chord after 35 seconds) largely unaffected. This behavior suggests that dynamic equalization could be used to selectively mitigate SD in dense harmonic contexts without unnecessarily altering the timbre elsewhere.

Inference-Time Optimization of Vocal Timbre. Singers deliberately adjust their vocal formants to support voice blending and intonation in choir singing [69]. Inspired by this behavior, one could combine a synthesis system based on a vocal tract model, as introduced in Chapter 4, with a suitable cost measure derived from a differentiable SD model. In this setting, gradients with respect to vocal tract filter parameters could be used to iteratively adapt vocal timbre in order to reduce SD. This idea of directly optimizing synthesis parameters during inference via gradient backpropagation has been introduced as *inference-time optimization* [200]. A central challenge, however, is that the SD model considered here relies on an explicit representation of partials in $\mathcal{P}$. Estimating these partial parameters typically involves peak-picking operations, which are non-differentiable and therefore prevent backpropagation through the full signal-processing chain. In contrast, dissonance models based on harmonicity [66, 126, 202] can be formulated to operate directly on spectral representations, and may offer a more promising alternative in this context.

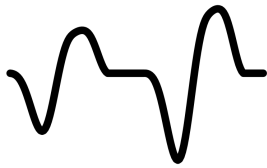
Synthesizer With Awareness for Other Musicians (SWARM). The methods introduced in Chapters 5 and 6 are all suitable for real-time applications, since they rely only on information within the current frame and allow for computationally efficient implementation. In particular, gradients with respect to the cost measures could be calculated for the current frame in a real-time polyphonic synthesizer and then inform the synthesis parameters, such as F0, individual harmonic frequencies, and harmonic amplitudes, in the next frame. To explore this idea, we implemented a prototype system called *SWARM* (*synthesizer*

with awareness for other musicians). The synthesizer takes MIDI input and produces polyphonic audio output through additive synthesis with a sawtooth-based timbre, allowing for multiple tones to be played at the same time, for example through MIDI input from a keyboard. As shown in the user interface in Figure 6.12, *SWARM* provides three adaptation mechanisms. First, *Global Pitch Shift* implements the intonation adaptation algorithm described in Chapter 5, shifting the F0 of each tone on a frame-by-frame basis according to gradients from the differentiable SD model. Second, *Harmonic Pitch Shift* applies the same principle at the level of individual partials, shifting their frequencies independently. Third, *Harmonic Reweighting* uses gradients of SD with respect to amplitude to reduce the contribution of harmonics associated with high dissonance, while compensating the amplitudes of other harmonics to preserve overall loudness. The system is implemented using JUCE³⁸ and can be compiled as a plugin for Digital Audio Workstations (DAWs), allowing it to be used in a real-time context within standard music production workflows. In addition, it includes a side-chain audio input that allows external tracks to be analyzed via peak picking, as described in Section 6.2.1, enabling the synthesized output to adapt in response to other simultaneously played signals. Overall, *SWARM* is a step towards leveraging the nuanced effects of real-time timbre adaptation based on SD in a creative music production context.

³⁸ <https://juce.com/>

7 Wind Instrument Synthesis with Differentiable Pulsetables

This chapter is based on [183]. I was the main contributor to developing the research concept, implementing and conducting the experiments, and preparing the manuscript. Christian Dittmar introduced the use of pulsetables for wind instrument synthesis, implemented a non-differentiable version (outlined in [40]), and contributed to the manuscript. Stefan Balke contributed to the development of ideas and manuscript preparation. Meinard Müller supervised the work and contributed throughout all stages of the research and manuscript preparation.



Pulsetable synthesis is a concatenative audio synthesis method that generates harmonic tones by repeating short waveform prototypes at regular intervals. While related to the more commonly used wavetable method, using pulsetables aligns closely with the physical excitation mechanism of some reed and brass instruments, making it well suited for efficient modeling of those sounds. However, a convincing synthesis

result typically requires extensive parameter tuning. In this chapter, we explore pulsetable synthesis in the context of differentiable digital signal processing (DDSP) and as a method for synthesizing tonal sounds. We introduce a differentiable pulsetable synthesizer in which pulse prototypes are directly optimized via gradient descent, jointly trained with a lightweight neural network to select pulses for a fixed instrument given a target pitch and velocity. Building on this approach, we propose a framework for wind instrument synthesis that provides interpretable control parameters that disentangle, pitch, loudness and timbre of the synthesized tones and can be trained in an unsupervised way using only a few minutes of recordings from a target instrument. Initial experiments with resynthesis and timbre transfer demonstrate that our approach enables expressive synthesis with only 60 000 trainable parameters, offering an efficient alternative to other differentiable instrument synthesis methods.

This chapter is organized as follows. In Section 7.1, we discuss the conceptual background and related work. Section 7.2 gives a comprehensive overview of pulsetable synthesis in comparison to the wavetable method. Section 7.3 then introduces a wind instrument synthesis system based on pulsetable synthesis and examines how different DDSP components and lightweight neural networks can be combined to disentangle control parameters for pitch, loudness, and timbre. Finally, Section 7.4 presents a preliminary evaluation, followed in Section 7.5 by a discussion of potential extensions and future directions.

7.1 Background

As previously discussed, advances in neural audio synthesis (e.g., [44, 112]) allow for generating high-quality instrument performances, but current models are often computationally expensive, require extensive training data, and lack controllability. One way to address these issues is to leverage prior knowledge about the audio synthesis process using the DDSP paradigm [42, 72]. Incorporating DDSP building blocks such as oscillators and filters into a machine learning pipeline introduces an inductive bias that can aid in learning efficient representations, reduces demands on training data, and provides interpretable parameters for analysis and modification. Several synthesis methods have already been explored in the DDSP context, including a combination of additive and subtractive synthesis [42], modal physical models [11, 104], Frequency Modulation (FM) [21], and wavetable synthesis [191].

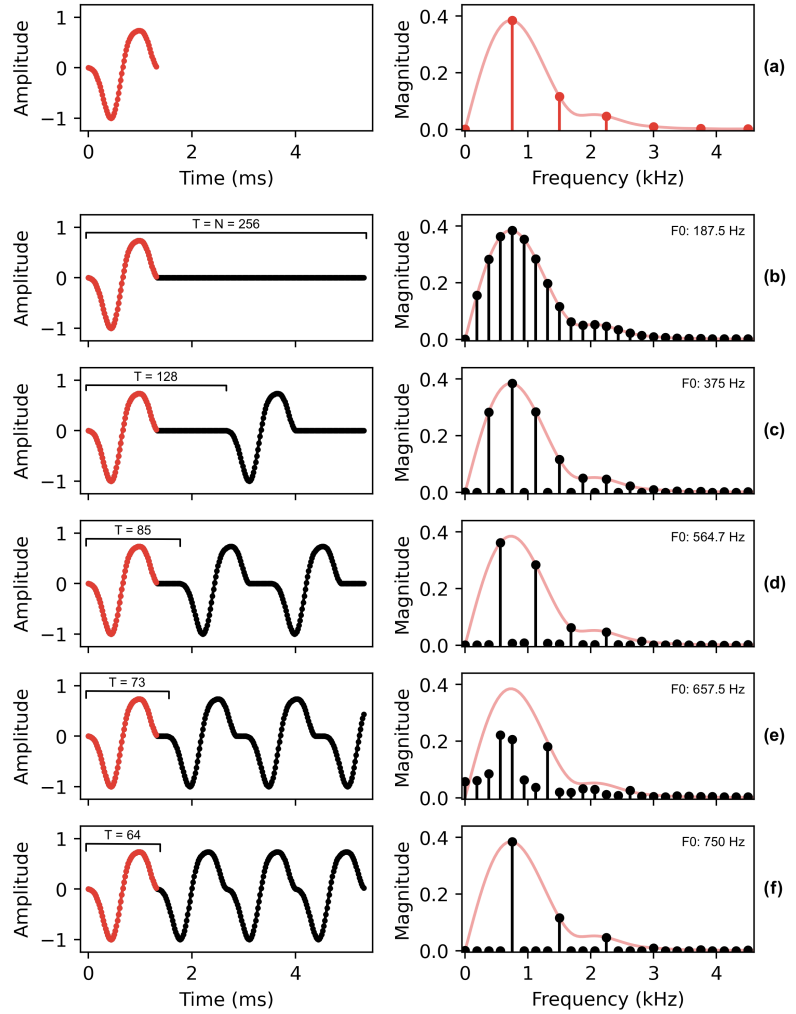
In this chapter, we introduce a differentiable variant of pulsetable synthesis. This method, also known as *pulsar* or *formant wave* synthesis [161, 162], is based on the concatenation of short prototype waveforms in regular intervals to create harmonic tones. In contrast to wavetable synthesis, where these waveforms are resampled to match the target period of the tone, the pulses are kept at their original length and padded with zeros. To a certain degree, this resembles the physical mechanism by which some wind instruments, such as the trumpet or oboe, generate sound, periodically producing impulses at the mouthpiece through opening and closing of the lips or reed. These impulses have nearly constant width and vary primarily in repetition rate according to the F_0 of the tone [52, 137]. With this, pulsetables have been proposed as a suitable alternative for efficiently synthesizing wind instrument sounds [40], contrasting earlier wavetable-based approaches [38, 78]. We will discuss the properties of pulsetable synthesis and the commonalities and differences to the more commonly used wavetable method in Section 7.2.

Our differentiable approach addresses several issues of pulsetable synthesis. First, by making both the pulses and their selection learnable (Section 7.2.1), we eliminate the need for tedious manual pulse extraction, which typically requires anechoic recordings of the target instrument across multiple registers and intensities. Second, through introducing a wind instrument synthesis framework based on pulsetables (Section 7.3), we enable an efficient generation of wind instrument sounds without manual parameter tuning. Our framework uses very lightweight neural network building blocks and can be trained in an unsupervised way, while disentangling the interpretable control parameters of pitch and velocity from the instrument’s timbre. In addition to experimentally showcasing the strength and weaknesses of the pulsetable method in comparison to wavetables, we present preliminary results on using our model for controllable timbre transfer between instruments (Section 7.4). Code and audio examples can be found on a supplemental website.³⁹

³⁹ <https://audiolabs-erlangen.de/resources/MIR/2026-ICASSP-DiffPulse>

Figure 7.1: Illustration of the pulsetable method using a single learned trumpet pulse with $L = 64$ (shown in red) and concatenations to length $N = 256$ with varying period T (shown in black). Each plot shows time-domain signals (left) and the corresponding frequency-domain spectra (right). All signals use a sampling rate of $F_s = 48\,000$. The red line in the frequency-domain plots shows the spectral envelope of the pulse, obtained via the DTFT. For (b–f), the concatenated signals ($N = 256$) and their discrete spectra are shown in black.

- (a) Showing only a single pulse with $L = 64$ and its spectrum.
- (b) Concatenated with $T = 256$ samples ($f = 187.5$ Hz).
- (c) $T = 128$ samples ($f = 375$ Hz).
- (d) $T = 85$ samples ($f \approx 564.7$ Hz).
- (e) $T = 73$ samples ($f \approx 657.5$ Hz).
- (f) $T = 64$ samples ($f = 750$ Hz).

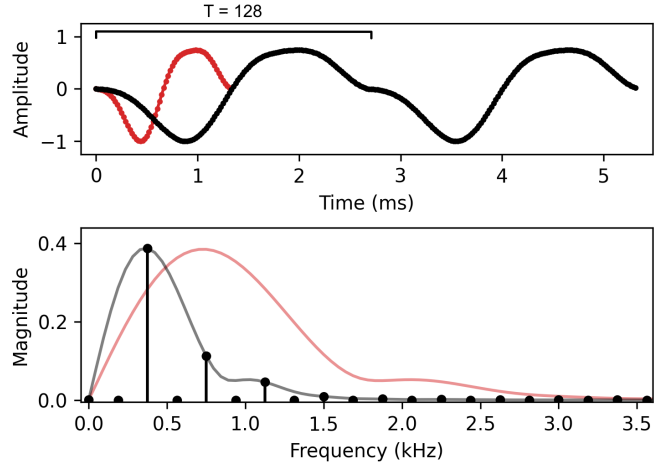


7.2 Pulsetable and Wavetable Synthesis

We have seen in Chapter 2 that an essential property of a harmonic tone is the periodicity of its waveform. This observation motivates a simple synthesis principle in which a tone with fundamental frequency f in Hz is synthesized by periodically repeating a short time-domain signal or *pulse* $p \in \mathbb{R}^L$ of length L samples, with a period of $T = F_s/f$ samples. If the pulses are simply concatenated, matching the period with zero-padding (requiring $T \geq L$), we call this method *pulsetable synthesis*.

Figure 7.1 illustrates the resulting waveform and its corresponding magnitude spectrum for a pulse p that is repeated with different periods to form a tonal signal of overall length $N = 256$. The waveform and spectrum of the pulse itself with length $L = 64$ (shown in red) are shown in Figure 7.1a. Using the Discrete-Time Fourier Transform (DTFT), we can calculate a *spectral envelope* (indicated by a red line in all frequency-domain visualizations), which effectively approximates the Fourier transform of a continuous pulse signal. In the simplest case, with $T = N = 256$ samples (Figure 7.1b), the resulting

Figure 7.2: Illustration of the wavetable method and its properties, showing time-domain signals (top) and the corresponding frequency-domain spectra (bottom). The same single period waveform as in Figure 7.1 is used, but it is time-stretched by resampling to the desired period of $T = 128$ samples. This way, the entire spectral envelope is compressed in frequency.

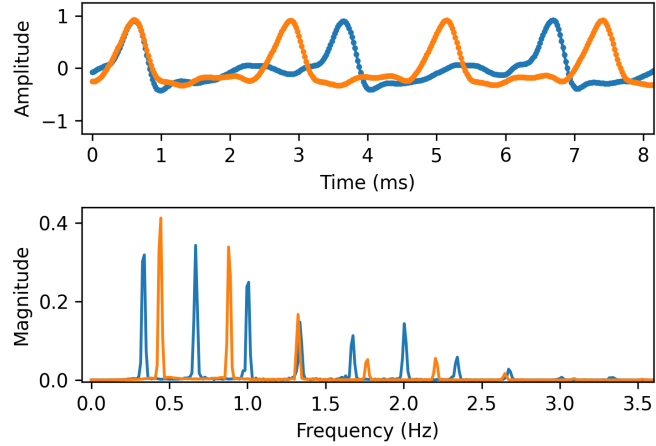


signal consists of a single zero-padded pulse, and its spectrum follows the spectral envelope of p exactly. Although defining an F_0 is not meaningful for such a short signal containing only one period, a period of $T = 256$ would correspond to a frequency of $f = 187.5$ Hz. Reducing the period to $T = 128$ (Figure 7.1c), corresponding to $f = 375$ Hz, yields a harmonic spectrum whose amplitudes continue to follow the same spectral envelope. The same behavior is observed for $T = 85$ (Figure 7.1d), corresponding to $f \approx 564.7$ Hz, which is close to three periods within the considered signal excerpt. This illustrates how the timbre of the synthesized tone is governed by the shape of the pulse p , as its spectral envelope determines the magnitudes of the harmonics in the resulting signal.

For illustrative purposes, the period lengths in the examples above were chosen such that the resulting F_0 is (approximately) an integer multiple of the DFT bin spacing, $F_s/N = 48\,000/256 = 187.5$ Hz, or equivalently, the corresponding period T evenly divides the overall signal length N . In this case, the spectrum exhibits exact zeros between harmonic components, a property we show formally in Appendix C. Importantly, the underlying principle of pulsetable synthesis is not limited to such special cases. For arbitrary period lengths T , the harmonic frequencies generally fall between DFT bins, resulting in spectral leakage (cf. Chapter 3). As an example, Figure 7.1e, shows a synthesized signal with period $T = 73$ samples, corresponding to $f \approx 657.5$ Hz. The energy of each individual harmonic is now distributed across multiple frequency bins due to spectral leakage, resulting in a less clear structure of the discrete spectrum. Nevertheless, the underlying harmonic amplitudes, as they would be revealed by a DTFT, continue to follow the spectral envelope imposed by the pulse p .

Furthermore, since generally $f \in \mathbb{R}$ (i.e., we want to synthesize a continuous F_0), T is not necessarily an integer period either. In this case, successive pulse repetitions are offset by a fractional number of samples, and interpolation between the samples of p is required. For instance, if $T = 85.5$, the second repetition of the pulse starts at a half-sample offset relative to the first, which can be realized by appropriately interpolating the pulse waveform. While linear interpolation is often sufficient to achieve reasonable synthesis quality and is therefore used in this work, more sophisticated interpolation methods may provide higher quality at the cost of increased computational complexity [103, 196].

Figure 7.3: Time-domain signals (top) and the corresponding frequency-domain spectra (bottom) of two recorded trumpet tones from [8] with $f \approx 330$ Hz (blue) and $f \approx 440$ Hz (orange). Even though their periods differ, the waveform shapes are similar for the two tones.



For the similar and more commonly used *wavetable synthesis*, the pulse p is not zero-padded but resampled to match the desired period length T . As a result, the waveform is stretched or compressed in time, resulting in the entire spectrum being compressed or stretched in frequency, as illustrated in Figure 7.2. In comparison with Figure 7.1c, both tones have the same F0 but a fundamentally different timbre, since wavetable synthesis preserves the *relative* amplitude between harmonics instead of prescribing a fixed spectral envelope.

Consequently, pulsetable synthesis is particularly well suited for synthesizing tones whose timbre is shaped by a frequency-selective attenuation or amplification that remains independent of the F0, for example, due to resonances in an instrument body. Wavetable synthesis, on the other hand, allows for efficiently synthesizing sounds where the spectral envelope changes depending on the pitch, such as a sawtooth wave, where the amplitude ratio between harmonics is constant. The recorded trumpet tone excerpts in Figure 7.3 demonstrate that the characteristic waveforms of the trumpet, for example, align more closely with the pulsetable method. Even though the two tones have different F0s ($f \approx 330$ Hz for the blue and $f \approx 440$ Hz for the orange signal), their pulse shapes remain similar. This is a direct result of the physical mechanism of sound generation in a trumpet [52].

From a technical perspective, both pulsetable and wavetable synthesis impose an upper limit on the fundamental frequency f , or equivalently, a lower limit on the period T . For pulsetables, if $T < L$, the pulse p is not read-out completely, but cut off by the start of the next pulse, which may lead to discontinuities in the resulting waveform. For wavetables, if the waveform is compressed, the spectrum is shifted towards higher frequencies, potentially resulting in aliasing if the highest components end up above the Nyquist frequency $F_s/2$. In our example from Figure 7.1, the pulse length is $L = 64$ samples, so that the lowest possible period for pulsetable synthesis is $T = 64$ samples ($f = 750$ Hz), whereas for wavetable synthesis (Figure 7.2), assuming a highest frequency component of 12 kHz in the pulse, the lowest possible period is $T = 32$ samples ($f = 1500$ Hz).

To account for time-varying timbre in both pulsetable and wavetable synthesis, it is common to tabulate M different pulses p_m as the rows of a matrix $P \in \mathbb{R}^{M \times L}$, hence the term *table* in the method’s name. Using a one-hot weighting vector $w_i \in [0, 1]^M$ for every time frame i , we can now select a pulse with $p_i = P^T w_i$. Depending on the specific application, this selection could be based on different parameters. For example, since wind instruments commonly exhibit a brighter timbre at higher loudness [52, 78, 93], different pulses could be selected based on the target amplitude. Alternatively, a selection based on F0 could account for differences of the timbre characteristics in different instrument registers. As a third option, timbre could also be varied independently of pitch and loudness, similar to the control exhibited by skilled musicians. In the context of pulse selection, this would require an additional control parameter for timbre (see, e.g., [236], where a wavetable for singing voice synthesis is parameterized by a “vocal effort” parameter).

Notably, if the individual pulses have dissimilar waveforms, further interpolation between pulses may be necessary to avoid discontinuities when the selected pulse changes between consecutive time frames. Finally, it is common to store the pulses independent of the output signal’s amplitude. If we then define a frame-wise gain parameter g_i and make the fundamental frequency f_i time-varying as well, a complete pulsetable or wavetable synthesizer as shown in the center of Figure 7.4 can be controlled by three time-varying parameters f_i , g_i , and w_i .

7.2.1 Learnable Pulsetable and Selection

Manually designing a wavetable or pulsetable synthesizer is often a labor-intensive process. For example, the wind instrument synthesis model proposed in [40] requires extracting pulses from an anechoic reference recording across different instrument registers, ensuring that the pulses are representative of the entire intended pitch and velocity range. For wavetable synthesis, this challenge motivated approaches that either learn the wavetable and selection jointly [191] or create a model-based fixed wavetable and learn to estimate the selection parameter from data [236], leveraging the DDSP paradigm of backpropagating a loss through the synthesis pipeline.

Analogously, we propose to learn the pulsetable P and the selection of pulses from recordings. As in [191], we optimize P directly via gradient descent using a reconstruction loss that compares the output of the pulsetable synthesis to a recording of a reference instrument (see Section 7.3 for details). We initialize all pulse prototypes in P with the same time-domain waveform, using a pulse that is designed in the frequency domain with quadratically decaying magnitude towards higher frequencies. In comparison to random initialization, this led to faster convergence in our experiments.

For the selection, we define a function $\mathcal{W} : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]^M$ that takes a known F0 f and gain g (as a proxy for the tone’s pitch and loudness, respectively) as input and outputs a weighting vector w for the pulse selection. $\mathcal{W}$ can then be applied for each time frame i , deriving the time-varying $w_i = \mathcal{W}(f_i, g_i)$ from the remaining two parameters of our entire synthesis system, f_i and g_i , further simplifying the set of

control parameters. f_i and g_i could either be learned separately, estimated with model-based algorithms, or synthetically generated, e.g., from MIDI. While the mapping from f_i and g_i to w_i is not necessarily unique or linear, we can use a lightweight neural network that can be optimized jointly with P (see Section 7.3 for details). Given the reduced control over the timbre of individual tones in the synthesis, we lose the ability to perfectly reconstruct an input signal, since the timbre variations in a real musical performance are not determined only by pitch and velocity. However, with this simple and interpretable control parameterization, our approach provides flexibility in how the learned pulsetable can be used, e.g., for altering pitch, loudness, or timbre of the synthesized sounds.

For a differentiable formulation of the pulse selection, we have to solve two separate problems. First, a one-hot weighting w_i , i.e., a discrete decision for a certain pulse in P , is not differentiable. Following a common relaxation, we approximate the one-hot encoding of w_i with a softmax as in [191], given by

$$\sigma(x)_j = \frac{\exp(x_j)}{\sum_{m=1}^M \exp(x_m)} \quad (7.1)$$

for $x \in \mathbb{R}^M$. To further encourage the selection of a single pulse per frame, we apply a *temperature-scaled* softmax which progressively increases the selectivity of the relaxed pulse weights by setting

$$w_i = \sigma\left(\frac{\tilde{w}_i}{\tau}\right), \quad (7.2)$$

where $\tilde{w}_i \in \mathbb{R}^M$ denotes the frame-wise unscaled logit outputs of the neural network, and τ is a temperature parameter. With $\tau \rightarrow 0$, the output of the temperature-scaled softmax approaches a one-hot vector and we linearly anneal τ from 1 to 0.01 over the first 30 training epochs. As a result, the final entries in P remain more interpretable, directly representing the pulses used for synthesis.

Second, jointly optimizing the set of pulses P and the selection network $\mathcal{W}$ poses a problem similar to the “codebook collapse” in discrete representation learning [138]. If the model converges early to selecting only a small subset or even a single pulse, gradient updates never reach the remaining pulses, making their subsequent optimization and usage increasingly unlikely. To avoid this, we introduce a regularization term that maximizes the entropy of the average output of $\mathcal{W}$ across all time frames and items in a training batch. While the temperature-scaled softmax (Equation 7.2) promotes the selection of a single pulse within each time frame, this regularization term encourages diversity across time frames and batch items. With a sufficiently large batch size and number of time frames, these objectives are compatible: individual weight vectors w_i can be highly selective (i.e., low entropy), whereas their average over time and batch remains close to uniform (i.e., high entropy). Together, these mechanisms guide $\mathcal{W}$ toward selecting a single pulse per time frame while maintaining a diverse usage of pulses across the batch.

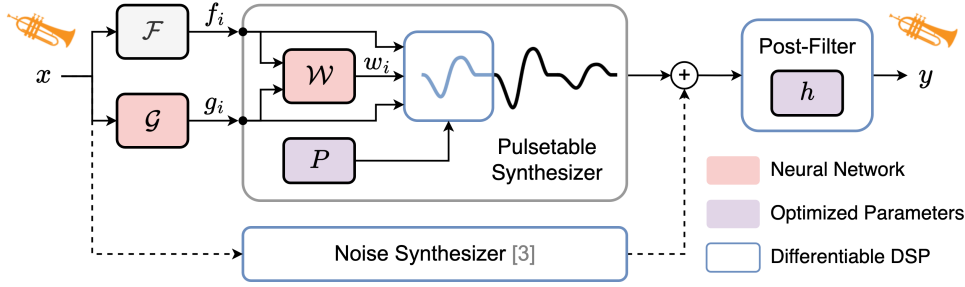


Figure 7.4: Overview of the wind instrument synthesis framework, with the fixed F0 estimator $\mathcal{F}$, trained gain estimator $\mathcal{G}$, the actual synthesizer with pulsetable P and a jointly trained table weight estimator $\mathcal{W}$, as well as the FIR post-filter with coefficients h . A filtered noise synthesizer as in [42] adds non-harmonic signal components.

7.3 Wind Instrument Synthesis

Our wind instrument synthesis framework aims to deliver natural-sounding audio with a small computational footprint and minimal training data. The full model has about 60 000 trainable parameters and can be trained from scratch using only a few minutes of recordings from a single target instrument (e.g., a trumpet recorded in a certain acoustic environment). Furthermore, we design the framework to provide an interpretable intermediate representation based on the time-varying control parameters f_i , g_i , and w_i . All components can be trained jointly in an unsupervised way, relying only on a reconstruction loss between input signal x and output signal y in an autoencoder setup. This enables applications such as instrument timbre transfer (by replacing P , $\mathcal{W}$, and h with one trained on a different instrument), or performance synthesis from symbolic data (e.g., by generating artificial F0 and gain control parameters).

An overview of the framework is shown in Figure 7.4. In addition to the learnable pulsetable synthesizer with selection network $\mathcal{W}$ as described in Section 7.2, we include the following components.

An **F0 estimator** $\mathcal{F}$ provides f_i and is not trained jointly with the other components due to the known training instabilities related to frequency-related parameters in DDSP [214]. Both model-based (such as [19]) or pre-trained estimators (e.g., [160]) can be used interchangeably here. When available, we use the F0 annotations from [8] directly in our experiments.

A **gain estimator** $\mathcal{G}$ predicts the time-varying gain control parameter $g_i \in \mathbb{R}_+$ for the pulsetable synthesis. While heuristics such as a smoothed RMS envelope [40] could be used instead, learning $\mathcal{G}$ jointly with the other components allows for focusing specifically on the envelope of the tonal component in x and exploiting the specific characteristics of pulsetable synthesis to minimize the reconstruction loss (see also Chapter 2). We implement $\mathcal{G}$ as a recurrent neural network with 15 000 trainable parameters. For details on the model architectures, see Section 7.3.1.

A **noise synthesizer** models non-harmonic signal components such as noisy note onsets or breathing, which cannot be generated with pulsetable synthesis (cf. Chapter 2). We use the same filtered noise

method as in [42], but with a smaller recurrent neural network with 40 000 trainable parameters to estimate noise magnitudes for each frame.

An **FIR post-filter** with coefficients $h \in \mathbb{R}^{4096}$ is used to capture time-invariant acoustic characteristics of the target instrument, such as instrument radiation, room acoustics, and the recording setup. While the filter length of 4096 samples (approx. 85 ms) is not sufficient to represent a full room impulse response, it effectively models global coloration effects not present in the pulses in P . Instead of using a neural network, we also optimize the FIR coefficients directly in time domain via gradient descent, encouraging sparsity with L1 regularization of the coefficients.

The general approach presented here is not limited to the use of pulsetable synthesis for modeling the tonal components of the output signal. For example, in the case of the wavetable method, the only modification is the table readout, where pulses are time-stretched rather than zero-padded to realize the desired period T . The strength of both the pulsetable and wavetable approaches lies in their efficient and interpretable representation of the target instrument’s timbre. Depending on the intended application, however, alternative synthesis methods such as additive synthesis could be employed as drop-in replacements, provided that the timbral characteristics can be derived from the parameters f_i and g_i . We will explore this possibility further in Section 7.4.3.

Furthermore, note that the pulsetable and wavetable method is inherently limited to generating harmonic tonal components, so that inharmonic tonal components (e.g., at onsets, see the trumpet examples in Section 7.4) are not modeled explicitly with this system. Potential extensions addressing this limitation are outlined in [40] and discussed in Section 7.5.

7.3.1 Training Details

We use a sampling rate $F_s = 48\,000$ Hz throughout all experiments, enabling a comparison of our results with high-quality reference recordings. The parameters f_i , g_i , and w_i are updated with a frame rate of 128 samples (2.67 ms) and linearly interpolated to sample accuracy to avoid discontinuities between frames, as described in Chapter 2.4.1. The pulsetable is configured with a pulse length $L = 64$ samples, which at $F_s = 48$ kHz corresponds to a maximum possible F0 of $F_s/L = 750$ Hz without cutting off parts of the pulse as described in Section 7.2. While this pulse duration of 1.2 ms aligns with the measured excitation length of double-reed instruments in their middle register [137], shorter pulses would allow for higher fundamental frequencies, whereas longer pulses would capture a more detailed spectral envelope at lower F0s. Consequently, learning pulses of varying lengths could improve resynthesis quality, which we leave for future work.

For the reconstruction loss, we use the multi-scale spectral (MSS) loss in the *smooth MSS* configuration (see Chapter 3), but with window sizes of (4096, 2048, 1024, 512, 64). This choice of window sizes is motivated by two considerations. First, since the optimization does not rely explicitly on gradients with

respect to frequency, we can prioritize the improved computational efficiency of power-of-two transform sizes over the use of co-prime sizes. Second, by omitting the window sizes 256 and 128 and focusing on larger windows, we increase the frequency selectivity of the loss, which discourages the noise synthesizer from reproducing harmonic signal components. All models are trained for 40 epochs with the same amount of training data (see Section 7.4.1).

Gain Estimator $\mathcal{G}$: The gain estimator $\mathcal{G}$ receives a mel-spectrogram representation (following, e.g., [42] for comparability) of the input signal and outputs a gain value g_i for each time frame, which serves as the envelope for the harmonic component of the signal. In this way, the network learns not only to estimate the overall signal envelope but also to disregard noisy or transient components. This behavior distinguishes the learned $\mathcal{G}$ from simpler signal-processing approaches, such as computing an RMS envelope as described in Section 2.2.3. The network has a total of 14 807 trainable parameters. It is composed of two 2D convolutional layers followed by a bi-directional LSTM, whose output is finally mapped to a scalar g_i via a fully-connected layer and a scaled sigmoid nonlinearity. While a minimum output value of 0 is required to represent unvoiced frames, the upper bound of g_i is less critical, as it can be compensated for by scaling the pulses in P . For our experiments, we set $g_i \in [0, 2]$, which empirically yields pulse amplitudes within the range of ± 1 .

Pulse Selection $\mathcal{W}$: To use f_i and g_i as inputs to the neural network $\mathcal{W}$, it is generally advisable to scale values to a controlled range, such as $[-1, 1]$, contributing to well-behaved gradients and improved numerical stability. Instead of applying an explicit scaling to these scalar inputs, we encode f_i and g_i as *binned distributions*, denoted by $\hat{f}_i \in [0, 1]^B$ and $\hat{g}_i \in [0, 1]^B$, where B is the number of bins. This representation maps scalar frequency and gain values to *pseudo-probability* vectors over their respective domains, a strategy commonly used to recast regression tasks as classification problems [48, 201]. As an input representation, it has the advantage that scaling is implicit: when properly normalized, the total weight of each input vector is fixed to one, ensuring a consistent value range. Moreover, this formulation allows the pseudo-probability vector to be deliberately “blurred”, which can improve robustness and generalization during training. Following [203], we apply Gaussian blurring to the binned representations by calculating the j th element in $\hat{f}_i$ for $j \in [1 : B]$ with

$$(\hat{f}_i)_j = \frac{1}{C} \exp \left(\frac{-(jR - 1200 \log_2(f_i/f_{\min}))^2}{2\sigma_f^2} \right), \quad (7.3)$$

where R is the width of each bin in cents, $f_{\min}$ is the smallest represented frequency, and C is a constant chosen to normalize $\hat{f}_i$ to sum up to one. σ_f (in cents) can be used to blur the representation. For $\hat{f}_i$, we set $B = 8$, $R = 569$ cents, $f_{\min} = 150$ Hz, and $\sigma_f = 300$ cents, resulting in a covered frequency range of 150–1500 Hz. Note that the goal is to choose an appropriate pulse (and therefore instrument timbre) based on the broad register of the tone, not the exact F0, so that this coarse resolution is sufficient. We calculate $\hat{g}_i$ analogously, with bins on a logarithmic scale covering a range from 0.01 to 1, i.e., 40 dB. $\hat{f}_i$ and $\hat{g}_i$ are concatenated and provided as input to a two-layer fully-connected network with a total of 1012

trainable parameters. As mentioned in Section 7.2.1, the parameter τ in the temperature-scaled softmax is gradually decreased from 1 to 0.01 over the course of 30 epochs.

Noise Synthesis: The noise synthesizer is based on the subtractive synthesis introduced by Engel et al. [42], using a white noise generator and a subsequent FIR filter with 256 coefficients, whose 129 linearly-spaced magnitudes are the output of the network. We apply the resulting linear-phase filter by frame-wise multiplication in the frequency domain, using the overlap-add method to ensure a continuous filtering of the time-domain signal. The noise magnitude estimator has a total of 39 315 trainable parameters and follows the same architecture as the gain estimator, using more convolution filters to account for the higher task complexity.

In total, this setup results in 59 486 trainable parameters, also including the 4096 post-filter coefficients and 256 pulsetable samples. All parameter counts above are calculated for the models with $M = 4$ (see Section 7.4.3 for a performance comparison of models with different M).

7.4 Experiments

We evaluate the proposed wind instrument synthesis system in a resynthesis setting, i.e., using the model as an autoencoder in which parameters are estimated from an input signal so that the output signal of the synthesizer approximates the input. Section 7.4.1 describes the training and evaluation data used in our experiments. In Section 7.4.2, we present a qualitative analysis by comparing the results of pulsetable and wavetable synthesis for trumpet and clarinet. Quantitative results, including more instruments and a comparison with an additive synthesis baseline, are reported in Section 7.4.3.

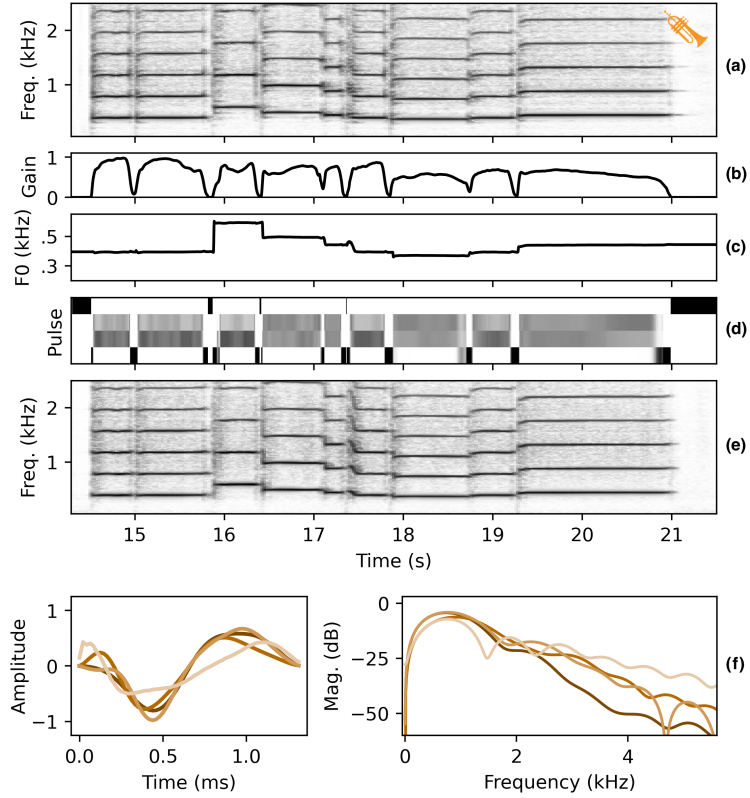
Note that the presented analysis should be viewed as preliminary. Instead of providing evidence for the superiority of one method over another, the goal is to assess the feasibility of the proposed approach and to identify general trends in the comparisons. A more comprehensive evaluation should particularly include subjective listening tests, since objective metrics can capture perceptually relevant differences between the synthesis results only to a limited extent.

7.4.1 Dataset

For our experiments, we use the ChoraleBricks dataset [8], which contains performances of ten four-part chorales (S: soprano, A: alto, T: tenor, B: bass). Each voice is recorded in isolation with a variety of different wind instruments guided by a conducting video, resulting in a diverse set of loosely synchronized monophonic recordings. We focus on four instruments representing distinct timbral and physical characteristics: the brass instruments trumpet (tp) and baritone (bar), and the woodwinds clarinet (cl) and oboe (ob). With its relatively homogeneous structure and playing style, the ChoraleBricks dataset is well suited for systematic analysis. While more diverse datasets with greater variability in pitch, loudness,

Figure 7.5: Synthesis results from the model using **pulsetables** with $M = 4$ for trumpet (**tp**) and the *same voice* (SV) test signal.

- (a) Spectrogram of the original signal.
- (b) Gain estimates g_i from $\mathcal{G}$.
- (c) F0 trajectory (from the dataset).
- (d) Pulse weight estimates w_i from $\mathcal{W}$.
- (e) Spectrogram of the resynthesized output signal.
- (f) Visualization of learned pulses in time domain (left) and frequency domain (right). Color encodes the pulse number (from light to dark).



and timbre would be required to evaluate our system in more challenging resynthesis scenarios, such settings are left for future work.

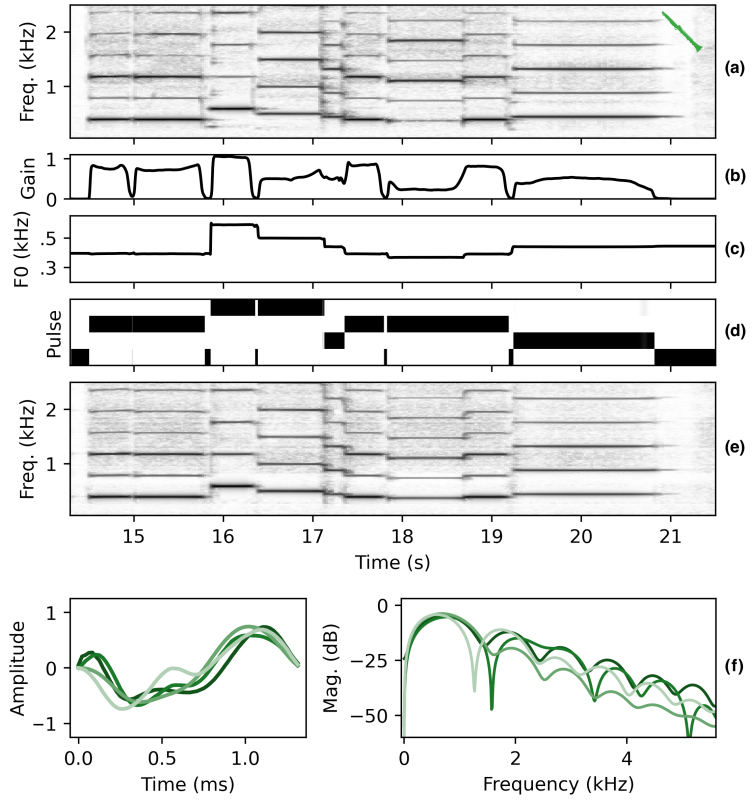
For training, we use recordings of a single voice from eight chorales (BA1, DR1, GE1, GE2, JA1, TE1, VU1, VU2). We select the S voice for **tp**, **c1**, and **ob**, and the T voice for **bar**, resulting in a total of 4 minutes and 18 seconds of material for each instrument. From each chorale, we randomly extract 500 potentially overlapping excerpts of 4 seconds (1500 frames), yielding 4000 training items per instrument. For evaluation, we consider two conditions. In the *same voice* (SV) condition, we use recordings of the same voice that was seen during training, but taken from a different, previously unseen chorale (AN1). In the *different voice* (DV) condition, we instead use recordings from a different voice of an unseen chorale (CR1). Specifically, we use voice A for the instruments **tp** and **c1** while the models are trained on S, and voice B for **bar** while the models are trained on T (no DV recording is available for **ob**). The DV condition allows us to assess how well the method generalizes to unseen timbres and pitch ranges, as these recordings contain notes at substantially lower pitches than those present in the training data.

7.4.2 Qualitative Comparison

The goal of the following evaluation is to assess the quality of the proposed model's resynthesis under controlled conditions. Based on the dataset described above, we train separate models for trumpet and

Figure 7.6: Synthesis results from the model using **pulsetables** with $M = 4$ for clarinet (c1) and the *same voice* (SV) test signal.

- (a) Spectrogram of the original signal.
- (b) Gain estimates g_i from $\mathcal{G}$.
- (c) F0 trajectory (from the dataset).
- (d) Pulse weight estimates w_i from $\mathcal{W}$.
- (e) Spectrogram of the resynthesized output signal.
- (f) Visualization of learned pulses in time domain (left) and frequency domain (right). Color encodes the pulse number (from light to dark).



clarinet using the corresponding training data, jointly optimizing $\mathcal{G}$, $\mathcal{W}$, P , h , and the noise synthesizer. Unless otherwise noted, we use the pulsetable method with $L = 64$ and $M = 4$. As we will show in the following, this relatively small number of pulses is generally sufficient to achieve competitive results with the ChoraleBricks dataset while highlighting the strength and weaknesses of the compared models.

Let us first consider the trumpet model and the SV condition. Figure 7.5 shows an excerpt of the resynthesis result. We can observe that the spectrum of the resynthesized signal (Figure 7.5e) closely resembles that of the original signal (Figure 7.5a). The gain trajectory g_i (Figure 7.5b) predominantly tracks the tonal components of the signal while largely disregarding noisy signal components, which is particularly evident between tones where g_i approaches zero despite the presence of noise. In addition, the model utilizes different pulses over the course of the excerpt with the pulse selection w_i (Figure 7.5d), while predominantly selecting a single pulse within each frame, as intended and encouraged by the regularization terms described in Section 7.2.1. Figure 7.5f shows the time-domain signal of the four learned pulses (left) and their respective spectra (right), showing that the pulses mainly differ in their attenuation of higher frequencies, i.e., the selection allows for controlling the brightness of the resulting tone. However, at least in the excerpt shown, the brightest pulse (corresponding to the lightest curve in Figure 7.5f and the first pulse in Figure 7.5d) is selected almost exclusively in unvoiced passages where g_i is close to 0. In contrast, the least bright pulse (the darkest curve in Figure 7.5f and the last pulse in Figure 7.5d) is predominantly used toward the end of individual tones. This behavior is consistent with the observation in Chapter 6

that timbre tends to become darker toward the end of tones. A prominent difference between the original recording and the resynthesized signal lies in the tone onsets of the trumpet, which are characterized by clearly visible inharmonic components. As this can neither be modeled by the pulsetable synthesis nor by the noise component, these characteristics are absent from the resynthesized signal.

Figure 7.6 shows the same excerpt from the SV test signal, but now played on a clarinet and resynthesized using the clarinet model. The harmonic energy distribution of the recorded clarinet tones is more complex in comparison to the trumpet. In particular, the even harmonics are consistently attenuated, as already discussed in Section 2.2.4. Furthermore, the strength of individual harmonics varies more noticeably across tones. For example, the tone starting around 18 seconds exhibits a weaker fundamental than the surrounding tones. While the corresponding estimate of g_i is lower for this tone, the pulse selection w_i remains unchanged, indicating that the model does not learn to represent this effect with a separate pulse. Another notable difference is that the spectra of the learned pulses are less smooth than those of the trumpet model and exhibit clear local minima, which account for the attenuation of even harmonics.

We can investigate how well the model generalizes to an unseen pitch range by resynthesizing the DV test signal. For the clarinet, an excerpt of this resynthesis is shown in Figure 7.7. The results clearly indicate that the pulsetable model is unable to reproduce the timbre of the original recording in this setting. For instance, in the recorded signal (Figure 7.7a), the second harmonic is consistently attenuated across all tones, whereas in the resynthesized signal (Figure 7.7e), it is instead the third harmonic that is attenuated. This mismatch is a direct consequence of the pulsetable synthesis method. As discussed in Section 7.2, the spectral envelope of a pulse remains fixed in frequency when synthesizing tones with different periods T . As a result, when lower-pitched tones are synthesized using the learned pulses shown in Figure 7.7f, the local minima of the pulse spectrum no longer align with the even harmonics. This leads to unintended attenuation of harmonics that are not suppressed in the original signal. Consequently, in this particular case of a clear relative attenuation of certain harmonics, wavetable synthesis appears to be more suitable for generalizing across pitch ranges. When a waveform is stretched to a longer period, its spectrum is correspondingly compressed, causing the local minima to remain aligned with the same integer multiples of the F_0 . This effect can be observed by training an alternative clarinet model with identical settings, except for replacing the pulsetable method with wavetable synthesis. Figure 7.8 shows the DV resynthesis result for this model, where the clarinet timbre is reproduced more faithfully. Moreover, the spectra of the learned pulses in Figure 7.8f now align more regularly with the even integer multiples of the pulse F_0 (recall $L/F_s = 64/48,000 = 750$ Hz).

For a final comparison, we can repeat this DV experiment and pulsetable–wavetable comparison for the trumpet. Figure 7.9 shows the DV resynthesis result for the pulsetable model, whereas in Figure 7.10, the wavetable approach was used. The trumpet DV test signal is characterized by a resonance around 1.2 kHz, which leads to increased harmonic energy in this frequency region largely independent of the current F_0 . This characteristic is more naturally reproduced by the pulsetable approach, as illustrated by the learned pulses shown in Figure 7.9f, which exhibit smoother spectra with a maximum around this

Figure 7.7: Synthesis results from the model using **pulsetables** with $M = 4$ for clarinet (c1) and the *different voice* (DV) test signal.

- (a) Spectrogram of the original signal.
- (b) Gain estimates g_i from $\mathcal{G}$.
- (c) F0 trajectory (from the dataset).
- (d) Pulse weight estimates w_i from $\mathcal{W}$.
- (e) Spectrogram of the resynthesized output signal.
- (f) Visualization of learned pulses in time domain (left) and frequency domain (right). Color encodes the pulse number (from light to dark).

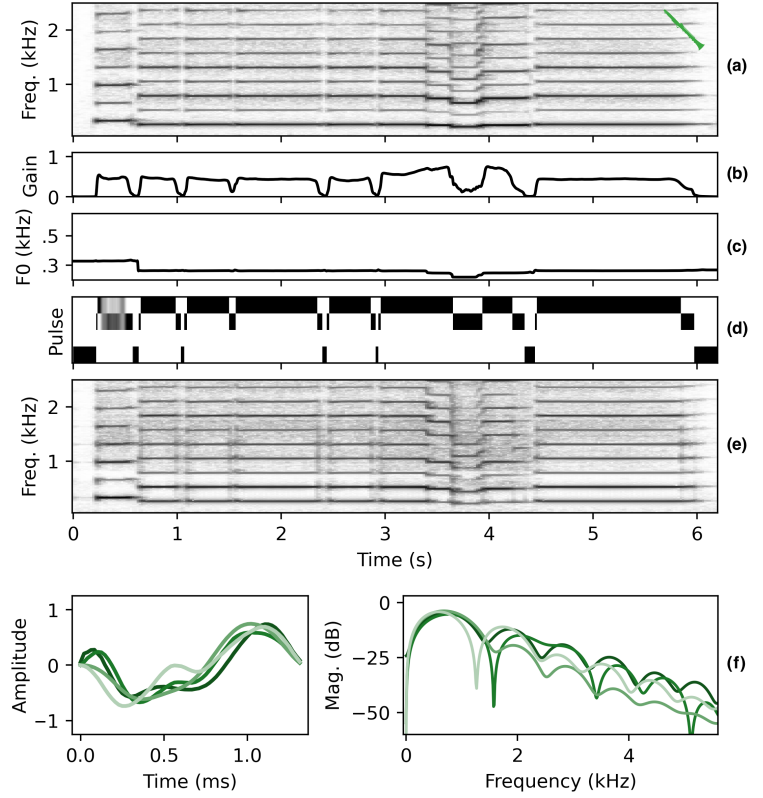


Figure 7.8: Synthesis results from the model using **wavetables** with $M = 4$ for clarinet (c1) and the *different voice* (DV) test signal.

- (a) Spectrogram of the original signal.
- (b) Gain estimates g_i from $\mathcal{G}$.
- (c) F0 trajectory (from the dataset).
- (d) Pulse weight estimates w_i from $\mathcal{W}$.
- (e) Spectrogram of the resynthesized output signal.
- (f) Visualization of learned pulses in time domain (left) and frequency domain (right). Color encodes the pulse number (from light to dark).

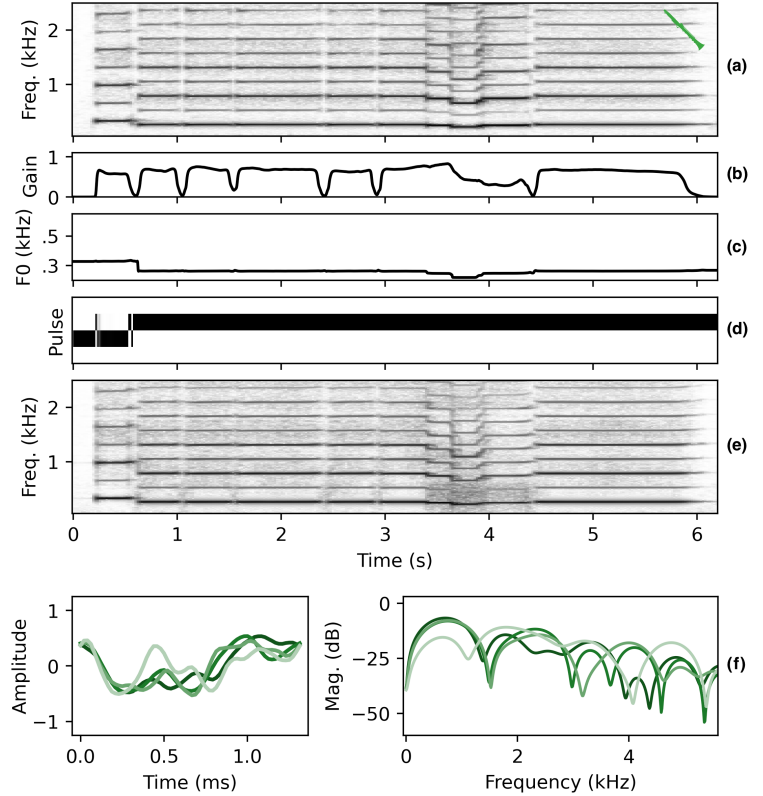


Figure 7.9: Synthesis results from the model using **pulsetables** with $M = 4$ for trumpet (tp) and the *different voice* (DV) test signal.

- (a) Spectrogram of the original signal.
- (b) Gain estimates g_i from $\mathcal{G}$.
- (c) F0 trajectory (from the dataset).
- (d) Pulse weight estimates w_i from $\mathcal{W}$.
- (e) Spectrogram of the resynthesized output signal.
- (f) Visualization of learned pulses in time domain (left) and frequency domain (right). Color encodes the pulse number (from light to dark).

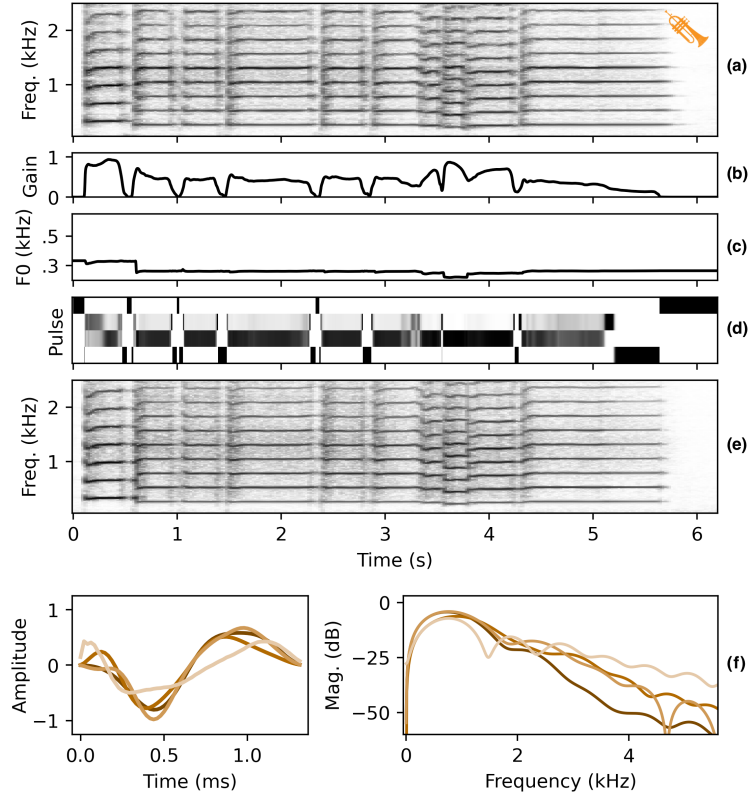
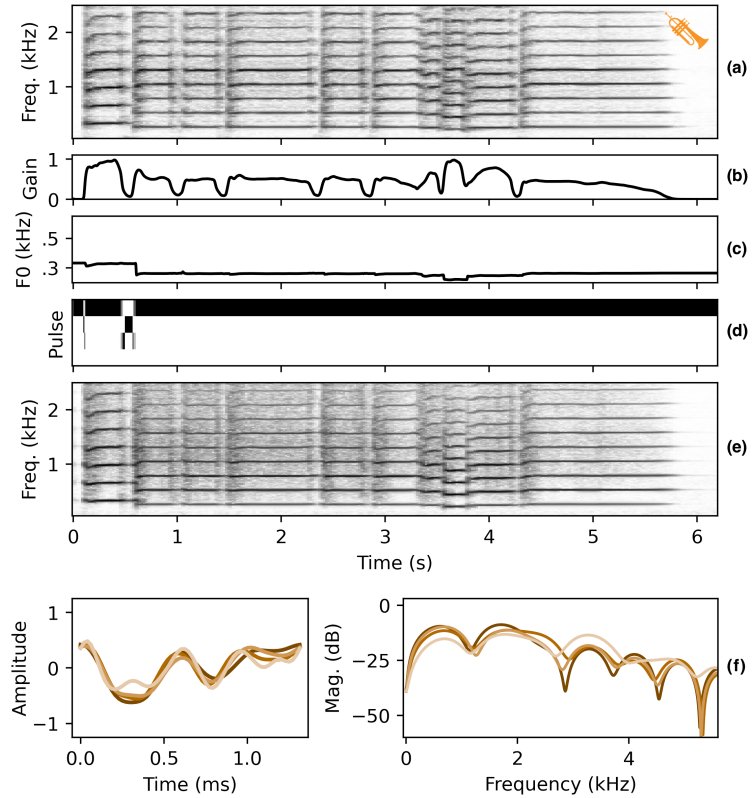


Figure 7.10: Synthesis results from the model using **wavetables** with $M = 4$ for trumpet (tp) and the *different voice* (DV) test signal.

- (a) Spectrogram of the original signal.
- (b) Gain estimates g_i from $\mathcal{G}$.
- (c) F0 trajectory (from the dataset).
- (d) Pulse weight estimates w_i from $\mathcal{W}$.
- (e) Spectrogram of the resynthesized output signal.
- (f) Visualization of learned pulses in time domain (left) and frequency domain (right). Color encodes the pulse number (from light to dark).



characteristic resonance. While this effect is less visible in the spectrum of the resynthesized signal than in the clarinet case, the quantitative analysis in Section 7.4.3 confirms that the pulsetable method generalizes more consistently to unseen pitch ranges for the trumpet.

Finally, it should be noted that the limitations discussed here are partly due to the small number of pulses used ($M = 4$). With a larger pulse set, the model could, for example, learn pitch-specific pulses, thereby mitigating the respective limitations of pulsetable synthesis for the clarinet or wavetable synthesis for the trumpet. However, these results highlight that selecting a synthesis model aligned with the physical characteristics of the target instrument can lead to more efficient representations with improved generalization.

7.4.3 Quantitative Comparison

For a more general view on the influence of different tonal signal synthesis methods in our wind synthesis system, we quantitatively compare the pulsetable and wavetable methods with a variant using additive synthesis for a range of wind instruments. To accommodate the additive synthesis, we replace the weight estimator $\mathcal{W}$ with a neural network that estimates the amplitudes of 100 sinusoids which are synthesized according to the method described in Chapter 2.4.1. The frequency of each sinusoid is determined by f_i , where each sinusoid models an harmonic at integer-multiples of the current F0. Since the number of outputs (100 amplitudes) and the modeling complexity is significantly larger, the number of trainable parameters is also higher, with 30 052 instead of 1012 parameters, resulting in 88 270 trainable parameters overall for the additive synthesis baseline, 48% more compared to the pulsetable model with $M = 4$.

Together with one additive synthesis model for each compared instrument, we train a total of 28 models, spanning four instruments, three different numbers of learned pulses $M \in \{2, 4, 16\}$, and two synthesis modes using pulsetables or wavetables. To compare these models in terms of objective metrics, we focus on the accuracy of the timbral reproduction, not considering other perceptually important factors for the overall resynthesis quality such as note onsets. To do this, we resynthesize the SV and DV chorales and calculate the frame-wise amplitude difference in dB between the first five harmonics that we obtain using F0-informed peak finding on x and y (see Section 6.2.1). This measure has been found to correlate well with perceived timbral differences, on par with metrics such as Mel-Frequency Cepstral Coefficient (MFCC) similarity [79]. As a low anchor, we additionally synthesize tones using additive synthesis, in which the harmonic amplitudes are replaced by those of a sawtooth timbre (see Figure 2.3). In this setting, the resulting harmonic amplitudes correspond to the actual instrument timbre only incidentally, showing how a clearly perceptually distinguishable synthesis compares to the other methods in terms of our metric.

Results are shown in Table 7.1. For the SV condition, only minor differences between pulsetable and wavetable synthesis can be observed, with a small tendency toward better performance with $M = 16$ in wavetable mode. For baritone, oboe, and clarinet, this model performs best, albeit with improvements of

	Instrument	Pulsetable			Wavetable			Add	Saw
	$M =$	2	4	16	2	4	16		
SV	Trumpet (tp)	2.84	2.67	2.57	2.80	2.71	2.66	2.80	3.71
	Baritone (bar)	3.78	3.80	3.89	3.81	4.14	3.78	4.11	6.88
	Oboe (ob)	3.65	3.55	3.58	3.86	3.69	3.09	3.32	4.71
	Clarinet (cl)	5.85	5.81	5.84	5.46	5.65	5.44	5.50	8.72
DV	Trumpet (tp)	4.90	4.96	4.96	5.22	5.39	5.62	6.50	7.04
	Baritone (bar)	3.67	3.88	3.61	5.24	4.35	5.18	5.19	6.31
	Clarinet (cl)	9.41	9.82	10.23	3.80	3.54	5.73	3.00	9.03

Table 7.1: Mean harmonic amplitude difference in dB between the first five harmonics of input and resynthesized signals. We compare a track from the same voice (SV) and a different voice (DV), a varying number of learned pulses M , and pulsetable vs. wavetable mode. “Add” denotes the additive synthesis model and “Saw” denotes the baseline with additive synthesis using a sawtooth timbre.

less than 1 dB in the mean harmonic amplitude difference between the best and worst model. This suggests that a smaller number of pulses suffices to generally capture the instrument timbre in the ChoraleBricks data. The additive synthesis model achieves a comparable performance, falling between the best and worst pulsetable or wavetable configurations across all evaluated instruments. Notably, for the clarinet, larger harmonic amplitude differences can be observed across all models, including the sawtooth anchor condition, ranging from 5 to 9 dB compared to 2 to 7 dB for the other instruments. This may be an additional indication for the more complex timbral characteristics of the clarinet that are to some degree also independent of f_i and g_i and therefore cannot be captured by our model.

In the DV condition, results confirm our qualitative observations from Section 7.4.2 that pulsetable models generalize more effectively to the unseen pitch range for the brass instruments, while for clarinet, the wavetable method generalizes significantly better. For the clarinet models, the higher mean amplitude difference for pulsetables (9.82 dB) than for wavetables (3.54 dB) with $M = 4$ particularly highlights the limitations of a fixed spectral envelope when few pulses are used. Notably, additive synthesis performs even better than the wavetable method for the clarinet, achieving a mean amplitude difference of 3.00 dB, whereas it performs worst for the trumpet (6.50 dB compared to 4.90 dB for pulsetable with $M = 2$).

7.4.4 Timbre Transfer

With this final experiment, we give an outlook on the concept of *timbre transfer*, where the idea is to resynthesize a melody or phrase played on one instrument to another. Similar to translation in language, there is no unique correct solution and “idioms” of one instrument have to be suitable translated to the other. For example, a strong vibrato in a violin tone (see, e.g., our running example in Chapter 2) would not directly translate to a flute, where it is more common to use tremolo, i.e., rhythmic variations in loudness instead of pitch.

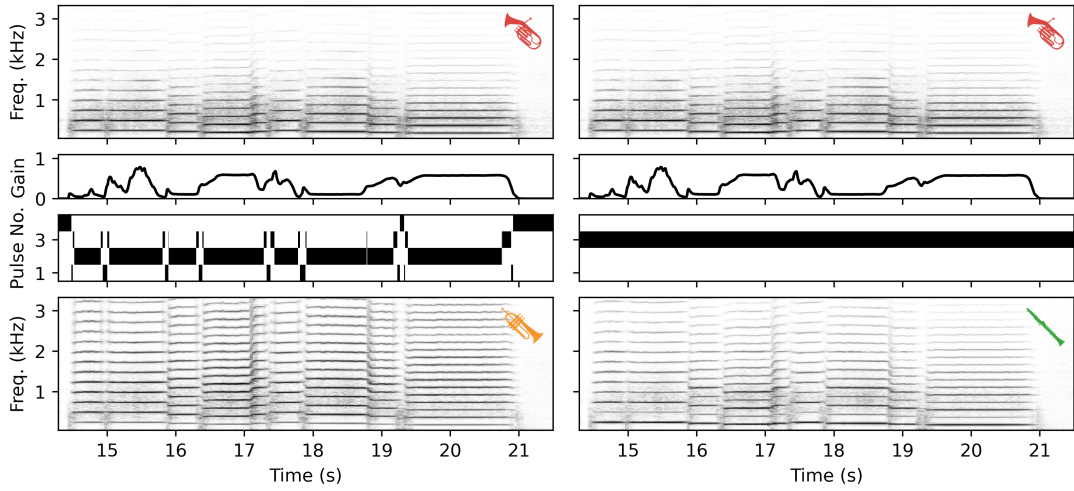


Figure 7.11: Synthesis results from the model trained on trumpet (tp) using **pulsetables** (left) and clarinet (c1) using **wavetables** (right) with $M = 4$ with a bar input signal (*timbre transfer*). Each side shows a spectrogram of the original signal (top), the gain estimated with $\mathcal{G}$ (second from top), the pulse weights estimated by $\mathcal{W}$ (third from top), and the spectrogram of the resynthesized output signal (bottom).

As an illustrative example, Figure 7.11 shows a timbre transfer experiment in which we take the control signals f_i and g_i estimated by the baritone model from the baritone SV track and resynthesize them using P , $\mathcal{W}$, and h from the trumpet–pulsetable (left) and clarinet–wavetable (right) models discussed in Section 7.4.2. Particularly the distinctive clarinet timbre, characterized by attenuated even harmonics and stronger high-frequency content compared to the baritone, is clearly visible in the spectrogram of the respective resynthesized output signal. However, this form of timbre transfer does not account for instrument- or player-specific characteristics of the control signals f_i and g_i , for example in terms of different intonation, pitch modulation, or dynamics [2]. The disentangled representation of our wind instrument synthesis system would allow the transfer of *playing style*, expressed in the temporal trajectories of the individual parameters, independently of timbre. This approach enables a more controlled and interpretable form of timbre transfer than that found in some DDSP models, where this effect is achieved by “overfitting” to the characteristics of the target instrument, so that any input signal (even a non-musical sound) is mapped to a sound that is naturally playable by the target instrument (see, e.g., [42]). Finally, the separate transformation of control signals, including noise magnitudes, could also be learned within our framework in an end-to-end fashion, for example through adversarial training. We leave the exploration of this aspect for future work.

7.5 Further Notes

In this chapter, we presented differentiable pulsetable synthesis and introduced a complete system for wind instrument modeling. The proposed framework disentangles pitch, loudness, and timbre in a parametric representation by combining physically informed assumptions with signal-based modeling. The model-based approach enables a synthesis method that is both computationally efficient and interpretable.

In the preliminary experimental evaluation, we did not find conclusive evidence that the pulsetable approach consistently outperforms wavetable synthesis or additive synthesis. Depending on the modeled instrument characteristics and the desired form of control, alternative synthesis methods for the tonal components of the sound, such as additive synthesis, can yield comparable results, provided that the method is sufficiently expressive. Nevertheless, the results indicate that pulsetable and wavetable synthesis generalize more reliably to unseen pitch ranges when the chosen method aligns with the physical sound generation mechanism of the modeled instrument, for example pulsetable synthesis for the trumpet and wavetable synthesis for the clarinet. Whether this behavior constitutes a perceptual advantage over alternative synthesis methods, for example due to more natural vibrato characteristics [137], remains an open question. Addressing such nuanced effects would require carefully designed perceptual listening experiments.

In addition to a more comprehensive evaluation, several promising directions for future work can be identified. Practically, training similar models with larger amounts of training data or employing pre-training will likely improve the overall synthesis quality and generalization. From a methodological perspective, the pulsetable method could be extended by learning pulses of varying length, increasing the ability to cover larger pitch ranges. Furthermore, as illustrated by the trumpet example in Figure 7.5, the current system struggles to reproduce the perceptually salient inharmonic components present at tone onsets that can neither be generated by the pulsetable method nor by the noise synthesizer. The onset is an important part of an instrument’s characteristic sound and it contributes substantially to instrument identification [193]. Previous work has addressed this challenge using sampled onset databases [38] or dynamic spectral shaping via frequency-modulated low-pass filters [40], and similar strategies could be integrated into this wind synthesis framework. Finally, linking back to Chapter 3, the frequency estimator $\mathcal{F}$ remains the only component in this framework that cannot be trained in an unsupervised fashion with the current setup. While separate F0 estimation is often a feasible alternative, jointly learning frequency estimation within the synthesis model may offer similar benefits to those observed for gain estimation, in that the model could potentially exploit small frequency fluctuations or systematic offsets to improve the perceptual plausibility of the synthesized output.

Beyond synthesis quality, the proposed system opens up several application-oriented possibilities. For example, the framework could be extended toward timbre or *playing style* transfer directly in the parameter space. By offering a more interpretable alternative to implicitly encoding such characteristics through model “overfitting”, this approach may not only allow for a more nuanced control over synthesis results,

but also facilitate a more systematic analysis of playing styles through analysis-by-synthesis, for example to compare different performers. Moreover, performance rendering in the parameter space could offer a lightweight and controllable alternative to end-to-end music synthesis systems. Such a rendering may also help mitigate data scarcity in wind instrument research by enabling the generation of controlled synthetic data, for example to augment training sets in tasks such as multi-pitch estimation.

Finally, it should be noted that, similar to the discussion in Chapter 4, purely data-driven approaches for high-quality sound synthesis are an active and productive area of research and development, often without exhibiting the same limitations with respect to specific modeling challenges such as tone onsets. The contribution of this work is therefore not intended as a direct alternative to these methods, but rather as a complementary approach that emphasizes controllability as well as efficiency in terms of data and computation.

8 Summary and Outlook

In this thesis, we made substantial contributions to the development and analysis of differentiable models for processing monophonic music signals and gained new insights into their practical use. We now recapitulate the main themes and summarize key outcomes, using four groups of three keywords, or *triads*, as a guiding principle to structure the discussion.

The first triad—*reconstruction, adaptation, synthesis*—divides the considered application scenarios into high-level tasks. For **reconstruction**, we showed how differentiable models enable the recovery of tone attributes from a degraded signal, selectively targeting timbre and loudness with the vocal tract model (Chapter 4). For **adaptation**, we modified specific attributes while preserving the remaining characteristics, targeting pitch with the intonation adaptation algorithm (Chapter 5) or timbre (Chapter 6) to reduce sensory dissonance. For **synthesis**, we generated signals from interpretable parameters and disentangled pitch, loudness, and timbre in an analysis-by-synthesis framework for wind instrument synthesis (Chapter 7). Overall, these case studies illustrate how differentiable models support the targeted and nuanced processing of music signals when combined with domain knowledge from audio signal processing, MIR, and psychoacoustics.

The second triad—*controllability, data efficiency, computational efficiency*—highlights recurring limitations of general-purpose deep learning approaches in music processing. We demonstrated that incorporating differentiable models into deep learning pipelines can effectively address these limitations (Chapters 4 and 7). Regarding **controllability**, a parameterization with musically meaningful parameters, such as F0, gain, and pulse weights in our pulsetable synthesizer, provides an interpretable way to manipulate the resulting signals. Regarding **data efficiency**, these parametric representations can be learned from comparably limited training data, with the unsupervised analysis-by-synthesis approach being particularly advantageous, as it does not require annotated training data. Regarding **computational efficiency**, our experiments showed that convincing reconstruction and synthesis results are achievable with lightweight neural networks when inductive bias is introduced through DDSP components. While large, general-purpose deep learning systems can yield high-quality results across a wider range of tasks without relying on domain-specific designs (as discussed in the fourth triad), our approaches offer clear advantages when controllability, data, or compute are limiting factors, making machine learning more accessible in settings where resources are scarce.

The third triad—*pitch, loudness, timbre*—comprises the tone attributes considered in this thesis and represents the interpretability inherent in those perceptually informed signal descriptors. Using differentiable models, we can leverage musical and perceptual concepts related to those attributes to process music signals.

With respect to **pitch**, we quantified intonation using a differentiable cost measure grounded in musical and perceptual principles and introduced a flexible framework for gradient-based adaptation (Chapter 5). With respect to **loudness**, while its direct control via the signal level is almost trivial, we considered the coupling between playing intensity and timbre changes in our wind synthesis system (Chapter 7). With respect to **timbre**, we identified its central role in sensory dissonance and explored methods to minimize dissonance in creative additive synthesis scenarios (Chapter 6). Overall, differentiable perceptual models offer strong potential to guide gradient-based optimization toward results and representations that align more closely with a human experience of music.

The fourth and final triad—*model, loss, data*—highlights the interplay between the three core components of any machine learning approach. Considering the **model**, we showed that explicit, differentiable signal models can restrict the space of possible outputs in a way that promotes meaningful solutions. However, these restrictions may also make optimization harder and can adversely affect convergence behavior. Considering the **loss**, we demonstrated that gradients with respect to individual model parameters can depend strongly on the interaction between model and loss function. We illustrated this with the concrete example of optimizing a sinusoid frequency parameter using the MSS loss (Chapter 3). Moreover, we observed that insights from the sinusoid model extend to a more general deep learning setting, where the smoothness of the loss with respect to the encoded parameters influences the organization of a VAE latent space. Considering the **data**, we showed that employing physically plausible models, such as pulsetable synthesis for the trumpet and wavetable synthesis for the clarinet (Chapter 7), can facilitate the learning of representations that generalize to unseen data. This is an example for the dependence between inductive bias and the amount of data that learning systems need to infer such structure implicitly. Overall, our results give insight into an inherent trade-off in the design of learning systems. On the one hand, imposing strong inductive biases restricts the space of possible outputs, which can improve controllability and reduce data requirements, but often comes at the cost of more difficult training. On the other hand, adopting more general output representations typically leads to more favorable training dynamics, while requiring larger amounts of training data and offering less direct control.

Based on these findings, we conclude this dissertation by outlining three promising directions for future research that build directly on the contributions of this work and discuss how each of these directions extends toward more general questions in the context of music signal processing. For more application-specific open problems, we refer to the summaries provided at the end of the individual chapters.

Understanding Model–Loss–Data Interactions. A first concrete direction for future work, building in particular on Chapters 4 and 7, is the systematic comparison of different synthesis methods and the inductive biases they introduce when used as differentiable components within a deep learning pipeline. While several synthesis methods have been explored in isolation in the context of DDSP [11, 21, 42, 104, 191, 236], systematic comparisons under controlled conditions are needed to characterize their respective strengths and limitations across synthesis tasks. More generally, such studies could

help clarify in which cases strong inductive biases (e.g., through physically informed models) support controllability and learning, and when they instead constrain expression or hinder optimization.

Perceptual and Musical Models in Music Synthesis. A second promising direction is to integrate the differentiable perceptual and musical models introduced in Chapters 5 and 6 into the training of deep learning-based music synthesis systems. Building on ideas from ensemble synthesis approaches such as [85], a synthesis model could be trained to “sing or play in tune” with a given accompaniment by incorporating the proposed models as auxiliary loss terms during training or inference. While this concept is not limited to systems that rely on explicit, differentiable signal models, combining it with an approach such as the wind instrument synthesizer proposed in Chapter 7 offers additional advantages. In this case, interpretable parameters (such as pitch, harmonic amplitudes, or other timbre-related controls) could be optimized directly and in isolation according to perceptual or musical criteria. At a broader level, integrating explicit perceptual or musical models into the loss function could enable the encoding of musical relationships that are difficult to learn from data alone, such as how well a synthesized voice blends with or fits into an existing accompaniment.

Differentiable Models for Music Representation Learning. Finally, a third direction for further research is to extend the experiments on latent space organization and its relationship to loss function properties. In particular, the VAE experiments from Chapter 3 should be broadened to include a wider range of real-world signals and more expressive, state-of-the-art encoder–decoder architectures (e.g., [36]), in order to more robustly assess how loss functions that are smooth with respect to frequency parameters shape latent representations. In addition, similar investigations could be conducted for other perceptually relevant properties, such as the timing of sound events in an audio signal, which are likewise associated with non-smooth loss landscapes. More broadly, these considerations are highly relevant to music representation learning, where learned representations [36, 107] aim to encode audio in a form suitable for diverse downstream tasks, including their use as inputs to multimodal large language models [28, 101]. Aligning such representations more closely with perceptually meaningful signal properties is another key step toward more interpretable and controllable interactions with music signals.

Appendix

A Periodicity of Spectral Loss with Rectangular Window

In this appendix, we show that the periodic behavior of the spectral loss between two sinusoid signals as shown in Figure 3.3 is a direct result of the frequency response of the window function.

The magnitude spectrum of the discrete rectangular window of length N is given by

$$W_{\text{rect}}(\omega) = \left| \frac{\sin\left(\frac{N}{2}\omega\right)}{\sin\left(\frac{1}{2}\omega\right)} \right|, \quad (\text{A.1})$$

where $\omega \in [0, 2\pi]$ is the normalized angular frequency in radians per sample [197].⁴⁰ We now take the spectral loss with configuration (WR, S2, C0, D2) as an example, using a rectangular window, $N = 512$, no compression, and squared ℓ^2 norm between the compared spectra (cf. Table 3.1). Analogously to Section 3.3, we assume the comparison of two sinusoids with unit amplitude and frequencies f_{tgt} and z in Hz, where z is the parameter we vary to obtain a loss landscape $\mathcal{L}_x(z)$. For convenient notation, we use angular frequencies in the following, setting $\omega_{\text{tgt}} = 2\pi f_{\text{tgt}}/F_s$ and $\omega_z = 2\pi z/F_s$. With this, we can express $\mathcal{L}_x(z)$ explicitly as the comparison of the magnitudes of two shifted versions of W_{rect} sampled at the $K = \lceil (N+1)/2 \rceil$ DFT bin center frequencies $b_k = 2\pi k/N$ for $k \in [0 : K-1]$, resulting in

$$\mathcal{L}_x(z) = \sum_{k=0}^{K-1} \left(W_{\text{rect}}(b_k - \omega_{\text{tgt}}) - W_{\text{rect}}(b_k - \omega_z) \right)^2. \quad (\text{A.2})$$

By inserting Eq. A.1 into Eq. A.2 and rearranging based on the trigonometric identity

$$\sin(\alpha) \sin(\beta) = \frac{1}{2} (\cos(\alpha - \beta) - \cos(\alpha + \beta)), \quad (\text{A.3})$$

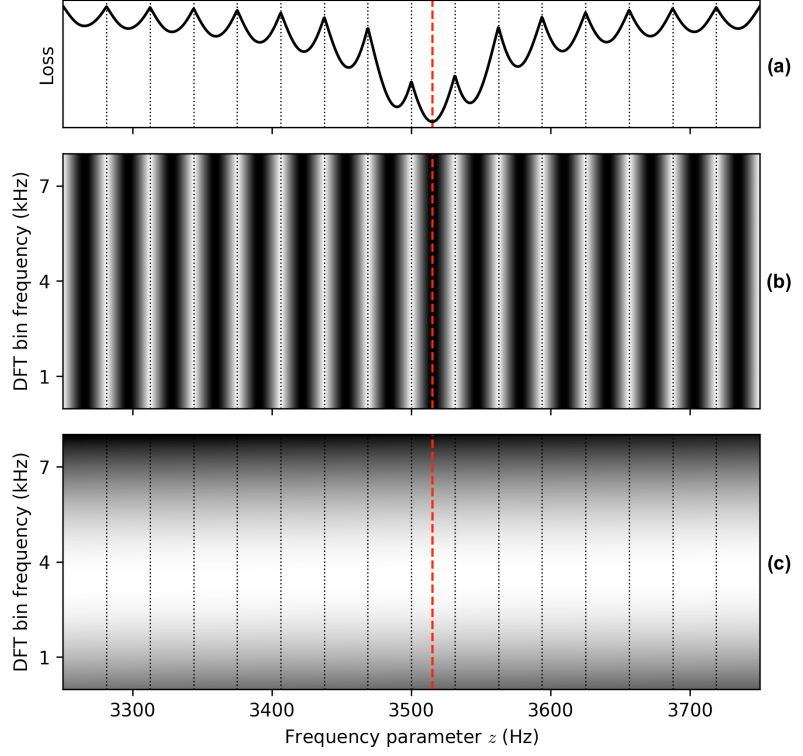
we obtain

$$\begin{aligned} \mathcal{L}_x(z) &= \sum_{k=0}^{K-1} W_{\text{rect}}(b_k - \omega_{\text{tgt}})^2 + \sum_{k=0}^{K-1} W_{\text{rect}}(b_k - \omega_z)^2 \\ &\quad - 2 \sum_{k=0}^{K-1} \left| \frac{\cos\left(\frac{N}{2}\omega_z - \frac{N}{2}\omega_{\text{tgt}}\right) - \cos\left(\frac{N}{2}\omega_z + \frac{N}{2}\omega_{\text{tgt}} - Nb_k\right)}{\sin\left(\frac{1}{2}(b_k - \omega_z)\right) \sin\left(\frac{1}{2}(b_k - \omega_{\text{tgt}})\right)} \right|. \end{aligned} \quad (\text{A.4})$$

⁴⁰ Note that $W_{\text{rect}}(0) := N$ is explicitly defined (similar to the sinc function) to avoid an undefined division by 0. This case must be handled accordingly also in Equation A.4.

Figure A.1: Visualization of the cross-term numerator and denominator from Equation A.4. Black dotted vertical lines indicate DFT bin frequencies. The red dashed vertical indicates the target frequency $f_{\text{tgt}} = 3515$ Hz.

- (a) Loss landscape $\mathcal{L}_x(z)$ for reference.
 (b) Visualization of the numerator term with varying the frequency parameter z (x-axis) and the DFT frequency bin b (y-axis). Dark color indicates larger values.
 (c) Visualization of the denominator term, analogously to (b).



While the first two terms can be directly interpreted as the overall energies of the two signals being compared, the cross-term determines the structure of the resulting loss landscape. In the numerator, Nb_k is always an integer multiple of 2π and therefore does not affect the value of the periodic cosine function. As a result, the numerator can be interpreted as a superposition of two cosine terms that depend only on z , each with a frequency of $N/(2F_s)$ in Hz. Applying half-wave rectification by taking the absolute value makes the numerator periodic with period N/F_s , corresponding exactly to the spacing of the DFT bins. This behavior is illustrated in Figure A.1b, which shows the numerator as a function of z (x-axis) and b (y-axis) for a fixed $f_{\text{tgt}} = 3515$ Hz. Higher values, indicated by darker colors, correspond to larger cross-terms being subtracted from the sum of signal energies in Equation A.4, leading to an overall smaller loss $\mathcal{L}_x(z)$.

The denominator decreases as b_k approaches either z or f_{tgt} . With this, the term primarily acts as a weighting of the relative contribution of the individual frequency bins b_k to the overall loss and it does not introduce additional periodicity to $\mathcal{L}_x(z)$. This effect is also visible in Figure A.1c, where small denominator values (which lead to a smaller loss overall) are largely determined by b_k .

B Differentiability of Dissonance Kernel

Proposition B.1. The function $d_h(f, f')$ (Eq. 5.5) with parameters $f, f', d_{\max} \in \mathbb{R}_{>0}$ is differentiable with respect to f .

Proof. We rewrite $d_h(f, f') = \phi(\psi(f, f'))$ as the composition of two functions

$$\psi(f, f') = \frac{\log_2(f/f')}{d_{\max}} \quad (\text{B.5})$$

with $f, f', d_{\max} \in \mathbb{R}_{>0}$ and

$$\phi(x) = \begin{cases} \exp(-\ln^2(|x|)) & x \neq 0 \\ 0 & x = 0 \end{cases} \quad (\text{B.6})$$

with $x \in \mathbb{R}$.

$\psi(f, f')$ is a composition of functions that are differentiable in its entire domain (including $f = f'$), so it is itself differentiable. The same is easily verified for $\phi(x)$ with $x \neq 0$, while for $x = 0$, we need to show that the limit

$$\lim_{h \rightarrow 0} \frac{\phi(0+h) - \phi(0)}{h} \quad (\text{B.7})$$

exists from both sides. For this, we rewrite the piece-wise definition as

$$\phi(x) = \begin{cases} \exp(-\ln^2(-x)) & x < 0 \\ 0 & x = 0 \\ \exp(-\ln^2(x)) & x > 0 \end{cases} \quad (\text{B.8})$$

and obtain

$$\begin{aligned} & \lim_{h \rightarrow 0^+} \frac{\exp(-\ln^2(h)) - 0}{h} \\ &= \lim_{h \rightarrow 0^+} \frac{\exp(-\ln^2(h))}{\exp(\ln(h))} \\ &= \lim_{h \rightarrow 0^+} \exp(-\ln^2(h) - \ln(h)) \\ &= \lim_{h \rightarrow 0^+} \exp(\underbrace{-\ln(h)}_{\rightarrow \infty} \underbrace{(\ln(h) + 1)}_{\rightarrow -\infty}) = 0, \end{aligned} \quad (\text{B.9})$$

$$\begin{aligned} & \lim_{h \rightarrow 0^-} \frac{\exp(-\ln^2(-h)) - 0}{h} \\ &= \lim_{h \rightarrow 0^+} \frac{\exp(-\ln^2(h)) - 0}{-h} = 0. \end{aligned} \quad (\text{B.10})$$

$d_h(f, f')$, as the composition of two differentiable functions $\phi(x)$ and $\psi(f, f')$, is therefore also differentiable.

□

C Zero-Interleaved Spectrum of Repeated Signals

Proposition C.1. Repeating any signal in time domain leads to interleaving the spectrum with zeros.

Proof. We can rewrite the Fourier transform of a length- N signal x that is repeated M times (i.e., has a total length of MN) as

$$\begin{aligned} X(k) &= \sum_{n=0}^{MN-1} x(n \bmod N) \exp\left(-i2\pi \frac{kn}{MN}\right) \\ &= \sum_{n=0}^{N-1} x(n) \sum_{m=0}^M \exp\left(-i2\pi \frac{k(n+mN)}{MN}\right), \end{aligned} \quad (\text{C.11})$$

where mod is the modulo operator. The sum $\sum_{m=0}^M \exp\left(-i2\pi \frac{k(n+mN)}{MN}\right)$ becomes zero for all k with $k \bmod M \neq 0$. This can be seen if we rewrite it again as

$$\begin{aligned} &\sum_{m=0}^M \exp\left(-i2\pi \frac{k(n+mN)}{MN}\right) \\ &= \sum_{m=0}^M \exp\left(-i2\pi \frac{kn}{MN}\right) \exp\left(-i2\pi \frac{kmN}{MN}\right) \\ &= \exp\left(-i2\pi \frac{kn}{MN}\right) \sum_{m=0}^M \exp\left(-i2\pi \frac{km}{M}\right), \end{aligned} \quad (\text{C.12})$$

which can be interpreted as the length- M Fourier transform of a constant. The Fourier transform of a constant is only non-zero for the zero-frequency component at $k = 0$, or, due to aliasing, at $k = 0, M, 2M, \dots$

□

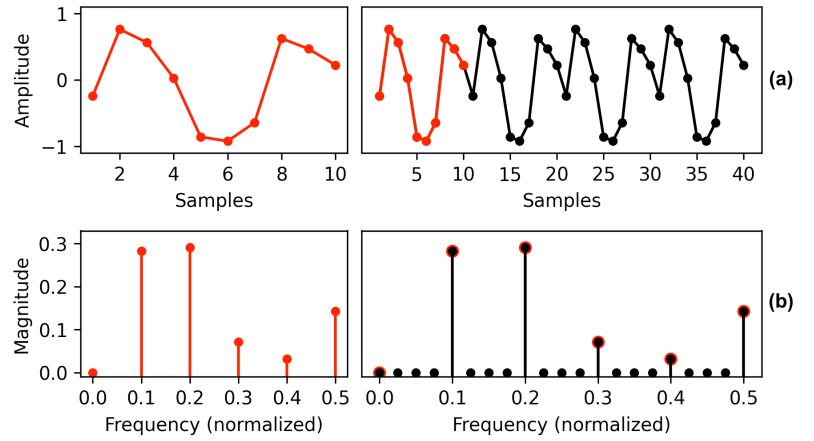
We can illustrate this property with an example. Figure C.2a shows a time-domain signal x with length $N = 10$ in red on the left side. This random signal is repeated $M = 4$ times to create a new signal y of length $MN = 40$ shown in black on the right side. Figure C.2b shows the magnitude spectra of x (red) and y (black). The non-zero frequency components of both signals have identical magnitude, but in the spectrum of y (shown in black), the additional bins in between have a magnitude of 0. The length N of the repetition period therefore determines the spacing of the non-zero frequency components, which are located at the bin center frequencies of a DFT with length N , i.e., at integer multiples of F_s/N . The

C. Zero-Interleaved Spectrum of Repeated Signals

Figure C.2: Example illustrating the zero-interleaved spectrum of a repeated signal.

(a) A signal x with length $N = 10$ (left side, red), and the same signal repeated $M = 4$ times (right side, black), both shown in time domain.

(b) Magnitude spectra of the two signals shown above.



resulting tone that can be heard when y is played therefore has an F0 of F_s/N Hz and exactly integer harmonics with magnitudes determined by the DFT of x .

Note that the behavior described in Proposition C.1 holds exactly only for repetitions of integer-length signals. As discussed in Chapter 7, modifying the repeated signal, for example through time stretching or zero-padding, still results in harmonic signals, but yields intermediate frequency bins with non-zero values. Moreover, if the length of y is not a multiple of N , the corresponding DFT bin center frequencies no longer align, and spectral leakage further distorts the result.

Abbreviations

12-TET	12-Tone Equal Temperament	LM-SVR	Singing Voice Reconstruction from Larynx Microphone
AM	Amplitude Modulation	LTI	Linear Time Invariant
BBWE	Blind Bandwidth Extension	MIDI	Musical Instrument Digital Interface
BPM	Beats Per Minute	MFCC	Mel-Frequency Cepstral Coefficient
CLMM	Cumulative Link Mixed Model	MIR	Music Information Retrieval
CM	Close-Up Microphone	MPE	MIDI Polyphonic Expression
CNN	Convolutional Neural Network	MSE	Mean Squared Error
DAW	Digital Audio Workstation	MSS	Multi-Scale Spectral
DFT	Discrete Fourier Transform	NAM	Non-Audible Murmur
DTFT	Discrete-Time Fourier Transform	NN	Neural Network
DSP	Digital Signal Processing	RMS	Root Mean Square
DDSP	Differentiable Digital Signal Processing	RPA	Raw Pitch Accuracy
ET	Equal Temperament	RTF	Relative Transfer Function
ERB	Equivalent Rectangular Bandwidth	SD	Sensory Dissonance
F0	Fundamental Frequency	SE	Speech Enhancement
FIR	Finite Impulse Response	SMS	Spectral Modeling Synthesis
FM	Frequency Modulation	SNR	Signal-to-Noise Ratio
GRA	Gradient-Sign Ranking Accuracy	SPL	Sound Pressure Level
HC	Harmonic Centroid	STFT	Short-Time Fourier Transform
i.i.d.	independent and identically distributed	SVR	Singing Voice Reconstruction
IIR	Infinite Impulse Response	SVS	Singing Voice Synthesis
JI	Just Intonation	TI-FIR	Time-Invariant FIR
KL	Kullback-Leibler	TV-FIR	Time-Variant FIR
LM	Larynx Microphone	VAE	Variational Autoencoder
		WER	Word Error Rate

Bibliography

- [1] Mark Abel. *Groove: An Aesthetic of Measured Time*, volume 73 of *Historical Materialism*. Brill, Leiden, Netherlands, 2014. ISBN 978-90-04-24293-7. doi: 10.1163/9789004242944.
- [2] Jakob Abeßer, Klaus Frieler, Estefanía Cano, Martin Pfeiderer, and Wolf-Georg Zaddach. Score-informed analysis of tuning, intonation, pitch modulation, and dynamics in jazz solos. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):168–177, 2017. doi: 10.1109/TASLP.2016.2627186. URL <https://doi.org/10.1109/TASLP.2016.2627186>.
- [3] Jakob Abeßer, Simon Schwär, and Meinard Müller. Pitch contour exploration across audio domains: A vision-based transfer learning approach. *CoRR*, abs/2503.19161, 2025.
- [4] Per-Gunnar Alldahl. *Choral Intonation*. Gehrmans Musikförlag, 2008. ISBN 9789177482567.
- [5] Juan Alonso and Cumhur Erkut. Latent space explorations of singing voice synthesis using DDSP. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 183–190, Online, 2021.
- [6] Samuel Applebaum and Thomas Lindsay. *The Art and Science of String Performance*. Alfred Publishing Company, Sherman Oaks, CA, 1986. ISBN 978-0882843520.
- [7] Anders Askenfelt, Jan Gauffin, Johan Sundberg, and Peter Kitzing. A comparison of contact microphone and electroglottograph for the measurement of vocal fundamental frequency. *Journal of Speech, Language, and Hearing Research*, 23(2):258–273, 1980.
- [8] Stefan Balke, Axel Berndt, and Meinard Müller. ChoraleBricks: A modular multitrack dataset for wind music research. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 8(1):39–54, 2025.
- [9] James Murray Barbour. *Tuning and Temperament: A Historical Survey*. Michigan State College Press, 1951.
- [10] Jacob Benesty, Jingdong Chen, and Yiteng Huang. *Microphone Array Signal Processing*. Springer Verlag, 1st edition, 2008.
- [11] Hans-Ulrich Berendes, Simon Schwär, Maximilian Schäfer, and Meinard Müller. Towards differentiable piano synthesis based on physical modeling. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Milano, Italy, 2023.
- [12] Hans-Ulrich Berendes, Simon Schwär, and Meinard Müller. Lyrics transcription in Western classical music with Whisper: A case study on Schubert’s Winterreise. In *Workshop on NLP for Music and Audio (NLP4MUSA)*, Oakland, CA, United States, 2024.
- [13] Jesse Berezovsky. The structure of musical harmony as an ordered phase of sound: A statistical mechanics approach to music theory. *Science Advances*, 5(5):eaav8490, 2019. doi: 10.1126/sciadv.aav8490.

Bibliography

- [14] Emmanuel Bigand, Richard Parncutt, and Fred Lerdahl. Perception of musical tension in short chord sequences: The influence of harmonic function, sensory dissonance, horizontal motion, and musical training. *Perception & Psychophysics*, 58(1):125–141, 1996. doi: 10.3758/BF03205482.
- [15] Paul Boersma. Praat, a system for doing phonetics by computer. *Glott International*, 5(9/10):341–345, 2001.
- [16] Xavier Boutillon. Model for piano hammers: Experimental determination and digital simulation. *The Journal of the Acoustical Society of America*, 83(2):746–754, 1988. doi: 10.1121/1.396194.
- [17] Daniel L. Bowling, Dale Purves, and Kamraan Z. Gill. Vocal similarity predicts the relative attraction of musical chords. *Proceedings of the National Academy of Sciences*, 115(1):216–221, 2018. doi: 10.1073/pnas.1707517114.
- [18] Fabian Burk, Louisa Traser, Michael Burdumy, Bernhard Richter, and Matthias Echternach. Dynamic changes of vocal tract dimensions with sound pressure level during messa di voce. *The Journal of the Acoustical Society of America*, 154(6):3595–3607, 2023. doi: 10.1121/10.0023080.
- [19] Arturo Camacho and John G. Harris. A sawtooth waveform inspired pitch estimator for speech and music. *The Journal of the Acoustical Society of America*, 124(3):1638–1652, 2008.
- [20] Estefanía Cano, Derry FitzGerald, Antoine Liutkus, Mark D. Plumbley, and Fabian-Robert Stöter. Musical source separation: An introduction. *IEEE Signal Processing Magazine*, 36(1):31–40, 2019. doi: 10.1109/MSP.2018.2874719.
- [21] Franco Caspe, Andrew McPherson, and Mark Sandler. DDX7: Differentiable FM Synthesis of Musical Instrument Sounds. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 608–616, Bengaluru, India, 2022.
- [22] Celemony. Melodyne. <https://www.celemony.com/en/melodyne/what-is-melodyne>, 2025. Accessed: 2025-11-11.
- [23] Yin-Ping Cho, Fu-Rong Yang, Yung-Chuan Chang, Ching-Ting Cheng, Xiao-Han Wang, and Yi-Wen Liu. A survey on recent deep learning-driven singing voice synthesis systems. In *Proceedings of the IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 319–323, 2021. doi: 10.1109/AIVR52153.2021.00067.
- [24] Hyeong-Seok Choi, Juheon Lee, Wansoo Kim, Jie Lee, Hoon Heo, and Kyogu Lee. Neural analysis and synthesis: Reconstructing speech from self-supervised representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 16251–16265, Virtual, 2021.
- [25] Hyeong-Seok Choi, Jinhyeok Yang, Juheon Lee, and Hyeongju Kim. NANSY++: Unified voice synthesis with neural analysis and synthesis. *Computing Research Repository (CoRR)*, abs/2211.09407, 2022. URL <https://arxiv.org/abs/2211.09407>.
- [26] John M. Chowning. The synthesis of complex audio spectra by means of frequency modulation. *Journal of the Audio Engineering Society*, 21(7):526–534, 1973.
- [27] Rune H. B. Christensen. *ordinal—Regression Models for Ordinal Data*, 2023. R package version 2023.12-4.1.

- [28] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen2-Audio technical report. *CoRR*, abs/2407.10759, 2024.
- [29] David Code. Groven.Max: An adaptive tuning system for MIDI pianos. *Computer Music Journal*, 26:50–61, 2002.
- [30] Perry R. Cook. *Real Sound Synthesis for Interactive Applications*. A K Peters / CRC Press, Natick, MA, 2002. ISBN 1-56881-168-3. doi: 10.1201/b19597.
- [31] Timothy James Cuffman. *A Practical Introduction to Just Intonation Through String Quartet Playing*. DMA dissertation, University of Iowa, Iowa City, IA, 2016.
- [32] Jiajie Dai and Simon Dixon. Intonation trajectories within tones in unaccompanied soprano, alto, tenor, bass quartet singing. *The Journal of the Acoustical Society of America*, 146(2):1005–1014, 2019. doi: 10.1121/1.5120483.
- [33] Peter Daniel and Reinhard Weber. Psychoacoustical roughness: Implementation of an optimized model. In *Acta Acustica united with Acustica*, volume 83(1), pages 113–123, 1997.
- [34] Meribeth Bunch Dayme. *Dynamics of the Singing Voice*. Springer, New York, NY, USA, 5 edition, 2009. doi: 10.1007/978-3-211-88729-5.
- [35] Alain de Cheveigné and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America (JASA)*, 111(4):1917–1930, 2002.
- [36] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023.
- [37] Tomas Dekens, Yorgos Patsis, Werner Verhelst, Frédéric Beaugendre, and François Capman. A multi-sensor speech database with applications towards robust speech processing in hostile environments. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco, 2008.
- [38] Istvan Derenyi and Roger B. Dannenberg. Synthesizing trumpet performances. In *Proceedings of the International Computer Music Conference (ICMC)*, Ann Arbor, Michigan, USA, 1998.
- [39] Christian Dittmar, Martin Pfeiderer, Stefan Balke, and Meinard Müller. A swingogram representation for tracking micro-rhythmic variation in jazz performances. *Journal of New Music Research*, 47(2):97–113, 2018. doi: 10.1080/09298215.2017.1367405.
- [40] Christian Dittmar, Johannes Zeitler, Stefan Balke, Simon Schwär, and Meinard Müller. PULSE-IT: Lightweight and expressive synthesis of wind instrument playing. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Daejeon, South Korea, 2025.
- [41] Tuomas Eerola and Imre Lahdelma. The anatomy of consonance/dissonance: Evaluating acoustic and cultural predictors across multiple datasets with chords. *Music & Science*, 4:1–19, 2021. doi: 10.1177/20592043211030471.

Bibliography

- [42] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. DDSP: Differentiable digital signal processing. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual, 2020.
- [43] Jesse Engel, Rigel Swavely, Lamtharn Hanoi Hantrakul, Adam Roberts, and Curtis Hawthorne. Self-supervised pitch detection by inverse audio synthesis. In *International Conference on Machine Learning (ICML), Workshop on Self-Supervision in Audio and Speech*, Vienna, Austria, 2020.
- [44] Jesse H. Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with WaveNet autoencoders. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 70, pages 1068–1077, Sydney, Australia, 2017.
- [45] Philippe Esling, Axel Chemla-Romeu-Santos, Alexis Bitton, and Théodore Caillon. RAVE: A variational autoencoder for fast and high-quality neural audio synthesis. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Vienna, Austria, 2020.
- [46] Behnam Faghih and Joseph Timoney. Annotated-VocalSet: A singing voice dataset. *Applied Sciences*, 12(18):9257, 2022. doi: 10.3390/app12189257.
- [47] Gunnar Fant. The LF-model revisited: Transformations and frequency domain analysis. Quarterly Progress and Status Report 2–3, Speech Transmission Laboratory, Royal Institute of Technology, Stockholm, 1995.
- [48] Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taïga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop regressing: Training value functions via classification for scalable deep RL. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.
- [49] Leonardo Fierro and Vesa Välimäki. SiTraNo: A MATLAB app for sines-transients-noise decomposition of audio signals. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 73–80, Vienna, Austria, 2021.
- [50] Derry FitzGerald. Harmonic/percussive separation using median filtering. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 246–253, Graz, Austria, September 2010.
- [51] Harvey Fletcher and Wilden A. Munson. Loudness, its definition, measurement and calculation. *Journal of the Acoustic Society of America*, 5:82–108, 1933.
- [52] Neville H. Fletcher and Thomas D. Rossing. *The Physics of Musical Instruments*. Springer, Berlin, Germany, 2nd edition, 1998.
- [53] Mary A. Florentine, Arthur N. Popper, and Richard R. Fay, editors. *Loudness*, volume 37 of *Springer Handbook of Auditory Research*. Springer, New York, NY, USA, 2011. doi: 10.1007/978-1-4419-6714-1.
- [54] Fred N. Fritsch and R. E. Carlson. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17(2):238–246, 1980. doi: 10.1137/0717021.
- [55] Sharon Gannot, David Burshtein, and Ehud Weinstein. Signal enhancement using beamforming and nonstationarity with applications to speech. *IEEE Transactions on Signal Processing*, 49(8):1614–1626, 2001. doi: 10.1109/78.934132.

- [56] John M. Geringer, Rebecca B. MacLeod, Clifford K. Madsen, and Jessica Napoles. Perception of melodic intonation in performances with and without vibrato. *Psychology of Music*, 43(5):675–685, 2015.
- [57] Beat Gfeller, Christian Frank, Dominik Roblek, Matthew Sharifi, Marco Tagliasacchi, and Mihajlo Velimirovic. SPICE: Self-supervised pitch estimation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1118–1128, 2020.
- [58] Brian R. Glasberg and Brian C. J. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing Research*, 47(1):103–138, 1990. ISSN 0378-5955.
- [59] Werner Goebel and Laura Bishop. Joint shaping of musical time: How togetherness emerges in music ensemble performance. In Clemens Wöllner and Justin London, editors, *Performing Time: Synchrony and Temporal Flow in Music and Dance*. Oxford University Press, 2023. ISBN 9780192896254.
- [60] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge and London, 2016. ISBN 978-0-262-03561-3.
- [61] Colin E. Gough. Measurement, modelling and synthesis of violin vibrato sounds. *Acta Acustica united with Acustica*, 91(2):229–240, 2005.
- [62] Martin Graciarena, Horacio Franco, Kemal Sonmez, and Harry Bratt. Combining standard and throat microphones for robust speech recognition. *IEEE Signal Processing Letters*, 10(3):72–74, 2003.
- [63] John M. Grey and John W. Gordon. Perceptual effects of spectral modifications on musical timbres. *The Journal of the Acoustical Society of America*, 63(5):1493–1500, 1978.
- [64] Jiarui Hai and Mounya Elhilali. Diff-Pitcher: Diffusion-based singing voice pitch correction. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2023. doi: 10.1109/WASPAA58266.2023.10248127.
- [65] Ulf A. Hamster. bwsample: Processing best-worst scaling data. In *Journal of Open Source Software*, volume 6(64), page 3324, 2021.
- [66] Peter M. C. Harrison and Marcus T. Pearce. An energy-based generative sequence model for testing sensory theories of western harmony. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 160–167, Paris, France, 2018.
- [67] Peter M. C. Harrison and Marcus T. Pearce. Simultaneous consonance in music perception and composition. *Psychological Review*, 127(2):216–244, 2020.
- [68] Christopher Harte, Mark B. Sandler, Samer Abdallah, and Emilia Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 66–71, London, UK, 2005.
- [69] Frank Havrø. ‘You Cannot Just Say: “I am Singing the Right Note”’. Discussing intonation issues with Neue Vocalsolisten Stuttgart. *Music & Practice*, 1, 2013. doi: 10.32063/0104.
- [70] Curtis Hawthorne, Ian Simon, Adam Roberts, Neil Zeghidour, Josh Gardner, Ethan Manilow, and Jesse H. Engel. Multi-instrument music synthesis with spectrogram diffusion. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 598–607, 2022.

Bibliography

- [71] Ben Hayes, Charalampos Saitis, and György Fazekas. Sinusoidal frequency estimation by gradient descent. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1–5, Rhodes Island, Greece, 2023. doi: 10.1109/ICASSP49357.2023.10095188.
- [72] Ben Hayes, Jordie Shier, György Fazekas, Andrew McPherson, and Charalampos Saitis. A review of differentiable digital signal processing for music and speech synthesis. *Frontiers in Signal Processing*, 3, 2024. ISSN 2673-8198. doi: 10.3389/frsip.2023.1284100.
- [73] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, Nevada, USA, 2016. doi: 10.1109/CVPR.2016.90.
- [74] Hermann L. F. Helmholtz. *On the Sensations of Tone as a Physiological Basis for the Theory of Music*. Longmans, Green, and Co., 4 edition, 1912.
- [75] Paula Henry and Tomaz R. Letowski. Bone conduction: Anatomy, physiology and communication. Technical report, Army Research Laboratory, 2007.
- [76] Christian Herbst. Electrolottography – an update. *Journal of Voice*, 34(4):503–526, 2020.
- [77] John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 31–35, Shanghai, China, 2016. doi: 10.1109/ICASSP.2016.7471631.
- [78] Andrew Horner and James Beauchamp. Synthesis of trumpet tones using a wavetable and a dynamic filter. *Journal of the Audio Engineering Society*, 43(10):799–812, 1995.
- [79] Andrew B. Horner, James W. Beauchamp, and Richard H. Y. So. Evaluation of mel-band and mfcc-based error metrics for correspondence to discrimination of spectrally altered musical instrument sounds. *Journal of the Audio Engineering Society*, 59(5):290–303, 2011.
- [80] David M. Howard. Intonation drift in a capella soprano, alto, tenor, bass quartet singing with key modulation. *Journal of Voice*, 21(3):300–315, 2007. doi: 10.1016/j.jvoice.2005.12.005.
- [81] David Miles Huber. *The MIDI Manual*. Focal Press, 3rd edition, 2006.
- [82] Eric J. Humphrey, Sravana Reddy, Prem Seetharaman, Aparna Kumar, Rachel M. Bittner, Andrew Demetriou, Sankalp Gulati, Andreas Jansson, Tristan Jehan, Bernhard Lehner, Anna Krupse, and Luwei Yang. An introduction to signal processing for singing-voice analysis: High notes in the effort to automate the understanding of vocals in music. *IEEE Signal Processing Magazine*, 36(1):82–94, 2019. doi: 10.1109/MSP.2018.2875133.
- [83] William Hutchinson and Leon Knopoff. The acoustic component of Western consonance. *Interface*, 7(1): 1–29, 1978. doi: 10.1080/09298217808570246.
- [84] Ji Sang Hwang, Sang Hoon Lee, and Seong Whan Lee. HiddenSinger: High-quality singing voice synthesis via neural audio codec and latent diffusion models. *Neural Networks*, 181:106762, 2025. doi: 10.1016/j.neunet.2024.106762.

- [85] Hiroaki Hyodo, Shinnosuke Takamichi, Tomohiko Nakamura, Junya Koguchi, and Hiroshi Saruwatari. DNN-based ensemble singing voice synthesis with interactions between singers. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, pages 660–667, Macao, China, 2024.
- [86] ISO 16:1975. Acoustics – Standard tuning frequency (Standard musical pitch), 1975.
- [87] Lloyd A. Jeffress. Beating sinusoids and pitch changes. In *The Journal of the Acoustical Society of America*, volume 43(6), pages 1464–1464, 1968.
- [88] Patrik N. Juslin and Petri Laukka. Communication of emotions in vocal expression and music performance: Different channels, same code? *Psychological Bulletin*, 129(5):770–814, 2003. doi: 10.1037/0033-2909.129.5.770.
- [89] Patrik N. Juslin and Petri Laukka. Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening. *Journal of New Music Research*, 33(3):217–238, 2004. doi: 10.1080/0929821042000317813.
- [90] Anuj Karpatne, Ramakrishnan Kannan, and Vipin Kumar, editors. *Knowledge Guided Machine Learning: Accelerating Discovery using Scientific Knowledge and Data*. Chapman and Hall/CRC, 1 edition, 2022. doi: 10.1201/9781003143376.
- [91] James M. Kates. On using coherence to measure distortion in hearing aids. *The Journal of the Acoustical Society of America*, 91(4):2236–2244, 1992. doi: 10.1121/1.403657.
- [92] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *CoRR*, abs/2006.12057, 2020.
- [93] Douglas H. Keefe. Physical modeling of wind instruments. *Computer Music Journal*, 16(4):57–73, 1992.
- [94] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, pages 2350–2354, Graz, Austria, 2019. doi: 10.21437/interspeech.2019-2219.
- [95] Daewoong Kim, Hao-Wen Dong, and Dasaem Jeong. ViolinDiff: Enhancing expressive violin synthesis with pitch bend conditioning. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1–5, Hyderabad, India, 2025. doi: 10.1109/ICASSP49660.2025.10890613.
- [96] Hounsung Kim, Soonbeom Choi, and Juhan Nam. Expressive acoustic guitar sound synthesis with an instrument-specific input representation and diffusion outpainting. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7620–7624, Seoul, South Korea, 2024. doi: 10.1109/ICASSP48485.2024.10446678.
- [97] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. CREPE: A convolutional representation for pitch estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165, Calgary, Canada, 2018. doi: 10.1109/ICASSP.2018.8461329.
- [98] Youngmoo E. Kim. Singing voice analysis, synthesis, and modeling. In David Havelock, Sonoko Kuwano, and Michael Vorländer, editors, *Handbook of Signal Processing in Acoustics*. Springer, New York, NY, 2008. doi: 10.1007/978-0-387-30441-0_23.

Bibliography

- [99] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.
- [100] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual, 2021.
- [101] Zhifeng Kong, Arushi Goel, Rohan Badlani, Wei Ping, Rafael Valle, and Bryan Catanzaro. Audio flamingo: A novel audio language model with few-shot learning and dialogue abilities. In *Proceedings of the International Conference on Machine Learning*, volume 235, pages 25125–25148, 2024.
- [102] William H. Kruskal and W. Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952. doi: 10.1080/01621459.1952.10483441.
- [103] Timo I. Laakso, Vesa Välimäki, Matti Karjalainen, and Unto K. Laine. Splitting the unit delay — Tools for fractional delay filter design. *IEEE Signal Processing Magazine*, 13(1):30–60, 1996. doi: 10.1109/79.482137.
- [104] Jin Woo Lee, Jaehyun Park, Min Jun Choi, and Kyogu Lee. Differentiable modal synthesis for physical modeling of planar string sound and motion simulation. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, BC, Canada, 2024.
- [105] Sungho Lee, Hyeong-Seok Choi, and Kyogu Lee. Differentiable artificial reverberation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2541–2556, 2022.
- [106] Xinyu Li, Venkata Chebiyyam, and Katrin Kirchhoff. Speech audio super-resolution for speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, pages 3416–3420, Graz, Austria, 2019. doi: 10.21437/interspeech.2019-3043.
- [107] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger B. Dannenberg, Ruiibo Liu, Wenhua Chen, Gus Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike Guo, and Jie Fu. MERT: Acoustic music understanding model with large-scale self-supervised training. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Vienna, Austria, 2024.
- [108] Justin London. What is musical tempo? In Clemens Wöllner and Justin London, editors, *Performing Time: Synchrony and Temporal Flow in Music and Dance*. Oxford University Press, 2023. ISBN 9780192896254.
- [109] Franz Loosen. Tuning of diatonic scales by violinists, pianists, and nonmusicians. *Perception & Psychophysics*, 56:221–226, 1994.
- [110] Jordan Louviere, Terry Flynn, and Anthony Marley. *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press, 2015. doi: 10.1017/CBO9781107337855.
- [111] Yin-Jyun Luo, Ming-Tso Chen, Tai-Shih Chi, and Li Su. Singing voice correction using canonical time warping. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018. doi: 10.1109/ICASSP.2018.8461280.
- [112] Ben Maman, Johannes Zeitler, Meinard Müller, and Amit H. Bermano. Multi-aspect conditioning for diffusion-based music synthesis: Enhancing realism and acoustic control. *IEEE Transactions on Audio, Speech, and Language Processing*, 33:68–81, 2025. doi: 10.1109/TASLP.2024.3507553.

- [113] Raja Marjieh, Peter M. C. Harrison, Harin Lee, Fotini Deligiannaki, and Nori Jacoby. Timbral effects on consonance disentangle psychoacoustic mechanisms and suggest perceptual origins for musical scales. *Nature Communications*, 15:1482, 2024. doi: 10.1038/s41467-024-45812-z.
- [114] Naotake Masuda and Daisuke Saito. Improving semi-supervised differentiable synthesizer sound matching for practical applications. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:863–875, 2023. doi: 10.1109/TASLP.2023.3237161.
- [115] Matthias Mauch and Simon Dixon. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, Florence, Italy, 2014.
- [116] Matthias Mauch, Klaus Frieler, and Simon Dixon. Intonation in unaccompanied singing: Accuracy, drift, and a model of reference pitch memory. *Journal of the Acoustical Society of America*, 136(1):401–411, 2014. doi: 10.1121/1.4881915. URL <https://doi.org/10.1121/1.4881915>.
- [117] Maranda McBride, Phuong Tran, Tomasz Letowski, and Rafael Patrick. The effect of bone conduction microphone locations on speech intelligibility and sound quality. *Applied Ergonomics*, 42(3):495–502, 2011.
- [118] Josh H. McDermott, Alan F. Schultz, Eduardo A. Undurraga, and Ricardo A. Godoy. Indifference to dissonance in native Amazonians reveals cultural variation in music perception. *Nature*, 535(7613):547–550, 2016. doi: 10.1038/nature18635.
- [119] M. E. McIntyre and J. Woodhouse. On the fundamentals of bowed string dynamics. *Acustica*, 43:93–108, 1979.
- [120] Peter Meier, Simon Schwär, Sebastian Rosenzweig, and Meinard Müller. Real-time MIR algorithms for music-reactive game world generation. In *Proceedings for the Workshop für Innovative Computerbasierte Musikinterfaces (ICMI)*, Darmstadt, Germany, 2022. doi: 10.18420/muc2022-mci-ws03-225.
- [121] Peter Meier, Simon Schwär, Gerhard Krump, and Meinard Müller. Evaluating real-time pitch estimation algorithms for creative music game interaction. In *INFORMATIK 2023 – Designing Futures: Zukünfte gestalten*, pages 873–882. Gesellschaft für Informatik e.V., Bonn, Germany, 2023. doi: 10.18420/inf2023_97.
- [122] Peter Meier, Simon Schwär, Gerhard Krump, and Meinard Müller. Real-time pitch estimation for creative music game interaction. In *Proceedings of the Deutsche Jahrestagung für Akustik (DAGA)*, pages 1346–1349, Hamburg, Germany, 2023.
- [123] Peter Meier, Simon Schwär, and Meinard Müller. A real-time approach for estimating pulse tracking parameters for beat-synchronous audio effects. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 314–321, Guildford, Surrey, UK, 2024.
- [124] Peter Meier, Meinard Müller, and Stefan Balke. Analyzing pitch estimation accuracy in cross-talk scenarios: A study with wind instruments. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 3–10, Graz, Austria, 2025.
- [125] Peter Meier, Sebastian Strahl, Simon Schwär, Meinard Müller, and Stefan Balke. Pitch estimation in real time: Revisiting SWIPE with causal windowing. In *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, pages 285–297, London, UK, 2025. doi: 10.5281/zenodo.17496630.

Bibliography

- [126] Andrew J. Milne. *A Computational Model of the Cognition of Tonality*. PhD thesis, City University London, London, UK, 2013.
- [127] Chris Mingard, Henry Rees, Guillermo Valle-Pérez, and Ard A. Louis. Deep neural networks have an inbuilt occam’s razor. *Nature Communications*, 16(220), 2025. doi: 10.1038/s41467-024-54813-x.
- [128] Yuki Mitsufuji, Giorgio Fabbro, Stefan Uhlich, Fabian-Robert Stöter, Alexandre Défossez, Minseok Kim, Woosung Choi, Chin-Yun Yu, and Kin-Wai Cheuk. Music Demixing Challenge 2021. *Frontiers in Signal Processing*, 1, 2022. doi: 10.3389/frsip.2021.808395.
- [129] Werner Mohrlock. Hermode Tuning. http://www.hermode.com/index_en.html, 2025. Accessed: 2025-11-11.
- [130] Brian C.J. Moore. *An Introduction to the Psychology of Hearing*. Brill Academic Publisher, 6th edition, 2013.
- [131] Meinard Müller. *Fundamentals of Music Processing – Using Python and Jupyter Notebooks*. Springer Verlag, 2nd edition, 2021. ISBN 978-3-030-69807-2. doi: 10.1007/978-3-030-69808-9.
- [132] Ahmed Mustafa, Nicola Pia, and Guillaume Fuchs. StyleMelGAN: An efficient high-fidelity adversarial vocoder with temporal adaptive normalization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toronto, Canada, 2021.
- [133] Aage R. Møller. Frequency analysis in the peripheral auditory system. In *Auditory Physiology*, pages 191–249. Academic Press, 1983. ISBN 978-0-12-503450-0.
- [134] Yoshitaka Nakajima, Hideki Kashioka, Kiyohiro Shikano, and Nick Campbell. Non-audible murmur recognition input interface using stethoscopic microphone attached to the skin. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 708–711, 2003. doi: 10.1109/ICASSP.2003.1200069.
- [135] Tomoyasu Nakano, Masataka Goto, and Yuzuru Hiraga. An automatic singing skill evaluation method for unknown melodies using pitch interval accuracy and vibrato features. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1706–1709, Pittsburgh, PA, USA, 2006.
- [136] Shahan Nercessian, Andy Sarroff, and Kurt James Werner. Lightweight and interpretable neural modeling of an audio distortion effect using hyperconditioned differentiable biquads. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 890–894, Toronto, ON, Canada, 2021. doi: 10.1109/ICASSP39728.2021.9413996.
- [137] Michael Oehler and Christoph Reuter. Dynamic excitation impulse modification as a foundation of a synthesis and analysis system for wind instrument sounds. In *Mathematics and Computation in Music (MCM)*, pages 189–197, Berlin, Germany, 2009.
- [138] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, Long Beach, CA, USA, 2017.
- [139] Alan V. Oppenheim, Alan S. Willsky, and Hamid Nawab. *Signals and Systems*. Prentice Hall, 1996.

- [140] Alan V. Oppenheim, Ronald W. Schaffer, and John R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1999.
- [141] Makoto Otani, Tatsuya Hirahara, and Seiji Adachi. Numerical simulation of sound originated from the vocal tract in soft neck tissues. *The Journal of the Acoustical Society of America*, 120(5):3352–3352, 2006. doi: 10.1121/1.4781428.
- [142] Yigitcan Özer, Simon Schwär, Viora Arifi-Müller, Jeremy Lawrence, Emre Sen, and Meinard Müller. Piano Concerto Dataset (PCD): A multitrack dataset of piano concertos. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 6(1):75–88, 2023. doi: 10.5334/tismir.160.
- [143] Yigitcan Özer, Leo Brütting, Simon Schwär, and Meinard Müller. libsoni: A Python toolbox for sonifying music annotations and feature representations. *Journal of Open Source Software (JOSS)*, 9(96):06524:1–6, 2024. doi: 10.21105/joss.06524.
- [144] Yigitcan Özer, Simon Schwär, and Meinard Müller. Piano concerto accompaniment creation. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, San Francisco, CA, USA, 2024.
- [145] Changhao Pan, Dongyu Yao, Yu Zhang, Wenxiang Guo, Jingyu Lu, Zhiyuan Zhu, and Zhou Zhao. Synthetic singers: A review of deep-learning-based singing voice synthesis approaches. *CoRR*, abs/2601.13910, 2026.
- [146] Etienne Parizet. Paired comparison listening tests and circular error rates. *Acta Acustica united with Acustica*, 88:594–598, 2002.
- [147] Richard Parncutt and Graham Hair. A psychocultural theory of musical interval: Bye Bye Pythagoras. *Music Perception*, 35(4):475–501, 2018. doi: 10.1525/mp.2018.35.4.475.
- [148] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, IRCAM, Paris, France, 2004.
- [149] Christopher J. Plack, Andrew J. Oxenham, Arthur N. Popper, and Richard R. Fay, editors. *Pitch: Neural Coding and Perception*, volume 24 of *Springer Handbook of Auditory Research*. Springer, New York, NY, USA, 2005. doi: 10.1007/0-387-28958-5.
- [150] Reinier Plomp and Willem Johannes Maria Levelt. Tonal consonance and critical bandwidth. *Journal of the Acoustical Society of America*, 38(4):548–560, 1965.
- [151] K. M. M. Prabhu. *Window functions and their applications in signal processing*. CRC Press/Taylor & Francis, Boca Raton, FL, USA, 2014. ISBN 978-1-4665-1583-3.
- [152] Eric Prame. Vibrato extent and intonation in professional Western lyric singing. *The Journal of the Acoustical Society of America (JASA)*, 102(1):616–621, 1997.
- [153] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. Technical report, OpenAI, 2022.
- [154] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. MIR_EVAL: A transparent implementation of common MIR metrics. In *Proceedings of the*

Bibliography

- International Society for Music Information Retrieval Conference (ISMIR)*, pages 367–372, Taipei, Taiwan, 2014.
- [155] Lenny Renault, Rémi Mignot, and Axel Roebel. Differentiable piano model for MIDI-to-audio performance synthesis. In *Proceedings of the 25th International Conference on Digital Audio Effects (DAFx)*, pages 232–239, Vienna, Austria, 2022.
- [156] Bruno H. Repp. Diversity and commonality in music performance: An analysis of timing microstructure in Schumann’s “Träumerei”. *The Journal of the Acoustical Society of America*, 92(5):2546–2568, 1992. doi: 10.1121/1.404425.
- [157] Christoph Reuter. Formant positions of musical instruments and recommendations for ensemble scoring in orchestration treatises. In *Proceedings of the Czech Conference on Acoustics*, pages 127–130, Prague, 1995.
- [158] Gaël Richard, Pierre Chouteau, and Bernardo Torres. A fully differentiable model for unsupervised singing voice separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seoul, South Korea, 2024. doi: 10.1109/ICASSP48485.2024.10447244.
- [159] Gaël Richard, Vincent Lostanlen, Yi-Hsuan Yang, and Meinard Müller. Model-based deep learning for music information research: Leveraging diverse knowledge sources to enhance explainability, controllability, and resource efficiency. *IEEE Signal Processing Magazine*, 41(6):51–59, 2024. doi: 10.1109/MSP.2024.3415569.
- [160] Alain Riou, Stefan Lattner, Gaëtan Hadjeres, and Geoffroy Peeters. PESTO: Pitch estimation with self-supervised transposition-equivariant objective. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 535–544, Milano, Italy, 2023.
- [161] Curtis Roads. Introduction to pulsar synthesis. *The Journal of the Acoustical Society of America*, 109 (5_Supplement):2401–2401, 2001.
- [162] Xavier Rodet. Time-domain formant-wave-function synthesis. *Computer Music Journal*, 8(3):9–14, 1984.
- [163] Sebastian Rosenzweig, Helena Cuesta, Christof Weiß, Frank Scherbaum, Emilia Gómez, and Meinard Müller. Dagstuhl ChoirSet: A multitrack dataset for MIR research on choral singing. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 3(1):98–110, 2020. doi: 10.5334/tismir.48.
- [164] Sebastian Rosenzweig, Simon Schwär, Jonathan Driedger, and Meinard Müller. Adaptive pitch-shifting with applications to intonation adjustment in a cappella recordings. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 121–128, Vienna, Austria, 2021.
- [165] Sebastian Rosenzweig, Simon Schwär, and Meinard Müller. libf0: A python library for fundamental frequency estimation. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022.
- [166] Sebastian Rosenzweig, Frank Scherbaum, and Meinard Müller. Computer-assisted analysis of field recordings: A case study of Georgian funeral songs. *ACM Journal on Computing and Cultural Heritage (JOCCH)*, 16(1): 1–16, 2023. doi: doi.org/10.1145/3551645.
- [167] Justin Salamon and Emilia Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, 2012. doi: 10.1109/TASL.2012.2188515.

- [168] Justin Salamon, Emilia Gómez, Daniel P. W. Ellis, and Gaël Richard. Melody extraction from polyphonic music signals: Approaches, applications, and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134, 2014. doi: 10.1109/MSP.2013.2271648.
- [169] Justin Salamon, Rachel M. Bittner, Jordi Bonada, Juan José Bosch Vicente, Emilia Gómez, and Juan P. Bello. An analysis/synthesis framework for automatic f0 annotation of multitrack datasets. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, pages 71–78, Suzhou, China, 2017. doi: 10.5281/zenodo.1415588.
- [170] Frank Scherbaum. On the benefit of larynx-microphone field recordings for the documentation and analysis of polyphonic vocal music. *Proceedings of the International Workshop Folk Music Analysis*, pages 80–87, 2016.
- [171] Frank Scherbaum and Nana Mzhavanadze. Svan funeral dirges (Zär): Musical acoustical analysis of a new collection of field recordings. *Musicologist*, 4:138–167, 2020. doi: 10.33906/musicologist.782094.
- [172] Frank Scherbaum, Nana Mzhavanadze, Sebastian Rosenzweig, and Meinard Müller. Tuning systems of traditional georgian singing determined from a new corpus of field recordings. *Musicologist*, 6(2):142–168, 2022.
- [173] Konstantin Schmidt and Bernd Edler. Blind bandwidth extension of speech based on LPCNet. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 426–430, 2021.
- [174] Michael Schoeffler, Sarah Bartoschek, Fabian-Robert Stöter, Marlene Roess, Susanne Westphal, Bernd Edler, and Jürgen Herre. webMUSHRA—a comprehensive framework for web-based listening tests. *Journal of Open Research Software*, 6(1), 2018.
- [175] Kilian Schulze-Forster, Gaël Richard, Liam Kelley, Clement S. J. Doire, and Roland Badeau. Unsupervised music source separation using differentiable parametric source models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1276–1289, 2023. doi: 10.1109/TASLP.2023.3252272.
- [176] Simon Schwär and Meinard Müller. Multi-scale spectral loss revisited. *IEEE Signal Processing Letters*, 30: 1712–1716, 2023. doi: 10.1109/LSP.2023.3333205.
- [177] Simon Schwär and Meinard Müller. Multi-voice intonation adaptation via gradient descent. *IEEE Transactions on Audio, Speech and Language Processing (TASLPRO)*, 34:2491–2503, 2026. doi: 10.1109/TASLPRO.2026.3675812.
- [178] Simon Schwär, Sebastian Rosenzweig, and Meinard Müller. A differentiable cost measure for intonation processing in polyphonic music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 626–633, Online, 2021.
- [179] Simon Schwär, Meinard Müller, and Sebastian J. Schlecht. A variational Y-Autoencoder for disentangling gesture and material of interaction sounds. In *AES 4th International Conference on Audio for Virtual and Augmented Reality (AES AVAR)*, Redmond, WA, USA, 2022.
- [180] Simon Schwär, Meinard Müller, and Sebastian J. Schlecht. Modifying partials for minimum-roughness sound synthesis. In *Proceedings of the 3rd International Conference on Timbre*, pages 130 – 134, Thessaloniki, Greece, 2023.

Bibliography

- [181] Simon Schwär, Michael Krause, Michael Fast, Sebastian Rosenzweig, Frank Scherbaum, and Meinard Müller. A dataset of larynx microphone recordings for singing voice reconstruction. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 7(1):30–43, 2024. doi: 10.5334/tismir.166.
- [182] Simon Schwär, Stefan Balke, and Meinard Müller. Measuring sensory dissonance in multi-track music recordings: A case study with wind quartets. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 117–124, Daejeon, South Korea, 2025.
- [183] Simon Schwär, Christian Dittmar, Stefan Balke, and Meinard Müller. Differentiable pulsetable synthesis for wind instrument modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 14792–14796, Barcelona, Spain, 2026. doi: 10.1109/ICASSP55912.2026.11462505.
- [184] Carl E. Seashore. The natural history of the vibrato. *Proceedings of the National Academy of Sciences of the United States of America*, 17(12):623–626, 1931.
- [185] Richard Seaton, David Sharp, and Dennis Pim. Pitch drift in a capella choral singing: the outcomes from an international survey. *Proceedings of Institute of Acoustics*, 36(3):312–319, 2014.
- [186] Xavier Serra and Julius Smith III. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24, 1990.
- [187] Joan Serrà, Santiago Pascual, Jordi Pons, R. Oguz Araz, and Davide Scaini. Universal speech enhancement with score-based diffusion. *Computing Research Repository (CoRR)*, abs/2206.03065, 2022.
- [188] William A. Sethares. Local consonance and the relationship between timbre and scale. *Journal of the Acoustical Society of America*, 94(3):1218–1228, 1993.
- [189] William A. Sethares. Adaptive tunings for musical scales. *The Journal of the Acoustical Society of America*, 96, 1994.
- [190] William A. Sethares. *Tuning, Timbre, Spectrum, Scale*. Springer, London, 1998. ISBN 1-85233-797-4.
- [191] Siyuan Shan, Lamtharn Hantrakul, Jitong Chen, Matt Avent, and David Trevelyan. Differentiable wavetable synthesis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4598–4602, Singapore, 2022.
- [192] Shota Shimizu, Makoto Otani, and Tatsuya Hirahara. Frequency characteristics of several non-audible murmur (NAM) microphones. *Acoustical Science and Technology*, 30(2):139–142, 2009.
- [193] Kai Siedenburg. Specifying the perceptual relevance of onset transients for musical instrument identification. *The Journal of the Acoustical Society of America*, 145(2):1078, 2019. doi: 10.1121/1.5091778.
- [194] Kai Siedenburg, Charalampos Saitis, Stephen McAdams, Arthur N. Popper, and Richard R. Fay, editors. *Timbre: Acoustics, Perception, and Cognition*, volume 69 of *Springer Handbook of Auditory Research*. Springer, Cham, Switzerland, 2019. ISBN 978-3-030-14831-7. doi: 10.1007/978-3-030-14832-4.
- [195] Julius O. Smith and Xavier Serra. PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation. In *Proceedings of the International Computer Music Conference (ICMC)*. Computer Music Association, 1987.

- [196] Julius Orion Smith. *Physical audio signal processing: For virtual musical instruments and audio effects*. W3K publishing, 2010.
- [197] Julius Orion Smith. *Spectral Audio Signal Processing*. W3K Publishing, <http://books.w3k.org/>, 2011. ISBN ISBN 978-0-9745607-3-1.
- [198] Karolin Stange, Christoph Wick, and Haye Hinrichsen. Playing music in just intonation: A dynamically adaptive tuning scheme. *Computer Music Journal*, 42(3):47–62, 2018. doi: 10.1162/comj_a_00478.
- [199] Christian J. Steinmetz and Joshua D. Reiss. auraloss: Audio focused loss functions in PyTorch. In *Proceedings of the Digital Music Research Network One-day Workshop (DMRN+15)*, 2020.
- [200] Christian J. Steinmetz and Joshua D. Reiss. ST-ITO: Controlling audio effects for style transfer with inference-time optimization. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 1–8, San Francisco, CA, USA, 2024.
- [201] Lawrence Stewart, Francis R. Bach, Quentin Berthet, and Jean-Philippe Vert. Regression as classification: Influence of task formulation on neural network features. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 11563–11582, Valencia, Spain, 2023.
- [202] Frieder Stolzenburg. Harmony perception by periodicity detection. *Journal of Mathematics and Music*, 9(3): 215–238, 2015. doi: 10.1080/17459737.2015.1034826.
- [203] Sebastian Strahl and Meinard Müller. dYIN and dSWIPE: Differentiable variants of classical fundamental frequency estimators. *IEEE Transactions on Audio, Speech, and Language Processing*, 33:2622–2633, 2025. doi: 10.1109/TASLPRO.2025.3581119.
- [204] Alex Stupakov, Evan Hanusa, Jeff Bilmes, and Dieter Fox. COSINE - a corpus of multi-party conversational speech in noisy environments. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4153–4156, 2009. doi: 10.1109/ICASSP.2009.4960543.
- [205] Johan Sundberg. The acoustics of the singing voice. *The Journal of the Acoustical Society of America*, 61(3): 726–744, 1977. doi: 10.1121/1.381336.
- [206] Johan Sundberg. *The Science of the Singing Voice*. Northern Illinois University Press, 1987. ISBN 9780875805429.
- [207] Jack E. Taylor, Guillaume A. Rousselet, Christoph Scheepers, and Sara C. Sereno. Rating norms should be calculated from cumulative link mixed effects models. *Behavior Research Methods*, 55:2175–2196, 2023. doi: 10.3758/s13428-022-01814-7.
- [208] James Tenney. *A History of 'Consonance' and 'Dissonance'*. Excelsior Music Publishing Company, 1988. ISBN 0-935016-99-6.
- [209] Ernst Terhardt. On the perception of periodic sound fluctuations (roughness). *Acta Acustica united with Acustica*, 30(4):201–213, 1974.
- [210] Sten Ternström, Peter Pabon, and Maria Södersten. The Voice Range Profile: Its function, applications, pitfalls and potential. *Acta Acustica united with Acustica*, 102(2), 2016. doi: 10.3813/AAA.918943.

Bibliography

- [211] William F. Thompson and Brigitte Robitaille. Can composers express emotions through music? *Empirical Studies of the Arts*, 10(1):79–89, 1992.
- [212] Neil Todd. A model of expressive timing in tonal music. *Music Perception*, 3(1):33–57, 1985. doi: 10.2307/40285321.
- [213] Maciej Tomczak, Min Susan Li, and Massimiliano Di Luca. Virtuoso Strings: A dataset of string ensemble recordings and onset annotations for timing analysis. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Milano, Italy, 2023.
- [214] Joseph Turian and Max Henry. I’m sorry for your loss: Spectrally-based audio distances are bad at pitch. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), I Can’t Believe It’s Not Better Workshop*, Vancouver, Canada, 2020.
- [215] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. In *Proceedings of the ISCA Speech Synthesis Workshop*, Sunnyvale, USA, 2016.
- [216] Pantelis N. Vassilakis and Roger A. Kendall. Psychoacoustic and cognitive aspects of auditory roughness: definitions, models, and applications. In *Proceedings of Human Vision and Electronic Imaging XV*, volume 7527, 2010.
- [217] Julián Villegas and Michael Cohen. Roughness minimization through automatic intonation adjustments. *Journal of New Music Research*, 39:75–92, 2010.
- [218] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006. doi: 10.1109/TSA.2005.858005.
- [219] Emmanuel Vincent, Tuomas Virtanen, and Sharon Gannot, editors. *Audio Source Separation and Speech Enhancement*. Wiley, 1st edition, 2018. ISBN 978-1-119-27989-1.
- [220] Sanna Wager, George Tzanetakis, Stefan Sullivan, Cheng-i Wang, John Shimmin, Minje Kim, and Perry Cook. Intonation: A dataset of quality vocal performances refined by spectral clustering on pitch congruence. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 476–480, Brighton, UK, 2019.
- [221] Sanna Wager, George Tzanetakis, Cheng-i Wang, and Minje Kim. Deep Autotuner: A pitch correcting network for singing performances. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 246–250, Barcelona, Spain, 2020.
- [222] Xin Wang, Shinji Takaki, and Junichi Yamagishi. Neural source-filter waveform models for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28: 402–415, 2020. doi: 10.1109/TASLP.2019.2956145.
- [223] Frank Wefers. *Partitioned convolution algorithms for real-time auralization*. PhD thesis, RWTH Aachen University, Aachen, Germany, 2015.

- [224] Weixing Wei, Peilin Li, Yi Yu, and Wei Li. HarmoF0: Logarithmic scale dilated convolution for pitch estimation. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, Taipei, Taiwan, 2022. doi: 10.1109/ICME52920.2022.9858935.
- [225] Christof Weiß and Geoffroy Peeters. Comparing deep models and evaluation strategies for multi-pitch estimation in music recordings. *IEEE/ACM Transactions on Audio, Speech & Language Processing*, 30: 2814–2827, 2022. doi: 10.1109/TASLP.2022.3200547.
- [226] Christof Weiß, Sebastian J. Schlecht, Sebastian Rosenzweig, and Meinard Müller. Towards measuring intonation quality of choir recordings: A case study on Bruckner’s Locus Iste. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 276–283, Delft, The Netherlands, 2019.
- [227] Sheng-Ying Isabella Weng and Erkki Huovinen. Expressive semitones: Music students’ perceptual preferences for melodic intonation on the violin. *Musicae Scientiae*, 28(3):558–572, 2024. doi: 10.1177/10298649231225777.
- [228] Nils Werner, Stefan Balke, Fabian-Robert Stöter, Meinard Müller, and Bernd Edler. trackswitch.js: A versatile web-based audio player for presenting scientific results. In *Proceedings of the Web Audio Conference (WAC)*, London, UK, 2017.
- [229] Julia Wilkins, Prem Seetharaman, Alison Wahl, and Bryan Pardo. VocalSet: A singing voice dataset. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 468–474, Paris, France, 2018.
- [230] Da-Yi Wu, Wen-Yi Hsiao, Fu-Rong Yang, Oscar Friedman, Warren Jackson, Scott Bruzenak, Yi-Wen Liu, and Yi-Hsuan Yang. DDSP-based singing vocoders: A new subtractive-based synthesizer and a comprehensive evaluation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 76–83, Bengaluru, India, 2022.
- [231] Yusong Wu, Ethan Manilow, Yi Deng, Rigel Swavely, Kyle Kastner, Tim Cooijmans, Aaron Courville, Cheng-Zhi Anna Huang, and Jesse Engel. MIDI-DDSP: Detailed control of musical performance via hierarchical modeling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.
- [232] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6199–6203, 2020. doi: 10.1109/ICASSP40776.2020.9053795.
- [233] Ran Yan, Ghazal Jessani, Elizabeth S. Spelke, Peter de Villiers, Jill de Villiers, and Samuel A. Mehr. Across demographics and recent history, most parents sing to their infants and toddlers daily. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 376(1840):20210089, 2021. doi: 10.1098/rstb.2021.0089.
- [234] Luwei Yang, Mi Tian, and Elaine Chew. Vibrato characteristics and frequency histogram envelopes in Beijing opera singing. In *International Workshop on Folk Music Analysis (FMA)*, pages 139–140, 2015.

Bibliography

- [235] Sangeon Yong and Juhan Nam. Singing expression transfer from one voice to another for a given song. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018. doi: 10.1109/ICASSP.2018.8461793.
- [236] Chin-Yun Yu and György Fazekas. Singing voice synthesis using differentiable LPC and glottal-flow-inspired wavetables. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 667–675, Milano, Italy, 2023.
- [237] Johannes Zeitler, Christof Weiß, , Vlori Arifi-Müller, and Meinard Müller. BPSD: A coherent multi-version dataset for analyzing the first movements of Beethoven’s piano sonatas. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 7(1):195–212, 2024.
- [238] Yu Zhang, Wenxiang Guo, Changhao Pan, Dongyu Yao, Zhiyuan Zhu, Ziyue Jiang, Yuhao Wang, Tao Jin, and Zhou Zhao. TC-Singer 2: Customizable multilingual zero-shot singing voice synthesis. *CoRR*, abs/2505.14910, 2025.
- [239] Xiaobin Zhuang, Huiran Yu, Weifeng Zhao, Tao Jiang, and Peng Hu. Karatuner: Towards end-to-end natural pitch correction for singing voice in karaoke. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Singapore, 2022. doi: 10.1109/ICASSP43922.2022.9746372.
- [240] Le Zhuo, Ruibin Yuan, Jiahao Pan, Yinghao Ma, Yizhi LI, Ge Zhang, Si Liu, Roger Dannenberg, Jie Fu, Chenghua Lin, et al. LyricWhiz: Robust multilingual zero-shot lyrics transcription by whispering to ChatGPT. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 343–351, Milano, Italy, 2023.
- [241] Eberhard Zwicker. Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *Journal of the Acoustical Society of America*, 33(2), 1961.