The Art and Science of the Electroacoustic Steelpan

by

Colin Malloy
B.S., Whitman College, 2008
B.A., Oregon State University, 2011
M.M., Southern Oregon University, 2015
M.Mus, University of Victoria, 2018

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Interdisciplinary Studies

The Art and Science of the Electroacoustic Steelpan

by

Colin Malloy
B.S., Whitman College, 2008
B.A., Oregon State University, 2011
M.M., Southern Oregon University, 2015
M.Mus, University of Victoria, 2018

Supervisory Committee

Dr. George Tzanetakis, Co-Supervisor
(Department of Computer Science)

Kirk McNally, Co-Supervisor
(Department of Music)

Dr. W. Andrew Schloss, Departmental Member
(Department of Music)

ABSTRACT

Hailing from the Caribbean nation of Trinidad and Tobago, the steelpan is one of the most significant acoustic instruments invented in the 20$^{\text{th}}$ century. Despite enjoying international popularity and attention from acoustics researchers, the steelpan has not received as much attention as an electroacoustic instrument. This dissertation presents my work in the field of music technology as it relates to the steelpan. The contributions presented here can broadly be put into three categories: acoustics, digital signal processing, and electroacoustic steelpan composition/performance. Relating to the acoustics of the steelpan, I present two research projects. First is a computation analysis that characterizes the timbral differences between mallets on the tenor steelpan. The other is a steelpan-specific audio dataset and pitch detection system that outperforms the available state-of-the-art methods. I then present two audio effects plugins I developed. Realstretch is a realtime time-stretcher—an audio effect that had previously never been implemented as a realtime effect. HarmonEQ is an audio equalizer that reframes how the equalizer's controls in terms of musical notes and chords rather than abstract frequency. HarmonEQ's change in control scheme allows it to be paired with a chord detection system so the equalizer can update its own settings based on the harmony of the incoming audio signal. Next I discuss approaches to using the steelpan as an electroacoustic instrument. The practical considerations of electroacoustic performance are discussed through summary and analysis of three new solo compositions for electroacoustics steelpan that were premiered on March 26, 2022 as part my of Ph.D. recital. These various contributions combine to progress the art and science of the electroacoustic steelpan.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

My family, for supporting me in the this long endeavor.

Charmaine, for your love, support, and willingness to put up me while writing this.

Kirk, George, and Andy, for your mentoring, encouragement, and insight.

University of Victoria, for funding me with a doctoral fellowship.

Dan, Jordie, Keon, Terry, and my other collaborators, for all of the help you've provided along the way.

DEDICATION

For advancement of the steelpan. Beat pan.

Chapter 1

Introduction

The steelpan, also commonly referred to as the steel drum, is a pitched percussion instrument known for its association with Caribbean dance music. It was invented in the early 20^{th} century in Trinidad and Tobago, but its pre-history dates back to the 19^{th} century colonial era of Trinidad. The steelpan has become the national instrument of Trinidad and Tobago and a point of national culture and pride. The main goal of this dissertation is to advance the state of steelpan analysis, audio signal processing for the steelpan, and electroacoustic performance with the steelpan. This includes new research into the acoustical properties of the steelpan, advancements in steelpan pitch detection, audio signal processing tools, and compositions for the electroacoustic steelpan.

The steelpan is often considered to be the most significant acoustic instrument invented in the 20^{th} century, but is still not well understood. Furthermore, it is an acoustically complex instrument with physically coupled notes that have various vibrational modes. Improving our understanding of the acoustics of the steelpan will help builders design better instruments for performers, help performers understand the intricacies of their instrument, and help composers write pieces better suited for the instrument. Further benefits include the study and development of related instruments. The handpan and tongue drums are newer instruments derived from the steelpan. While they have distinctive sounds, they do share some acoustical properties. Benefits made to the science of the steelpan will further benefit the study of these instruments as well.

The steelpan, like many percussion instruments, poses a difficult challenge in recording and processing its audio. With sharp transients and a complex harmonic structure, it can be difficult to capture its sound as the performer hears it. The

overlapping sustain of notes also makes more creative audio processing challenging since there is no practical way to isolate the audio for individual notes—even when playing monophonic music. Developing new creative processing chains for the steelpan opens new timbral possibilities in both recording and performance contexts. The electroacoustic steelpan repertoire is also very limited. The work presented in this dissertation is an effort to further this nascent field and inspire others to contribute to it as well. This work provides a pathway to make it easier for others to explore the possibilities of the electroacoustic steelpan and expand the range of our artistic understanding of the instrument.

In the effort to develop these workflows, I needed to develop new signal processing tools as well. Although I designed these tools for myself to use while performing and recording my steelpans, they will be of benefit to countless other musical contexts as well. My primary contributions in this area are *HarmonEQ* and *Realstretch*. *HarmonEQ* reframes the traditional parametric equalizer in terms of musical harmony rather than abstract frequency. Realstretch is a time-stretching audio effect that operates on live, realtime audio, making it useable in live performance settings.

1.1 Document Overview

This dissertation is interdisciplinary in nature, reflecting the degree program it is written as part of: Interdisciplinary Studies in Music and Computer Science. All of the subject matter falls under the broad category of "music technology," but can broadly be categorized as research in acoustics, computer science, and music composition/performance. There is no singular contribution that is the main focus of this dissertation. Instead, the various research projects work together to improve our understanding of the steelpan as an electroacoustic instrument.

Chapters 2 and 3 contain new work based on the acoustics of the steelpan. Chapter 2 presents an acoustical analysis of steelpan mallets. Audio feature extraction was performed on five different steelpan mallets. By comparing the results, we can see what differentiates the timbres of the various mallets. In Chapter 3 I present a dataset of steelpan audio and my work in steelpan pitch detection using machine learning. A new architecture for pitch detection is proposed that improves on baseline state of the art methods while also reducing latency.

Chapters 4 and 5 include the new signal processing tools I developed during the course of my thesis research. Chapter 4 presents Realstretch, a realtime time-stretcher

based on the Paulstretch algorithm. It uses a novel buffer method that allows it to continuous stretch incoming audio while still maintain temporal coherence with the input. Chapter 5 presents HarmonEQ, a chord-based equalizer. HarmonEQ eschews the traditional frequency controls in equalizers in favor of controls based on musical chords. This changes the way in which a user interacts with the equalizer. Another important feature made possible through this change is the inclusion of an automatic chord detection mode that analyzes the incoming audio in real time and dynamically adjusts the equalizer to match the input audio.

Chapters 6-9 concern the use of the electroacoustic steelpan in composition and performance. Chapter 6 begins with an overview of the extant repertoire for the electroacoustic steelpan. It then presents an overview of my approach to electroacoustic performance on the steelpan. Prior to writing this document, I presented a recital of compositions for electroacoustic steelpan on March 26, 2022. The remaining chapters present original electroacoustic compositions from that program. Chapter 7 covers *Crushed Atmos*. This piece explore the timbral possibilities resulting from using eight parallel instances of a bitcrusher audio effect. Chapter 8 presents *Steelpanopticon*, which explores the use of distortion with the steelpan. It puts the steelpan, a literal piece of heavy metal, in a heavy metal musical context. Using such heavy distortion effects with the steelpan present unique challenges that needed to be addressed through creative signal processing work. Chapter 9 covers *(De/Con)struction in Steel and Electricity*. This piece was originally written for steelband and electronic percussion ensemble. I create a new solo version of the piece for the recital. This adaptation necessitated creating a new solo part, constructing a backing track to cover the original steelband and electronic parts, and doing sound design for both the backing track and the live steelpan. These compositions present different ways to approach electroacoustic steelpan composition and add new entries to the repertoire.

1.2 Contributions to the Art and Science of the Electroacoustic Steelpan

In Chapter 2, I apply music information retrieval techniques to characterize the timbral qualities of five different mallets on a tenor steelpan. Most acoustics research on the steelpan is focused solely on the vibrational qualities of the steelpan and this research adds significant contributions to better the understand of the complete

mallet-instrument system. This expands our knowledge of the steelpan and how it is analyzed acoustically.

Chapter 3 contains two primary contributions. First is SASS-E, the Steelpan Audio Sample Set for Evaluation. This is an audio dataset containing over 13,000 audio samples of notes from three different tenor steelpans. This audio dataset is then used to train Steelpan-Pitch, a machine learning pitch detection system that outperforms the baseline state-of-the-art pitch detection methods. Steelpan-Pitch shows that despite shortening the analysis window significantly and decreasing the size of the neural network, results can be improved by constricting the use case of the pitch detection system.

Chapter 4 makes one theoretical and one practical contribution. The theoretical contribution is the modification of the circular buffer to not always write the input to the buffer. Making the write process be triggered by a note onset and re-synchronizing the write pointer to the read pointer allows for time-stretching as a live, realtime audio effect. The practical contribution is the prototype, Realstretch. This audio effect utilizes the modified circular buffer to implement the Paulstretch time-stretching algorithm as an AU/VST plugin. Realstretch is the first-of-its-kind live time-stretching effect.

In Chapter 5, I introduce the concept of the harmonic equalizer. This concept is then implemented in the form of HarmonEQ, an AU/VST plugin. HarmonEQ is an equalizer whose controls are in terms of musical notes and harmony instead of abstract frequency. HarmonEQ also features an automatic chord detection mode that analyzes the incoming audio signal to estimate the current chord. It then updates its parameters to match the audio signal. This is a unique new feature for an audio equalizer.

The contributions presented in Chapter 6 relate to the electroacoustic steelpan in general. The first contribution is a repertoire list of all of the pieces for electroacoustic steelpan I have found in several years of searching. The next contribution is my approach to performing on the electroacoustic steelpan. This begins with how I incorporate guitar pedals, in particular the Line 6 Helix floorboard, into my processing chain and various methods of capturing the steelpan's sound. Then I present how I accomplish quadraphonic processing in Ableton Live—a digital audio workstation that only supports stereo tracks. In the future, the list of repertoire and my methodology can serve as a reference for composers and performers looking for information on methods of electroacoustic steelpan performance and composition.

Chapters 7-9 chronicle my practical contributions to the electroacoustic steelpan repertoire. Each chapter covers an original electroacoustic composition presented as part of my PhD recital on March 26, 2022. The pieces explore different ways to utilize the steelpan in electroacoustic performance. The repertoire list in Chapter 6 reveals that prior to my work, there were about 10 pre-existing electroacoustic works for steelpan. Considering this, the compositions presented here serve to significantly expand the nascent electroacoustic steelpan repertoire.

Chapter 2

Acoustical Analysis of the Timbral Characteristics of Different Steelpan Mallets

2.1 Introduction

Although the steelpan has been the focus of significant acoustical research (e.g., an entire chapter was dedicated to the instrument in Thomas Rossing's book, *Science of Percussion Instruments* [113]), there is still much to be studied regarding it. Most previous studies analyzing the acoustical properties of steelpans and other percussion instruments are concerned with the modes of vibration and sound radiation of the instruments. The priority has been the acoustical properties of the sounding object with much less attention given to the mallets used to activate the instrument. While it is important to understand these properties of the steelpan, this situation does ignore one half of the mallet-instrument system. This chapter presents analysis of the timbral effects five different mallets have on a steelpan.

2.1.1 Background and Rationale

Overview of the Steelpan

The steelpan, shown in Fig. 2.1, was invented and developed in Trinidad and Tobago in the 1930's and 40's. Its precursors were made from old frying pans and biscuit tins, but modern versions are made from 55-gallon oil drums. This gives the builder

increased control over the physical properties of the barrel. The family of steelpan instruments consists of many different instruments with voice ranges from soprano to bass. Since there is little standardization of the different steelpan types, each voice range includes many different note layouts and pitch ranges—often with major variations.

Figure 2.1: A low-C tenor steelpan.

The tenor steelpan is the highest voiced instrument consisting of a single pan. Several variations of the tenor steelpan exist, but the most common layout arranges the notes in the circle of fifths . This note layout is commonly referred to as the "fourths and fifths" or spiderweb layout. In North America, tenor steelpans typically have a range of C4-E6 while in Trinidad it is more common for tenor steelpans to have a range of D4-F♯6. These are referred to as "low C" and "high C"/"low D" tenor steelpans respectively. Other tenor steelpan note layouts with varying ranges and sometimes completely unrelated note placements exist as well, such as the Invader lead, thirds and fourths tenor, and the Aubrapan [76].

The next voices down from the tenor steelpan are the double tenor and double seconds steelpans. They are referred to as "double" steelpans because a single instru-

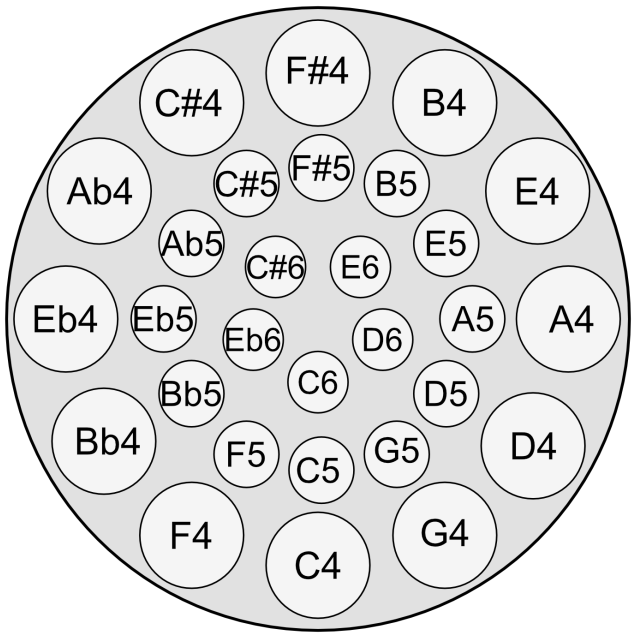


Figure 2.2: Low C tenor steelpan note layout.

ment consists of a pair of pans. Both instruments split the twelve chromatic pitch classes between the two pans, but in different ways. The double tenor's range can vary from F3-G3 at the bottom to A5-D6 at the top. The double seconds typically is voiced slightly lower than the double tenor ranging from E3-F♯3 at the bottom to B♭5-F♯6.

The guitar and cello steelpans come in many variations. Depending on the builder, these instruments can have double, triple, or quad setups. Although these instruments are similarly voiced, when there is a difference in range, guitar steelpans are the higher voiced of the two. The other main difference between the two is that guitar steelpans usually have significantly shorter skirts than cello steelpans. This has a significant effect on the timbre, making them brighter sounding.

Bass steelpans are the lowest voiced instruments in the steelpan family. Different configurations consist of between 6 and 12 pans made from full oil barrels for a single instrument. Outside of Trinidad, six bass (consisting of 6 pans) is the most common with a range of B♭1-E♭3.

The construction process for steelpans is complex. They usually begin as a 55-gallon oil drum, but high end instruments are also constructed from custom-made barrels. First, the bottom of the oil drum is sunk through hammering to create a

bowl. Then the builder hammers upward on the underside of the bowl to create small convex areas which correspond to the individual notes. An "outline" of each note is then scored into the metal to help acoustically isolate the notes. After the notes are shaped, the pan is heat treated and the notes are tuned. The skirt of the oil barrel is also cut to a suitable length for the instrument range—this varies from about 10 cm for tenor steelpans to nearly the full oil barrel for bass steelpans. Construction is typically finished by painting or chroming to prevent rust forming on the instrument [31], [115], [89], [37], [90], [91].

Acoustical Properties of the Steelpan

The acoustics of the steelpan are complex due to several factors. The notes have flat (or semi-flat) elliptical shapes [120], [114]. All of the notes of a single pan are also acoustically coupled since they share a vibrating surface. This coupling causes significant acoustic interference between notes. Striking a note activates nearby notes—especially those that are harmonically related. Due to this, the vibrational behavior of the steelpan is complicated and non-linear with significant inharmonic partials [1], [4], [2], [115], [3]. Tuners will typically tune a vibrational mode of a note to the second harmonic (an octave above the fundamental). Sometimes a second vibrational mode can be tuned to the third harmonic (an octave and a fifth above the fundamental), but, especially in the high range of the instrument, the higher vibrational modes will vibrate at unrelated partials [119].

The vibrational modes of various members of the steelpan family have been analyzed in many different ways. Hansen et al. analyzed the vibrational modes and coupling—both direct linear and nonlinear—on a double tenor steelpan [43]. [113] and [116] used TV holography to characterize the vibrational modes of tenor, double second, cello and bass steelpans. Recently, deep learning techniques have been adapted for acoustic analysis of the steelpan. The geometry of a tenor steelpan and modal analysis has also been performed using a 3D laser doppler vibrometer [77]. Hawley et al. used a convolutional neural network to analyze electronic speckle pattern interferometry images of a steelpan to track the vibrational modes of steelpans notes [45].

The acoustics of the steelpan family has also been investigated through numerical modelling. Finite element modeling consists of dividing a physical body into a finite number of discrete elements with associated material and geometric properties. Using

specialized software, it is possible to explore the acoustical properties of the model in ways that are not possible with physical instruments. However, this approach is computationally expensive and requires simplifications and assumptions to be made about the properties of the instrument modeled. Gay made models of tenor bass and bass steelpans using the finite element method. The tenor bass steelpan is an instrument voiced between the cello and bass steelpans [41]. The relationship between the shell geometry and note localization has also been studied through finite element modeling. The results from [19] suggests that the curvature of the steelpan's bowl allows for localized vibrational modes in the notes.

In most acoustic analyses of the steelpan, the vibrations are being driven using some kind of mechanism—often electromagnets. This does not improve our understanding of how the instrument behaves in playing situations, however. In a sound radiation study done by Copeland et al., an electromagnetic driver is used to measure sound radiation from various types of steelpans [27]. In their article, they note that the frequency response of the steelpan differs significantly between when it is electromagnetically driven and mallet activated. In a study by Kronman and Jansson, they use mallets to excite a steelpan and measure the average frequency response from several hits [55]. However, that study was published in 1988 and there were technical constraints of the time that prevent the level of analysis that is possible now. Muddeen et al. have also analyzed the steelpan using cepstral analysis [86]. This type of analysis allows for nuanced anechoic-style measurements without the need for an actual anechoic chamber.

Rationale

Many instruments have a default playing method and make occasional changes for special effects. For example brass players typically use only one mouthpiece in a concert (or at least for a single piece) and violinists rarely carry more than one bow on stage. Therefore, they typically make timbral changes by adjusting playing technique. While percussionists can also make timbral shifts through playing technique to some extent, they are more limited. More often, they rely on mallet choice for timbral shifts and regularly use multiple mallets in a single performance. It is also becoming increasingly common for composers to call for percussionists (including steelpannists) to utilize a wide variety of mallet types beyond what is considered "standard." What once were thought of as special effects mallets are becoming commonplace in perform-

ers' mallet bags. Many of these mallets are now chosen as the primary implement for new works where they would not have in the past. As such, it is important to understand how mallet design, materials, and other properties interact with the instrument and affect its timbre.

2.1.2 Overview

This study uses Music Information Retrieval (MIR) tools to perform spectral analysis on single note audio samples of a tenor steelpan using a variety of mallets. MIR is an interdisciplinary field concerned with the extraction and analysis of data from musical audio signals using computational approaches. This data is often used in conjunction with machine learning to solve problems such as genre classification, tempo estimation, song structure analysis, and more. The data extracted using these tools, referred to as audio features, is presented here to give insight into the properties of the complete acoustical system of the steelpan and its mallets.

In undertaking this study, I recorded over 500 audio samples of a steelpan being played with five different mallets. Five smaples per note were selected as being representative of the mallets and prepared for analysis. The steelpan used had 29 notes, resulting in a collection of 145 audio samples. The Essentia audio analysis library was used in a Python script to automate the analysis process [14]. The data extracted using the Python script provides a basis on which to make objective comparisons between different mallets on the same instrument.

2.2 Instrument and Mallets

The instrument used for this study was a silver painted low C tenor built by Anthony Duncan in Trinidad and Tobago in 2015 (see Fig. 2.1). It was last tuned by Chris Wabich, a steelpan tuner based in Los Angeles, in spring 2019—shortly before the recording took place. This instrument follows the standard low C tenor circle of fifths layout with a 29 note range from C4 to E6 (as in Fig. 2.2).

In this study five different mallets, shown in Fig. 2.3, were analyzed. First were two different rubber-tipped mallets. One had a 1/2" wooden dowel as its core (far right mallet in Fig. 2.3). This was made by Mark Hosten in Trinidad. The other rubber-tipped mallet had aluminum tubing as its core and was made by Innovative Percussion based in Nashville, TN (far left mallet in Fig. 2.3). Rubber-tipped mallets

like these are considered the standard mallets for tenor steelpan, regardless of the shaft material. The attributes that contribute the most to the tone produced by these mallets are the hardness and thickness of the rubber tip, the overall weight of the mallet, and the stiffness of the shaft. Some mallets (including the aluminum Innovative Percussion mallet) have rubber tips on both ends with different thicknesses that generate subtly different timbres. The side with the thinner rubber would be more appropriate for melodic lines–especially when playing the highest notes—while the end with the thicker rubber would be more suited for harmonic accompaniment on lower notes. Only the thinner rubber is analyzed here.

Figure 2.3: Mallets analyzed.

The next two mallets were made of wood. One was a home-style wooden chopstick (the center mallet in Fig. 2.3). This chopstick is harder and denser than the typical disposable chopstick which is also often used. The other was a custom made dowel rod bundle mallet that I made myself (second from the right in Fig. 2.3). This mallet consisted of a seven 1/8" dowels surrounding a 3/16" dowel in the center. Chopsticks are common mallets used in more experimental, contemporary steelpan playing. The

bundle rod mallet was modeled after similar, larger bundle rods used on the drum set. As far as I am aware, I am the first person to make a version of the bundle rod for use on the steelpan.

The final mallet was made of cardboard tubing (second mallet from the left in Fig. 2.3). This was made by removing the cardboard tube from a wire clothes hanger (like the kind used at dry cleaners) and cut down to approximately 7.75 inches. Cardboard tubes have often been used as practice mallets due to their softer sound, but are also commonly called for in experimental, contemporary steelpan playing.

2.3 Audio Sample Acquisition

The audio samples used for this study were recorded at the University of Victoria Create Lab Recording Studio. The microphone used was an Earthworks M50, an omnidirectional small diaphragm condenser measurement microphone with a flat frequency response from 3 Hz to 50 kHz. The recording interface used was a Focusrite Rednet 4 with built-in preamplifiers. The audio was recorded at a samplerate of 192 kHz and 24-bit depth using Avid Pro Tools.

Due to the M50's sensitivity below 20 Hz, the audio signal contained significant ultra low frequency content. This low frequency noise is caused by normal movement in the air that is inaudible to humans, but is picked up by a microphone that is sensitive in infrasonic range. It was necessary to filter this subsonic content in order to prevent it from skewing the audio feature measurements. Avid's Channel Strip plugin was used to pre-process the audio with a -24 dB per octave high pass filter centered at 30 Hz before exporting individual audio files.

When recording the samples, I played the notes on the steelpan while an assistant operated the computer. We recorded 3-5 "forte" hits per note per mallet across the entire range of the steelpan. I then used my expert judgement to select the best representative for each mallet on each note for analysis. Each note was evaluated based on its tone quality and loudness. From the 3-5 recorded candidates, we then selected a single audio sample to represent a standard note at that pitch so that notes near each other in pitch were had similar tonal qualities and loudness.

This methodology could be viewed as a potential source of error, but I am a highly experience performer and qualified to make these subjective decisions. It could be suggested to employ a mechanical device to strike the instrument in a more tightly controlled manner. However, such devices are often too cumbersome to make

reaching all of the notes feasible, too expensive for the author to have access to, and difficult to set up to reliably produce high quality sound. Different notes on a steelpan behave differently. Some notes have a large "sweet spot" where the note speaks well and others have smaller, more particular ones. The amount of force coming from the proper direction at the correct angle necessary to achieve a high-quality "forte" tone can also vary across notes. I have the extensive performance experience on the steelpan to achieve this.

2.4 Audio Analysis using the Essentia Library

2.4.1 Background on Essentia

The Essentia audio library is an open-source C++ library designed to enable robust audio analysis and feature extraction. It was developed and is maintained by the Music Technology Group at Universitat Pompeu Fabra. Essentia also includes Python bindings which allows it to be used in Python scripts. Since it can be easily incorporated into a Python script, it is an excellent tool for analyzing large quantities of audio samples efficiently with consistent tolerances and settings.

There were two Python scripts written to conduct the analysis. The first performed the analysis on the entire audio sample while the second analyzed only the attack portion. The same basic process was used for both scripts. Fewer algorithms were used in the audio sample attack analysis because some algorithms were not appropriate for analyzing that portion of the audio. The purpose of analyzing the attack portion in isolation is that future projects intend to use this data for realtime audio processing where the computer will have to produce its output based solely on data from the attack portion of a sound in order to reduce latency.

The scripts begins by downsampling each audio sample from 192 kHz to 44.1 kHz. This is Essentia's default samplerate and many of the algorithms are optimized for it. Each audio sample is then normalized to -6 dBFS and the start/stop points are trimmed with a -60 dBFS power level tolerance. Since the notes were similar in loudness and peak value, the normalization does not have a strong effect on the outcomes other than to match the peak levels. Once this is done, the samples are ready for analysis. Once the analysis operations are complete, the Python script saves the results in a comma separated value (CSV) file. The analysis process took approximately 90 seconds to analyze all 145 samples.

2.4.2 Algorithms Used

This analysis used algorithms implemented in the Essentia C++ library as outlined by Geoffrey Peeters in his 2004 report [102]. Essentia is a mature MIR library whose algorithms are robust and follow the general guidelines set out by Peeters and subsequent MIR research. Table 2.1 gives a list of the algorithms used and a brief description of what they measure.

Algorithm	Description
Spectral Centroid	Weighted average of the frequency spectrum; the "center of mass" of a spectrum
Spectral Deviation	Deviation of the amplitude harmonic peaks from a global spectral envelope
Spectral Spread	The spread of the spectrum around its mean value
Spectral Skewness	The measure of the asymmetry of the spectrum around its mean value
Spectral Kurtosis	The measure of the flatness of a distribution around its mean value
High Frequency Content	Computes the high frequency content of the spectrum
LogAttackTime	Logarithm of the time duration between 10% and 90% of the maximum amplitude
Temporal Centroid	The temporal balancing point of the spectral energy
Duration	The measure of the time a signal is more than 40% of its maximum amplitude
Loudness	Sum of the individual loudnesses for each Bark band

Table 2.1: Description of low-level audio feature algorithms.

As mentioned, a subset of these algorithms were used to analyze only the attack portion of each of the audio samples (where the audio goes from 10% to 90% of its maximum amplitude). The omitted algorithms were generally temporal in nature and make more sense in the context of a complete audio sample.

2.4.3 Expectations

The expectation for this study was that the clear audible differences between the mallets should be reflected in the low-level audio feature data. The two rubber-tipped mallets should have similar results across the features since they sound very similar. The rubber mallets also act as an experimental control in a way. If they have

similar results as each other, but differ from the other mallets then that indicates that the software is performing the analysis successfully.

The chopstick and bundle rod should have somewhat similar results in the data since they produce similar timbres. For there to be an audio feature that can clearly separates the two would be an ideal result. While they sound similar, they are distinguishable to the ear. Since the cardboard tube shares different timbral qualities with both rubber mallets and the wooden mallets, this should be reflected in the data. However, there should be enough differences in the various features to distinguish them from both categories.

2.5 Results

2.5.1 Whole sample analysis

This section consists of commentary on the results of the analysis of the whole audio samples. The data is presented in Figs. 2.4-2.23. For each audio feature, two figures are shown: one with the raw data and a second with trendlines calculated using linear regression.

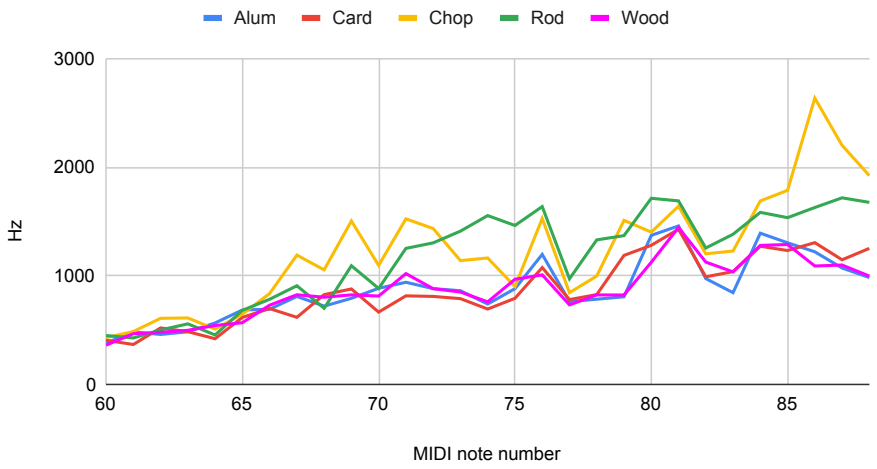


Figure 2.4: Spectral centroid data.

Figs. 2.4 and 2.5 shows plots of the spectral centroid values and trendlines. The plots reveal clear patterns. In the instrument's low range, the values are similar, but they spread out with the higher notes. The chopstick and dowel rods have consistently higher spectral centroid values than the other mallets that becomes more pronounced

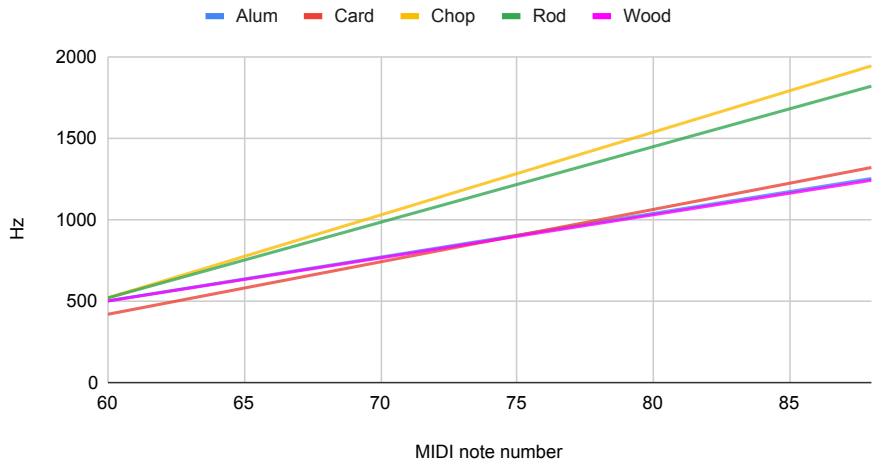


Figure 2.5: Spectral centroid trendlines.

in the higher range. The two rubber-tipped mallets and the cardboard tube yielded similar results. Also, certain notes on this steelpan, for example E5 (MIDI note 76) and A5 (note 81), have higher centroid values for all mallet types.

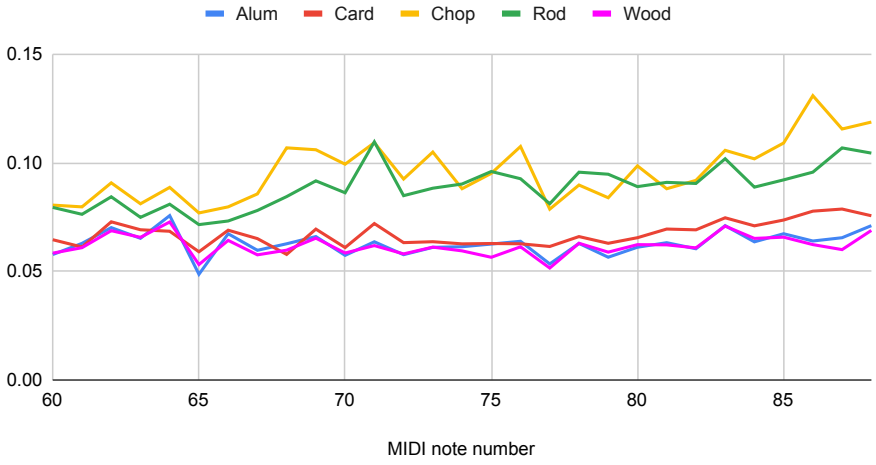


Figure 2.6: Spectral deviation data.

The values and trendlines for the spectral deviation, shown in Figs. 2.6 and 2.7, are much more consistent than for the spectral centroid. The mallets are clearly split into two groups: the "wooden" mallets (chopstick and dowel rods) and the "softer" mallets (the two rubber-tipped mallets and the cardboard tube). Although the wooden mallets are grouped together, the chopstick typically has a higher deviation value than the dowels rods. Also, while the two rubber-tipped mallets are tightly

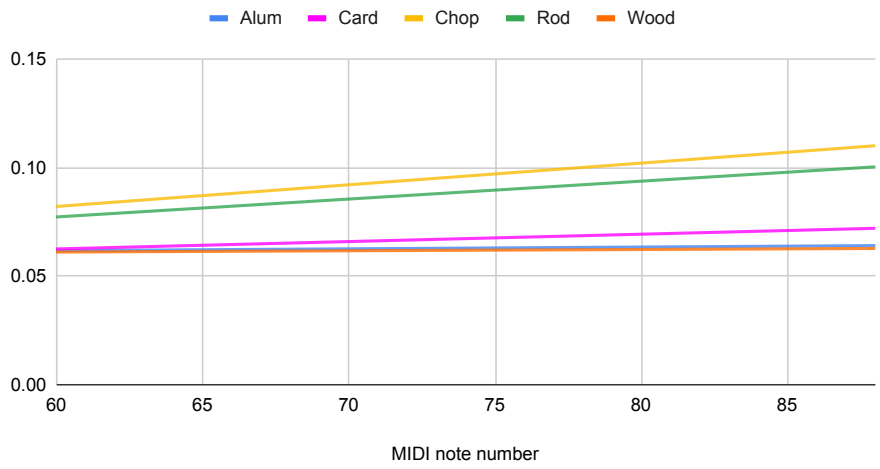


Figure 2.7: Spectral deviation trendlines.

coupled, the cardboard tube has generally higher values above MIDI note 69 (A4).

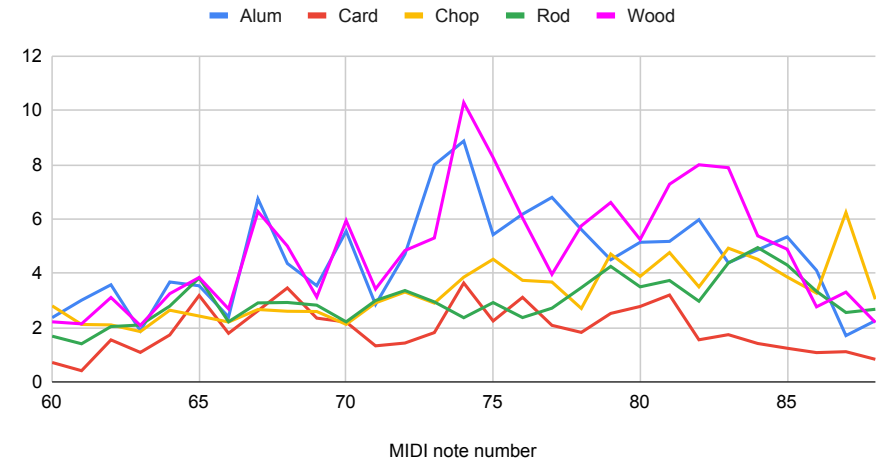


Figure 2.8: Spectral spread data.

The data for the spectral spread (see Figs. 2.8 and 2.9) shows that in the middle of the instrument's range, the rubber-tipped mallets have significantly higher values. The cardboard tube has the lowest values overall with the flattest trendline. The data for the spectral skewness (see Figs. 2.10 and 2.11) and spectral kurtosis (see Figs. 2.12 and 2.13) shows similar trends. The skewness and kurtosis decrease from the low range to the high range of the instrument, but the values for the different mallets are consistent relative to each other. The cardboard tube has the highest values, the two rubber-tipped mallets are tightly coupled, and the wooden mallets have lower values

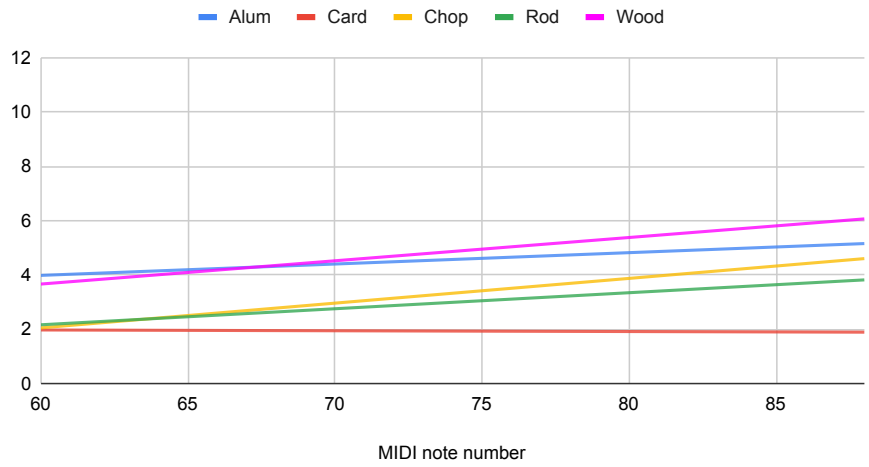


Figure 2.9: Spectral spread trendlines.

with the chopstick having the lower of the two. The data for high frequency content (see Figs. 2.14 and 2.15) shows a clear separation between the rubber-tipped mallets and the rest. There is more high frequency content for those mallets for almost every note with a large differential in the upper-mid range of the steelpan.

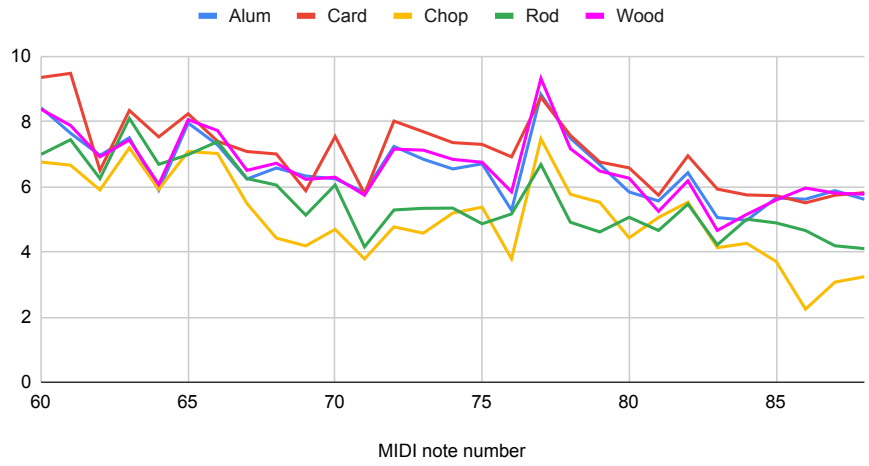


Figure 2.10: Spectral skewness data.

Figure 2.16 shows that all of the mallets have similar log attack time values with very similar trendlines. This suggests that attack time is primarily a property of the note and the mallet type does not have a strong influence on the value. The data and trendlines for temporal centroid in Figs. 2.18 and 2.19 show inconsistent variation among the mallets. More analysis will be required to establish a pattern.

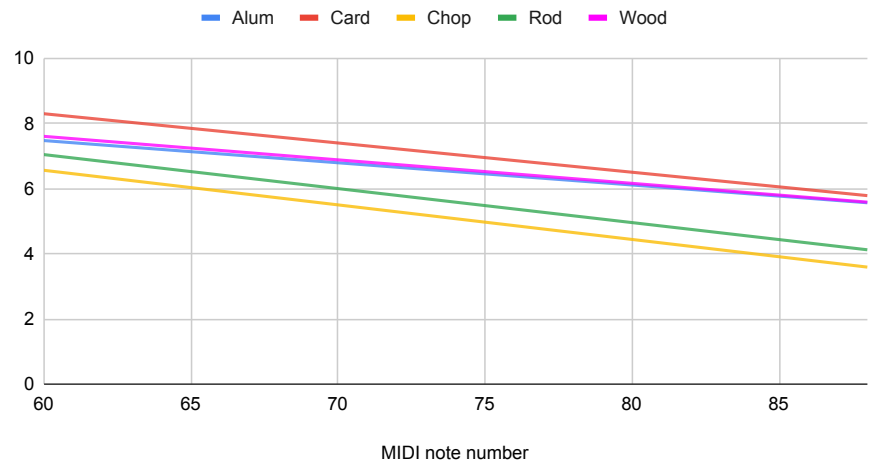


Figure 2.11: Spectral skewness trendlines.

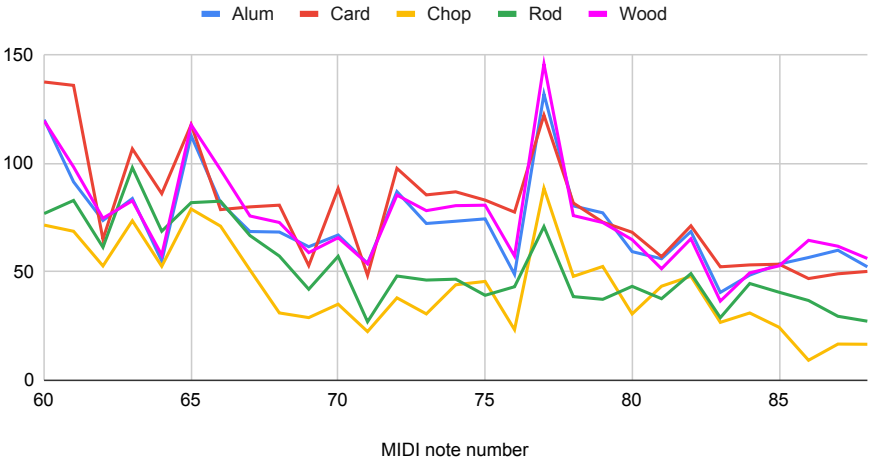


Figure 2.12: Spectral kurtosis data.

The data and trendlines for duration in Figs. 2.20 and 2.21 is interesting and makes sense given the tone of the mallets. The cardboard tube has the highest duration values in general. It has the least pronounced attack of all of the mallets and the attack has the highest amplitude of the sample. With a lower max amplitude, more of the sample will be above 40% of the maximum. This also explains why the two rubber-tipped mallets have the next highest values. The two wooden mallets have the sharpest and loudest attacks with a pronounced drop-off. Based on the definition for the duration measure, this causes the samples to drop below 40% of their max values quicker than the other mallets.

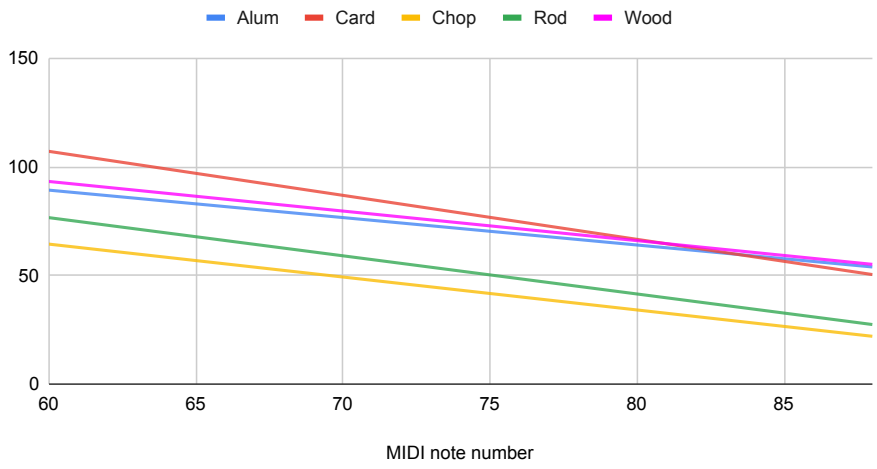


Figure 2.13: Spectral kurtosis trendlines.

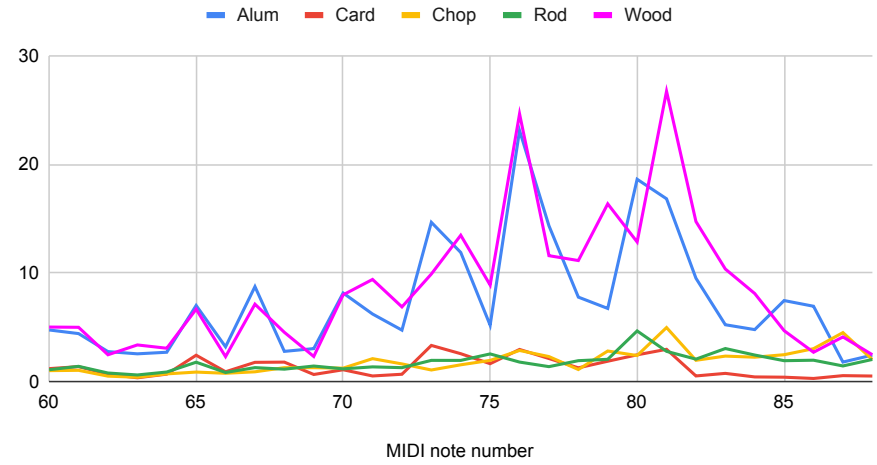


Figure 2.14: High frequency content data.

This is reflected in a different way in Figs. 2.22 and 2.23. Recall from Table 2.1 that for the loudness analysis, this is defined as the sum of loudnesses of the individual Bark bands. Thus the loudness algorithm is indirectly summing the spectral energy. The rubber-tipped mallets have the most blended tone of low and high frequency partials so they have the highest loudness of the mallets.

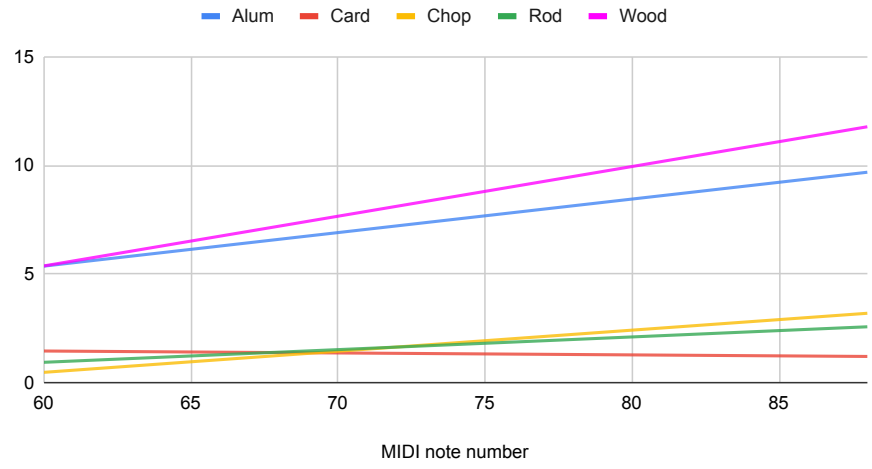


Figure 2.15: High frequency content trendlines.

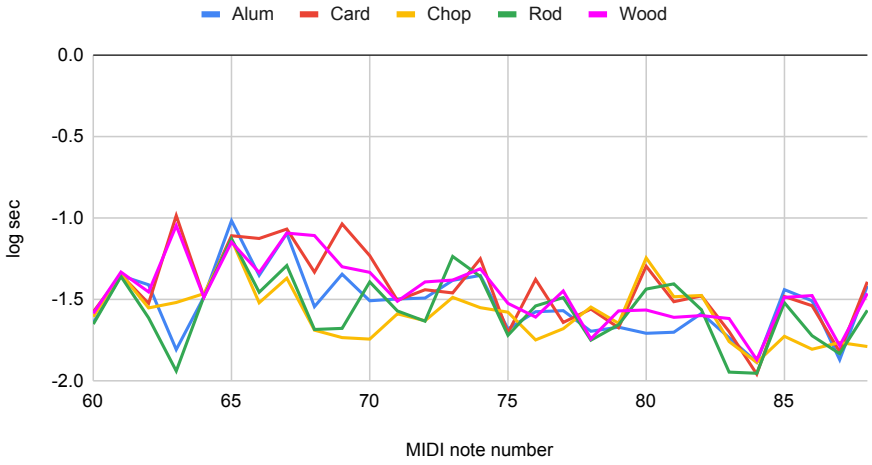


Figure 2.16: Log attack time data.

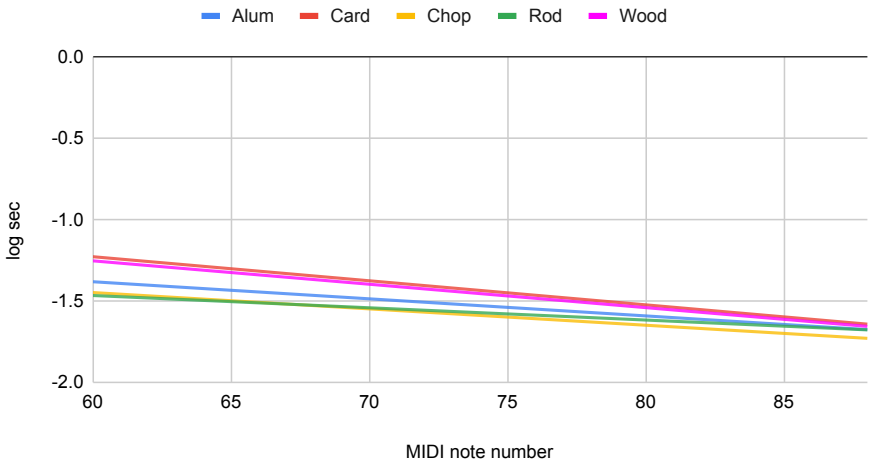


Figure 2.17: Log attack time trendlines.

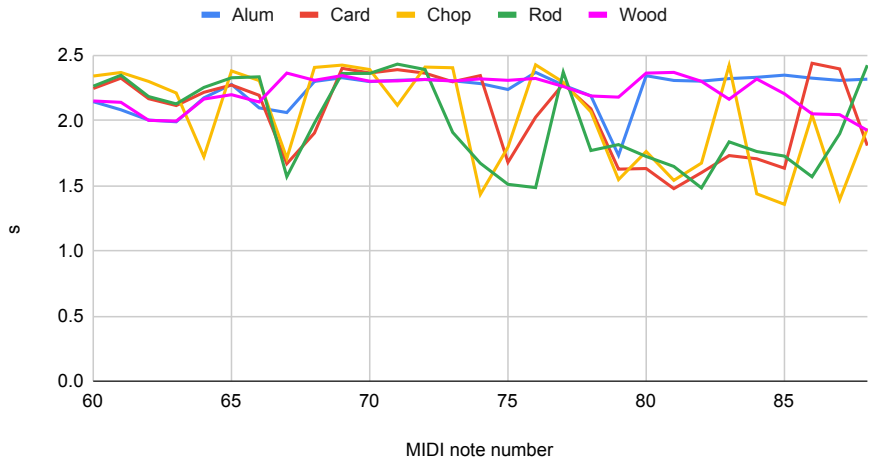


Figure 2.18: Temporal centroid data.

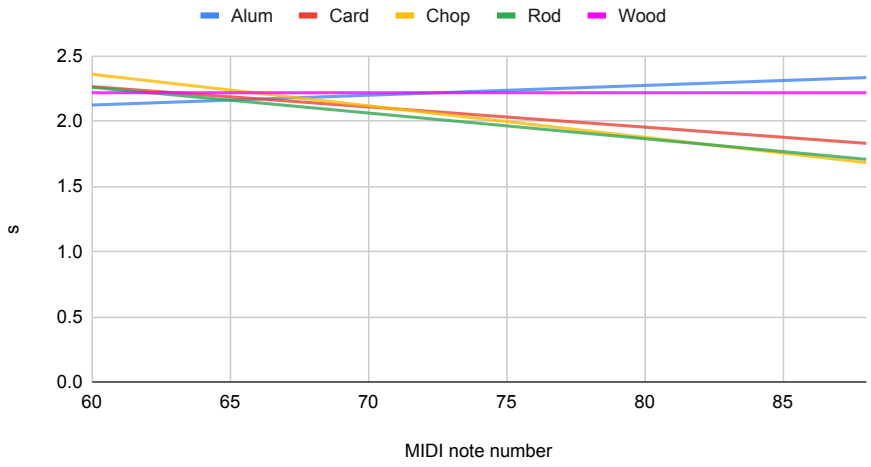


Figure 2.19: Temporal centroid trendlines.

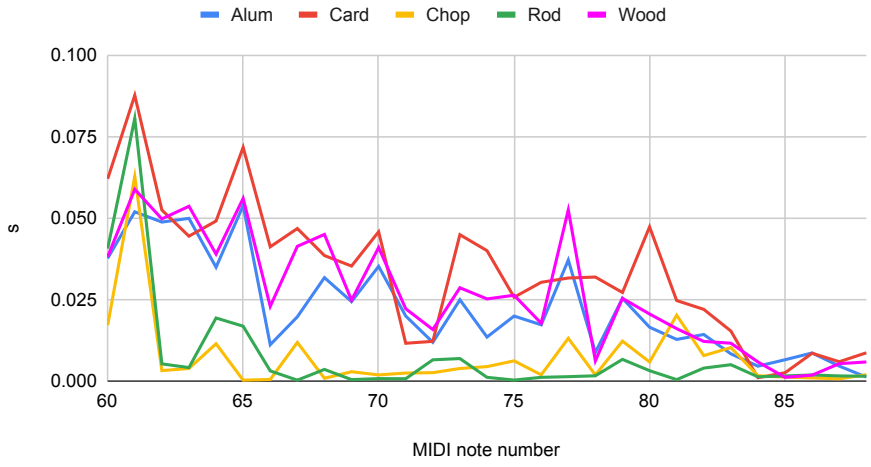


Figure 2.20: Duration data.

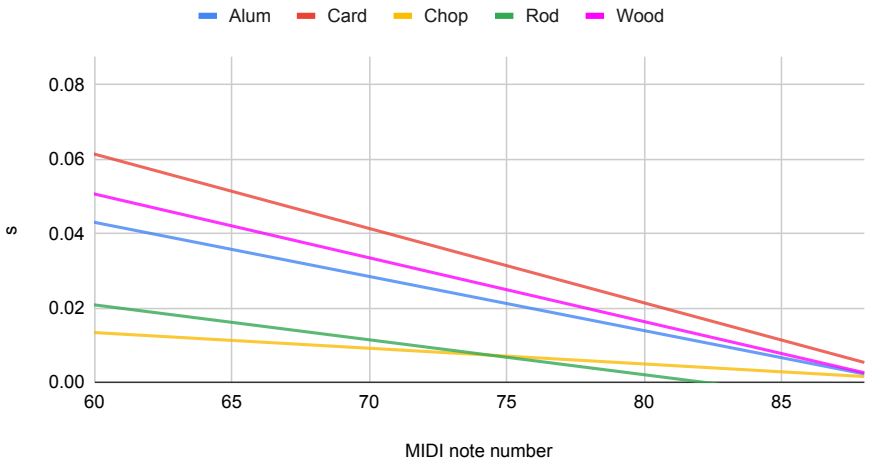


Figure 2.21: Duration trendlines.

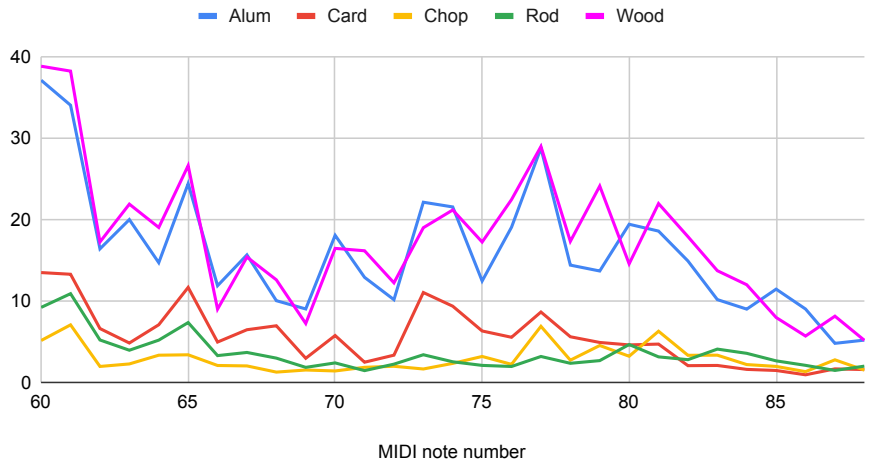


Figure 2.22: Loudness data.

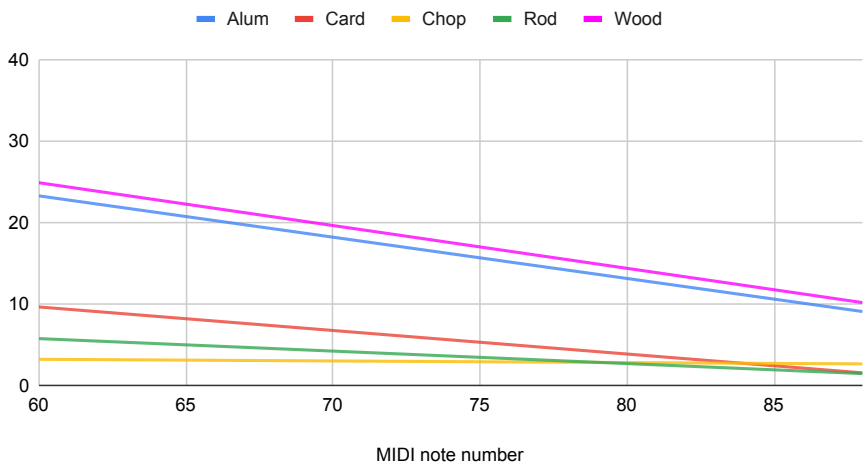


Figure 2.23: Loudness trendlines.

2.5.2 Sample attack analysis

This section gives commentary on Figs. 2.24-2.35. The figures here follow the same formatting as above: first the raw data and then a second with trendlines generated using linear regression. These figures show the results of the analysis on the attack portion of the audio samples. The audio samples were trimmed based on the attack start and end times produced by the log attack time algorithm. The trimmed audio was then used to perform another round of analysis to look for insight at only the attack portions of the audio samples. Only the non-temporal forms of analysis are included here.

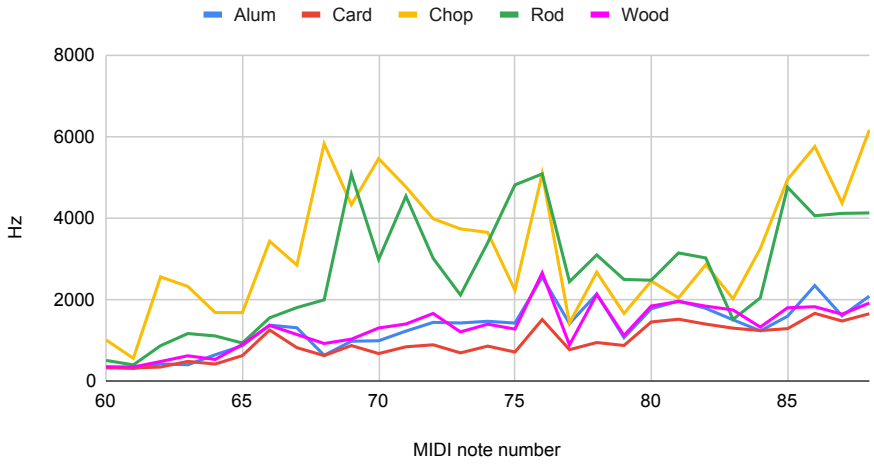


Figure 2.24: Attack spectral centroid data.

Figs. 2.24 and 2.25 show the attack spectral centroid data. It shows that the separation between the wooden mallets and the rest is even more pronounced in the attack. This is also reflected in the attack spectral deviation data in Figs. 2.26 and 2.27. The wooden mallets activate more high frequency partials than the other mallets which cause the spectral centroid and deviation values to trend higher than they do for the other mallets.

The data for the attack spectral spread, shown in Figs. 2.28 and 2.29, shows that the clearest trend is for the cardboard tube to have significantly lower values than the other mallets across the entire range of the instrument. This was true when considering the whole audio sample as well, but is more pronounced during the attack. The other mallets have similar values except in the middle of the steelpan's range where the rubber-tipped mallets sometimes have significantly higher values.

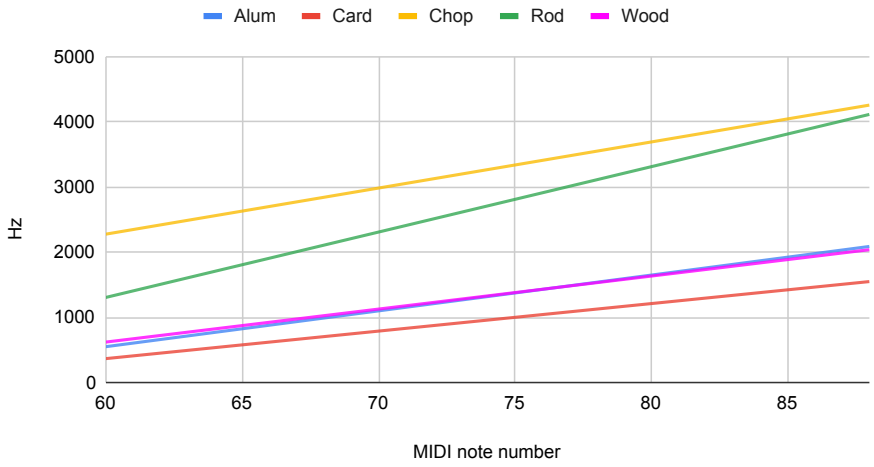


Figure 2.25: Attack spectral centroid trendlines.

The data for the spectral skewness and kurtosis of the attack in Figs. 2.30, 2.31, 2.32, and 2.33 continue the trend of making patterns from the whole sample analysis more pronounced. The wooden mallets have significantly lower values for both measures than the other three mallets across the entire range of the steelpan.

This pattern shifts with the high frequency content measure in Figs. 2.34 and 2.35. The rubber-tipped mallets have significantly higher values across the range than the other three mallets. The high frequency content algorithm is an audio feature that is not based on human sound perception. It is a weighted sum of the magnitude spectrum for a signal. One might think that the wooden mallets should have the highest values in this category since they have a "brighter" sound. However, while they have higher spectral centroid values, the rubber-tipped mallets activate more partials at higher levels across the entire spectrum. When weighted and summed, these partials contribute a large amount to the high frequency content value.

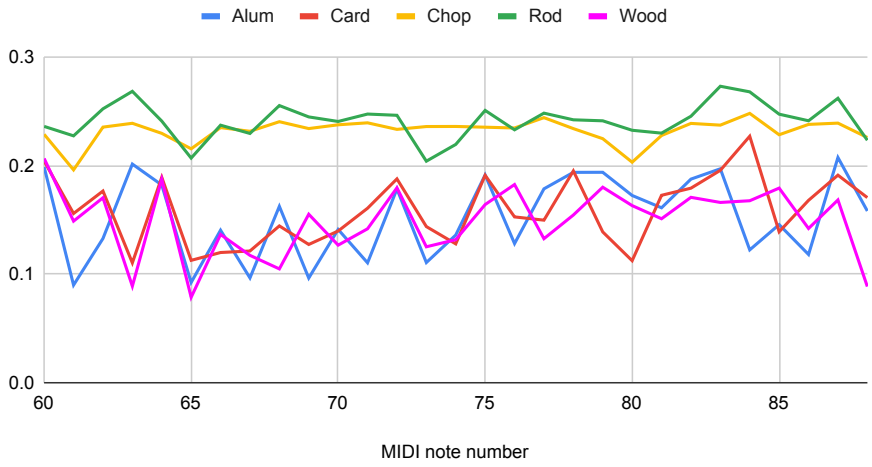


Figure 2.26: Attack spectral deviation data.

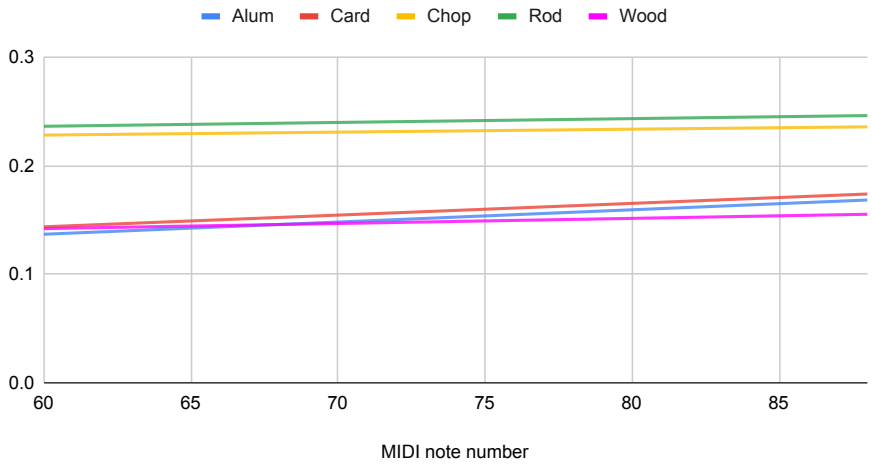


Figure 2.27: Attack spectral deviations trendlines.

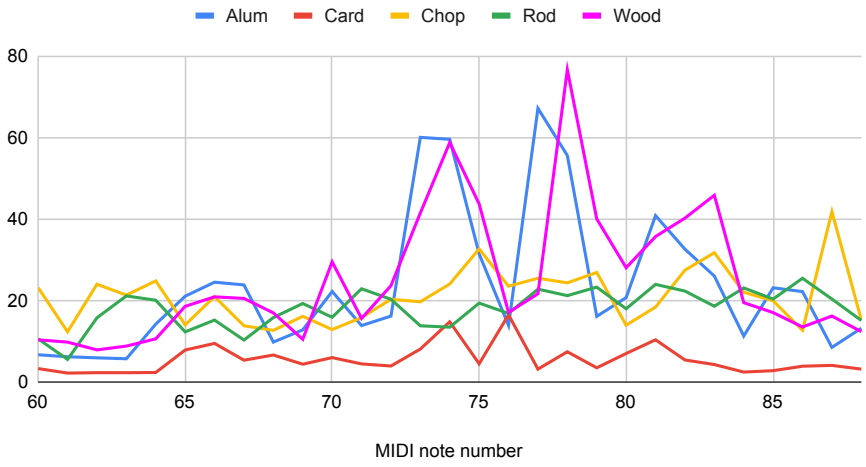


Figure 2.28: Attack spectral spread data.

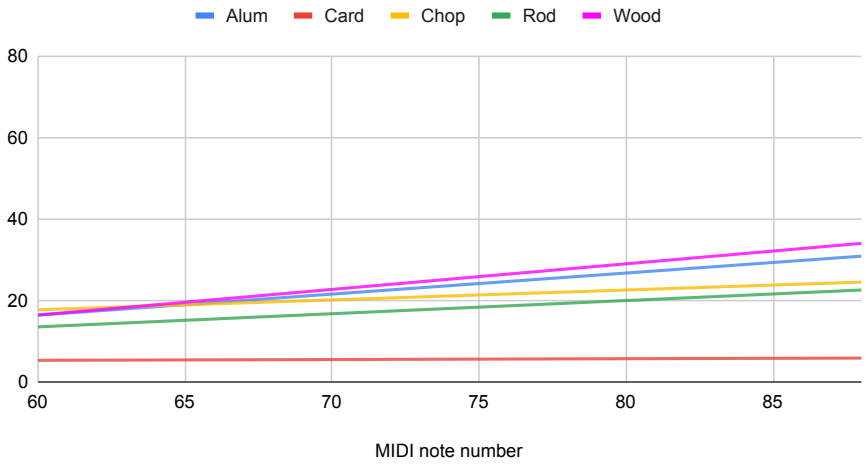


Figure 2.29: Attack spectral spread trendlines.

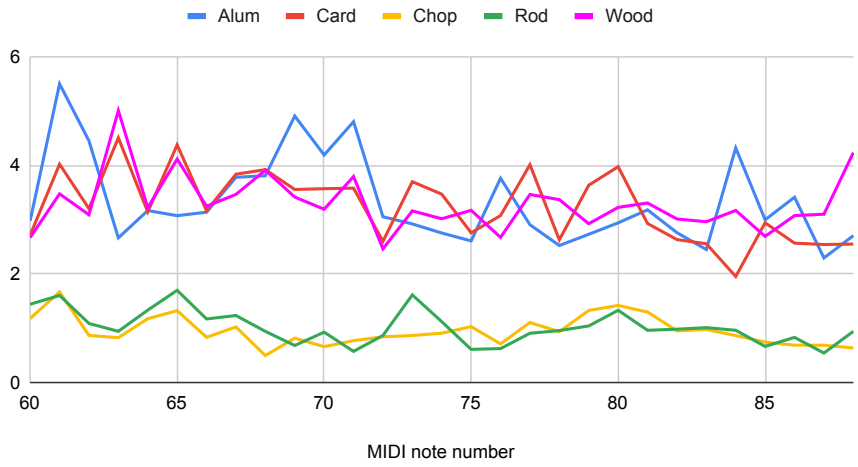


Figure 2.30: Attack spectral skewness data.

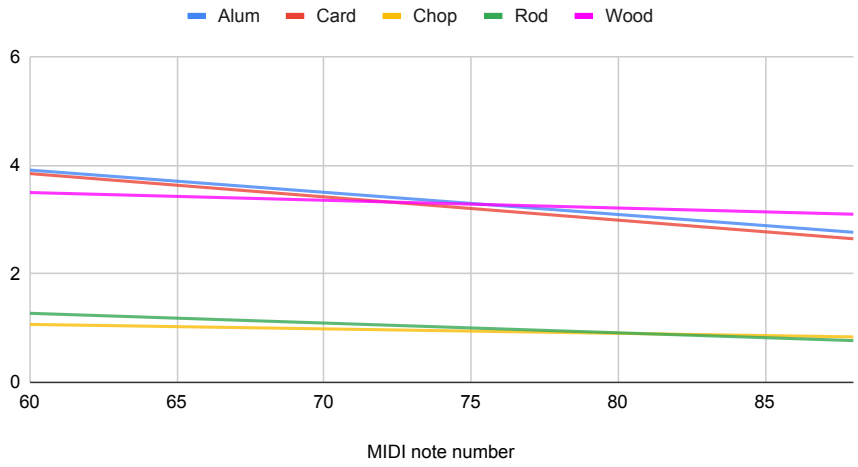


Figure 2.31: Attack spectral skewness trendlines.

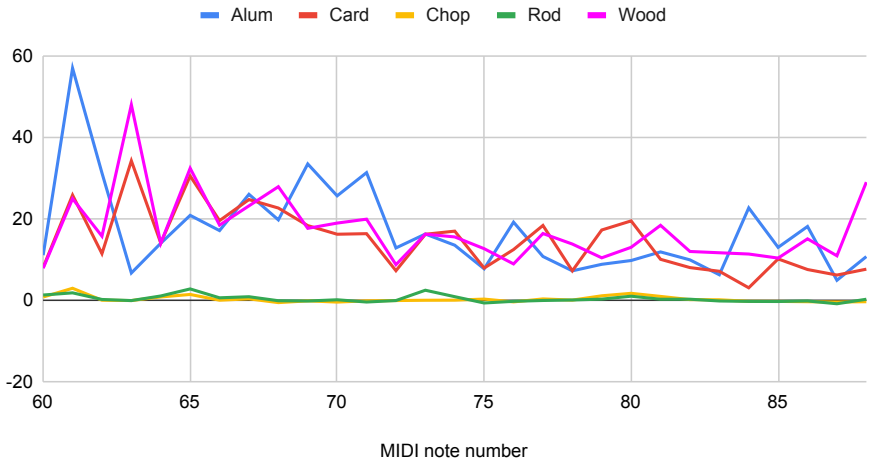


Figure 2.32: Attack spectral kurtosis data.

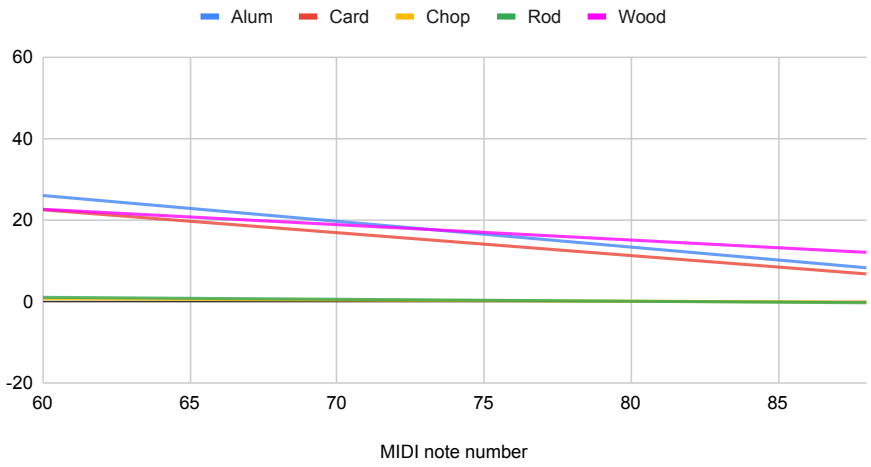


Figure 2.33: Attack spectral kurtosis trendlines.

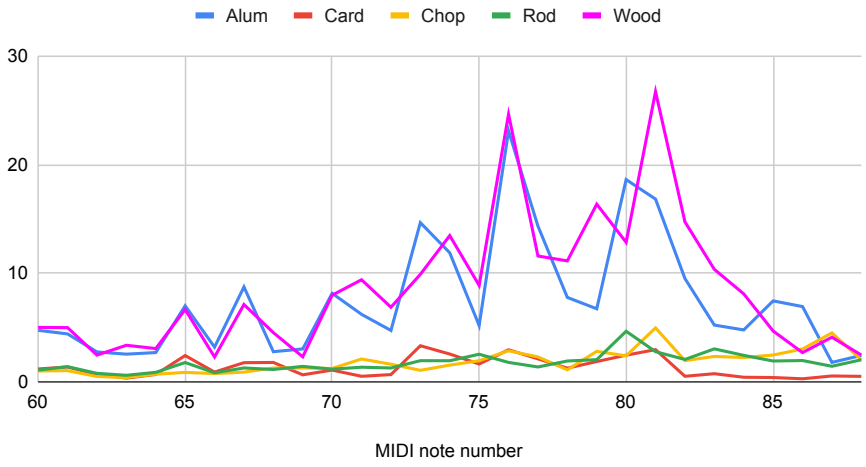


Figure 2.34: High frequency content data.

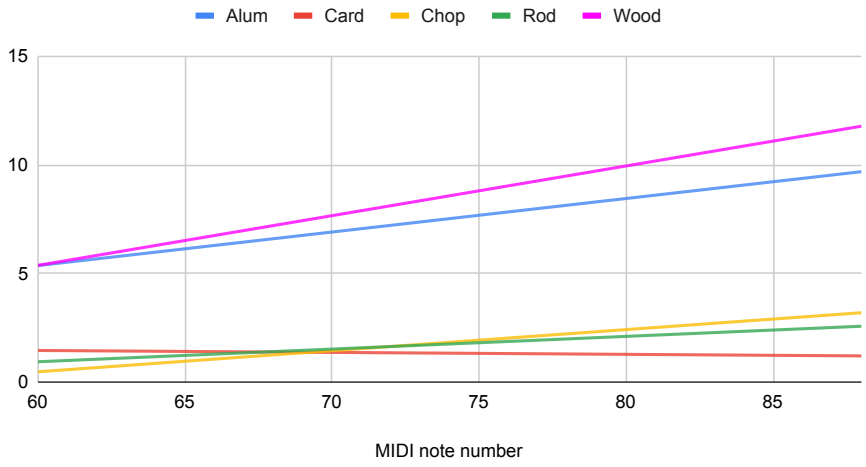


Figure 2.35: High frequency content trendlines.

2.6 Conclusion

Using the Essentia audio analysis library to extract the low-level audio features provides a novel way to analyze the acoustical properties of an instrument. In this study, focusing on the differences between different mallets on a single steelpan reveals the influence the different mallets types have on timbre.

The numerous audio features extracted both from whole note audio samples and from the attack portion of samples provide analytical descriptions of the various mallets. As predicted, the results for rubber-tipped mallets were tightly coupled in general. Only a few features such as high frequency content, spectral spread and duration had noticeable deviation between the two. The chopstick and bundle rods were similarly coupled as predicted. The cardboard is the wildcard. Often its values are similar to the rubber-tipped mallets, but sometimes it has more similar values to the chopstick and bundle rods. Although the aural differences between the various mallets is apparent to a listener, the extracted audio feature data provides a framework for analytical description and comparison that goes beyond what is possible based on subjective aural descriptions.

2.7 Future Work

While this data is useful, it will become more so once a larger dataset encompassing more steelpans and mallets is established. The intention is to increase the scope of this project by conducting the same analysis on more instruments. As more steelpans are made available for analysis, their data will be added to a database that will eventually cover the entire steelpan family. Another area for future work is to further develop the analysis process. There are more audio features that can be added, such as mel-frequency and Bark-band cepstral coefficients.

This method of analysis can be applied to other instruments as well. This can then provide another way to compare instruments and discuss their differences in an objective way.

MIR is a field often paired with machine learning to create automated music classification systems. An example of this would be automatic genre classification. With more data, a machine learning model can be trained to classify a steelpan note's pitch and the type of mallet used. An approach to steelpan pitch detection is discussed in Chapter 3 although it is not based on audio feature extraction.

Based on the data presented above, it would be reasonable to expect a machine learning model to be able to categorize an audio sample by mallet into generalized categories like “rubber,” “wooden,” and “cardboard”. Such a model would ideally also be able to differentiate between chopsticks and dowel bundle rod mallets.

As an electroacoustic steelpan performer, my ultimate goal for this research is to implement these machine learning models as realtime objects that can run in Max/MSP (or Max for Live) so that they can be used to control realtime processing in performance. If implemented as a module that outputs note onset triggers, pitch estimation, and mallet estimation data would allow for enhanced performer-computer interaction with intelligent audio processing based on what the performer plays in realtime.

This analysis may also be relevant to general steelband audio analysis. There has been some computational ethnomusicology research regarding steelband music such as in [107]. This is another avenue of enquiry to pursue in future research.

Chapter 3

Steelpan Pitch Detection

3.1 Introduction

The pitch of a melodic instrument is a primary characteristic of a musical sound. Pitch detection, also referred to as fundamental frequency estimation, is an important task for audio processing and analysis. General pitch detection methods, such as PYIN [79] and CREPE[51], work well in many situations, but there are many tasks for which custom-tailored methods may perform better. In these situations, a dataset of appropriate audio is needed in order to design a solution.

Performing pitch detection on the steelpan is a situation in which a custom solution can potentially outperform state-of-the-art methods such as PYIN and CREPE. In this chapter I present SASS-E, the Steelpan Audio Sample Set for Evaluation, and Steelpan-Pitch, a deep learning model for steelpan pitch detection. A version of this chapter has been published in the Proceedings of the International Conference on New Interfaces for Musical Expression[75]. The architecture for Steelpan-Pitch is designed to minimize latency so that it can be implemented in realtime processing for live audio. SASS-E is open for public use for training and analysis of steelpan audio. Steelpan-Pitch is evaluated on the test set from SASS-E and the results are compared against PYIN and CREPE. The evaluation shows that the custom solution, Steelpan-Pitch, outperforms the baseline state-of-the-art solutions. To further show that Steelpan-Pitch's algorithm generalizes beyond the steelpan data represented in SASS-E, it is evaluated on steelpan samples taken from the commercial sample library Andy Narell Steel Pans produced by Ilio. I also conduct an experiment to find the balancing point between minimizing latency and maintaining accuracy. Although Steelpan-Pitch is

designed for pitch detection on steelpans it can be adapted to model custom pitch detectors for other instruments. With a suitable training dataset the Steelpan-Pitch architecture can be adapted to achieve low latency, high accuracy pitch detection for other instrument types.

This chapter is structured as follows: Section 3.2 presents background information on pitch detection methods. Section 3.3 presents the details of the SASS-E dataset. Section 3.4 presents the proposed architecture of Steelpan-Pitch. The experiments and results are discussed in Section 3.5. Section 3.6 concludes the chapter.

3.2 Background

The pitch of an audio signal is a perceptual property that generally has a strong relationship to its fundamental frequency (F_0). This relationship is so strong that the two terms are often used interchangeably. As a fundamental property of a signal, pitch detection has received significant attention and many different approaches have been proposed. There are three main categories of approaches: time-domain, frequency-domain, and data-driven [108]. Most time-domain approaches are based on the autocorrelation function where a signal is correlated with itself at various time lags [128]. Such methods were implemented in digital hardware as early as 1976 [33]. The average magnitude difference function was proposed as a variation on autocorrelation that eliminates multiplication operations by using the absolute value of the difference between the signal and itself at various time delays [112]. The YIN algorithm was proposed as a further refinement of the autocorrelation method that combines it with various methods of error prevention to improve accuracy [30]. Subsequently, PYIN was developed as a probabilistic version of YIN that uses multiple pitch candidates and a Viterbi-decoded hidden Markov model [79]. In contrast to the autocorrelation-based methods, SWIPE compares the input signal's spectrum with the spectra of sawtooth waveforms [21]. The Cepstrum approach uses the cepstrum of a signal (the power spectrum of the logarithm of the power spectrum) to perform pitch detection [93]. In 2018, CREPE (Convolutional REpresentation for Pitch Estimation) emerged as the best performing data-driven machine learning method [51].

Historically, most pitch detection methods generate pitch candidates algorithmically and use heuristics for selecting the final output. The best performing of this group of algorithms was PYIN. In 2018, CREPE presented a new approach to pitch detection by designing a convolutional neural network that takes a raw audio signal as

input and outputs a fundamental frequency estimation based on how it was trained. In their landmark paper, Kim et al. show that CREPE significantly outperforms PYIN and SWIPE on the RWC-synth and MedlyDB-stem-synth audio datasets [51]. Another convolutional neural network-based pitch estimation system is DeepPitch [135]. DeepPitch contrasts with CREPE by using a high-resolution log-frequency spectrum as input and actually consists of two neural networks: one for pitch activity detection and the other for pitch value estimation. Another recent contribution is DA-Net: Dual Attention Network for pitch estimation [61]. DA-Net is based on CREPE's structure, but uses dual attention modules instead of convolutional layers. These examples confirm that a data-driven approach to pitch detection is a viable method. I have also been active in presenting preliminary explorations of applying these techniques to the steelpan [66], [67], [71]. Although CREPE performs well, Singh et al. showed that the number of network parameters could be reduced while simultaneously improving performance by increasing the filter kernel size and using residual blocks [126]. The CREPE architecture and variants of it have been re-used in other audio machine learning contexts and performed well [130].

3.3 SASS-E v1.0

SASS-E (Steelpan Audio Sample Set for Evaluation) is a new audio dataset constructed for evaluating music information retrieval tasks on steelpan audio. SASS-E includes samples from three steelpans that are the personal instruments of the author and were all recently tuned before recording the audio samples. One is a professional quality instrument built by Kyle Dunleavy in the United States. It is a semi-bore, chrome-plated steelpan. This means each note has four small holes drilled around it. Its range is C3-F6 (30 notes). The second instrument is a full-bore, nickel-plated steelpan from Trinidad. The final instrument is a non-bore, painted steelpan of unknown origin.

The three instruments were recorded at different times over the course of several years, but all in the same professional quality recording studio environment with the same setup. Microphones used for the sessions included the Earthworks M50 measurement microphone, Beyerdynamic M 160 ribbon microphone, and the Schoeps CMC 6 small diaphragm condenser microphone with MK 4 cardioid capsule. The M 160 and CMC 6 were positioned underneath the bowl of the steelpans while the M50 was positioned above the steelpan. The microphone positions are informed by the

Figure 3.1: Sample recording session.

recommendations in [88] as well as the author's own extensive experience performing on and recording steelpans. All instruments were recorded using a Focusrite RedNet 4 audio interface.

At least 50 strikes were recorded per note per instrument using rubber tipped mallets. The strikes were recorded at dynamic levels varying from pianissimo to fortissimo. The performer attempted to strike the notes in a variety of locations in order to capture all of the possible timbral nuances of each note. Notes in the high range of the steelpan sometimes do not fully activate if not struck in their "sweet spot". This results in a dull, thudding sound. Non-activating hits were not included in the dataset.

The samples from the three instruments are mixed in the dataset. The files identify their source note at the beginning of their filename. The filenames are formatted in the form `<MIDI note number>_<set>_sample_<number>.wav` where
`<MIDI note number>` refers to the integer MIDI value of the note's fundamental frequency, `<set>` can be "train", "val", or "test" depending on which pre-set split it belongs to, and `<number>` assigns a unique number to each sample at a given MIDI note value. The audio samples are pre-split into training, validation, and test sets with 7,931 samples in the training set, 2,680 samples in the validation set, and 2,702 samples in the test set. The dataset contains 13,313 samples for a total of 9 hours and 25 minutes of audio. The audio samples are trimmed so there is minimal leading silence and allows for the full release of each note to ring out with some trailing silence. Depending on the situation, users of the dataset should use automatic trimming on the samples. The shortest audio sample is 1.19 s and the longest is 12.46 s long.

3.4 Proposed network architecture

The proposed network architecture of Steelpan-Pitch is based on CREPE, but modified to reduce both the number of parameters and the minimum latency of the system. Steelpan-Pitch is a deep convolutional neural network that takes a monophonic time-domain audio signal as input and produces a pitch estimation as output. Figure 3.2 shows a block diagram of the proposed architecture. The input is a 128-sample audio frame at a samplerate of 16 kHz (represented on the far left of Fig. 3.2). There are six convolutional layers and then the network terminates with a fully connected layer. Each triangle in Fig. 3.2 gives the details for the layers from left to right. The first convolutional layer consists of 512 filters of size 64. The second through fourth layers

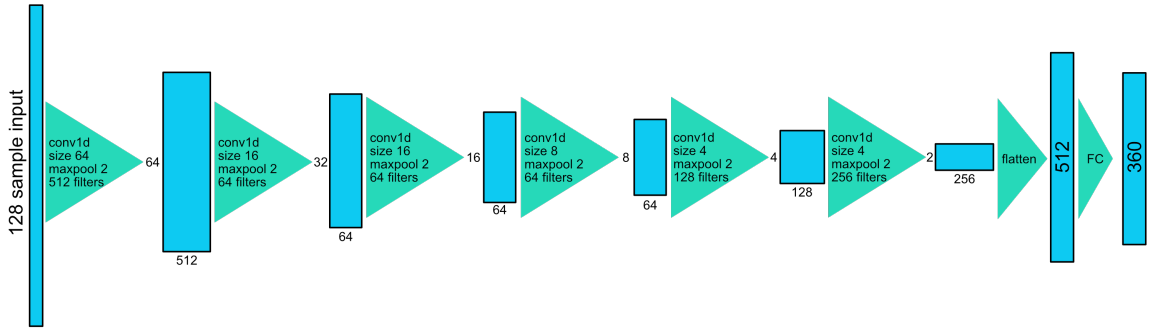


Figure 3.2: The network architecture of Steelpan-Pitch.

each consist of 64 filters of size 16, 16, and 8 respectively. The fifth and sixth layers increase the number of filters to 128 and 256 – both of size 4. After every convolutional layer is a maxpooling layer of size 2. The output of the final convolutional layer is flattened to a 512-dimensional latent representation. The fully connected output layer consists of 360 neurons with sigmoid activations as in [51]. The output vector from the fully connect layer, $\hat{y}$, is used to calculate the final pitch estimate.

Based on [51], the 360 output nodes of the fully connected layer represent frequency bins with 20 cent logarithmic spacing. The 360 bins are denoted as $c_1, c_2, \ldots, c_360$ and span six octaves of audio from note C1 (32.70 Hz) to B7 (1975.5 Hz). The output of the system, $\hat{c}$, is the frequency estimation given by the weighted mean of the bin frequencies by the corresponding values of the output vector, $\hat{\mathbf{y}}$:

$$\hat{c} = \frac{\sum_{i=1}^{360} \hat{y}_i c_i}{\sum_{i=1}^{360} \hat{y}_i}. \tag{3.1}$$

The pitch estimate, $\hat{f}$, is then given by

$$\hat{f} = f_{\text{ref}} \cdot 2^{\hat{c}/1200} \tag{3.2}$$

where f_{ref} is the reference frequency of 10 Hz.

The network is trained using Gaussian blurring on the target vectors, $\hat{y}$, to encourage the system to prefer nearly correct predictions when it does make an incorrect prediction [13]. The bin corresponding to the ground truth is given a value of one and then decays over the surrounding bins with a standard deviation of 25 cents according to

$$y_i = \exp\left(-\frac{(c_i - c_{\text{true}})^2}{2 \cdot 25^2}\right). \tag{3.3}$$

Steelpan-Pitch uses a binary cross entropy loss function to determine the error between the target, $\mathbf{y}$, and the prediction, $\hat{\mathbf{y}}$, as in Equation 3.4.

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^{360} \left(-y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)\right) \tag{3.4}$$

The model is optimized using the Adam optimizer with a learning rate of 0.0002 [52]. Batch normalization [48] and dropout [129] are used at each convolutional layer as well. The dropout layers have a dropout probability of 0.25. The training set

was split into 128-sample frames with an overlap of 64 samples. A batch size of 128 was used in training. An early stopping callback was implemented that watched the validation loss value and stopped the training after five epochs had elapsed with no reduction greater than the threshold of 0.00005. The architecture was implemented in TensorFlow using Keras and trained using Google Colaboratory GPUs. Figures 3.3 and 3.4 show the loss and accuracy values during training. The accuracy values in Fig. 3.4 are those reported by TensorFlow during training and do not follow the same methodology that is used to evaluate the model. The results of the evaluation are presented in Section 3.5.3.

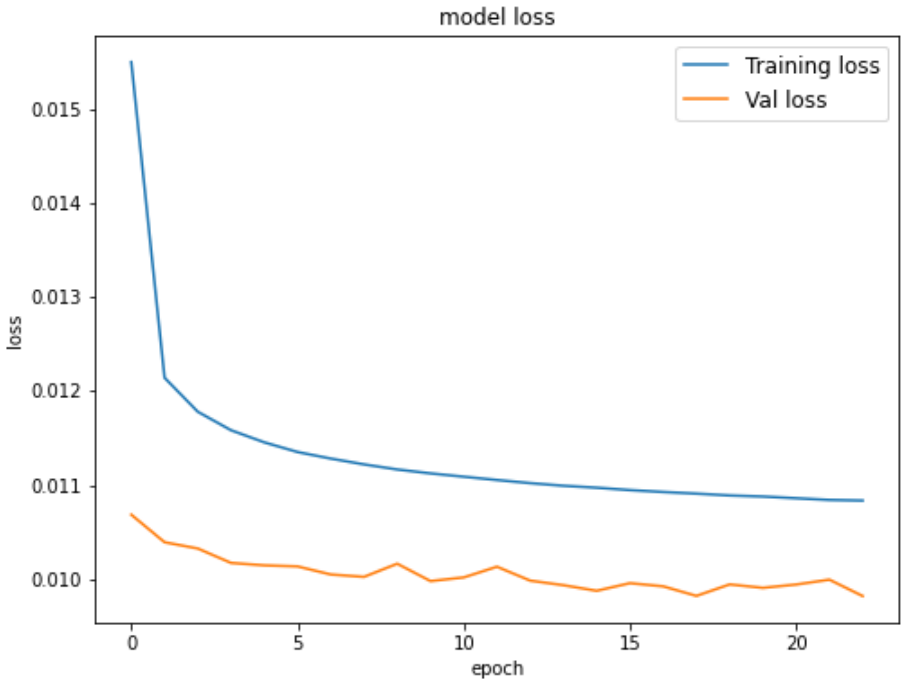


Figure 3.3: Training and validation loss for Steelpan-Pitch.

Although the proposed architecture is based on CREPE, there are some significant differences. One of the most important differences in the architecture is the accepted input signal. CREPE uses a 1,024-sample audio frame as input while our architecture uses a 128-sample audio frame. Both systems use a sample rate of 16 kHz. Thus when operating in a realtime system, this reduces the minimum latency of the system from 64 ms for 1,024 samples to 8 ms for 128 samples. In the original CREPE architecture, a stride of 4 was used at the first layer and a stride of 1 for each subsequent layer.

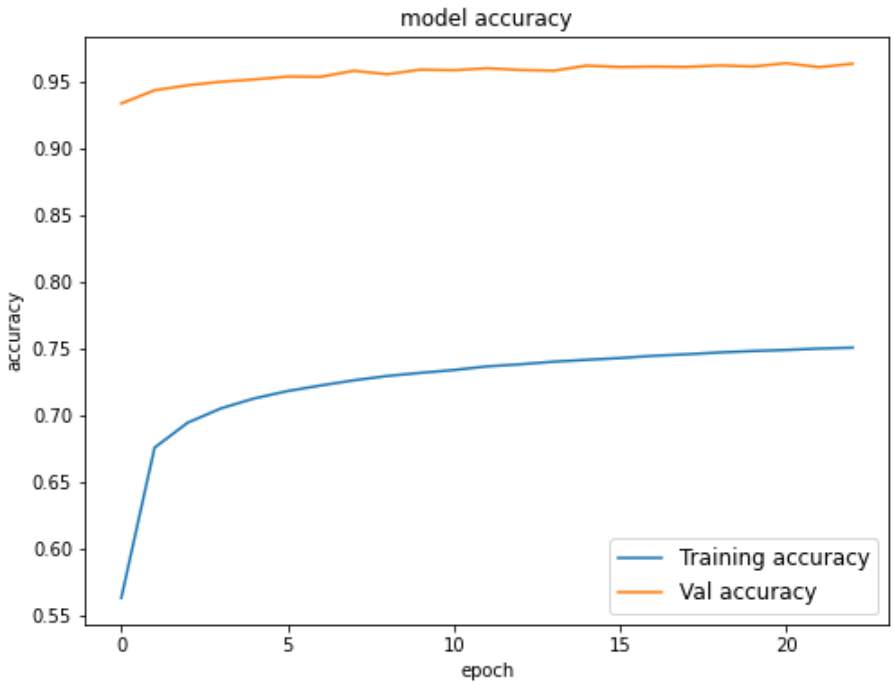


Figure 3.4: Training and validation accuracy for Steelpan-Pitch

Since the input frame to Steelpan-Pitch is smaller, it uses a stride of 1 for all layers and instead relies solely on maxpooling for downsampling in the network. The next main difference is the size of the latent representation at the end of the convolutional portion of the network. CREPE has a 2048-dimensional latent space while Steelpan-Pitch has a 512-dimensional latent space. Steelpan-Pitch also uses fewer filters per convolutional layer than CREPE. All of this results in significantly fewer parameters. The proposed architecture for Steelpan-Pitch has 1,009,320 total parameters whereas CREPE has over 22 million. The benefits of the network having fewer parameters include reduced training time, faster inference time, and a lower likelihood of overfitting the data.

CREPE also uses a hidden Markov model with the Viterbi algorithm to probabilistically determine the most likely output. This method is the "p" in the PYIN algorithm. However, hidden Markov models are computationally expensive. I chose to omit the hidden Markov model in Steelpan-Pitch's architecture because it is designed to be useable in realtime processing.

3.5 Experiments

3.5.1 Training Dataset

Steelpan-Pitch was trained solely on the training set from SASS-E. The recording sessions for constructing SASS-E were scheduled for shortly after the steelpans were tuned by a professional steelpan tuner. As such, I make the simplifying assumption that the notes can act as a representative of notes from their respective pitch classes. In this way, I treat this more as a classification task despite estimating intermediate values at the final output.

The SASS-E dataset was augmented for training using pitch shifting. A copy of each note in the dataset was randomly pitch shifted up or down between 20 cents and 2 semitones in 20 cent increments. Since the steelpan samples in SASS-E are all tuned to integer MIDI values, this allowed us to train the system on intermediate values as well. Augmenting the core SASS-E training set in this way effectively doubles the training set size to nearly 19 hours of audio. Each audio sample has the leading and trailing silence trimmed and is then sliced into 128-sample audio frames (8 ms at 16 kHz samplerate) with a step size of 4 ms. This resulted in 3,521,632 training audio frames and 611,197 validation audio frames.

3.5.2 Methodology

For these training, validating and testing Steelpan-Pitch, I used SASS-E's pre-split training, validation, and test sets. These sets were split using a 60/20/20 random split. All three sets contain instances of every note from every instrument in SASS-E. The danger here will be that the model may overfit the dataset and not generalize to other steelpans. However, I show that the model does generalize successfully in 3.5.5. The model is evaluated in terms of both raw pitch accuracy (RPA) and raw chroma accuracy (RCA) with 50, 25, and 10 cent thresholds. RPA measures the percentage of estimations that are within 50, 25, and 10 cents of the target. RCA measures the percentage of estimations that are within 50, 25, and 10 cents of a member of the target's pitch class. In other words, RCA accounts for octave errors.

Steelpan-Pitch is then compared against CREPE and PYIN. In every test instance, the pre-trained weights for CREPE provided by the developers are used. These methods are the current state-of-the-art machine learning (CREPE) and time-domain (PYIN) methods available. For reference, PYIN is evaluated with frame lengths of

both 1,024 samples and 128 samples. CREPE only operates with a frame length of 1,024 samples while Steelpan-Pitch only operates with a frame length of 128 samples. All three methods are evaluated on the test set from SASS-E.

3.5.3 Results

Table 3.1 and Figure 3.5 show the RPA results for the pitch detection methods on the SASS-E test set. We can see that Steelpan-Pitch significantly outperforms both PYIN and CREPE. At the 50 cent threshold, Steelpan-Pitch beats the next best performer by 22 percentage points. A surprising result here is that PYIN performed somewhat better with the shorter frame length. Table 3.2 and Figure 3.6 show the raw chroma accuracies on the SASS-E test set. In Table 3.2, we can see similar results as in Table 3.1. Steelpan-Pitch once again outperforms PYIN and CREPE by over 21 percentage points. An important point to reiterate here is that CREPE and PYIN both use Hidden Markov Models to probabilistically predict the next value.

These results are stark, but also understandable. Steelpan-Pitch was tested on audio samples from the same instruments that it was trained on. In order to show that Steelpan-Pitch has the potential to generalize beyond the training instruments, I present another experiment in Section 3.5.5.

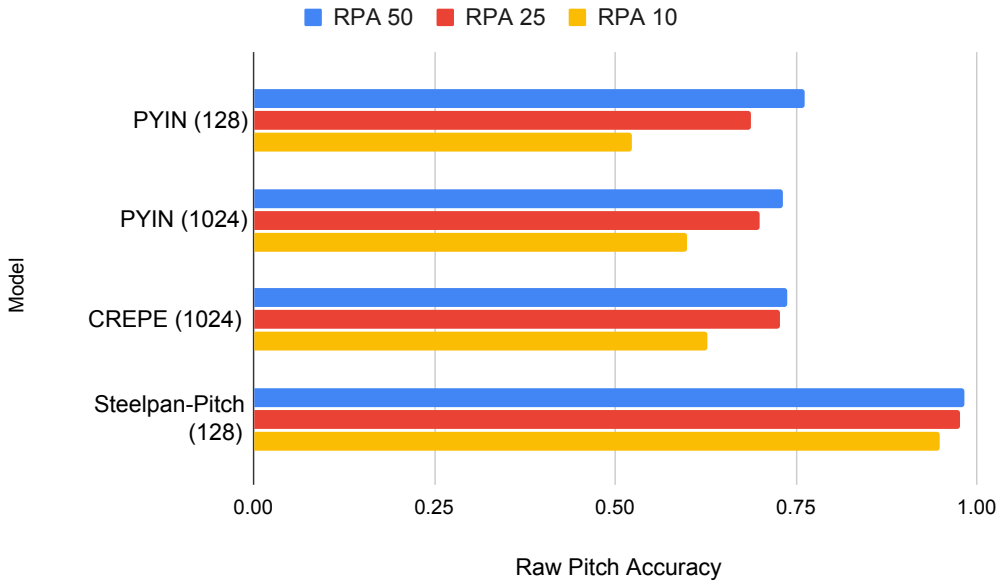


Figure 3.5: Raw Pitch Accuracy on SASS-E test set.

Model	Frame length	Params	RPA Threshold 50 cents	25 cents	10 cents
PYIN	128	-	0.761 ± 0.0009	0.688 ± 0.0010	0.523 ± 0.0012
PYIN	1024	-	0.731 ± 0.0015	0.699 ± 0.0016	0.599 ± 0.0017
CREPE	1024	22.2M	0.738 ± 0.0015	0.727 ± 0.0015	0.626 ± 0.0016
Steelpan-Pitch	128	1M	**0.982 ± 0.0003**	**0.976 ± 0.0004**	**0.948 ± 0.0005**

Table 3.1: Raw Pitch Accuracies and their standard deviations.

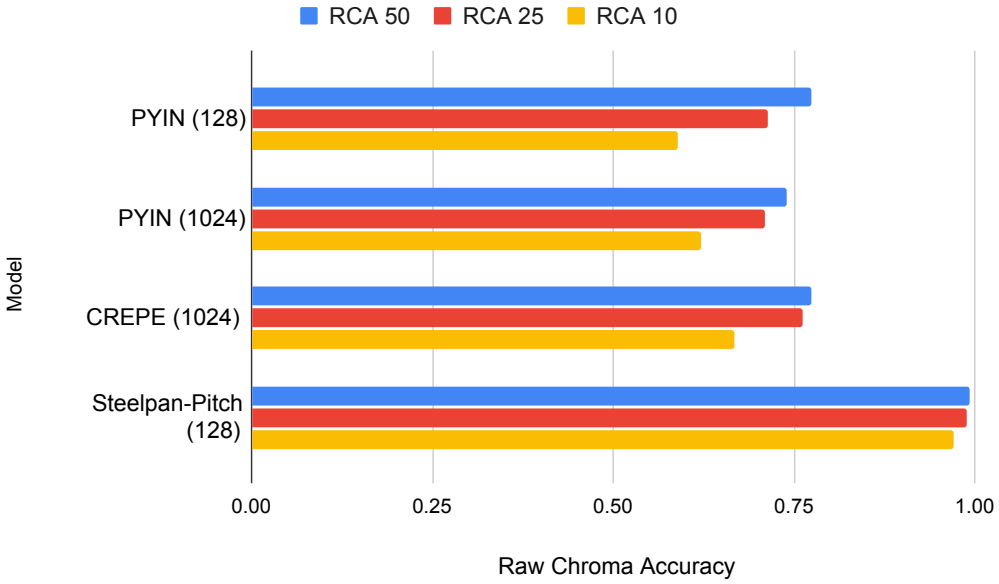


Figure 3.6: Raw Chroma Accuracy on SASS-E test set.

3.5.4 Frame Length Comparison

In adapting Steelpan-Pitch from Crepe's architecture, one of the most important changes made was changing the frame length of the input signal from 1,024 samples to 128 samples. The networks are designed to work on audio with a samplerate of 16 kHz. At this samplerate, 1,024 samples is 64 milliseconds. This is well beyond the generally accepted 10 ms threshold for human perception of a signal [136], [84]. If one were to implement CREPE in a realtime situation, the minimum latency would be 64 ms which is readily apparent to human users. In designing Steelpan-Pitch, I decided to work towards making a system that is ready for realtime processing situations by reducing the size of the input frame length while still maintaining a high level of accuracy.

To demonstrate this, I trained different versions of the Steelpan-Pitch architecture

Model	Frame length	Params	50 cents	RCA Threshold 25 cents	10 cents
PYIN	128	-	0.774 ± 0.0001	0.713 ± 0.0010	0.589 ± 0.0011
PYIN	1024	-	0.739 ± 0.0015	0.709 ± 0.0016	0.620 ± 0.0017
CREPE	1024	22.2M	0.773 ± 0.0014	0.761 ± 0.0014	0.668 ± 0.0016
Steelpan-Pitch	128	1M	**0.992 ± 0.0002**	**0.988 ± 0.0003**	**0.970 ± 0.0004**

Table 3.2: Raw Chroma Accuracies and their standard deviations.

on SASS-E. All versions of the system are based on the proposed architecture as in Fig. 3.2. For each different version, the size of the input layer is changed to a value from [64, 128, 256, 512, 1024] while the rest of the architecture remains the same. The change in input size causes a change in the dimensions at each layer, but the number of filters at each layer, the filter sizes, maxpooling, and other parameters are all kept consistent in order to make the models as comparable as possible. Each of the architecture were then trained on the SASS-E training set with the pitch shifting data augmentation as in 3.5.1 using appropriately sized audio frames.

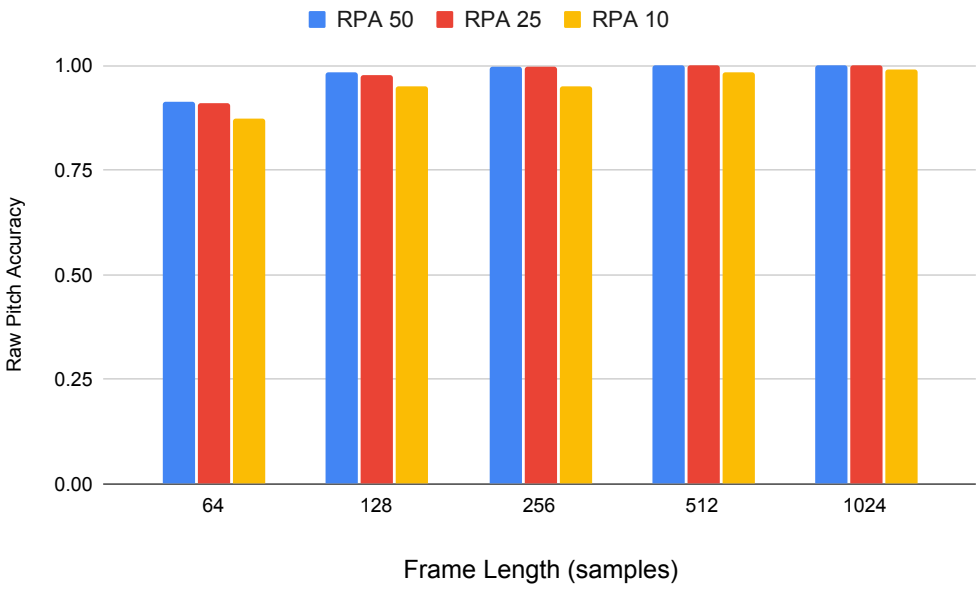


Figure 3.7: RPA for different frame lengths.

The results of the experiment are shown in Table 3.3 and Figs 3.7 and 3.8. As expected, the longer the input frame length, the better the network performs. However, we can see that there is not a significant reduction in 50-cent raw pitch accuracy until the frame length is reduced to 64 samples. With a 50-cent raw pitch accuracy of 0.982

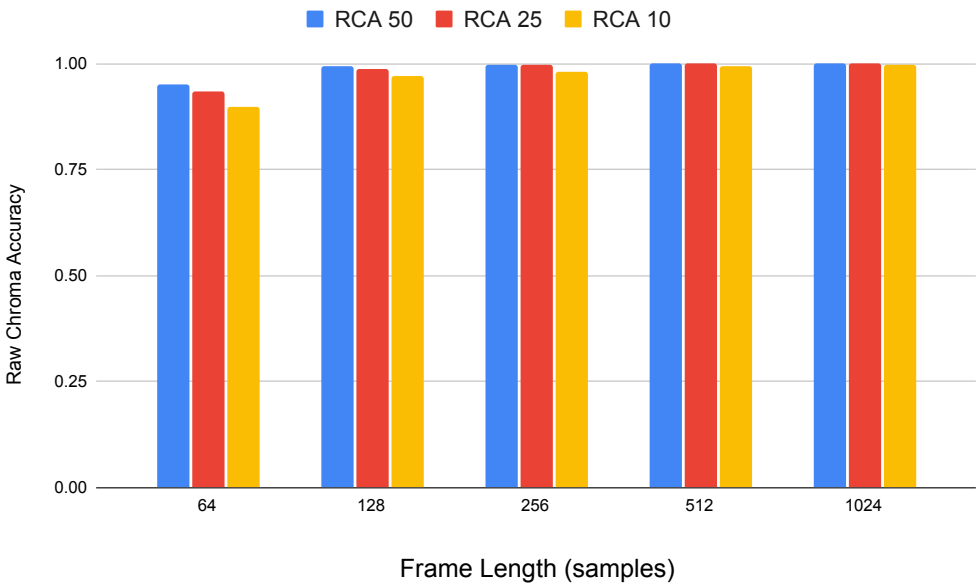


Figure 3.8: RCA for different frame lengths.

and raw chroma accuracy of 0.992, a frame length of 128 samples was selected as the optimal balance between latency and accuracy. At the system's 16 kHz samplerate, 128 samples is 8 ms in length which is within the tolerance of human perception for latency. Table 3.3 also shows that reducing the frame length can also significantly reduce the number of network parameters. Reducing the frame length from 1,024 samples to 128 results in a 56% reduction in parameters while sacrificing less than two percentage points of 50-cent threshold raw pitch accuracy. A frame length of 64 samples was not selected because, despite further reducing the latency to 4 ms, there is a significant drop in accuracy and only a 4% more reduction in parameters than a 128 sample frame length. Due to these factors I determined a frame length of 128 samples to be the optimal balance between accuracy, number of parameters, and latency.

3.5.5 Generalization

In order to demonstrate the ability for Steelpan-pitch to generalize to other instruments outside of SASS-E I also evaluated it on samples recorded from the commercial sample library Andy Narell Steel Pans – The Ellie Mannette Collection produced by Ilio. These samples were recorded at eight velocity levels per note across the entire

Frame length (samples)	Frame length (ms)	Params	Threshold (cents)	RPA	RCA
64	4	0.9 M	50	0.912 ± 0.00065	0.949 ± 0.00051
			25	0.910 ± 0.00067	0.935 ± 0.00057
			10	0.872 ± 0.00086	0.899 ± 0.00070
128	8	1.0 M	50	**0.982 ± 0.00031**	**0.992 ± 0.00021**
			25	**0.976 ± 0.00036**	**0.988 ± 0.00025**
			10	**0.949 ± 0.00052**	**0.970 ± 0.00040**
256	16	1.2 M	50	0.996 ± 0.00015	0.998 ± 0.00010
			25	0.996 ± 0.00016	0.998 ± 0.00011
			10	0.951 ± 0.00051	0.980 ± 0.00034
512	32	1.6 M	50	0.999 ± 0.00008	0.999 ± 0.00007
			25	0.999 ± 0.00008	0.999 ± 0.00007
			10	0.984 ± 0.00030	0.993 ± 0.00020
1024	64	2.3 M	50	0.999 ± 0.00016	0.999 ± 0.00015
			25	0.999 ± 0.00016	0.999 ± 0.00015
			10	0.990 ± 0.00062	0.996 ± 0.00036

Table 3.3: Comparison of RPA and RCA for different frame lengths.

range of the instrument in Ableton Live at 48 kHz/24 bits and later downsampled to 16 kHz. This set totaled 232 audio samples. Since this is a commercial sample library, I cannot include the audio in SASS-E and only use it here to demonstrate Steelpan-Pitch's performance on a steelpan it never analyzed in training.

Model	Frame Length	Metric	50 cents	Threshold 25 cents	10 cents
PYIN	128	RPA	0.848 ± 0.0032	0.726 ± 0.0041	0.502 ± 0.0046
		RCA	0.858 ± 0.0032	0.762 ± 0.0039	0.594 ± 0.0044
PYIN	1024	RPA	0.825 ± 0.0097	0.765 ± 0.0108	0.607 ± 0.0125
		RCA	0.825 ± 0.0097	0.767 ± 0.0108	0.650 ± 0.0122
CREPE	1024	RPA	**0.872 ± 0.0046**	0.815 ± 0.0054	0.627 ± 0.0067
		RCA	0.916 ± 0.0039	0.838 ± 0.0051	0.662 ± 0.0066
Steelpan-Pitch	128	RPA	0.862 ± 0.0048	**0.833 ± 0.0052**	**0.723 ± 0.0062**
		RCA	**0.959 ± 0.0028**	**0.945 ± 0.0032**	**0.880 ± 0.0045**

Table 3.4: RPA and RCA accuracies and standard deviations of PYIN, CREPE, and Steelpan-Pitch on novel steelpan samples.

The results of this test are shown in Table 3.4 and Figs 3.9 and 3.10. Although CREPE slightly outperforms Steelpan-Pitch at the RPA 50 cent metrics, Steelpan-Pitch performs the best for the rest of the metric categories. Furthermore, Steelpan-Pitch's performance drops off the significantly less than PYIN's and CREPE's at 10 cent thresholds. The accuracy of Steelpan-Pitch does drop somewhat from its performance on the SASS-E test set. However, the results here do show that Steelpan-Pitch

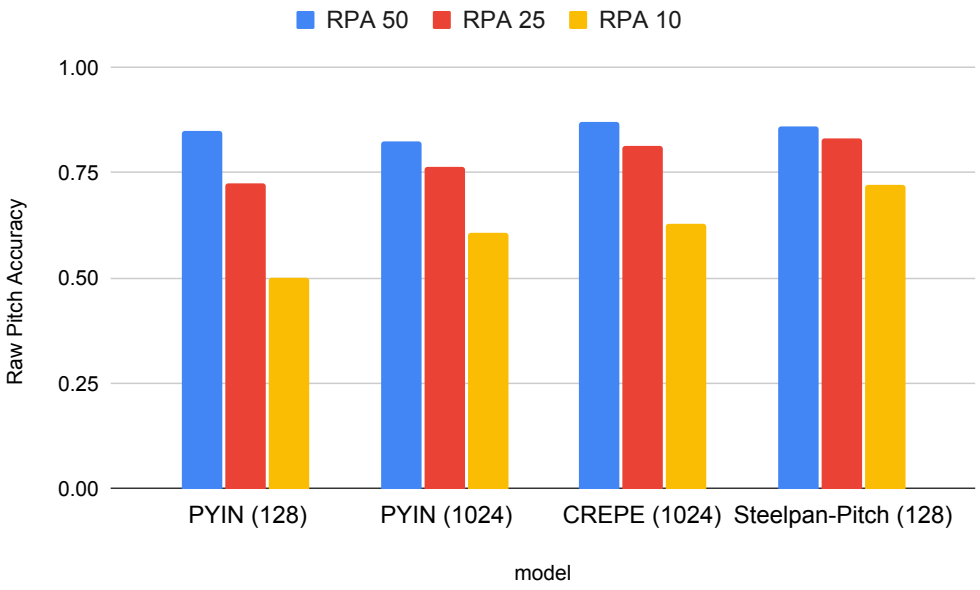


Figure 3.9: RPA on novel steelpan samples.

generalizes to other steelpans beyond those in SASS-E. Steelpan-Pitch's RCA results in particular are excellent with of 95% accuracy within 50 cents of the chroma value. This shows a significant portion of Steelpan-Pitch's errors on this instrument are octave errors. This further validates our initial simplifying assumption for generating the training targets. The ability for Steelpan-Pitch to generalize will likely continue to improve as SASS-E is expanded and Steelpan-Pitch is trained on a wider variety of steelpans.

3.6 Conclusion and Future Work

In this chapter I presented a new steelpan-specific data-driven method for pitch detection and a new audio dataset consisting of steelpan audio samples. Steelpan-Pitch shows that by limiting the scope of a pitch detection system, it can outperform state-of-the-art systems like CREPE and PYIN despite working on short audio frames with 8 ms of audio. While the performance does suffer to an extent on novel steelpan audio, Steelpan-Pitch is not simply overfitting to the dataset since it still outperforms the CREPE and PYIN on audio from a steelpan outside the dataset.

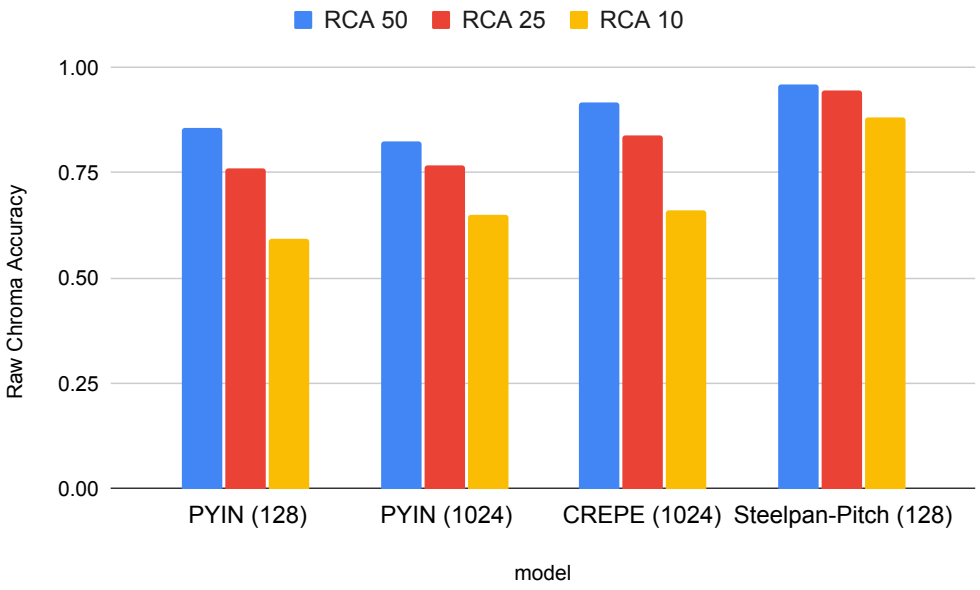


Figure 3.10: RCA on novel steelpan samples.

3.6.1 Future of SASS-E

In the future I plan to expand the SASS-E dataset by adding more instances of tenor steelpans, instances of all members of the steelpan family, and to add more articulation types using mallets like those used in Chapter 2. The tenor steelpan is the main melodic instrument in the steelpan family and the most common, but the other variations should also be represented. Since the acoustics of each different instrument and note layout vary, it will be ideal to include at least 2-3 instances of each type of steelpan as it is added to the dataset. The other primary members of the steelpan family of instruments include (from high to low voice): double tenor, double second, guitar, cello, tenor bass, and bass steelpans.

The type of mallet used to activate a steelpan note can have a drastic effect on the timbre of the instrument [64]. Steelpans are typically played with rubber tipped mallets, but there are many alternatives that are becoming increasingly common in performance such as cardboard tubing, chopsticks, brushes, yarn-wrapped mallets, and bundle rods. The dataset should also be augmented to include samples with as wide a variety of mallets as well.

3.6.2 Future of Steelpan-Pitch

The accuracy results for Steelpan-Pitch demonstrate that a custom-trained neural net for instrument-specific pitch detection can significantly outperform general pitch methods such as PYIN or CREPE. The next step for Steelpan-Pitch, however, is to further train it on a wider variety of steelpans covering the full range of the steelpan family. This is necessary in order to improve the generalization of the system to all types of steelpans. While the model currently performs well on other instances of tenor steelpans, it is likely that Steelpan-Pitch will not perform well on lower voiced steelpans.

Steelpan-Pitch is trained only on recently tuned instruments, but old, "janky" steelpans have an iconic sound of their own. The overtones are often wildly out of tune from the fundamental, but humans can still generally recognize the pitches. I do not yet have any samples from such instruments in the dataset to evaluate any of the pitch detection methods on, but in all likelihood the accuracy will suffer. With a data-driven approach like in Steelpan-Pitch, these kinds of sounds can be incorporated into the representation by adding them to the training dataset. After enough such "janky" instruments have been included, the system should be able to perform well on these instruments as well. Due to the esoteric nature and availability of these instruments, they are also unlikely to gain representation in general audio datasets.

The goal for Steelpan-Pitch is to maintain a well performing, but lightweight model that can be used for realtime pitch detection. The reason for this is that as a regular performer of electroacoustic steelpan music, my longterm goal is to build a low latency, realtime steelpan pitch transcription system that can analyze my playing live and perform audio processing tasks based on what I plays.

Steelpan-Pitch is provided as a pre-trained model for analyzing steelpan audio. However, another plan is to extend the system so that another performer can easily customize the model for their specific instrument. In this case, the provided pre-trained weights would be used for transfer learning. The final training for the system would be on audio provided by the performer from their specific audio setup to improve adaptability to other instruments and audio situations.

Chapter 4

Realstretch: Time-compensating realtime Time-stretching

4.1 Introduction

Similar to most percussion instruments, the steelpan is not capable of sustaining notes. When a note is struck, it sounds for as long as it sounds and the performer has minimal direct influence on its duration. Advanced players adjust their playing technique in order to subtly influence the timbre and activation of a note, but this effect is minimally effective.

Percussionists approximate sustained notes through rolls—that is, striking a note repeatedly. When done well, this imparts a suggestion of a sustained note. But steelpan notes have such obvious transient sounds that no matter how fast and gracefully a steelpannist rolls on a note, it will always have the essences of what it actually is: a note being struck repeatedly. However, what if a steelpan note (or any sound for that matter) could be sustained using audio signal processing?

This is a problem that has been explored in various ways. A simple solution is to use reverb (possibly with other effects) to create a sense of sustain. I used reverb in this way for my piece, *Steelpanopticon*, presented in Chapter 8. An audio effect designed specifically for this is the spectral freeze. This is an audio effect where an audio grain (very short segment) of a sound is captured and used to create "infinite" sustain for a note by repeatedly outputting that grain. There are various DSP techniques to implement a spectral freeze so that it sounds less unnatural and has a slow morphing quality to it. An envelope can also be applied to the frozen

sound in order to give it a specific length. This is an effective solution, but only utilizes a small portion of the source note's sound. It takes a more in-depth approach to follow the natural evolution of a sound.

This is the inspiration for my first audio plugin: a realtime version of the popular Paulstretch effect. Paulstretch is an extreme time-stretcher. The algorithm is designed for very long time-stretches with minimal artifacts. It has commonly been used in YouTube videos to take well known pieces of music and stretch them to be many times their original length. It gained notoriety when a video was posted stretching a Justin Bieber song to 8 times its original length. The video[1] currently has 3.2 million views.

The challenge in turning Paulstretch into a realtime plugin is that it is a time-stretcher. When time-stretching has been implemented in realtime applications in the past, it has been in a time-compensating way—usually in communication technologies. When transmitting audio over a digital communication service, there is no guarantee of data throughput. In order to improve the experience for the user, the system will time-stretch the audio when there is a slow-down in the data speeds. When the data transfer speeds increase again, this time-expansion of the audio is compensated for by time-compressing the audio.

The goal here, however, was to create a time-stretcher that can operate on live, realtime audio with a constant stretch factor that also maintains temporal coherence between the source sounds and its time-stretched version. In order to accomplish this, the circular buffer, a common memory method in audio effects, needed to be modified[65]. This modification includes a different form of time compensation that allows for the time-stretch effect to run continuously as desired. The result of this work is Realstretch, an AU/VST plugin that performs extreme time-stretching on live, realtime audio.

4.2 Background

4.2.1 Circular buffers

In digital signal processing, a circular buffer is a memory buffer that acts as a single contiguous memory allocation which treats the first and last memory spaces as neighbors. In a standard implementation, there is a write pointer and a read pointer. The

[1] `https://youtu.be/QspuCt1FM9M`

write pointer indicates which memory space the next input audio sample should be written to while the read pointer indicates which stored audio sample should be read next for output. One sample is written and one sample is read for each process cycle and then both pointers are iterated to the next memory space. If one of the pointers reaches the end of the buffer, it wraps back to the beginning (see Fig. 4.1 for a visual representation).

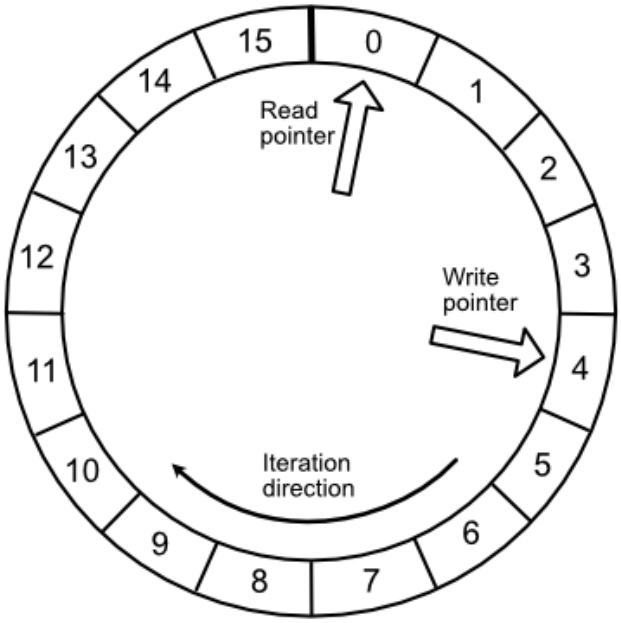


Figure 4.1: Visual representation of a circular buffer.

4.2.2 Time-stretching

Time-stretching is a common audio effect used in digital audio processing. Generally, time-stretching is used to change the length of an audio signal while maintaining the signal content. Many contemporary systems are designed to rescale the signal while maintain the original pitch, not affecting other spectral characteristics, and minimizing the introduction of audio artifacts. There are two main methods upon which time-scale modification systems are based: audio granulation and the phase vocoder.

Granulation-based time-stretching

Audio granulation is a process by which an audio signal is divided into very short segments, called grains, for processing [111]. These grains can vary in length, but are generally in the range of 1-100 milliseconds in length. A very simple method of time-stretching is to repeat grains using an overlap-and-add (OLA) methodology. A modification to the OLA method called synchronized overlap-and-add (SOLA) was developed by Roucos and Wilgus [117]. This system uses cross-correlation to maximize the similarity of the overlapped portions when performing the OLA.

Moulines et al., refined this process with time-domain pitch-synchronous overlap-and-add (TD-PSOLA) [42], [85]. The TD-PSOLA algorithm takes into account the momentary pitch of a signal to better define and align audio grains in the OLA process. Another variant of the SOLA algorithm is waveform-similarity-based overlap-and-add (WSOLA) proposed by Verhelst and Roelands [132]. Instead of focusing on pitch, the WSOLA algorithm maximizes waveform similarity to the original signal at segment joints.

The Discrete Fourier Transform

The discrete Fourier transform (DFT) is a tool that is frequently used in signal processing and is a intrinsic part of phase vocoder-based time-stretching. The DFT converts a time-domain audio signal into a sequence of complex numbers, $X_0, X_1, \ldots, X_{N-1}$, according to the formula

$$X_k = \sum_{n=0}^{N-1} x(n) \cdot e^{-\frac{i2\pi}{N}kn} \tag{4.1}$$

[127]. The subscripts, k, are frequently referred to as bin numbers and are each associated with a bin frequency dependent upon the number of bins, N, and the sampling rate of the discrete audio signal. If f_s is the sampling frequency, then the bin frequency for bin k is given by

$$f_k = k \cdot \frac{f_s}{N}. \tag{4.2}$$

The complex output of the DFT is difficult to visualize and instead we commonly visualize the bin magnitudes, $|X_k|$, for each frequency bin. As shown in Eq. 4.2, each of these bins are associate with a specific frequency. The magnitude gives an approximation of the strength of this frequency in the input signal to the DFT. When

combined with the complex phase for each bin, $\phi(X_k)$, the original signal, $x(n)$, can be reconstructed using the inverse DFT:

$$x_n = \sum_{k=0}^{N-1} |X_k| \cdot \phi(X_k). \tag{4.3}$$

This equation is frequently written in a different, but equivalent way. I have written it this way in anticipation of how the Paulstretch algorithm works.

Phase vocoder-based time-stretching

The phase vocoder is a system for decomposing audio into its frequency components [38] and then reconstituting it using the short-time Fourier transform (STFT) [104]. The STFT is term used when applying the DFT to a short span of an audio signal. Short, overlapping frames of the input audio signal are analyzed using the STFT, processed, and then resynthesized into output audio frames.

Time-stretching with the phase vocoder can be accomplished by mismatching the spacing of the analysis and synthesis audio frames. If the spacing of analysis audio frames is smaller than the spacing of the synthesis audio frames, then upon reconstruction the resynthesized output will be a stretched version of the input (see Fig. 4.2).

The problem with the traditional phase vocoder approach to time-stretching is that the frame spacing mismatch causes misalignments in the phase information which introduces audio artifacts often referred to as "phasiness" [57]. The issue of phase unwrapping and calculating instantaneous frequency has led to many variations on the phase vocoder to improve time-stretch performance [58], [36], [105], [106].

The Paulstretch algorithm

The Paulstretch algorithm[2] is a time-stretch algorithm developed by Paul Nasca for extreme time-stretching of audio signals. It uses a fast Fourier transform (FFT) approach with mismatched frame spacing like the phase vocoder. However, instead of attempting to maintain phase alignment, the phase information is randomized.

[2]Presented at: `http://hypermammut.sourceforge.net/paulstretch/`

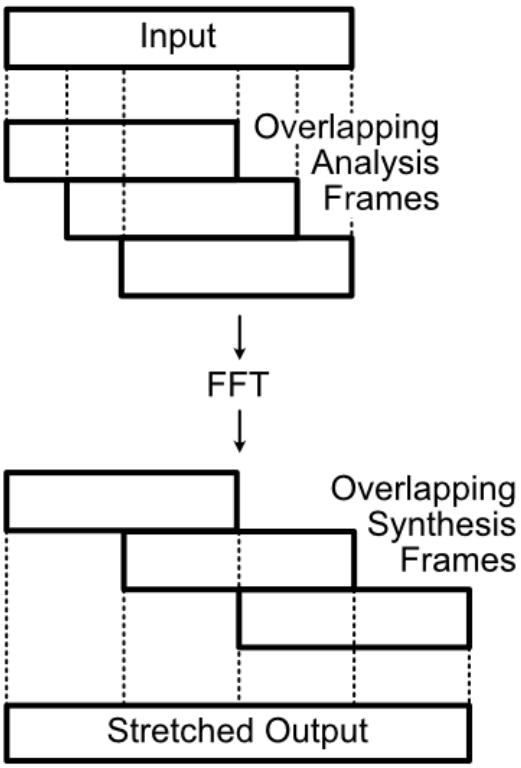


Figure 4.2: Phase vocoder time-stretching.

The reconstruction equation becomes a modified version of the inverse DFT

$$x'_n = \sum_{k=0}^{N-1} |X_k| \cdot \phi_r(k) \tag{4.4}$$

where $\phi_r(k)$ is a random phase value for each frequency bin k such that $0 \leq \phi_r(k) < 2\pi$. The phase randomization eliminates the phase alignment artifacts, but also spreads the energy of an input frequency bin across nearby output frequency bins. When the analysis frame size is small, this effect is called "whisperization".

Increasing the analysis frame size (and thus the number of frequency bins) improves pitch coherence and is the basis of the Paulstretch algorithm. The pitch coherence is improved since for a larger DFT, each bin represents a smaller frequency range. Thus when the energy gets spread to nearby bins, it is spread accross a smaller frequency range [74]. Typical phase vocoder frame sizes are between 1 and 20 mil-

liseconds. Typical Paulstretch frame sizes start at about 20 milliseconds and can be several seconds long. Long frames sizes like these eliminate the whisperization effect, but also introduce a temporal smearing effect where instantaneous pitch information is scattered across the synthesis audio frame.

4.3 Problem Formulation

Time-stretching as a live realtime audio effect has two main problems. The first is a memory problem. In a naive implementation, constant time-stretching of live audio has an unbounded memory requirement. The second is a temporal-perceptual problem. In the same naive implementation, the longer a constant time-stretching system runs the greater the gap between a source sound and its time-stretched version. These problems are discussed in more detail below.

4.3.1 The memory problem

A traditional time-stretching process in a live realtime application has a fundamental problem. Since the effect "expands" time, the output will always be longer than the input. If, for example, a system is stretching audio by a factor of two (doubling its length), then an input sound that lasts for five seconds will have a stretched output that lasts for ten. If the system were to begin outputting the stretched version as soon as it received the input, when the input sound finished five seconds later the system would only have played back half of the stretched version. At any point in time only half the processed input will have been output. The remaining half must be stored in a memory buffer. As the stretch factor increases, the amount of audio that must be stored increases.

This is not an issue for an offline, non-realtime time-stretch effect since the system could process the audio at whatever pace it can handle. In a live, realtime system, however, the amount of time for which the effect will run is unknown. If the system were to use a standard circular buffer to process the audio, the buffer size required would be unknown. The longer the system runs, the more memory that would be required. Regardless of how much memory is allocated, it would be possible for it to be insufficient. Live audio effects need to be able to operate stably for an arbitrary amount of time, but this memory problem prevents that in a naive time-stretching implementation.

4.3.2 The temporal-perceptual problem

There is also a temporal-perceptual problem associated with the increasing delay between input audio and the corresponding time-stretched output. As discussed, if a live, realtime time-stretching system is stretching the audio by a factor of two, then it takes 10 seconds to playback five seconds wroth of input. This means that if a sound is made five seconds after the system is started, there will be a 10 second delay before playback of its time-stretched version has begun. As the system runs for longer, this distance between input and output grows larger. This asynchrony between input and output is a significant temporal-perceptual problem.

4.3.3 An ideal system

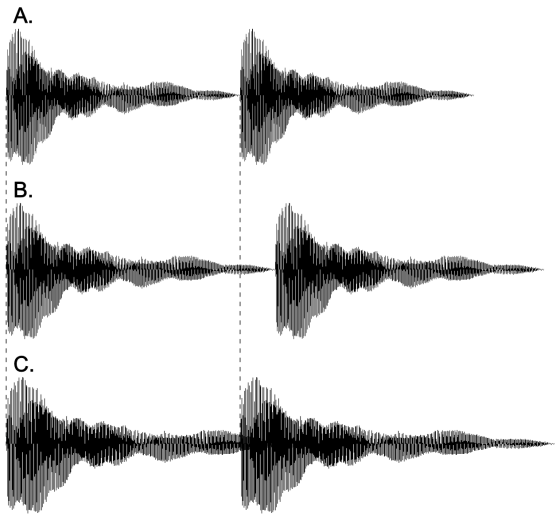


Figure 4.3: A. Original input. B. Traditional time-stretch. C. Overlapping live realtime time-stretch.

The goal of the technique presented here is to enable a system that addresses these issues. It should run with a fixed amount of memory and maintain the temporal relationship between the input and stretched output. The onsets of the input sound and the time-stretched version should be temporally aligned. This then implies that even if two input audio sounds do not overlap, the tails of their stretched versions may overlap. See Fig. 4.3 for a comparison of a traditional time-stretch and the how this ideal system would keep the time alignment between the original and stretched

onsets. Note in the figure that version C keeps the alignment of the onsets by allowing the time-stretched version of the first sound to overlap with the onset of the second sound.

4.4 Methods

The solution presented here employs a circular buffer with modified read and write pointer behaviors to solve the previously discussed issues. The read pointer iterates through in a continuous fashion, but at a rate that is the inverse of the stretch factor. E.g., if the stretch factor is four then the read pointer iterates at one-quarter of the system audio rate. The data read from the buffer is time-stretched and then output from the system. After an audio sample is read and processed, the value at its buffer location is cleared.

In contrast to how circular buffers typically operate, the buffer write process is inactive to start. When the audio input level crosses the activation threshold, the write point is repositioned to the read pointer's location and then audio input is written to the buffer at the system audio rate. Due to the periodic re-synchronization with the read pointer, it is necessary to write to the buffer in a non-destructive (i.e., additive) fashion.

When the audio input level drops below the deactivation threshold, the write process and iteration of the write pointer stop. When the input signal level crosses the activation threshold again, the write point re-synchronizes again and the write process resumes.

4.5 Implementation

An implementation pairing this system with the paulstretch time-stretch algorithm, called Realstretch, is available on GitHub[3]. This section discusses its design. Realstretch was implemented in MATLAB using the audio toolbox to generate AudioUnits (AU) and Virtual Studio Technology (VST) plugin versions. The user interface has six controllable parameters (shown Fig. 4.4).

[3]https://github.com/malloyca/realstretch/

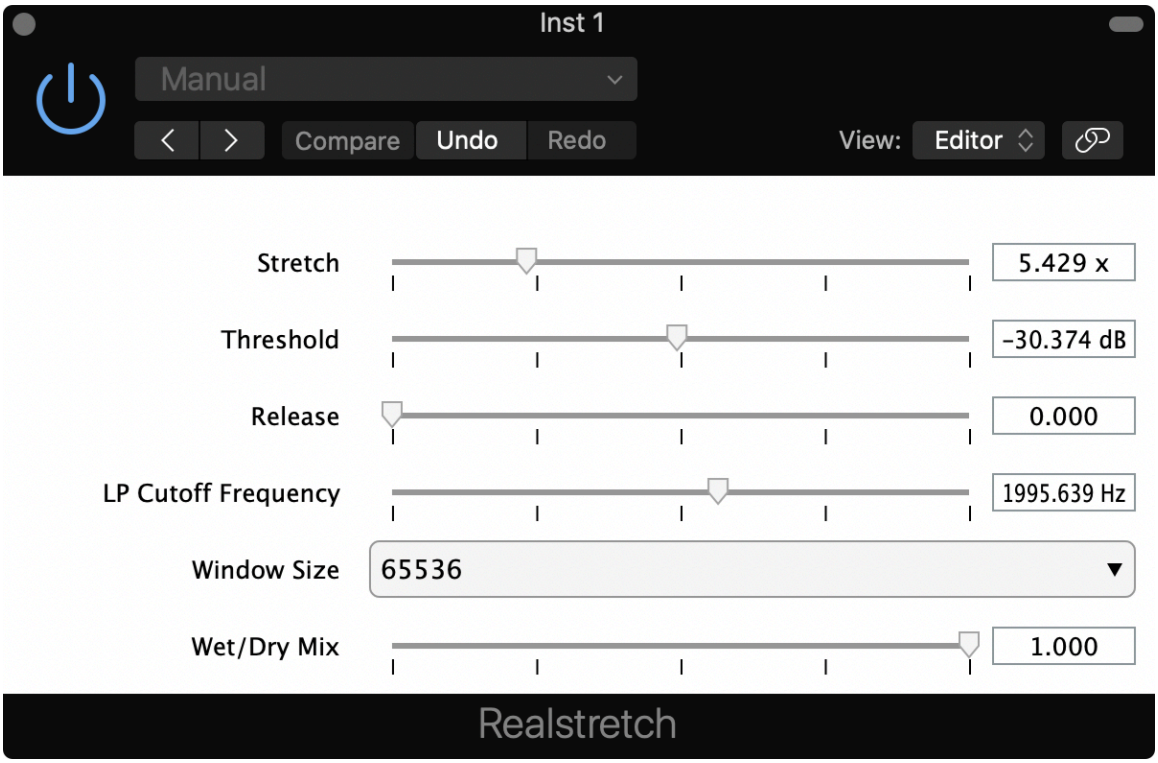


Figure 4.4: Realstretch user interface.

4.5.1 Stretch

The 'Stretch' parameter varies the stretch factor from a minimum value of 1 to a maximum of 20. When set to 1, the plugin does not perform time-stretching, but the phase information is still randomized. At a small window size, this will result in a whisperization effect. At larger window sizes, it creates what could be described as a spectral blur effect. The stretch parameter can be adjusted while the system is running.

4.5.2 Threshold

The 'Threshold' parameter sets the signal level required to trigger a re-synchronization. It has a range from -60 to 0 decibels relative to full scale (dBFS). This also indirectly controls the deactivation threshold which is approximately 3 dBFS lower than the activation threshold. Setting the deactivation parameter to a level lower than the activation threshold prevents multiple activations from occurring in a quick succession which can cause high frequency noise.

4.5.3 Release

The 'Release' parameter adjusts the sensitivity of the level detector to reductions in level. A low value decreases the sensitivity and a quick series of sounds will trigger a single re-synchronization (they will be stretched in sequence). A high value causes re-synchronizations to occur more often. A quick series of sounds will be stretched in parallel (overlapping).

4.5.4 Lowpass Cutoff Frequency

A -6 dBFS/octave lowpass filter was added to the system to assist in smoothing high frequency artifacts. The 'LP Cutoff Frequency' parameter adjusts the cutoff frequency from 100 Hz to 20 kHz.

4.5.5 Window Size

The 'Window size' parameter adjusts the FFT analysis window from 256 to 65,536 samples. This can be changed live, but will reset the memory buffers. A side effect of the Paulstretch algorithm is latency due to the large FFT window sizes. The latency is dependent on the length of the FFT analysis window size. This varies with the samplerate, but for 44.1 kHz it ranges from 5.8 ms to 1.49 s. The Paulstretch algorithm was selected as the time-stretch portion of this system for aesthetic reasons despite the latency. A different time-stretch algorithm could be used in place of Paulstretch to minimize latency.

4.5.6 Wet/Dry

The 'Wet/dry' control adjusts the balance of processed and unprocessed signal being output.

4.6 Discussion and Future Work

Realstretch was originally implemented in MATLAB using the MATLAB Audio Toolbox because it was entered in the Audio Engineering Society Student Competition: MATLAB Plugin in 2020, which required all entries be programmed in MATLAB. It won third place in that competition. MATLAB is a useful environment for developing

prototypes, but there are many limitations. The next step in developing Realstretch is to re-implement it in C++ using the JUCE Framework. JUCE is a C++ library designed for building audio applications like AU/VST plugins.

Once ported to C++ and JUCE, there are several other improvements to the algorithm that can be made. Realstretch would benefit greatly from having multi-threading functionality added. When the FFT analysis window is large, Realstretch has high, unbalanced computational. With multi-threading, the FFT computations can happen in background processing which will spread that load over time. Realstretch should also report its latency to the digital audio workstation (DAW). The current version does not report any latency so the DAW cannot compensate for the latency. Once reporting is implemented, the DAW can automatically compensate which will improve the alignment between input and output.

For live performance, a lower latency version can be implemented as well. The simple way to reduce latency is to change the time-stretching algorithm. Granulation-based time-stretching methods generally have less than 1 ms in latency. The Paulstretch algorithm, however, can also be modified to reduce the latency. In the basic version of the algorithm, the latency is equivalent to the FFT window size. If the window is is N samples long, then the added latency from the Paulstretch algorithm is N/f_s seconds. For example, a window size of 8,192 samples at 48 kHz samplerate will add approximately 171 ms of latency. The way to reduce the latency is to compute the same size FFTs on shorter audio segments with zero padding. Continuing the example, the latency can be reduced by half without greatly affecting the timbre by splitting that between two 8,192 sample FFTs where the first uses the first $N/2$ samples (then zero padded) and the second FFT uses the last $N/2$ samples (also zero padded). The zero padding is important to maintain the pitch coherence of this time-stretch method. With this method, the timbre of Paulstretch with larger window sizes can be achieved with lower latency. The drawback, however, is that this increases the computational costs. The next version of Realstretch will include this lower latency approach as well as other time-stretch algorithms so that users can choose the best approach for their needs.

Realstretch is a successful experiment in realtime audio processing. Time-stretching had never been implemented in this way before and has already been used in several recordings as well as live performances.

4.7 Conclusion

In this chapter, I presented Realstretch, a realtime time-stretcher based on the Paulstretch algorithm. In order to implement time-stretching as a realtime effect, I had to develop a new buffer management technique based on the circular buffer. This audio effect was inspired by the desire to make percussive sounds sustain and an affection for Paulstretch.

Chapter 5

HarmonEQ: Chord-based Equalization

5.1 Introduction

In this chapter I present a new approach to audio equalization that reframes the controls in terms of musical notes and harmony. This new approach enables new methods of interaction as well as the automation of equalizer settings to match the harmonic content by using automatic chord detection.

Since their introduction, innumerable versions of parametric equalizers have been developed [78]. They have become a ubiquitous tool in music production with both analog and digital implementations. Despite their wide usage, the parametric interface used to control them remains rooted in the equalizer's electrical engineering origins. The primary controls for each filter are frequency, gain, and Q factor. This is true even in modern digital implementations with advanced graphical user interfaces such as the FabFilter Pro-Q 3[1]. While audio engineers who use equalizers on a regular basis become familiar with frequency ranges and learn how to EQ around them, if they want to center a filter on the frequency for a specific musical note they still have to look up that frequency. Matters are further complicated if they want to apply filters to the frequencies of a full chord — especially across multiple octaves. Changing notes or chords results in many convoluted automations in the DAW. By providing the user the ability to control the center frequency of an EQ filter directly, the user has the freedom to set the frequency parameter where they wish, but is

[1] `https://www.fabfilter.com/products/pro-q-3-equalizer-plug-in`

forced to think in terms of abstract frequency. However, most musicians do not think in terms of frequency – they think in terms of notes and chords.

This chapter first introduces the concept of parametric equalizer frequency controls in terms of notes and harmony instead of frequency directly. Although parametric equalizers have been available for about 50 years, to the best of my knowledge there has never been an equalizer before with a note and chord control paradigm instead of direct frequency.

This is demonstrated through HarmonEQ, a novel harmonic equalizer that reframes the traditional frequency control for the parametric equalizer in terms of musical notes and chords. The frequency controls are removed in favor of 'Root Note' and 'Chord Type' controls. When either of these parameters are changed, HarmonEQ updates the center frequency for all of the filters automatically to match the selected chord. Thus although there is a loss of fine grained control, the user gains the ability to quickly and easily change many filters simultaneously to reflect the musical harmony.

Reframing the frequency controls in this way allows for the last main contribution of this HarmonEQ: an automatic mode in which a realtime automatic chord detection system analyzes incoming audio and updates the root note and chord type parameters automatically. The inclusion of this automatic mode makes the plugin an intelligent, adaptive audio effect.

HarmonEQ won first place in the 2021 Audio Engineering Society Student Competition: MATLAB Plugin. HarmonEQ was developed using MATLAB and the MATLAB Audio Toolbox [32]. The code is open source and available on both GitHub[2] and MathWorks File Exchange[3]. Also available is a YouTube video[4] made for the Audio Engineering Society competition that explains how HarmonEQ works, some of its potential applications, and gives a demonstration of the automatic chord detection mode.

[2] `https://github.com/malloyca/HarmonEQ`

[3] `https://www.mathworks.com/matlabcentral/fileexchange/92643-harmoneq`

[4] `https://youtu.be/qlQ4hnX0gDU`

5.1.1 Background

Parametric Equalizers

Most research regarding digital equalizers has been focused on filter design and the equations necessary to implement them. The most common way to implement equalization filters is through the use of biquadratic (biquad) filters. Christensen presents an in-depth analysis of biquads in [24]. However, higher order designs are used as well [95]. Often an idealized version of the parametric equalizer is sought where adjusting one parameter has no perceptible effect on other filter properties [110], [138]. However, research has also been conducted on how best to interact with a digital parametric equalizer using a graphical interface [60]. Another trend in equalizer design has been the development of dynamic equalizers that incorporate concepts from dynamics processing [137]. Interesting recent applications of neural networks to equalization also include the use of differentiable biquads in parametric equalizer matching [92].

There have been equalizers based on the harmonic series, such as the similarly named Voxengo HarmoniEQ[5] and Hunterwolf HW-HarmonEQ-1[6]. These two equalizers work in a different way from my HarmonEQ. The Voxengo HarmoniEQ works more like a traditional equalizer, but claims to make harmonic enhancements to the audio. This appears to be through nonlinear processing that adds harmonics to the audio. The Hunterwolf HW-HarmonEQ-1 is described as a "single band filter" that allows the user to "enhance or dim relative resonances up to the tenth harmonic partial, with selectable sharpness". HW-HarmonEQ-1 clearly processes based on the harmonic series. The HarmonEQ presented here, as will be described in the rest of this chapter, does not operate in the same way despite the similar name.

Automatic Chord Detection

The traditional approach to automatic chord detection is to perform spectral analysis and then compare the results to chord templates to find a match [98], [23]. The spectral analysis typically consists of applying either the Short-time Fourier Transform (STFT) or the Constant-Q Transform (CQT) and then extracting audio features from the frequency domain representation of the signal. A common method is to summarize the spectral representation of an audio signal into a chroma vector which organizes the spectral energy by pitch class [44], [59]. Another method projects the spectral in-

[5] `https://www.voxengo.com/product/harmonieq/`
[6] `https://www.hunterwolf.com/product/hw-harmoneq-1/`

formation into a Tonnetz-space for analysis [47]. These methods are often paired with Hidden Markov Models for improved performance [125], [131]. More recent methods of automatic chord detection tend to focus on Deep Learning approaches. An early method in this area was to train a convolutional neural network to perform chord detection [46]. Another method has been to combine feature extraction with deep neural networks [53]. Even the recent transformer architecture has been applied to chord detection in the form of a bi-directional Transformer [101] and as part of a curriculum learning training scheme [118].

5.2 HarmonEQ Equalizer Design

HarmonEQ uses up to 36 cascaded biquad peak filters covering a nine octave span of audio from C_1 (32.70 Hz) to B_9 (15,804 Hz). The biquad filters are defined by the transfer function:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}. \tag{5.1}$$

The biquad coefficients are calculated based on the equations from [17] and are given by:

$$\begin{aligned} b_0 &= 1 + \alpha A \\ b_1 &= -2\cos\omega_0 \\ b_2 &= 1 - \alpha A \\ a_0 &= 1 + \frac{\alpha}{A} \\ a_1 &= -2\cos\omega_0 \\ a_2 &= 1 - \frac{\alpha}{A} \end{aligned}$$

with

$$\begin{aligned} A &= 10^{\mathrm{dBgain}/20} \\ \omega_0 &= 2\pi\frac{f_0}{f_s} \\ \alpha &= \frac{\sin\omega_0}{2Q} \end{aligned}$$

where dBgain is the desired gain in dB, f_0 is the center frequency for the filter, f_s is the sampling frequency, and Q is the quality factor. Each filter has a gain range from -12 to 12 dBFS and a Q range from 0.5 to 100. Only the gain and Q values are directly controllable by the user.

The replacement for the typical frequency control is discussed below. The user interface is shown in Fig.5.1. It is important to note that the control methodology employed here can easily be used for other types of equalization filters. While we show how it can be used with peak filters, it can work with any other parametric filter type.

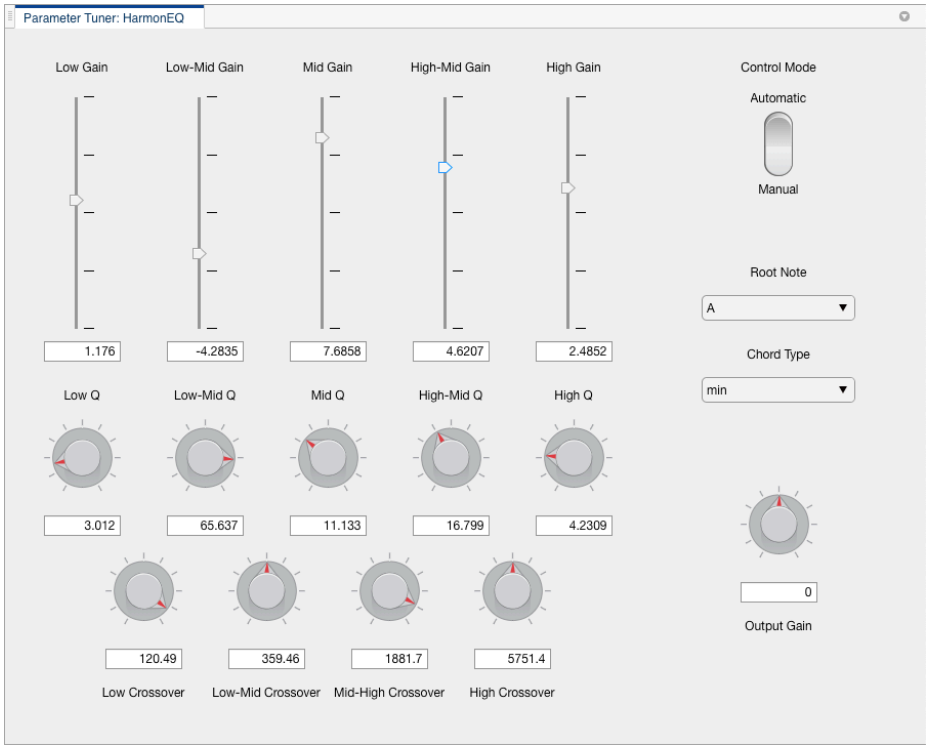


Figure 5.1: HarmonEQ user interface.

5.2.1 Filter Frequency Control Methodology

HarmonEQ uses two global parameters to control its filter frequencies: 'Root Note' and 'Chord Type'. There are four groups of nine filters spread across nine octaves of audio for a total of 36 filters. Each group of filters is associated with a part of the harmonic structure of a chord: the root, the third, the fifth, and the seventh. Note that the center frequencies of the filters are based on the musical harmony the plugin

is set to – not on the harmonic series. The root filters are always active, but the other three groups may be deactivated depending on the chord type selected. Nine octave-spaced root note filters are shown in Fig. 5.2.

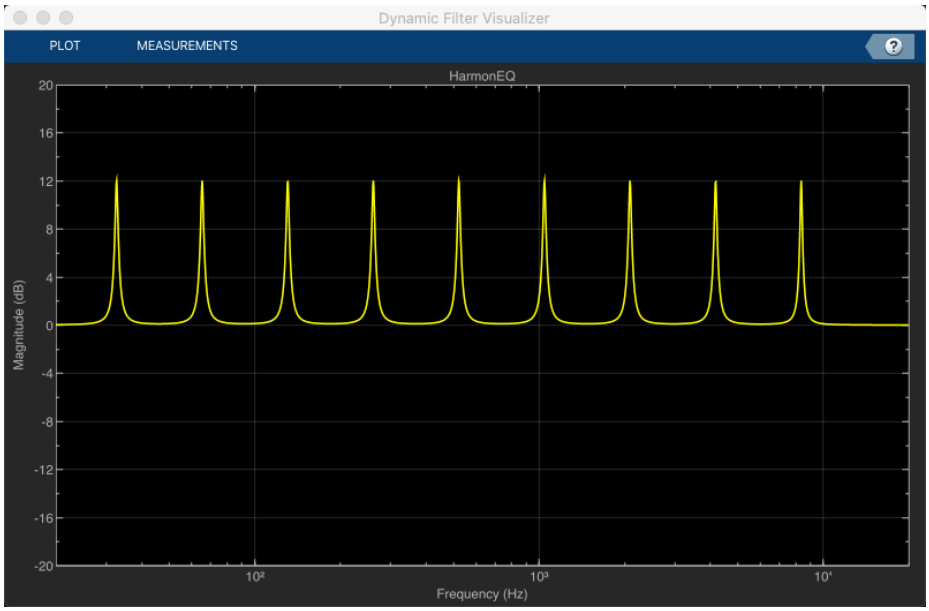


Figure 5.2: Nine octave-space root note filters.

The options available for the root note parameter are the twelve chromatic notes used in Western musical harmony: A, A♯/B♭, B, C, . . . , G♯/A♭. The frequencies of these notes used in HarmonEQ are based on the equal temperament tuning system with A = 440 Hz as a reference pitch. The available Chord Types and the associated musical intervals they contain are given in Table 5.1. The table uses standard abbreviations for musical intervals including m3 for minor third, M3 for major third, d5 for diminished fifth, P5 for perfect fifth, A5 for augmented fifth, d7 for diminished seventh, m7 for minor seventh, and M7 for major seventh.

In Western harmony, chords are made of combinations of these various musical intervals. The simplest chord included is a 5 chord, which consists of a root and a perfect fifth. Thus the filters for the thirds and sevenths of the chord are deactivated when a 5 chord is selected. The next set of chords include the root, a third and a fifth. These include the minor, major, diminished and augmented chords. For this set, the filters for the sevenths of a chord are deactivated. The last set of chord include the root, a third, a fifth and a seventh. This set includes the minor 7, dominant 7, major 7, minor 7 ♭5, and diminished 7 chords. All 36 filters are active for these chords. See Fig. 5.3 for an example with all 36 filters activated. The chord is set to

chord type	musical intervals
no chord	none
5	P5
Minor	m3, P5
Major	M3, P5
Diminished	m3, d5
Augmented	M3, A5
Minor 7	m3, P5, m7
Dominant 7	M3, P5, m7
Major 7	M3, P5, M7
Minor 7 ♭5	m3, d5, m7
Diminished 7	m3, d5, d7

Table 5.1: HarmonEQ chord types.

C minor 7 with 12 dB of gain and a Q setting of 60 for all filters. All of the chord tone frequencies are calculated based on equal temperament tuning. With Q settings less than 60, this is close enough to encompass the just intonations for chord tones. Incorporating the ability to switch between equal temperament and just intonation is beyond the scope of the current version of HarmonEQ, but something that will be considered for future iterations.

5.2.2 Crossover Controls

The nine octaves of audio are divided into five control regions: low, mid-low, mid, mid-high, and high. Each region has its own gain and Q controls for all filters that fall within the control region. The user cannot directly control the gain and Q for an individual filter. All of the plugins parameters can be automated within a DAW - including the gain and Q controls for each region, the crossover parameters, and the global root note and chord type controls.

The first octave of filters are always part of the low region, the third octave part of the mid-low region, the fifth part of the mid region, the seventh part of the mid-high region, and the ninth part of the high region. There are four crossover controls that allow the user to adjust where the split between neighboring regions is defined. Through these controls, the user can decide how much of the second octave is controlled by the low region controls and the mid-low region controls. The fourth, sixth, and eighth octave can similarly be split between the mid-low and mid, the mid and mid-high, and the mid-high and high regions respectively. By using different

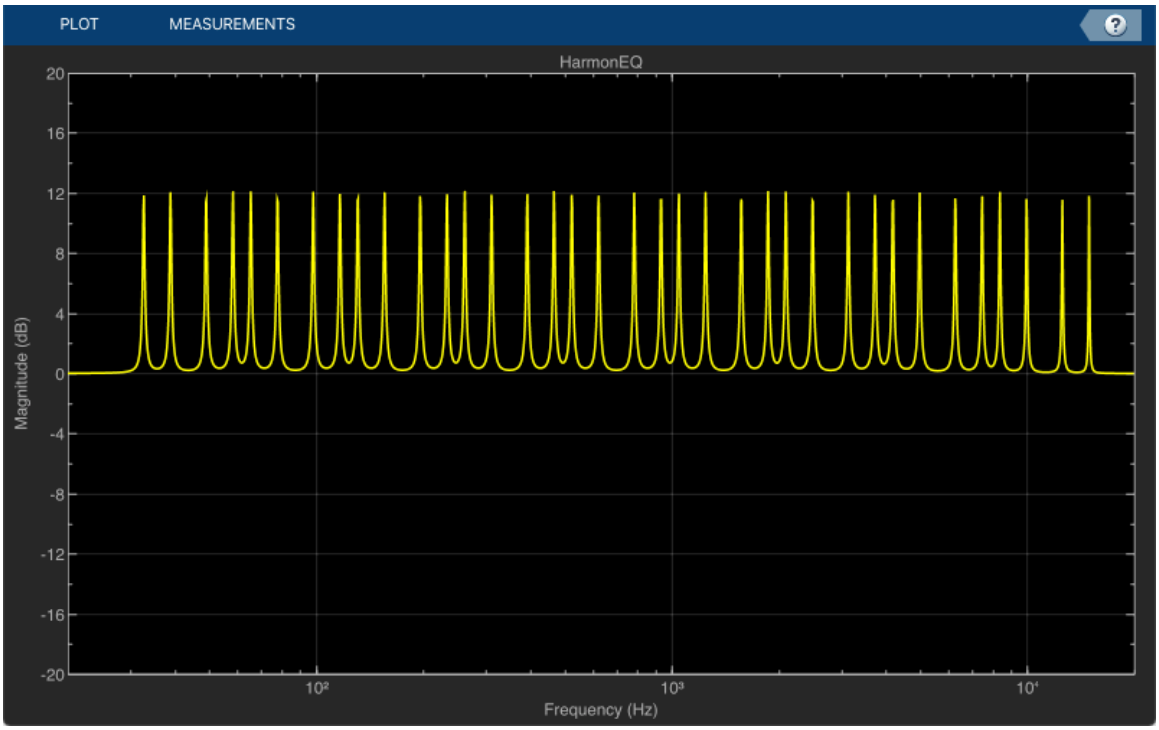


Figure 5.3: All 36 filters active with high Q setting.

settings in the different control regions, the user can create complex equalization curves (see Fig. 5.4).

5.2.3 Filter Updating

When the user changes the root note or chord type parameters, all of the filters' center frequencies are updated accordingly. However, the method for changing many filters simultaneously is important. Like in other parametric EQs, if the center frequency of a peak filter is changed instantaneously to another value, there will likely be a pop due to the destabilization of the filter. To avoid this, most EQs adjust filter center frequencies over several audio buffers. This is not a good solution for HarmonEQ, however, since there are many more filters than on a typical EQ. The user has no direct control over their movement and some may need to move up in frequency while others move down. This would cause undesired artifacts. HarmonEQ's solution is to quickly fade out the filters that need to change frequency and then fade them back in at the new proper frequencies. In the current implementation, this takes place over the course of eight audio buffers so the timing varies depending on the samplerate and buffer size. For example, the time to fade in and fade out at a samplerate of 48

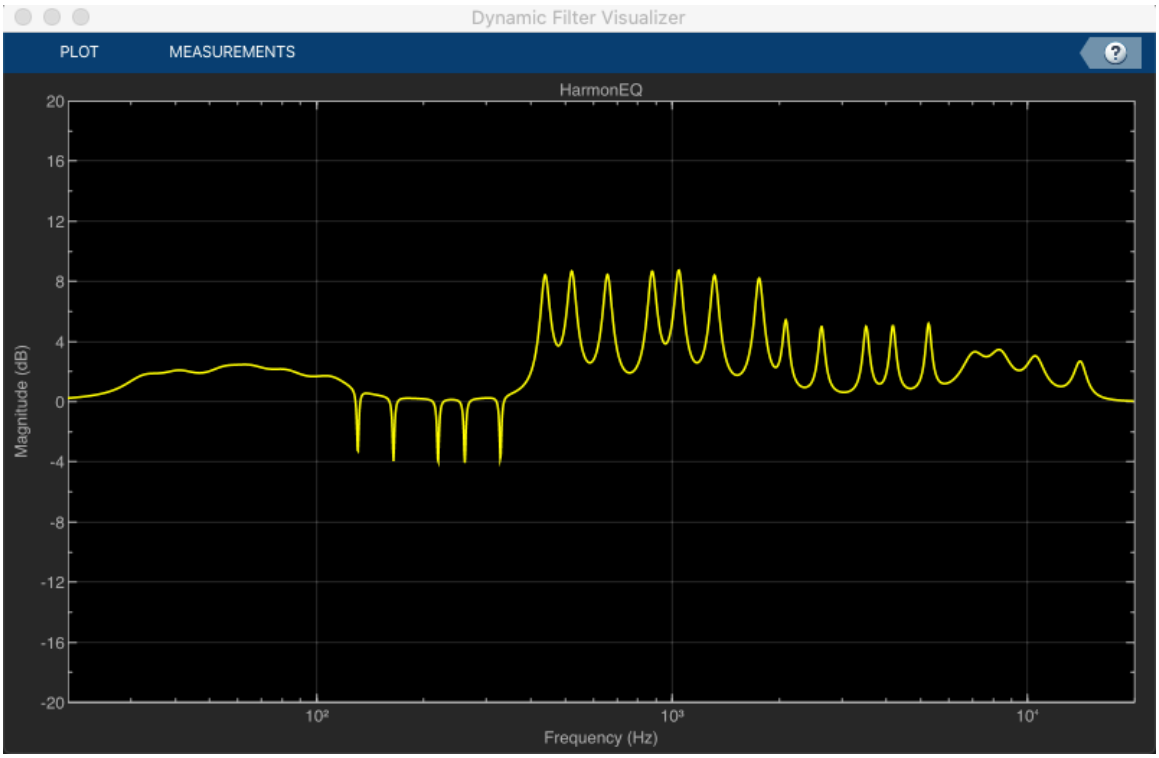


Figure 5.4: Complex equalization curve.

kHz with a buffer size of 128 samples is about 21.3 ms.

It is also important to note that with HarmonEQ's current region-based control method, it is possible that when a new root note or chord type is selected, a filter may cross the threshold from one region to another. In this case, the updated filter fades in with the gain and Q parameter settings for its new control region.

5.3 Evaluation

HarmonEQ has an automatic mode that disables the user's ability to manually set the root note and performs automatic chord detection on the incoming audio stream. The automatic chord detection system uses chromagrams and template-based chord recognition. The main method is adapted from [**?**] with custom heuristics for smoothing the update process.

The general chord detection method is as follows:

1. Sum the audio to mono.

2. Apply highpass and lowpass filters.
3. Calculate the frame power spectrum.
4. From the power spectrum, calculate the normalized chroma vector.
5. Calculate similarities from the chord template list.
6. Calculate the similarity weights based on heuristics.
7. Update the chord setting if the new similarity score outperforms the similarity score for the current chord.

It is assumed that both channels of a stereo signal will have similar harmonic content so a stereo audio signal is summed to mono to simplify the analysis process. The analysis system accumulates samples and uses larger buffer sizes to increase frequency resolution. At a sample rate of 48 kHz or below, a buffer size of 2,048 samples is used, 4,096 samples for between 48 and 96 kHz, and 8,192 samples for sample rates above 96 kHz. The mono analysis signal then has high-pass and low-pass filters applied to reduce low and high frequency content. Sixth-order Butterworth filters are used for both filters. The cutoff frequency for the high-pass filter is 110 Hz and the cutoff for the low-pass filter is 1,200 Hz. In testing, it was observed that these filters enhanced the effectiveness of the chord detection system since they filter out non-harmonic noise. Next the power spectrum of the audio frame is calculated from the STFT. This is then used to calculated the normalized chroma vector based on the method used by the `chroma_stft` function from [82]. A similarity measure, s, between the normalized chroma vector, x, and a chord template, y, is calculated using the inner product of normalized vectors according to:

$$s(x, y) = \frac{\langle x|y\rangle}{||x|| \cdot ||y||}. \tag{5.2}$$

Custom heuristics were developed in testing to further smooth the chord selection process. In testing, it was found that for major and minor chords, the version of the chord with the 3rd omitted (a 5 chord) often achieved a higher score than the true chord. For example, a D5 chord (consisting of the notes D and A) might outperform D minor (consisting of D, F, and A) even if D minor is the true chord. To address this, the similarity scores of '5' chords are scaled by 0.9 to lightly de-emphasize their selection. In a similar vein, the similarity scores for diminished, augmented, and

seventh chords are scaled by 0.75 to de-emphasized their selection unless their score is strong. This reduced false positives for seventh chords in testing. The reasoning behind this is that it is preferable for the system to be wrong by choosing a major or minor version of a seventh chord than it is to choose a seventh chord instead of a major or minor chord. Finally, in an effort to reduce incorrect chord changes and jitteriness, the similarity score between the current audio frame and the current chord is scaled by 1.035 to give a slight weighting to the current chord. This gives a slight preference to the current chord and prevents the system from oscillating between chords with similar scores. All of these heuristics were tuned manually in testing.

The system then selects the chord with the highest similarity score and updates the chord type for the plugin accordingly. This then updates all of the center frequencies for all of the plugin filters. Through this, the center frequencies for the filters will reflect harmonic changes in the incoming audio. More sophisticated methods for automatic chord detection such as the ones mentioned in the Sec. 5.1.1 can also be utilized provided that they can be computed in realtime.

In order to evaluate the performance of HarmonEQ, we have created several audio tracks presenting potential use cases. The first two use cases show how HarmonEQ can be effective at helping notes and chords cut through in a noisy mix. The third case presents a case for mixing bass and drums to reduce masking. The fourth case presents a creative way to use HarmonEQ to imbue non-harmonic sounds or instruments with harmonic character. All of the audio tracks for this analysis are available in the demos folder of the HarmonEQ GitHub repository.

5.3.1 Case 1: notes with noise

This example uses various instances of the note A played on the piano. The audio for these notes is then combined with pink noise. Although the notes are still audible, the noise has a strong masking effect on the piano notes.

The mixed audio was then processed using HarmonEQ with settings designed to give more prominence to the piano notes. HarmonEQ's root note was set to 'A' while the chord type was set to 'no chord'. The gain settings were +4.5 dB in the mid-low range, +6 dB in the mid range, and +4.5 dB in the mid-high range with Q settings of 14. These settings are a bit aggressive, but are used to show how HarmonEQ can easily target specific note frequencies. Spectrograms and chromagrams of this audio is shown in Fig. 5.5.

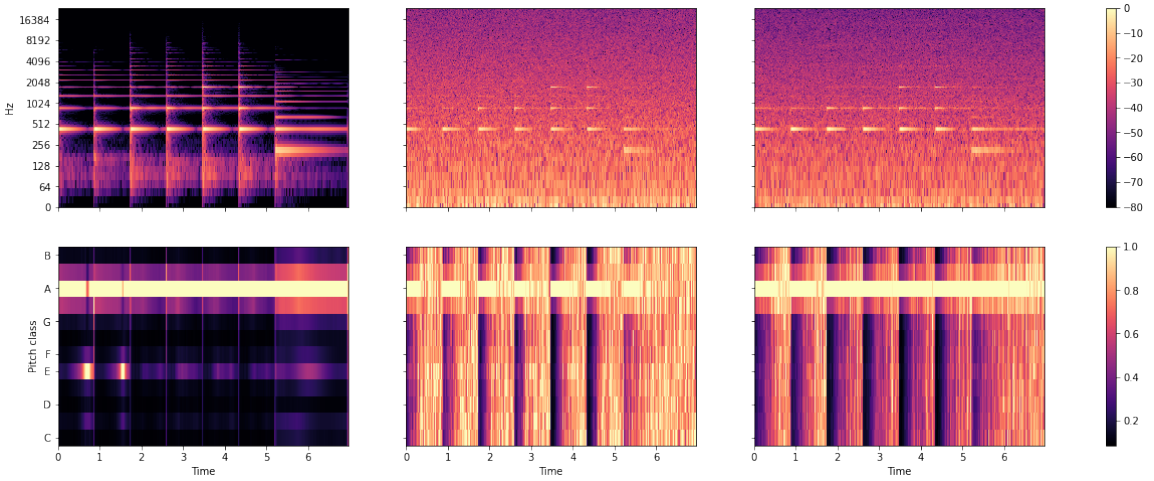


Figure 5.5: Time-aligned spectrograms (top row) and chromagrams (bottom row) of the clean piano notes (left column), piano notes + pink noise (center column), and then the mixed signal processed with HarmonEQ (right column).

In the left column, we see that the audio has clear tonal properties and the chromagram shows that the audio is strongly tuned to the note A. However, in the center column we can observe the masking effect of the pink noise. While the piano notes have a strong sense of tonality at their attacks, that becomes obscured by the pink noise while the notes sustain and decay. In the right column of the figure, we see the mixed audio processed by HarmonEQ. The coloring of the pink noise on the spectrogram is darkened showing that the piano notes are being reinforced and the masking is being reduced. This conclusion is reinforced by the chromagram. It clearly shows that the energy of the track has become more focused on the piano notes – even during the sustained portions of the notes.

5.3.2 Case 2: chords with noise

This example uses A major chords played on the piano. The piano audio is again combined with pink noise. Although the chords are still audible, the noise has a strong masking effect on the piano notes. The mixed audio is processed using HarmonEQ with settings designed to give more prominence to the piano notes. HarmonEQ's root note was set to 'A' while the chord type was set to 'maj' (for major). The gain settings were +4.5 dB in the mid-low range, +6 dB in the mid range, and +4.5 dB in the mid-high range with Q settings of 14. Again, these settings are a bit aggressive and chosen for demonstration purposes. Spectrograms and chromagrams of the audio

are shown in Fig. 5.6.

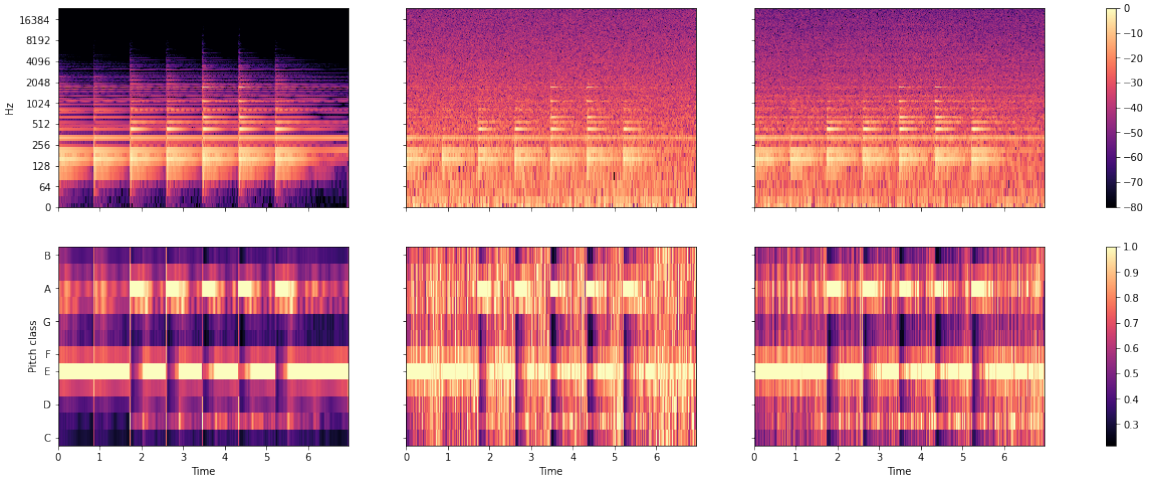


Figure 5.6: Time-aligned spectrograms (top row) and chromagrams (bottom row) of piano chords unprocessed (left column), mixed with pink noise (center column), and mixed with pink noise and processed with HarmonEQ (right column).

The left column of Fig. 5.6 shows that the spectral energy is concentrated around the notes A, C♯, and E — the notes that constitute an A major chord. The pink noise once again masks the piano chords although they are still observable on the spectrogram and chromagram in the center column. The right column shows that the relative strengths of the piano chords and the pink noise have shifted. Although the center and right spectrograms look very similar (and they should since the goal is to use the EQ as transparently as possible), the center and right chromagrams reveal the masking is reduced since we can see that the chroma energy is more concetrated in the chord tones in the right chromagram. Mixing in this way with HarmonEQ cannot completely overcome masking and noise, but will reinforce the tonality.

5.3.3 Case 3: mixing bass and drums

HarmonEQ can also be used to reduce masking between different audio tracks. This example consists of a track with drums and electric bass. The kick drum is masking the bass line. The typical solution to this is to use compression or dynamic equalization. HarmonEQ, however, can also used to reduce the kick drum strength at only the frequencies that align with the bass notes while also strengthening the bass signal at the appropriate frequencies.

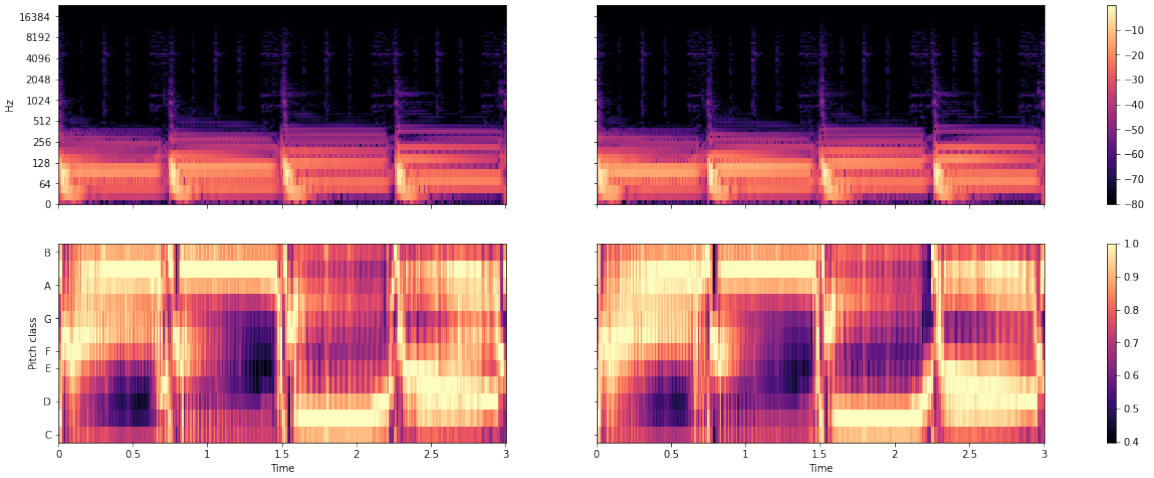


Figure 5.7: Time-aligned spectrograms (top row) and chromagrams (bottom row) of both unprocessed (left) and processed (right) drum and bass tracks

Instances of HarmonEQ are used on both the drum and bass tracks. The bass line pattern is G, A, C, D and the root note setting for both tracks was automated to follow those notes. The drum track instance of HarmonEQ was set to 'no chord' with gain settings of -7 dB in the low and low-mid ranges and -3 dB in the mid range. The bass track instance of HarmonEQ was set to '5' chord to give the notes a little harmonic boost. The gain settings were +3 dB in the low region, +6 dB in the low-mid region, and +3.5 dB in the mid region.

Spectrograms and chromagrams of the unprocessed and processed tracks are shown in Fig. 5.7. Again, the differences in the plots are subtle, but the results are audible in the sound files. The spectrograms appear nearly identical, but this is expected since the mixing being performed on the tracks is designed to be as transparent as possible while still accomplishing the goal of unmasking the bass. HarmonEQ is reducing masking by cutting the lower frequencies of the drum track at the root note frequencies of the bass line. The more prominent dark areas in the right chromagram of Fig. 5.7 show that the mixed audio has become more tonally focused on the notes of the bass line. This is particularly visible from time 1.5 s to the end of the example. The automation for this was set up in less than a minute. To do similar mixing using a traditional equalizer would take significantly longer. This method is also more narrowly targeted than using typical solutions like compression or a dynamic shelving filter.

5.3.4 Case 4: making drums harmonic

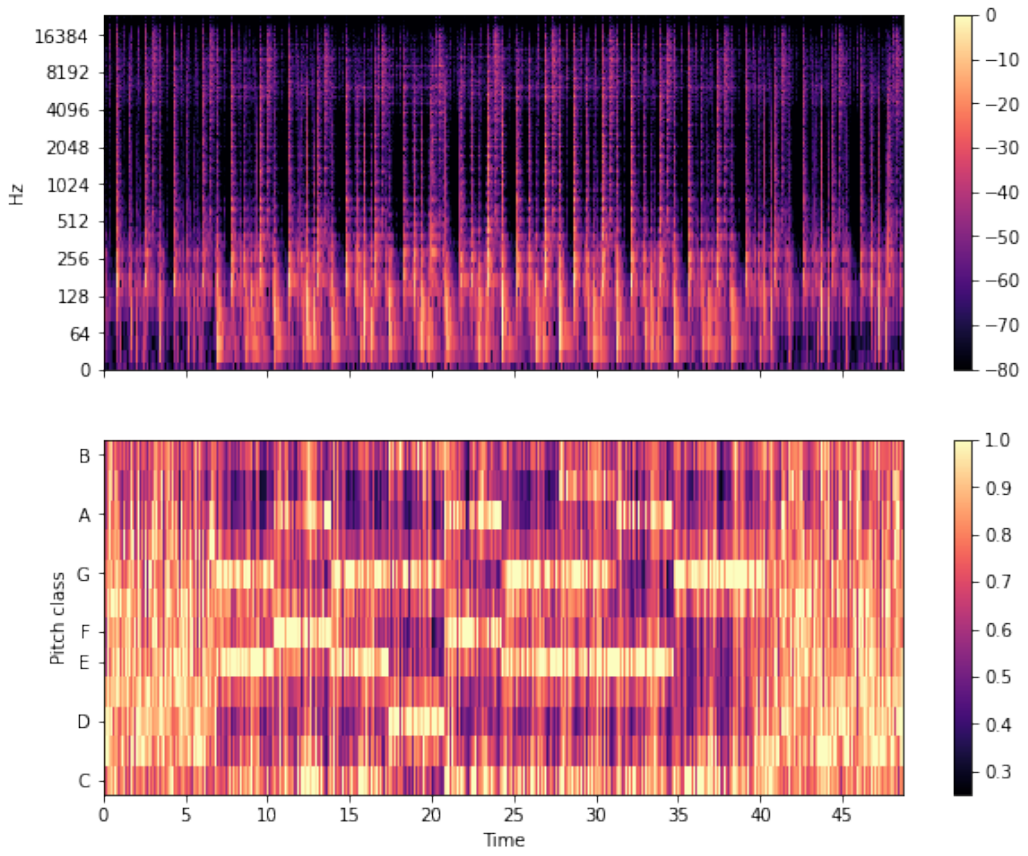


Figure 5.8: Time-aligned spectrogram (top) and chromagram (bottom) of drums processed through HarmonEQ.

HarmonEQ can also be used in creative ways to make non-harmonic, broad-spectrum noises sound harmonic. In this example, we processed a drum loop through HarmonEQ using a sequence of chords and aggressive filter settings to make the drums sound "harmonic". The drums undeniably sound like drums throughout, but when HarmonEQ is activated, they take on a synthesized, tonal quality and the chord changes are clearly audible.

A visualization of the processed drum loop is shown in Fig. 5.8. For approximately the first seven seconds, the drum loop is unprocessed. From seven seconds to about 41 seconds in the track, the drum loop is processed through HarmonEQ using automated chord changes and then returns to the raw drum loop at the end. The EQ settings are necessarily aggressive at +12dB of gain and a Q setting of 26 for all five control regions. The spectrogram shows that the spectral energy is increased while being processed by

HarmonEQ, which is expected since it is adding significant gain (although with narrow peak filters). However, the important information is contained in the chromagram. The chroma vectors are individually normalized to clearly show where the maximum energy is at any given point in time. It is clearly observable that the unprocessed drum loop consists of broad-spectrum, non-harmonic noise since the chroma vectors at those times have roughly even energy values. However, during the portions that are processed through HarmonEQ, it is clear that the spectral energy becomes much stronger at specific tones giving the drums a clear harmonic attribute. Processing the drum loop in this way took less than half an hour using automation tracks and making changes in the design choices is quick and simple. Manually processing a track in this way using a traditional equalizer would not be practical. This is one example of how a harmonic equalizer like HarmonEQ could be used in a creative way to open new possibilities in music production.

5.4 Discussion

The concept of the harmonic equalizer was inspired by an engineer who encountered a problem mixing audio tracks. He had trouble with conflicting tones and had to create numerous octave-spaced EQ filters that were boosting and cutting various tracks to reduce conflicting tones and masking. The process was very time-consuming and difficult to automate to reflect chord changes. With a harmonic equalizer like HarmonEQ, however, the time required to accomplish this task would be reduced greatly. HarmonEQ is not intended in any way to replace traditional parametric equalizers. The intention is to solve problems that traditional equalizers struggle with and enable new methods of creativity.

HarmonEQ can be used in a wide range of audio mixing situations. Examples include easily creating signal cuts in a kick drum at the root frequency of the bass note or in the drum track at the frequency of melody notes. It can also be used to enhance the tonal frequencies of a track. Or, as seen in Sec. 5.3.4, it can be used in all new creative ways to easily create new sounds.

While the concept of the harmonic equalizer is an important contribution, it becomes even more powerful when combined with automatic note/chord detection. Audio equalizers are regularly used to enhance the harmonic content of music, but this is the first instance of an equalizer that can do so intelligently based on the harmony of the audio being equalized.

5.5 Future work

Due to the change in control paradigm, a harmonic equalizer is well-suited for solving problems that can be very difficult with traditional parametric equalizers, but also makes some typical uses for equalizers impossible. There is a need for further study to understand how best to use a harmonic equalizer in normal situations and how it can also be used for creative purposes. The version of HarmonEQ presented here was also a first attempt at a harmonic equalizer and as such many standard features of equalizers have been left out. There are no types of filters besides peak filters because more testing is necessary to determine how to best incorporate highpass, lowpass, shelving, bandpass, bandstop, and other filters into the harmonic paradigm. Another feature that should be including in a future version is the ability to fine tune the placement of filters. One potential solution to this is a global tuning knob that would adjust the reference tuning from 440 Hz to $\pm$ 50 cents.

A weak point of the current version of HarmonEQ is the user interface. The goal of this version was to demonstrate the harmonic equalizer concept, but in future iterations a more advanced GUI will enable more modes of interaction that can allow for more flexible equalizer programming. For example, HarmonEQ currently allows only one gain and Q setting per control region, but it is likely that a user would want more fine control over these or to have the ability to define their own control regions. Just as research into what methods of interaction with parametric equalizers was conducted previously as in [60], new research is necessary in order to understand the optimal ways to interact with this new type of equalizer.

Another space where there is room for development is how to improve the automatic chord detection system. The system presented here was chosen to demonstrate the concept because it can be reliably run in a realtime system, but a comprehensive survey of different methods was not conducted. The recent trend in automatic chord detection is to use deep learning methods. As computational power increases, the suitability of convolutional neural networks and recurrent neural networks in realtime audio plugin applications becomes feasible. However, due to the realtime nature of audio plugins, it will be important to find the right balancing point for the trade-off between accuracy and computational efficiency requirements.

5.5.1 Playable version

I am also actively developing a prototype of a playable processor based on HarmonEQ in Max/MSP. This version uses MIDI-controllable bandpass filters to process an audio signal. When a MIDI note message is sent, it dynamically creates up to sixteen bandpass filters associated with that note. By default the bandpass filters are centered on the frequencies of the harmonic series of the trigger note, but can also be manually set by the user. The prototype is currently monophonic, but the final version is planned to be polyphonic. Since we are using bandpass filters in this version, the user can then mix filtered audio with the dry audio signal to achieve the desired sound. This version can process noise, sound files, or a live audio signal.

5.6 Conclusions

HarmonEQ is a new approach to audio equalization based on musical notes and harmony. It has been fully implemented in MATLAB. The code is open source and publicly available – as are pre-compiled versions of it. This new approach to audio equalization enables new methods of interaction as well as the application of automatic chord detection to equalization. This is a new direction for audio equalizers with the potential to lead to more intelligent assistive technologies for producers and audio engineers. While HarmonEQ is not suitable for every mixing situation, it makes solving some problems simpler and enables new and creative uses that were not practical with traditional parametric equalizers.

Chapter 6

The Electroacoustic Steelpan

This chapter begins with an overview of preexisting electroacoustic pieces for steelpan in Sec. 6.1. After the summary of the electroacoustic repertoire, Sec. 6.2 gives an overview of digital musical instruments and augmented acoustic instruments in particular. I then discuss two important developments in my practice for electroacoustic performance with the steelpan. The first development was the adoption of the Line 6 Helix as the primary front-end sound design device discussed in Sec. 6.4.4. The second was adopting Ableton Live as my performance Digital Audio Workstation (DAW) discussed in Sec. 6.6. In Chapters 7-9 I present new works I composed for electroacoustic steelpan that utilize the techniques presented in this chapter to expand the repertoire.

6.1 Electroacoustic Steelpan Repertoire

Let us begin with an overview of the state of the electroacoustic steelpan repertoire. There is a limited set of electroacoustic pieces for steelpan. This list attempts to be comprehensive since the electroacoustic steelpan is a small niche in the world of music. This list contains every work that I could find that includes both a steelpan and electronics. The size of this list shows that while composers are interested in the electroacoustic steelpan, there is also significant space for composers to make important new contributions to this nascent repertoire. Brief descriptions of the pieces are presented below grouped by composer in alphabetical order.

6.1.1 Akiho

21[5] by Andy Akiho is a duet for tenor steelpan (plus tambourine) and cello (plus kick drum) composed in 2009. Andy recorded the piece with cellist Mariel Roberts for his album, *NO one To kNOW one*[6]. In *21*, the cellist uses a pedal to control a looper to create a loop consisting of a cello melody and percussive parts played on kick drum, tambourine, and the skirt of the steelpan.

6.1.2 Alvarez

Javier Alvarez has composed several electroacoustic works for steelpan. The first, *Así El Acero*[7], was composed in 1988 and is scored for solo tenor steelpan with fixed media. This is one of the earliest electroacoustic works for steelpan and features frenetic, virtuosic lines. The fixed media portion includes many processed sounds including processed steelpan samples. Alvarez has released a recording of *Así El Acero*[8] with Simon Limbrick performing.

Alvarez later composed *Offrande*[9] in 2000. This is a duet for tenor steelpan and baritone steelpan (presumably double seconds or triple cello) with a fixed media accompaniment. This piece seeks to blend unorthodox steelpan timbres with the electronic accompaniment. *Offrande* frequently calls for the use of non-traditional implements such as plastic brushes to achieve these unorthodox timbres.

The final piece by Alvarez is *Mantis Walk in Metal Space*[10], an electroacoustic concerto for steelpan soloist, chamber winds, and live electronics. *Mantis Walk in Metal Space* is scored for a tenor steelpan soloist with chamber wind ensemble consisting of two flutes, two clarinets, oboe, bassoon, French horn, trumpet, trombone, and tuba. It was premiered in 2003 at IRCAM with Miquel Bernat as soloist.

6.1.3 Berry

Composed in 2009, *Mare Tranquillitatus*[12], is a work for tenor steelpan, crotales, and live electronics by Mark Berry. It uses rhythmically timed delays and pitch shifting to create a rhythmic and harmonic counterpoint to the soloist's performance. *Mare Tranquillitatus*, Latin for Sea of Tranquility, is named after a lunar mare and is the first location on the moon visited by humans. With new age aesthetics to the compositional style, this piece musically portrays a journey from Earth to the nearest celestial body.

6.1.4 Boyd

Blues steelpannist and vocalist Gregory Boyd has been experimenting with using guitar pedals with his steelpans since at least 2016 with this album *Transformation*[16] and single EP *Beating on a Drum*[15]. Boyd performs on double second steelpans uses a variety of distortions, fuzzes, and modulation effects. I have chosen not to present individual songs here. Boyd follows a different compositional tradition than contemporary classical music. Rather than creating "electroacoustic steelpans concert works", Boyd has developed a personal electroacoustic steelpan practice and incorporated it into the Blues musical tradition. Nevertheless, his work is important as an example of electroacoustic steelpan performance and how it can be implemented in a musical genre with which it does not have a traditional association.

6.1.5 Dyke

In 2007 steelpannist and tuner Darren Dyke released the album *Resonance: Steelpan in the 21st Century*[34]. This album is constructed on an interesting premise. Dyke recorded a session of traditional acoustic steelpan performances. He then collaborated with a variety of composers from around the world who used his recordings as the source material for new electronic compositions. Yoshio Machida (discussed in Sec. 6.1.7) is one of the composers included on this compilation.

6.1.6 Kreiger

In 1974, Arthur Kreiger composed *Dialogue*, a solo piece for steelpan and fixed media. This is the earliest electroacoustic piece for steelpan I know of. The sounds and processing used for the fixed media accompaniment reflect the era in which it was made due to the technological limitations. The compositional style is atonal with frequent shifts in rhythm, tempo, and mood. *Dialogue* was recorded by New York Philharmonic Associate Principal Percussionist and Juilliard faculty member Daniel Druckman for New York New Music Ensemble's 2003 album, *Meeting Places*[54], featuring all works by Kreiger.

6.1.7 Machida

Yoshio Machida is an artist who has incorporated the steelpan into a personal electroacoustic improvisational practice. Machida frequently combines steelpans with

synthesizers, Max/MSP patches, and various other sound sources to create experimental, minimalistic works. Machida's albums were previously only available in Japan, but are now available through Bandcamp and SquidCo.

The 2004 album *Infinite Flowers*[62] and 2008's *Steelpan Improvisations: 2001-2008*[63] feature his improvisational electroacoustic practice throughout.

6.1.8 McClure

Passacaglia on a Theme by Mark Rothko[80] is a 2009 piece composed by Robert McClure and commissioned by percussionist Eric Hines for solo steelpan and fixed media accompaniment. McClure has released a recording of the piece with Hines performing[81]. It is composed as a reflection on the paintings of Mark Rothko. The often sparse steelpan part is primarily improvised with some minimally composed sections. McClure blends the electronic sounds with the steelpan to evoke how Rothko uses paint to blend colors and shapes.

6.1.9 Ortiz

Mexican composer Gabriela Ortiz composed *Magna Sin*[97] for solo tenor steelpan and fixed media in 1992. The fixed media part is constructed from processed audio of steelpans, flute multiphonics, metal objects, guitar, and other percussion instruments. This piece explores the connection between the steelpan and the oil industry. In Mexico City, the unleaded gas designed to minimize pollution is called "magna sin" [96]. The steelpan has been connected to the oil industry since its modern version was developed in the late 1940's by Ellie Mannette using oil barrels (as discussed in Sec. 2.1.1).

6.1.10 Sekhon

Regeneration[124] is a 2005 composition by composer Baljinder Sekhon for solo tenor steelpan and fixed media. Phillip O'Banion released a recording of *Regeneration* on his 2016 album, *Digital Divide*[94]. *Regeneration* calls for many non-standard techniques, playing on all parts of the steelpan as well as the stand. The piece depicts the development of the steelpan in Trinidad. It uses brake drum rhythms (a traditional steelband accompaniment) as well as other allusions to Trinidadian music.

6.1.11 Walters

Composed by Adam Walters in 2019, *Prayer*[133] is a duet for trombone and steelpans with an optional looper part and fixed media. The steelpannist performs on both tenor and double second steelpans. The fixed media part features nature soundscapes and electronically generated sounds. To thicken the orchestration, the steelpannist can choose to use a looper to add background loops. Prayer was premiered and recorded by trombonist Aidan Chamberlain and steelpan virtuoso Mia Gormandy-Benjamin. The recording was released as a single by Walters in 2019[134].

6.2 Background on Digital Musical Instruments

Digital Musical Instruments (DMIs) are an increasingly common class of instruments that rely on computers in different ways to produce their sound. The earliest DMIs were digital synthesizers. Due to computational limitations these early DMIs required specialized microchips to process audio. In modern DMIs, the computation can now be performed on traditional computers, micro-computers (like the Raspberry Pi), or micro-controllers (like the Daisy) [39]. The breadth of possibility for DMIs is as wide as our imaginations. An early innovator in the space, Cook laid out his philosophy for developing new DMIs in 2001—often from everyday objects like a coffee mug [25].

6.2.1 Steelpan-Inspired MIDI Controllers

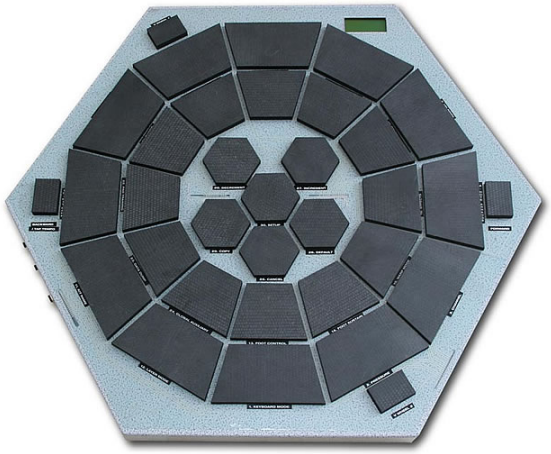


Figure 6.1: The PanKAT by Alternate Mode.

Several MIDI-based electronic steelpans have been produced since the mid-2000's. The PanKAT (see Fig. 6.1) was a steelpan-inspired electronic percussion instrument that leveraged manufacturer Alternates Mode's extensive expertise in using force sensitive resistors in electronic percussion instruments [83]. NAPE Inc. produces the E-Pan (shown in Fig. 6.2), an invention of Salmon Cupid [29]. The E-Pan is a steelpan-inspired digital instrument that closely follows the note layouts of several different varieties of acoustic steelpans. The E-Pan is known for its built-in steelpan samples, but can also be used as a MIDI controller. The Percussive Harmonic Instrument (P.H.I.) was also a digital musical instrument based closely off of a tenor steelpan [28]. An interesting aspect of the P.H.I. is that it had three octaves of sensors (36 note pads) while an acoustic tenor steelpan typically has only 29 notes. The electronic nature of the P.H.I. allowed it to have an extended range that would not work on an acoustic version of the instrument. Most recently, Persad et al. proposed the roll-up digital steelpan inspired by similar roll-up piano and drum pads [103]. The roll-up digital steelpan is the most portable of these instruments and intended to be used for learning and practice. As of this writing in 2023, the only instrument from this list in production is the E-Pan.

Figure 6.2: The E-Pan with tenor steelpan note layout by NAPE Inc.

These steelpan-inspired digital instruments and MIDI controllers have not been utilized extensively in electroacoustic performance. They are primarily used to replace

an acoustic steelpan for quiet practice or for convenient, consistent amplification. Sec. 6.1 explored the repertoire of electroacoustic works including the steelpan, but none of them include any of these instruments.

Nonetheless, these instruments have been important milestones for the steelpan. They provide many advantages over acoustic steelpans. They do not go out of tune, are easily amplified, and can be used to play sounds from other steelpan layouts or completely different instruments. Despite these advantages, though, they cannot replace the acoustic steelpan.

6.2.2 Augmented Acoustic Instruments

Augmented instruments are acoustic instruments that use electronics to expand their capabilities[35]. These augmentations can include additional sensors, audio processing, visualizations, and more. There are many considerations designers of augmented instruments must take into consideration as they develop their instruments. Some sensors might generate useful and interesting data streams, but interfere with performance. Others might affect the acoustical properties of the instrument. Sometimes these effects are negligible, but other times potential sensor options must be eliminated because of these kinds of issues.

A further issue to consider is between that of the input and the output. In modern music, audiences often hear with their eyes. Percussionists are known to use gesture to affect how the audience perceives their performances [123]. It is similarly important for the input to a processing system to relate to the resulting output of the system [122]. It can be jarring for the audience if they are unable to connect the cause and effect of sounds. Augmented instruments leverage the existing gestures and sounds of acoustic instruments to build a bridge for the audience from the acoustic input to the digitally manipulated output.

There are many examples of augmented instruments. Burtner developed the Metasaxophone at Stanford's Center for Computer Research in Music and Acoustics [20]. The Metasaxophone includes six pressure sensors on six keys, two pressure sensors off of the keys, five triggers, and an accelerometer for tilt sensing,. Palacio-Quintin created the Hyper-Flute, an augmented flute with attached sensors [99], [100]. The added sensors include an ultrasound transmitter, tilt sensor, magnetic field sensors on the player's fingers, and force sensing resistors. The SABRe, an augmented bass clarinet was developed by Schiesser and Schacher [121]. The SABRe features

four groups of sensors for key positions, inertial movement, mouth pressure, and trigger switches. Reid et al. developed MIGSI, the Minimally Invasive Gesture Sensing Interface for trumpet [109]. MIGSI add three types of sensors to the trumpet: optical sensors on the valves, an accelerometer for sensing pitch and roll, and force sensing resistors to detect physical contact from the player's hand. An earlier experiment in augmenting the trumpet was conducted by Jenkins et al [50]. Carvajal created Augkit, an augmented drum set system designed for live performance [22]. Augkit uses an acoustic drum set and a MIDI controller with audio processing achieved with custom software built using Max/MSP.

6.3 Introduction to My Processing Setup

Electroacoustic performance has two basic musical requirements: an acoustic instrument and electronic processing of some nature. For my work, I am using the steelpan as the acoustic instrument, but the possibilities for electronic processing are still countless. Most of the pieces discussed in Section 6.1 are composed for an acoustic steelpan to be played with an amplified fixed media backing track. This setup is often favored because it has the fewest variables. The backing track is fixed and only needs to be amplified. The steelpan does not need to be processed and is usually not amplified. Using dynamic backing tracks, live processing, or amplification of the steelpan significantly increase the complexity of the setup.

My goal, however, was to develop a practice that would support music with all three of those attributes—a practice that is informed by the acoustical analysis of the steelpan and can utilize the audio processing tools in performance and recording. To accomplish this I also decided that I needed to design an audio processing setup that was powerful, flexible, and portable. It needed to be able to support my custom audio effects as well as standard ones. Furthermore, in order to allow for performing in a variety of situations, it needed to be compact and portable. Finding the right balance to maximize all three dimensions took time and experimentation, but I have settled on a system that has proven effective in a variety of situations.

6.4 The Front End: Line 6 Helix

The front end of my system is built around the Line 6 Helix. The Helix allows for flexible signal routing in both the analog and digital domains. Figure 6.3 shows an

overhead view of a tenor steelpan and the Helix. In this section, I will discuss the general setup I have developed for using the Helix with the steelpan. In each of the following chapters discussing the new electroacoustic works for steelpan, there will be a section that discusses the specific setups used for that piece.

Figure 6.3: Overhead view of steelpan and Line 6 Helix.

6.4.1 Background on the Helix

The Helix is a guitar multi-effects floorboard audio processor designed by Line 6. Featuring 93 amplifier models, 266 audio effects, and 65 speaker cabinet emulations, the Helix is designed to be an all-in-one solution for guitarists. The effects categories include distortion, dynamics, equalization, modulation, delay, reverb, pitch/synth,

filter, wah, and amp/preamp/cabinet emulations. Each category has a wide variety of options available with all major effects used by guitarists included. It has flexible input/output options, completely customizable effects chains, an intuitive user interface with a color screen, and the ability to interact with computers over USB. The Helix uses a Digital Signal Processing (DSP) budget approach to sound design rather than having a fixed number of spaces for audio effects. The user can continue to add any audio effect block to their signal chain in any available position as long as the DSP budget is not exceeded. Although the Helix is designed for use with guitar and bass processing, it can be adapted for other instruments like the steelpan.

6.4.2 Capturing the Steelpan's Audio

I use two different input sources to capture the sound of the steelpan: a microphone and a piezo pickup. These are the most common methods of amplifying the steelpan. Copeland has also explored the used of magnetic pickups [26]. When working in the studio, I use a Beyerdynamic M160 ribbon microphone positioned under the bowl of the steelpan as the primary method for capturing the steelpan's sound as shown in Fig. 6.4. When performing live, I use a Blue Microphones enCORE 100i dynamic microphone in the same position. I prefer the tone of the ribbon microphone when recording, but choose to use the dynamic microphone onstage since it is more robust and easier to replace if damaged. The enCORE 100i is similar to the ubiquitous Shure SM57 which can be used if the enCORE 100i is not available. The microphone is plugged into the "MIC IN" XLR input on the Helix and uses the Helix's built-in microphone preamplifier to raise the level prior to processing.

This microphone position was selected based on several factors. Research has shown that the top of the steelpan radiates sound stronger, but with a darker tone [27], [87]. In [88], Muddeen and Copeland present specific microphone recommendations based on the steelpan's sound radiation. In personal conversation, well respected percussion recording producer Ray Dillard recommended the Beyerdynamic M160 with a placement under the steelpan. This is also in line with the recommendations of Bartlett [11].

I tested several different configurations and I prefer placing the microphone under the steelpan for two reasons: it de-emphasizes the low frequencies which I think produces a better tone and it gets the microphone out of the performer's way. When the microphone is placed above the pan, there is usually a trade-off: place the mi-

Figure 6.4: Side view of the microphone placement connected to the Line 6 Helix.

crophone in a position that captures a high quality, even sound, but is possibly in the performer's, or place the microphone in a position that compromises the sound quality, but is out of the performer's way. Placing the microphone below the bowl

allows for closer placement without interfering with the performer. Furthermore, by placing the microphone closer to the bowl of the steelpan, less microphone preamplifier gain is necessary which will reduce the likelihood of feedback in live performance. When recording in the studio, adjustments to the position may be made depending on microphone choice, room tone, and other factors.

The piezo pickup is an EnSoul PPEX10MT Pan Pickup (shown on its own in Fig. 6.5 and attached to the underside of a steelpan bowl in Fig. 6.6). EnSoul makes a range of piezo pickups specifically for steelpans and handpans with magnets embedded in a rubber casing. They currently produce three main models. In each of the models, the piezo is the same, but the first includes a 150 Hz highpass filter, the next a 250 Hz highpass filter, and the final model has no filter. The highpass filters are useful for reducing the "thud" noise of the mallet striking the steelpan – especially in live performance situations where the player is simply plugging the piezo into a system for amplification. Since I further process the audio from the steelpan, I prefer a piezo without filtering. I still often use a highpass filter in performance, but this allows me to tailor it to the specific musical situation instead of only having a single option. There are also times that I do not highpass filter the signal so that I can use the thuds and low frequency ring of the pan in processing. The piezo can be plugged into either the "GUITAR IN" or "AUX IN" 1/4" jacks on the Helix. The difference between the two is that the auxiliary input has a higher input impedance at 10 kOhm that will attenuate the high frequencies to a degree. Depending on the specific sound of a pan and preference of the performer, the high frequency attenuation can be thought of as a feature. I use the auxiliary input.

A significant advantage to the piezo as an input source is that it is not sensitive to extraneous noise. It will not pick up wind noise, sound from monitor speakers, other instruments, etc—unless they cause the steelpan to vibrate sympathetically. When using the Helix in performance, the footswitches do make clicking sounds when pressed. If tapped aggressively, these clicks will be picked up by the microphone, but would not be by the piezo pickup. Also, since the EnSoul pickups have magnets built in, they are simple to attach securely. I prefer the sound of a microphone so I use that as my primary input method. In some situations, I use both input simultaneously and process them with different signal chains. If the sound conditions of the performance space cause problems like feedback, however, I will use the piezo in place of the microphone.

Figure 6.5: EnSoul steelpan piezo pickup.

6.4.3 Motivation

There are many advantages to using the Helix over a typical guitar pedalboard with individual pedals. The most obvious advantage is the number of effects offered by the Helix. The variety gives me the ability to design precisely the sound I desire for a particular piece. The disadvantage of this many effects is the amount of time and experimentation it takes to become familiar with all of the different options and combinations. While it is possible to perform sound design on the Helix without being familiar with the various effects, in order to work quickly when designing a new patch, one must know what all of the (often euphemistically named) effects are, what they do, and how they will likely interact.

Another advantage of the Helix in performance is the ability to store and recall presets. Each piece can use a completely different DSP signal chain that can be changed almost instantly mid-performance. This can be accomplished manually using the footswitches or can be automated through MIDI—both wired and over USB from a computer.

Figure 6.6: EnSoul steelpan piezo pickup attached to bowl of tenor steelpan.

6.4.4 Routing and Controls in the Helix

The Helix can have two separate audio processing chains active at the same time. This allows me to send the microphone audio to one signal chain while the piezo pickup can be sent to the other. The two signal chains be sent to the same or different outputs on the Helix. Each of the two processing paths can also be split internally. An A/B split on a signal chain can be recombined later—allowing for parallel processing—or the split audio can be routed to different outputs.

Another option that increases the Helix's flexibility is that the output of processing path A can be set as the input to processing path B. This routing option allows for only one input source, but doubles the amount of DSP processing that can be applied to it. This option is especially helpful when using processing intensive effects like pitch shifting.

Most of the effects in the Helix can be selected to operate in either mono or stereo and a signal path within Helix can combine both. This is true even of effects that are typically mono only when used in a guitar pedal format such as distortion or compression. The user must thoughtfully decide when and where to use mono and stereo effects to achieve the sound they desire. An important consideration here is

that if a stereo signal is sent to a mono effect, it will sum the stereo signal to mono prior to processing. The Helix includes many "legacy" effects from previous Line 6 products that pre-date the Helix. Many of these effects, unfortunately, only operate in mono. Despite this, they are still excellent processing tools.

The Helix also has flexible send/receive routing options. There are four mono (or two stereo) sends and receives for interfacing with other equipment. Where the sends and receives are placed in the signal processing chain is also flexible since this is accomplished through Send/Receive effects blocks. The user can select the placement, level, panning, and dry/wet balance for each send and receive. I often use a secondary pedalboard with other audio effects not available in the Helix. Since the send/receive routing is saved as part of the Helix patch, all that is required is to connect the second pedalboard, make sure the pedals have the correct settings and it operates correctly every time.

Figure 6.7: Top view of the Helix.

Another helpful feature of the Helix is the flexible effect controls. The Helix has twelve touch-sensitive footswitches with customizable digital scribble strips (see Fig. 6.7). Each footswitch can be set to enable/disable the bypass of multiple effects at once or to change the value of an effect parameter. The Helix has a built-in expression pedal and can also have up to two more plugged in. Each expression pedal can adjust any effect parameter mapped to it and like the footswitches can have effect multiple parameters mapped simultaneously with custom settings for each parameter. For example, the wet/dry parameter of two different effects could be mapped to the same expression pedal, but with opposite settings. Then adjusting

the expression pedal would crossfade between the two effects. With this setup, the player could smoothly transition from one effect to another with a single expression pedal. Creating mappings like these with individual pedals on a pedalboard may sometimes be possible, but would usually be prohibitively complicated.

6.4.5 Helix as an Audio Interface

One of the most important features of the Helix, however, is its ability to be used as an 8-channel USB audio interface. Each of the internal audio paths inside of Helix can be routed to USB channels 1 through 6. Channels 7 and 8 are reserved for the dry input signals. This was an unfortunate design choice on Line 6's part. The Helix can handle up to four internal stereo audio paths by splitting each of the two main processing chains, resulting in eight audio channels in total. The Helix is also an 8-channel audio interface, but only six of those eight channels are available to be assigned to. If the performer splits the audio into four signal paths inside of the Helix, two of those paths will have to be mixed together before being sent to the computer over USB. Using the Helix as an audio interface will be discussed further in Section 6.5.

A benefit to having the Helix connected to the computer over USB is that it also communicates via MIDI. The Helix can send and receive MIDI messages from the computer to perform several tasks. Note that the Helix can act as a USB MIDI device even if it is not being used as an audio interface. A Helix setlist, preset, or snapshot can be selected over MIDI. These are all different ways to automate changing from one patch to another on the Helix. Helix can send and receive MIDI clock information as well. This allows the Helix to sync time-based effects to the session clock in Ableton (or another DAW). Finally, with MIDI control change messages, the computer can control the values for the expression pedals, footswitches, control a looper, and perform other tasks. The most common usage for MIDI control of the Helix would be to automate program changes on the Helix timed to or triggered by musical events happening in the DAW. All of this can be accomplished with MIDI signals sent over MIDI cables as well if the Helix is not plugged into a computer over USB.

6.5 Input/Output Audio Devices

System latency is an important consideration in electroacoustic performance. The generally accepted limit for latency is 10 ms [136]. This limit can vary based on context (e.g., whether the performer is using a gestural controller or an acoustic instrument as input). For electroacoustic performers, this limit is often lower if they are actively monitoring their processed sounds. It has been shown that percussionists can play to a metronome with a synchronization error of 2 ms or less [40]. Designers of DMIs should strive to minimize latency as much as possible [84], [56]. Jack et al. showed that increasing a system's latency from 10 ms to 20 ms, or adding 3 ms of jitter to 10 ms of latency significantly degraded the perceived quality of DMIs [49].

The latency of the internal processing of the Helix can vary, but in most situations it adds less than 2 ms of latency. Sending the audio to the computer over USB adds a bit more latency. This is all unavoidable, but important to minimize in order to maintain coherence between the live instrument and processed audio.

6.5.1 Input/Output Device Experiment

In an effort to minimize latency, I experimented with many different interface setups. Before adopting the Helix, my main audio interface was a Universal Audio Apollo Twin. At first I debated about whether to use the Helix as an audio interface or to send its output to the Apollo. The Helix works well for my setup as an audio input device, but the Apollo had many advantages over the Helix as an output device. To make a decision, I used Ableton Live to measure the reported latency for different configurations with different audio interfaces. All of the measurements were taken with the buffer size set to 128 samples and the samplerate set to 48 kHz. These are my preferred settings when performing, but the results should remain consistent relative to each other at different samplerates. This is demonstrated below. The audio interfaces included are the MacBook Pro's built-in microphone and speakers (used as a baseline), a MOTU M4, a Universal Audio Apollo Twin, and the Line 6 Helix.

The results of this are shown in Table 6.1. For reference, with a buffer size of 128 samples and samplerate of 48 kHz, the theoretical minimum latency possible is 2.67 ms. This is consistent with 2.67 ms consistently being the lowest input latency reported by the system. The frequency at which the lowest output latency (3.17 ms) appears suggests it is the lowest possible output latency. An interesting result is

Input Device	Output Device	Input Latency	Output Latency	Overall Latency
MacBook Pro Built-In Microphone	MacBook Pro Built-In Speakers	**2.67 ms**	4.38 ms	7.04 ms
UA Apollo Twin	UA Apollo Twin	8.75 ms	**3.17 ms**	11.9 ms
Line 6 Helix	Line 6 Helix	8.67 ms	10.7 ms	19.3 ms
MOTU M4	MOTU M4	3.17 ms	**3.17 ms**	6.33 ms
UA Apollo Twin	Line 6 Helix	**2.67 ms**	10.7 ms	13.3 ms
MOTU M4	Line 6 Helix	**2.67 ms**	10.7 ms	13.3 ms
Line 6 Helix	MOTU M4	**2.67 ms**	**3.17 ms**	**5.83 ms**
Line 6 Helix	UA Apollo Twin	**2.67 ms**	**3.17 ms**	**5.83 ms**

Table 6.1: Latency measurements of different audio interface combinations in Ableton Live on 2019 MacBook Pro with a 128-sample buffer size and 48 kHz samplerate.

that the latencies for both the Helix and the Apollo Twin change based on whether they are used for only input or output versus both. In fact, the Helix has the worst overall latency of any device when used for input and output simultaneously. In order to minimize the overall system latency for performance, a separate audio interface should be used for output. According to Table 6.1, both the Universal Audio Apollo Twin and MOTU M4 have an output latency of 3.17 ms for the chosen settings. The choice then to use the Apollo Twin for performance came down to capabilities. While both interfaces have four output channels, the Apollo Twin is able to output a separate headphone mix while the M4 headphone output is always the same as the main outputs. In performance, I often use cues, click tracks, or custom mixes that should not be amplified for the audience. Since all four outputs of the M4 would be used for quadraphonic playback, that would make it impossible to set up a separate headphone mix. When using stereo playback, it is possible to use one of the M4's pairs of output channels as a separate headphone mix so either interface should work equally well in that situation.

6.5.2 Buffer Size vs Samplerate Experiment

Another consideration when using a DAW in realtime performance is the computer CPU load. Beyond the obvious of amounts and types of plugins, there are a few parameters that affect this greatly. First is samplerate. The higher the samplerate, the more samples per second that must be processed. A samplerate of 96 kHz requires the system to process twice as many samples as 48 kHz. However, there are many

Samplerate	Buffer size	Input Latency	Output Latency	Overall Latency
44.1 kHz	32 samples	0.73 ms	2.00 ms	2.72 ms
44.1 kHz	64 samples	1.45 ms	2.00 ms	3.45 ms
44.1 kHz	128 samples	2.90 ms	3.45 ms	6.35 ms
44.1 kHz	256 samples	5.80 ms	6.35 ms	12.2 ms
44.1 kHz	512 samples	11.6 ms	12.2 ms	23.8 ms
48 kHz	32 samples	0.67 ms	1.83 ms	2.50 ms
48 kHz	64 samples	1.33 ms	1.83 ms	3.17 ms
48 kHz	128 samples	2.67 ms	3.17 ms	5.83 ms
48 kHz	256 samples	5.33 ms	5.83 ms	11.2 ms
48 kHz	512 samples	10.7 ms	11.2 ms	21.8 ms
96 kHz	32 samples	0.33 ms	1.00 ms	1.33 ms
96 kHz	64 samples	0.67 ms	1.00 ms	1.67 ms
96 kHz	128 samples	1.33 ms	1.67 ms	3.00 ms
96 kHz	256 samples	2.67 ms	3.00 ms	5.67 ms
96 kHz	512 samples	5.33 ms	5.67 ms	11.00 ms

Table 6.2: Input/Output latencies by samplerate and buffer size with Line 6 Helix as input device and a Universal Audio Apollo Twin as output device on a 2019 MacBook Pro.

other factors and doubling the samplerate does not necessarily imply a doubling of CPU. Another factor is the buffer size. The shorter the buffer size, the more "computational overhead" there is in the system. It must move data around and call functions more frequently which increases the CPU load even when the samplerate is left consistent. Since these parameters are controllable by the user, they should be selected to ensure glitch-free audio processing while minimizing the latency since both buffer size and samplerate will affect the system latency. To demonstrate this, I measured the system's latency using the Line 6 Helix and an input device and the Universal Audio Apollo Twin as an output device. I measured the input, output, and overall latencies for buffer sizes of 32, 64, 128, 256, and 512 at samplerates of 44.1 kHz, 48 kHz, and 96 kHz. The results are presenting in Table 6.2.

There are several interesting results from this experiment. First, there is clearly a minimum output latency. At all three samplerates, the output latency is the same for buffer sizes of both 32 and 64 samples. This suggests that while Ableton can process buffer sizes of 32 samples, macOS likely has a minimum output buffer size of 64 samples. Another interesting result is that the latency difference between corresponding buffer sizes for 48 kHz and 96 kHz samples rates is not quite a factor of

2. This is due to the output latencies. The input latencies are at their theoretical limits. The output latencies, however, require some additional time. At 44.1 kHz, this is 0.55 ms; at 48 kHz, it is 0.50 ms; and at 96 kHz, it is 0.33 ms. One can see that at buffer sizes greater than 64, the output latency is consistently higher than the corresponding input latency by these amounts.

A samplerate of 48 kHz and buffer size of 128 samples was chosen as a compromise between several factors. First, at buffer sizes smaller than 64 samples, audio glitches started to occur. Second, 48 kHz raises the Nyquist limit a small amount without significantly increasing the CPU load. This is important in avoiding audio aliasing. A samplerate of 96 kHz would raise the Nyquist limit greatly, but also significantly raises the CPU load. Based on personal experience, these settings were found to be the most consistently reliable. When working on a project with greater computational requirements, I increase the buffer size to 256 samples. This raises the overall latency to 11.2 ms. This is still under the general threshold for perception, but considering other sources of latency in the pre-computer signal chain from the Helix, other pedals, etc. it is on the borderline. I have not experienced noticeable issues due to latency when using a buffer size of 256, but I still prefer a buffer size of 128 provided the CPU is not being overloaded by audio processing demands.

6.6 The Back End: Ableton Live

As mentioned, the next step in my signal processing chain after the Helix is Ableton Live. I will discuss here my motivation for adopting Ableton Live as my primary DAW and the experience I gained composing and performing new electroacoustic works for steelpan. The new pieces I present in the following chapters were all performed with quadraphonic playback. I will describe my method for achieving that in Ableton Live as well as other recommendations for successful performance with Live.

6.6.1 Motivation

Prior to 2022, I did not have significant experience with Ableton Live, but used Max/MSP extensively. For a DAW, I usually used Apple Logic. However, I had encountered a frustrating dilemma: I wanted the non-linear and signal processing power of Max/MSP, but also sometimes wanted to make music that used audio effects/instrument plugins or that relied on a timeline. While Max/MSP is technically

capable of hosting plugins and following a timeline, both of those tasks are not convenient in the environment. For these reasons I decided to experiment with use Ableton Live due to its ability to host Max for Live, a version of Max/MSP that runs inside of Live, and immediately knew it was the right DAW for my purposes. As will be discussed in the coming chapters, I have created many custom audio effects that would not be possible in other environments.

6.6.2 Quadraphonic Output in Ableton

Something that differentiates Live from most other DAWs is that every track is always stereo. There are no mono tracks in Live. This simplifies signal routing since there is no need to consider the implications of passing audio between mono and stereo tracks. However, it also means that Live is unable to have a 4-channel track. This makes quadraphonic processing difficult and often convoluted. Other DAWs, such as Reaper and Logic have been working to improve their multi-channel track support in recent releases.

Live, however, remains stereo only. Due to this limitation, I have had to develop techniques for quadraphonic processing. In most arrangements for quadraphonic audio, the speakers are numbered clockwise in a circular fashion starting with the front left speaker (from the listener's perspective). This results in a numbering as in Fig. 6.8. In a standard quadraphonic track, the user should be aware of this to make sure audio is being processed and routed correctly, but for the most part this is transparent to the user. The track behaves like a stereo track, it just has four channels of audio being processed.

In Live, I have had to develop a custom method for quadraphonic processing. I use two stereo tracks: one for the front speakers and one for the back speakers. If one uses the standard numbering system as in Fig. 6.8, however, this makes panning a sound confusing since it reverse the pan direction. For this reason, I use a speaker numbering as in Fig. 6.9. This allows for coherent panning between left and right for the front and rear speakers. However, blending a sound between the front and rear speakers is still more difficult.

The accomplish this routing inside of Ableton, I create two return tracks. I label these "Out 1/2" and "Out 3/4" and route them to outputs 1/2 and 3/4 respectively. I then create a third return track for monitoring that I label "HP Mix" and route it to the headphone output channel. A custom headphone mix can be made through

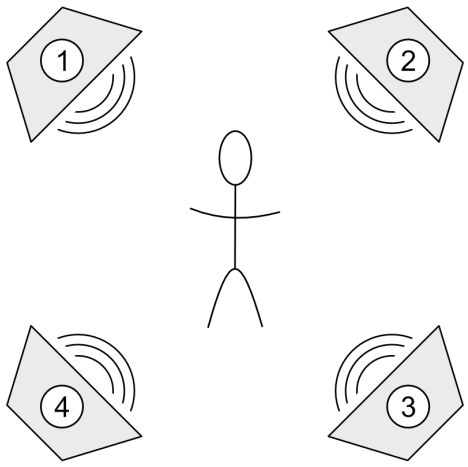


Figure 6.8: Standard quadraphonic channel numbering.

sends to the "HP Mix" return track. This basic setup can be seen in Fig. 6.10.

An important guideline to follow in this setup is to never send audio directly to the master output as one normally would. Instead, each audio and MIDI track's audio output setting should be 'Sends Only'. Then the output levels are controlled using the appropriate sends. The balance between the front and back speakers must be controlled manually through the two send controls. This method of adjusting the front/back balance of a signal can be awkward since it involves controlling two parameters simultaneously. This limitation, however, has not impacted my practice since I have explored spatialization in other ways.

Envelop for Live

Another potential solution to using spatial audio in Live is the Envelop for Live library. This is a free, open source suite of Max for Live devices designed to enable ambisonic mixing in Live. The suite is very powerful, fully featured, and an excellent option for someone doing ambisonic mixing.

I experimented with using Envelop for Live to create quadraphonic mixes of my compositions, but ran into a significant issue. Envelop for Live is designed for mixing

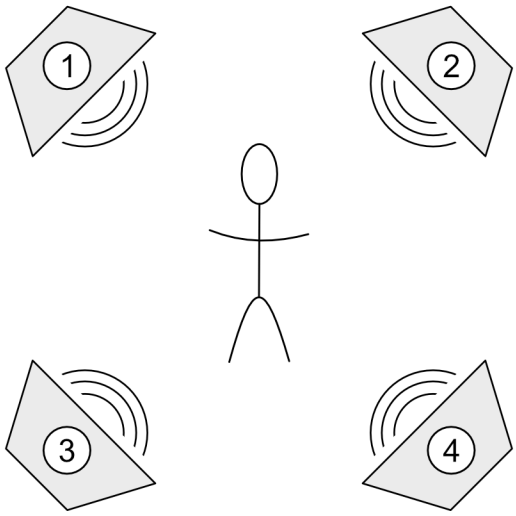


Figure 6.9: Custom quadraphonic channel numbering.

and not to accommodate a separate headphone mix. I use a custom headphone mix on almost every piece so that I can hear the important things I need to hear while performing as well as audio only for me such as cues and click tracks. Envelop for Live uses creates a new audio track to act as the master track and only supports a single ambisonic decoder on that channel. I could not send a quadraphonic mix to the speakers and a binaural mix to the headphones simultaneously. Due to this limitation, I decided not to use Envelop for Live and developed my own method presented above.

6.6.3 Performing with Ableton Live

As its name suggests, Ableton Live is a DAW designed with performing live in mind. It makes checking latency simple, allows for advanced parameter mapping, and has tools for reducing CPU load. Live even has a version of Max/MSP (called Max for Live) embedded within it. All of these features combine to make Ableton an excellent DAW for live performance.

Figure 6.10: Return tracks in Ableton Live for quadraphonic output and headphone mix.

Max for Live

I use Max/MSP regularly in my electroacoustic compositions. It provides a flexible visual programming environment for audio, video, and MIDI processing. Max for Live is a version of Max/MSP that runs inside of Ableton Live. The main strength of Max for Live is that it can be used to create what are called Max for Live devices – which are essentially audio or MIDI plugins that can be edited in Max/MSP. In this way, a user of Max/MSP can conveniently make custom plugins that are customized for their particular situation. Fully describing the breadth of possibilities Max for Live enables is beyond the scope of this document. Where applicable, in future sections I will describe any specific tools I developed in Max for Live for use with the electroacoustic steelpan.

Parameter Mappings

An aspect of Ableton that differentiates it from many other DAWs is that almost any parameter in any virtual instrument, audio effect, or Max for Live device can be automated or be mapped to. A plugin parameter can easily be mapped to a be controlled by a MIDI device or by a device within Live. This is a convenient feature for performance. I do not use it, but Ableton's Push controller also maps seamlessly with Live. If a performer needs to interact with software hosted within Live, but wishes to avoid the mouse and keyboard, Push is an excellent option. Of course, there are also countless other powerful MIDI controllers than may suit a particular situation even better than Push.

A collection of Max for Live devices are also provided by Ableton specificically for modulating other plugins' parameters. The devices bundled with Live Suite include Envelope Follower, LFO, Shaper, and Vector Map. I use Envelope Follower and LFO regularly add a subtle dimension of liveliness to my pieces. Many effects, in particular reverbs, offer built-in modulation in order give the effect a bit of randomness and variety. That way even if the same sound is sent through the effect twice, it should sound subtlely different each time. With the Envelope Follower and LFO Max for Live devices, a similar feature can be applied to almost any plugin. For this use case, it is best to use subtle settings.

Freezing Tracks

As mentioned in Section 6.5.2, CPU load is an important concern while performing live. When mixing audio, if the CPU is overloaded then the user can increase the buffer size to reduce the CPU load. In live performance, the performer is not afforded this option since increasing the buffer size will also increase the system latency. Ableton provides a convenient option for overcoming this situation: freezing tracks.

In Ableton, freezing a track is similar to bouncing in place. All of the settings for virtual instruments, plugins, and other utilities are "frozen" and the audio for the track is rendered in the background. The frozen audio includes all audio processing, automations, delay/reverb tails, and other aspects of the audio. It sounds just like it would have before, but when frozen the audio processing does not have to take place in realtime. The user cannot adjust any settings (other than track settings like level, panning, sends, etc.) unless they unfreeze the track. The user also cannot manipulate

the frozen audio.

Freezing audio is reversible. If the user decides that they wish to adjust a parameter, they can unfreeze the track, make the adjustment, and then refreeze the track. Live will then have to re-render new frozen audio.

If the user desires to manipulate the frozen audio, they can proceed to "flatten" the audio. This removes all effects from the track's effect chain, if it is a MIDI track it is converted to an audio track, and then the audio track is populated with the frozen audio track. While this action can technically be undone, it is only recommended if the user is confident they do not need the processing chain, MIDI regions, or other data from the original source. Once the flatten process is far enough back in the undo history, it becomes impractical to undo without losing all subsequent actions. A frozen track can be unfrozen and refrozen at any time.

The primary benefit of this is that for a DAW to play an audio track takes minimal computational power. So if an audio effect is too computationally intensive for a short buffer size, that can be mitigated by freezing the track. The one caveat, however, is that the processing of live audio cannot be frozen. For this reason, it is recommended that all non-live tracks are frozen for performance. This reserves as much CPU processing power as possible for any processing that must occur live. Flattening a track is only recommended if the user needs direct access to the rendered audio of the frozen track.

Minimizing Plugin Latency

As discussed in Section 6.5, latency is an important issue when performing with live electronics. There we explored latency associated with input and output devices. Plugins hosted inside of the DAW can also cause latency.

The most common reasons for plugins to cause latency is if they are using the Fast Fourier Transform (FFT) or "look ahead". An FFT is a process for converting a time-domain signal to the frequency-domain. The benefit of this is that some forms of processing is made simpler or it can allow processing that is not possible in the time-domain. The frequency resolution of the FFT increases as its frame length increases. However, a longer frame length means that the plugin must accumulate more samples before it can perform the next FFT to fill the frame.

Other effects have the ability to "look ahead" into the future to affect how it processes audio in the present. This, of course, is not how it truly works, but it can

give that impression to the user. In reality, if a plugin like a limiter has a 10 ms look ahead feature, it adds 10 ms of latency to the audio signal so it can "look ahead" and then process the audio based on what it measures from 10 ms ahead.

Ableton can conveniently display any audio effect's reported latency if the 'Delay Compensation' option is enabled (under the 'Options' menu). To check a plugin's latency, hover over the title bar of the plugin in a track's audio effect chain. In the lower left hand corner of Live's main window, it will display a message like "Latency: xxx samples (xx.x ms)". This is the amount of latency reported to Live by the plugin.

A warning to take note of is that plugins can lie to the DAW about their actual latency. The reported latency is typically used by the DAW for latency compensation. In Live, this is the 'Delay Compensation' option mentioned before. When enabled, Live will measure the latency for every audio chain and delay all tracks by that amount so that they remain synchronized despite having different amounts of latency.

This is very useful when mixing, but can be an issue when performing live. When working in Ableton, it is important to check the latency values for all plugins used. Many plugins have no or minimal latency, but some can have significant latency. For example, Eventide's power Physion is an amazingly power audio effect, but on my system it adds 7,440 samples (or 77.5 ms) of latency. This amount of latency would be clearly audible to an audience and would likely lead to a lack of temporal coherence between the acoustic instruments and the electronics.

When plugins add an unacceptable amount of latency, there are a few options to explore. First, can a similar effect that does not add latency be found? For example, the Arturia Comp FET-76 compressor plugin adds 512 samples (5.3 ms) of latency on my system, but the Universal Audio UADx 1176AE Compressor (which is a model of the same analog compressor) only adds 118 samples (1.23 ms). Or if even that much latency is too much, the user could use Ableton's Compressor which adds no latency. It will sound different than the 1176 emulations, but that is the trade off.

If no suitable replacements can be found, the next option is to freeze the track if the signal being processed is not live audio. As discussed in Section 6.6.3, Live renders a frozen track to an audio file and plays that back instead of processing the audio in realtime. Since the plugin is then deactivated by Live, it no longer adds latency. If the user must use, for example, the Eventide Physion plugin, this is best option to use.

The final option is to disable Live's "Delay Compensation" option. This is generally not recommended because then each track in Live will have a different amount

of latency depending on what plugins are active. The situation where this may make sense, however, is if the majority of tracks are using plugins with minimal latency and the track that does have significant latency will not sound incoherent due to the latency. For example, depending on the window size, my audio effect Realstretch can add over a second of latency. However, the nature of the effect also makes it still sound okay in most situations, even with the latency. In this case, the user should disable the Live's delay compensation, however, so that the other tracks are not delayed due to Realstretch's latency.

There is also a trick for processing "live" audio with audio effects that have a high amount of latency. This trick can be used if the live part is fully composed for the duration of the processed portion and the performer knows what they will play and when. In this situation, the performer can pre-record that part, freeze the output of the audio effect (fully wet), and then in the live performance instead of processing it live the frozen track is played back. This method is also generally not recommended, but sometimes is the best option for achieving the best performance possible.

6.7 Recital

On March 26, 2022, I presented a recital of new electroacoustic works for steelpan in PTY Recital Hall at University of Victoria as part of my PhD requirements [68], [69], [70], [72], [73]. The program consisted of four original compositions and a commission from composer Matthew Burtner. Links to video recordings of the performances are included in the citations. These pieces were composed to showcase the potential of the steelpan for electroacoustic performance and expand the repertoire. All of the pieces were presented with quadraphonic audio playback. The four original works all used the Line 6 Helix and Ableton Live for audio processing. In the following chapters I present analysis and reflection on the various aspects of designing electronics for, composing, and performing the pieces.

Chapter 7

Crushed Atmos

Crushed Atmos is an electroacoustic piece for solo tenor steelpan composed in early 2022. It was premiered on March 26, 2022 in the Phillip T. Young Recital Hall at University of Victoria as part of my PhD recital [68]. It featured quadraphonic playback and utilized the Line 6 Helix for the audio signal processing. The score and a link to a video recording of the premiere are included in Appendix A.1.

7.1 Audio Signal Chain

All of the audio processing for *Crushed Atmos* was performed on the Line 6 Helix. The Helix was still used as a USB audio interface for Ableton Live, but this was necessitated by Ableton being used for other pieces during the concert. Ableton received two stereo audio tracks from the Helix and output them to the quadraphonic speaker array without modification. One stereo track was routed to the front speakers and other to the back speakers.

7.1.1 Bitcrusher

Crushed Atmos was born out of experimenting with the distortion algorithms included in the Line 6 Helix. A bitcrusher is an effect that distorts an audio signal primarily by lowering the digital bit depth or samplerate resolution. When these parameters are lower enough, it adds digital quantization noise to the signal. Depending on the settings, this can add "warmth" or "brighgtness" to the signal [18].

The Helix's Bitcrusher algorithm, aside from being a high quality bitcrusher effect, has a feature that distinguishes it from typical distortion effects. The key feature of

Bitcrusher that *Crushed Atmos* takes advantage of is that the effect has a built-in noise gate that is opened or closed based the input signal level. The open and close levels are user controllable parameters. When in the input audio signal is greater than the 'Open' threshold setting, then the bitcrushing effect is applied. When the input signal's level drops below the 'Close' threshold setting, the bitcrushing effect is stopped and the dry input signal is passed through. There are two more associated parameters on Bitcrusher: 'Hold Time' and 'Decay'. The hold time is the minimum length of time that the gate will stay open even if the signal level drops below the gate close threshold while 'Decay' is the amount of time it takes for the gate to close. While the gate is closing, it blends between the bitcrushed signal and the dry signal.

As an experiment, I decided to see what would happen if I set the gate open threshold lower than the gate close threshold. For example, suppose the gate for the bitcrusher opens at -35 dB, but the gate closes at -10 dB. Then if the input signal exceeds -35 dBFS, it will trigger the bitcrushing effect, but unless it also exceeds -10 dBFS, then it will only last for as long as the hold parameter is set to. When the hold time is very short (around 25-100 ms) this results in what sounds like a temporary audio glitch. With settings like these, the effect will in essence "flicker" on and off at random while the level of the input signal is above the gate open threshold. Steelpan notes generally have sharp attacks and releases which makes it convenient for achieving the randomly triggered bitcrushing. More can be done to accentuate this affect and the full signal processing chain is discussed in Section 7.1.2.

7.1.2 Full Helix Processing Chain

The signal processing on the Helix consists of two nearly identical stereo processing chains. The two chains consist of almost all the same audio effects blocks and use the microphone for audio input. The processing chain as displayed by the Helix UI is shown in Fig. 7.1.

The first chain is sent to the front speakers while the second is sent to the back speakers. The processing blocks for the first chain consist of:

LA Studio Comp → Simple EQ → Minotaur → Searchlights → Shimmer → AM Ring Mod → parallel Bitcrushers → 10 Band Graphic EQ

and for the second chain:

LA Studio Comp → Simple EQ → Kinky Boost → Searchlights → Shimmer → AM Ring Mod → parallel Bitcrushers → 10 Band Graphic EQ.

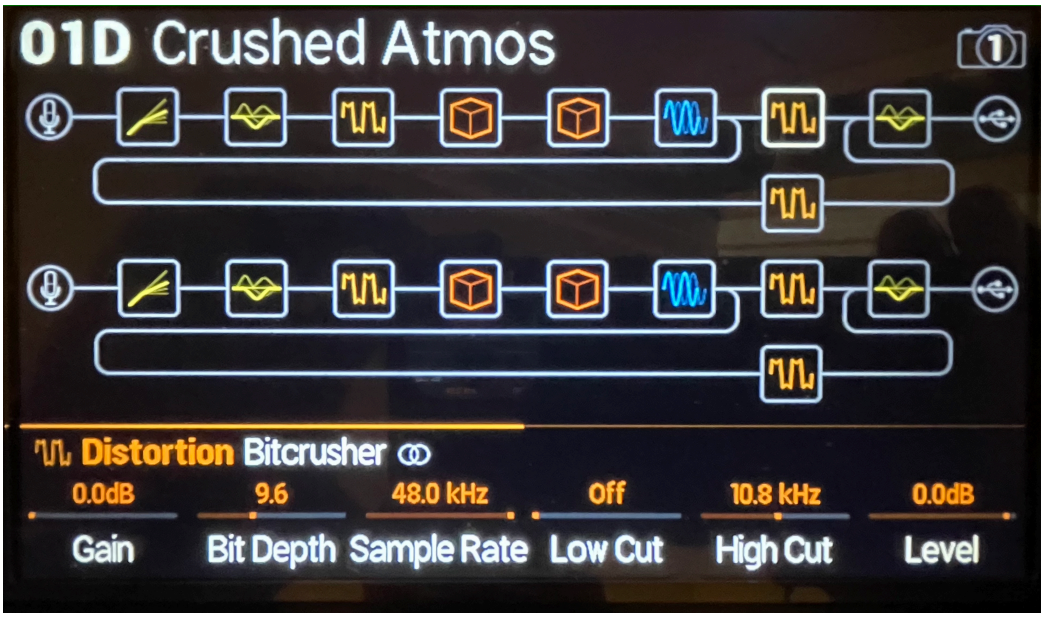


Figure 7.1: Processing chain on the Helix.

The only difference in audio effect block between the two chains is the third block. On the first chain it is the Minotaur distortion—a model of the Klon Centaur pedal—while on the second chain it is the Kinky Boost—a model of the Xotic EP Booster, which is itself an analog boost pedal designed to emulate the tone of the preamp in the Echoplex EP-3 tape delay.

The LA Studio Comps at the front of the processing chains are used to tame note transients a bit and boots the tails of notes to add some sustain. The Simple EQ blocks are used to cut the high frequencies a bit. The Minotaur and Kinky Boost effects are then used to boost the signals a bit for their respective chains. On the Minotaur, the 'Gain' parameter is set to 3.5, the 'Tone' to 3.4, and the 'Level' to 4.5. On the Kinky Boost, the 'Drive' parameter is set to 1.5 while both 'Boost' and 'Bright' set to 'off'. These effects boost and filter the signals a bit, but do not cause the signals to actually distort yet.

After that, both effects are pass through two reverb effects blocks: first the Searchlights and then Shimmer. Searchlights is a reverb effect similar to the Cloud algorithm from the Strymon Big Sky reverb pedal. It produces big, ambient reverbs with modulation. Its purpose on the signal chain is to greatly lengthen notes, elongating the substance of the steelpan. Its 'Decay' setting on is set to 9.1 (out of 10). This reverb has a long decay tail. Searchlights's modulation settings are designed to make it heavy,

but slow. 'Modulation' is set to 8.7, 'Speed' to 1.9, 'Intensity' to 8.3, and 'Spread' to 9.5. The reverb sounds like it is slowly, but constantly shifting. The settings for the second processing chain are all similar, but with very small variations.

The next effect block for both processing chains is Shimmer reverb. A shimmer reverb is an ambient reverb effect that includes pitch shifting in the feedback path. As the reverberating sound flows through the feedback networked of the reverb, it is gradually shifted up or down in pitch. Depending on the pitch shift intervals, this can create a mood that varies from beautiful and ethereal to unsettled. For *Crushed Atmos*, the pitch shift interval parameters are mapped to the Helix's expression pedal. At the expression pedal's starting position, the front Shimmer reverb's pitch shift intervals are set to shift down a perfect fourth and up and perfect fifth. As the expression pedal is titled forward, these intervals continuously shift to up 30 cents and up 40 cents. For the back Shimmer reverb, the intervals start at a perfect fifth both up and down. They then end at a major second down and up when the expression pedal is at the toe position. By mapping the pitch shift intervals to the expression pedal, the character of the reverb can be actively controlled by the performer. At the heel position, the reverb has a nice, lustery sheen to it, but at the heel position it becomes a bit more tense due to the smaller pitch shift intervals. The reverberant signal with the smaller shifts clashes more with the dry signal.

The two reverb effects combine to create a fluffy pillow of ambient sound. The next effect in the chain is the AM Ring Mod—a ring modulation effect. The front AM Ring Mod's 'Frequency' parameter is set to 1.1 kHz, the 'AM' parameter is set to on, and the 'AM Freq' parameter is set to 997 Hz. For the back instance, the 'Frequency' and 'AM Freq' parameters are set to 989 Hz and 1.2 kHz. These ring modulators are set to create a subtle sense of "electric sizzle" to the sound. Their effect is not nearly as prominent as the bitcrushing effect, but they do keep the sound from becoming too lush and vibrant from the reverbs. The 'Mix' parameters are mapped to control so that they can be disabled after the introduction of *Crushed Atmos*.

Next, both processing chains are split into A and B paths. On each path is a stereo instance of Bitcrusher. All four Bitcrushers have similar settings, but have intentional, small variations so that they will behave slightly differently on the live audio signal. The settings for the 'Bit Depth', 'Hold', 'Decay', and 'Mix' settings for each instance are shown in Table 7.1. The 'Open' parameter for all four Bitcrushers was mapped to be controlled by the expression pedal while the 'Close' was set to -15.0 dBFS for all four. The mappings are such that when the expression pedal is

in the heel down position, the Bitcrushers will not activate. When the pedal is in the toe down position, the Bitcrushers will activate very easily. As mentioned in Section 7.1.1, since the gate hold parameter settings are so short, the Bitcrushers will repeatedly activate and deactivate when the audio signal level crosses the gate open threshold.

Bitcrusher	Bit Depth	Hold	Decay	Mix
Front A	9.6	23 ms	50 ms	35%
Front B	8.3	47 ms	50 ms	35%
Back A	8.4	27 ms	50 ms	35%
Back B	10.1	54 ms	50 ms	35%

Table 7.1: Settings for the four stereo Bitcrushers.

For each chain, the outputs of its two Bitcrushers are then mixed together and run through a 10 Band Equalizer effect block. This equalizer performs a small cut in the upper-mid frequencies to compensate to prevent the distortion from becoming too prominent. The two stereo processing chains are then sent to Ableton Live over USB on channels 1/2 and 3/4. Ableton does not perform any further processing. It just routes the audio from the Helix to appropriate speaker.

7.2 Analysis

Crushed Atmos begins with a long, slow roll on the low C and G notes inspired by process pieces like Tenney's *Having Never Written a Note for Percussion* and Lucier's *Silver Streetcar for the Orchestra*. The roll (shown in Fig. 7.2) starts at a dynamic triple piano and the performer is instructed to crescendo to double forte over a long period of time—at least a minute. As the performer slowly builds the volume, they also gradually tilt the Helix's expression pedal forward. As the pedal creeps forward, the pitch shift intervals for the Shimmers adjust, but so too do the gate open thresholds for the Bitcrushers. They begin at 0 dBFS—too high to possible trigger the bitcrush effect at low volume levels—and slowly work their ways down to -55 dBFS—at which point almost any sound will trigger them.

At first, the Bitcrushers are not triggered at all. There is just a wash of luxurious reverb. As the volume of the steelpan increases and the level at which the bitcrushing is triggered decreases, little pops of distortion gradually begin to poke through the texture. At the mid-way point, the Bitcrushers are still only being intermittently

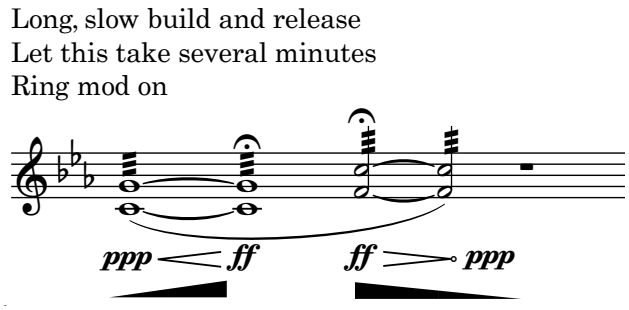


Figure 7.2: Opening of *Crushed Atmos.*

triggered by the input audio—but there are also four stereo instances. That adds up. By the time the performer reaches a dynamic level of forte the sound is more distorted than not, but there are still holes in the wall of static. At its peak however, the four Bitcrushers are so active that even though they are set up to only flicker on and off, their constant activations stack to complete the barrage of distorted steelpan audio.

At this point, the performer holds the hold at its peak before suddenly changing to the notes F3 and C4. The descent back to triple piano has begun. The difficulty here is to diminuendo in a controlled fashion both in terms of the instrument's tone as well as the audio processing. The change in dynamic must be gradual and smooth because of the reverb effects. Any dips or bumps in the dynamic level will be persist and then be passed to the Bitcrushers. The rate of pops of distortion will reveal any inconsistencies in the performer's touch.

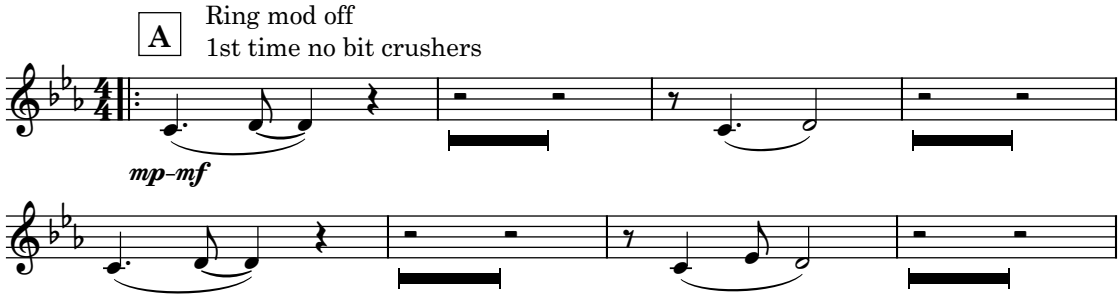


Figure 7.3: Mm. 2-9 of Section A.

After a pause to let the effect of the long introduction wash over the audience, Section A begins. The whole section is 16 measures long and played twice. It features a simple melody designed to show off sound processing for the piece. The first eight

measures are shown in Fig. 7.3. To contrast with the introduction, the first time Section A is played, the Bitcrushers should be disabled. The ring modulation is also disabled for the remainder of the piece.

On the second time Section A is played, the dynamic increases a bit and the Bitcrushers are enabled, but they are still controlled using the expression pedal. The instructions for controlled the pedal are given in the score. The pedal should default to heel position (Bitcrushers off). In the score, thick black horizontal lines below the staff indicate when the pedal should be shifted to toe position. Figure 7.3, shows that in Section A, the performer puts the pedal into toe position on beat 1 and the releases it on beat 3 in the second, fourth, sixth, and eighth measures. In this way, the bitcrushing effect will catch the reverb tail and flicker periodically as the reverb tail fades. In Section A, this effect is subtle because the melody focuses on the lowest notes of the steelpan and is only played at a mezzo forte level.

Section B is a recapitulation of Section A, but up an octave and louder at a dynamic level of forte. Aside from the instruction to increase volume, playing the notes in the middle range of the steelpan also causes them to pop out a bit more in the wash of reverb. When the expression pedal is pressed down every other measure, the level is noticeably higher than before and the Bitcrushers now have a clear activation and deactivation point controlled by the expression pedal of the performer. They crackles of distortion would continue past beat 3 if the performer did not return the expression pedal to heel position.

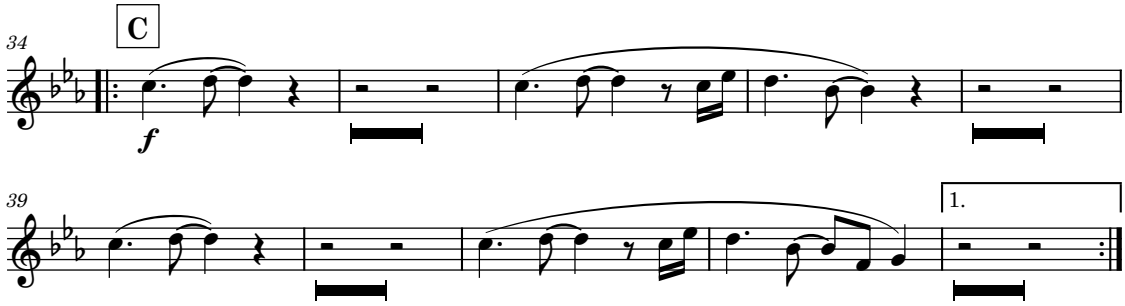


Figure 7.4: Section C, mm. 34-43.

Section C is similar to B, but features variations on the melody that elongates the phrases by a measure. This can be seen in mm. 34-43 in Fig. 7.4. The second phrase of Section C sets up the transition to Section D which uses the resolution from F to G in m. 42 as a focal point for the next melodic device.

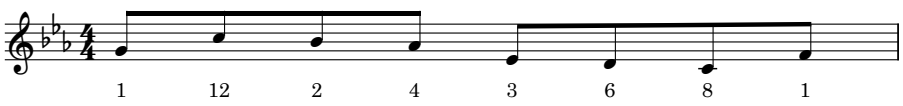


Figure 7.5: Section D melody source.

Section D was composed using an algorithmic approach. I composed an eight-note melody and assigned a number to each note as in Fig. 7.5. The number assigned to each note gives the frequency with which it reappears in the same rhythmic position. The first and last notes of the melody (G and F) both have 1 assigned and thus appear in every measure. The second note in the melody, A, has a 12 assigned to it and it reappears every twelfth measure. The third note in the melody, B♭, has a 2 assigned so it shows up in every second measure. This method continues for the remainder of the note and number pairs.

The overall length of the pattern generated by this algorithmic approach (before it would repeat itself) is determined by the least common multiple (lcm) of the assigned numbers. In this case,

$$\text{lcm}(1, 2, 3, 4, 6, 8, 12) = 24.$$

There is, however, some added repetitions after the 24th measure that make Section D actually 31 measures long in total. This method creates a sense of familiarity combined with movement and unveiling. Figure 7.6 shows the first 24 measures of Section D with all of the notes labeled.

This method of composition has a certain convenience, but there are some challenges as well. The main convenience is clear: once the system is set up, the composer just has to engrave the output. Defining a system like this that will work well musically is not straightforward. First, one must compose a short melody (one measure long in this case) that will work well when portions of it are masked. Although the masking is mathematically determined, it can sound random to the listener due to the long time scale on which the masking pattern occurs. The melody and masking pattern we designed to work well together to create melodic fragments that make sense on their own. This required significant trial and error. Most arrangements did not produce satisfactory results and I nearly eliminated the whole idea because I could not find patterns that I thought suited the piece.

Since Section D is focused on this new, contrasting material to that used in the earlier sections, I chose to have the bitcrushing effect present, but no longer be the

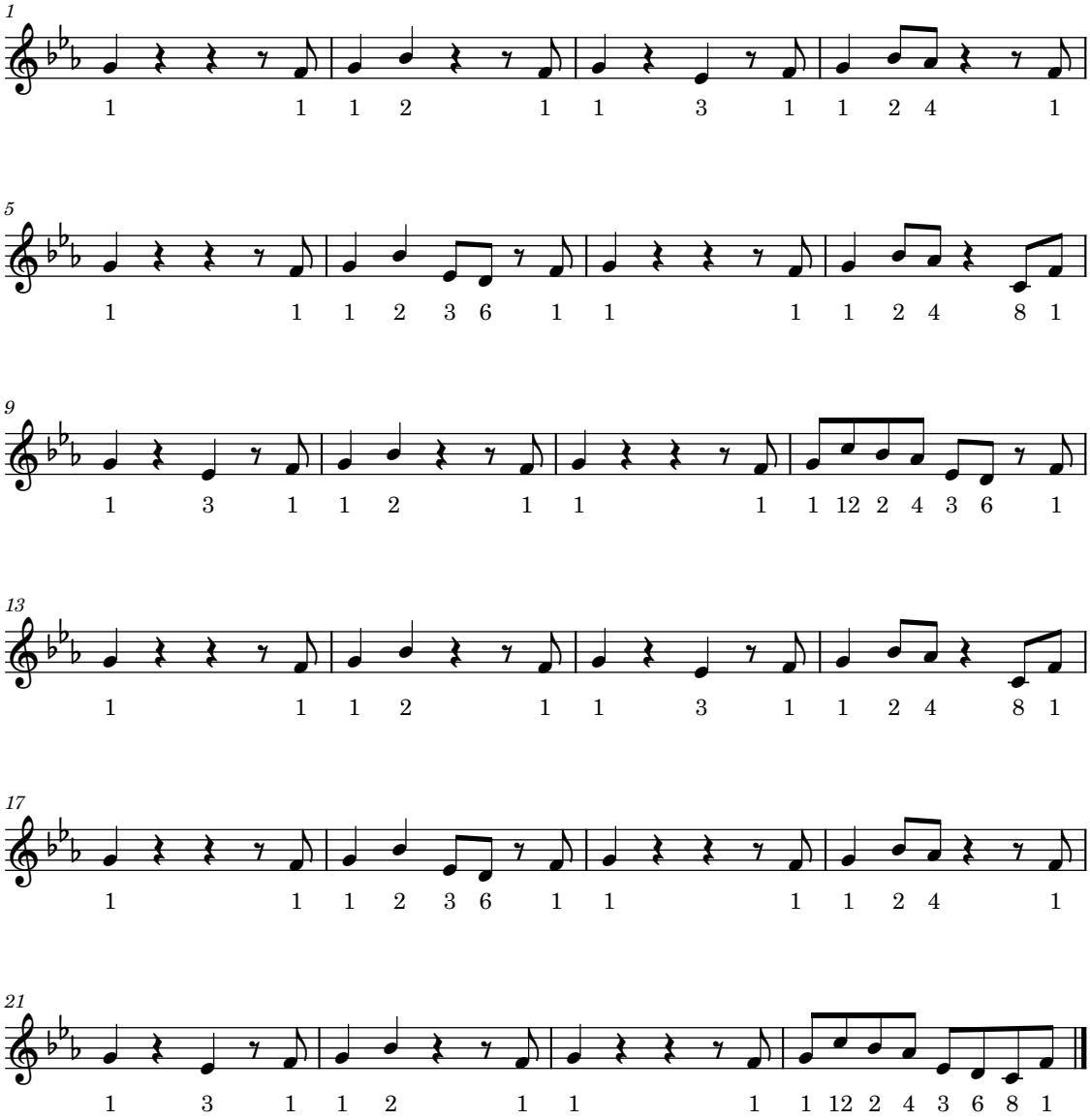


Figure 7.6: Mm. 55-78 of Section D annotated and renumbered for reference.

focus. The score instructs the performer to set the bitcrush effect to a moderate level (i.e., adjust the expression pedal to about 50%).

Section E is a recapitulation of Section C, but the pattern for activating and deactivating the Bitcrushers has shifted. This can be seen in Fig. 7.7. The black lines under the staff that indicate when the bitcrush effect should be active now starts with the beginning of the melodic phrase and continues only partially into the space between. While bitcrushed interjections become more sparse once the playing stops, they have not disappear completely by the time they are deactivated. This still

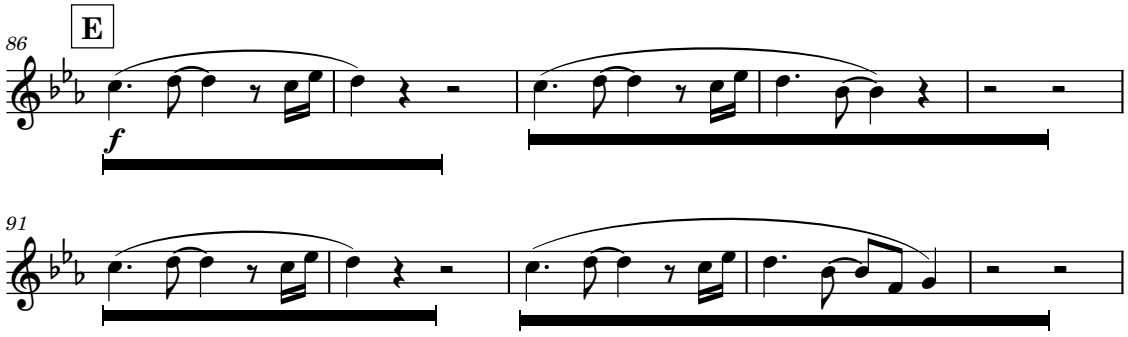


Figure 7.7: Section E.

results in a sudden cutoff which keeps this section interesting despite reusing melodic material from earlier in the piece.

Section F, shown in Fig. 7.8, is the final section of *Crushed Atmos*. The melodic material is a recapitulation of Section A, but once again the timing of the bitcrushing has shifted. For the first eight measures of Section F, it follows the pattern of Section E. However, as the dynamic level the performer decreases, the expression pedal can be left static with the Bitcrushers active since if everything is balanced properly they will only trigger intermittently.

This allows the piece to fade out gracefully—if with a melancholy mood—while still featuring the central element of the piece: the bitcrush effect. If it wasn't clear by this point, *Crushed Atmos*'s title is a reference to the importance of the reverb (atmosphere) and bitcrushing. *Crushed Atmos* explore some interesting audio processing territory. It features eight parallel bitcrushers (two stereo instances per processing chain = 8 mono bitcrushers) and extensive use of atmospheric reverb. The way that the bitcrushers are triggered, however, is likely the most interesting aspect of the piece. Utilizing an audio effect in a way that the designer probably never intended resulted in such a provocative sound that I was inspired to compose this work just to feature the sound design.

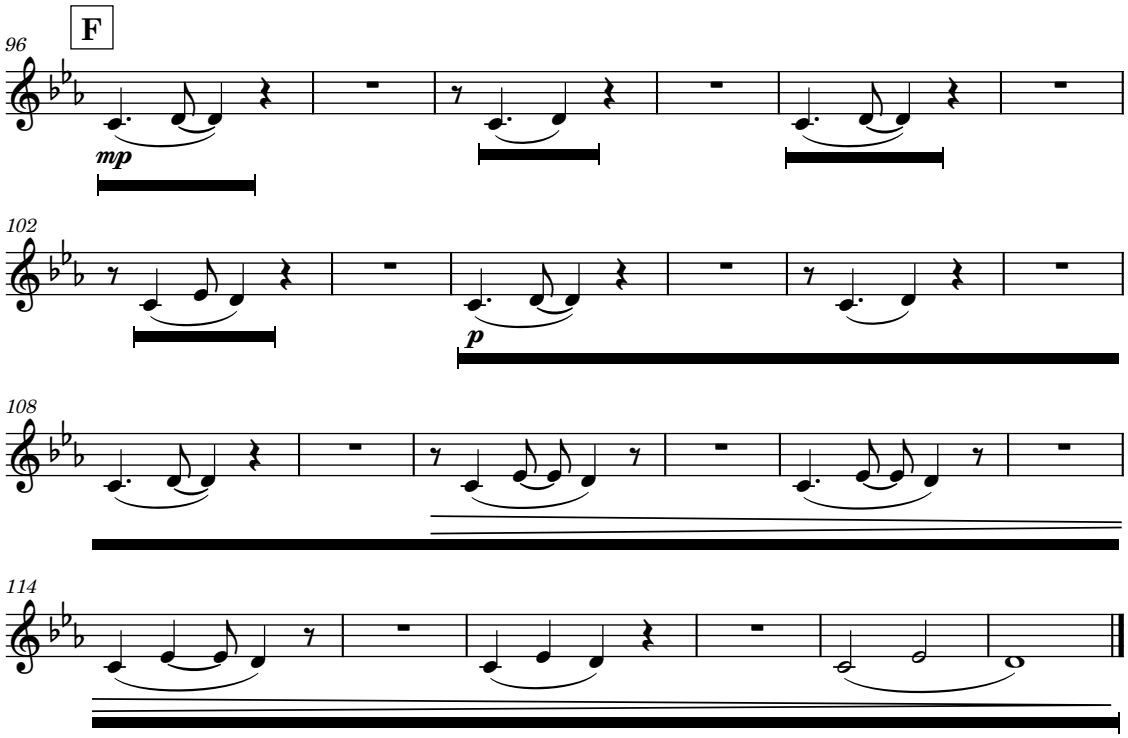


Figure 7.8: Section F.

Chapter 8

Steelpanopticon

Premiered at my PhD recital on March 26, 2022, *Steelpanopticon* is a piece designed to explore how distorted a steelpan audio signal can become [73]. Composed to be played with four mallets on a tenor steelpan, this is a loud, aggressive piece that is not commonly found in classical concert halls. Its compositional style draws from both the slow, organic grit of doom metal and the fast, odd-timed virtuosity of progressive rock. Throughout the piece, however, the acoustic timbre of the steelpan is always present to counterbalance the waves of distortion coming out of the speakers. The full score and a link to a recording of the premiere performance are included in Appendix A.2.

8.1 Audio Signal Chains

Steelpanopticon uses two different Helix patches for its processing: one based around fuzz distortion and the other around ambient reverb. The first patch uses the two Helix processing chain in serial. The output of the first chain is the input of the second chain. This gives the user a larger computational budget for processing, but limits them to a single processing chain.

8.1.1 Fuzz Distortion Chain

The fuzz distortion signal chain is shown in Fig. 8.1. About 2/3 of the way through the signal chain, it splits and is not recombined. The specific effect blocks up to the split are:

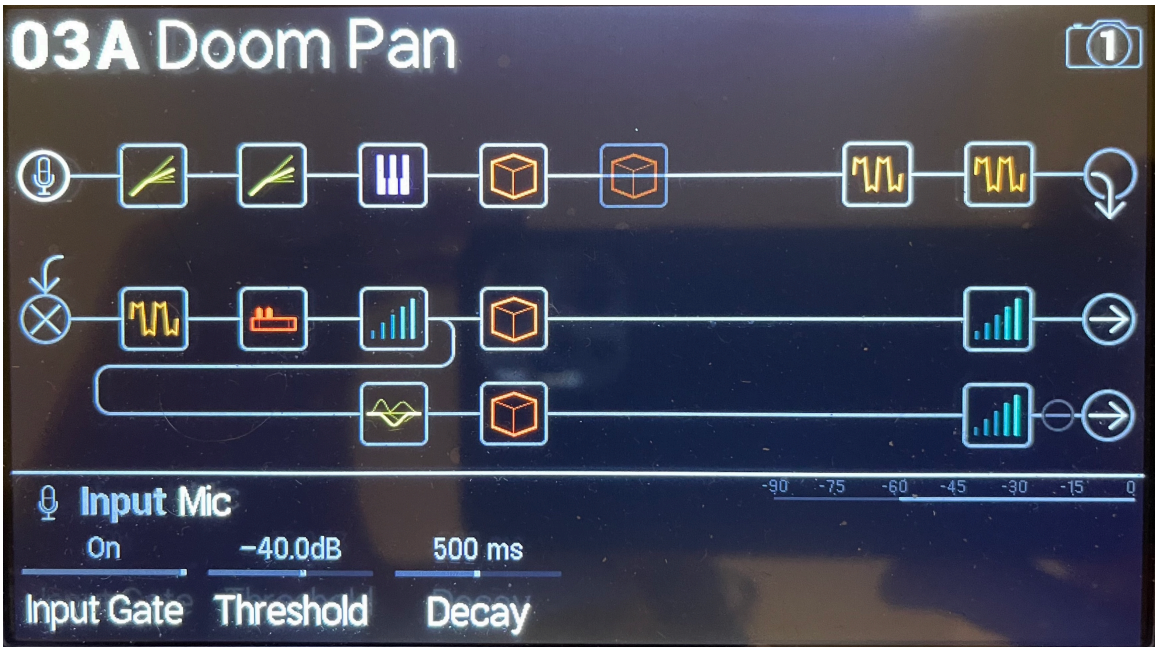


Figure 8.1: *Steelpanopticon* fuzz distortion signal chain.

LA Studio Comp → Noise Gate → Dual Pitch → Dynamic Plate → Dynamic Plate → Bighorn Fuzz → Wringer Fuzz → Vermin Dist → Line 6 Doom → Gain → Split A/B.

The effect blocks of split A are:

Split A → Dynamic Room → Volume Pedal → USB 1/2,

while the effect blocks of split B are:

Split B → Simple EQ → Dynamic Hall → Volume Pedal → USB 3/4

Splitting the signal chain into two stereo signals without re-combining allows the two splits to be sent to two stereo tracks in Ableton Live. Similar to *Crushed Atmos*, Live is not performing any processing on the computer and is just routing the audio to the proper speakers. The audio is being passed through Live because other pieces on the recital program required this setup.

The LA Studio Comp is a model of the Teletronix LA-2A compressor. Here it is used to control the steelpan's transients and to increase the sustain. The Noise Gate is present because further down the signal chain, multiple fuzz effects are used. Without the noise gate, the setup would occasionally feedback. The Dual Pitch effect is used to pitch shift the steelpan down both one and two octaves. The dry signal

is always passed through while the pitch shifted signal's level is controlled by the expression pedal. The next two blocks are both Dynamic Plate reverb effects. They have different settings and only one is used at a time. Two instances were necessary to facilitate a specific transition. After the reverb is the Bighorn Fuzz, which is a model of a 1973 Eletro-Harmonix Ram's Head Big Muff Pi fuzz pedal. The Bighorn Fuzz then feeds into the Wringer Fuzz. This fuzz model is based on a modified Boss FZ-2 owned by the band Garbage.

The output of the Wringer Fuzz is then routed to the second processing chain. At the beginning of this chain is the Vermin Dist—a model of the Pro Co RAT distortion pedal. This third distortion is only activated for the end of the piece. The next processing block is the Line 6 Doom guitar amplifier model. Next is a Volume block that reduces the signal by 4.5 dB. This is for setting levels in the performance space without having to adjust any of the earlier effects blocks. The output of the Volume block is the split into A and B signal paths.

Path A consits of two blocks: a Dynamic Room reverb with a short decay time of 0.9 seconds and then a Volume block. Path B consists of a Simple EQ, a Dynamic Hall reverb, and then another Volume block. The Dynamic Hall reverb is used add diffusion to the heaviness of the fuzz effects. Its 'Decay' times is set to 2.5 seconds and 'Mix' is set to 70%. Both Volume blocks are mapped to an expression pedal to control the volume level during performance. Path A is routed through Ableton Live to the front speakers while path B is routed to the rear speakers.

This effects chain turns the steelpan into a doom metal-inspired distortion machine. The Dynamic Plate reverb near the beginning of the signal chain is used add sustain to the notes. When combined with the two main fuzz effects, it makes the steelpan's notes last for seconds. The Dual Pitch reinforces the low end of the steelpan by lowering the pitch down both one and two octaves. This creates a violent tone that is completely unexpected from the steelpan. Since the acoustic sound of the steelpan will still be present in a reverberant concert hall, the processed signal in some ways becomes a distorted reverb tail for the acoustic steelpan. In order to achieve this effect in a large, non-reverberant space where the audience will only hear the amplified sound the performer will need to include dry steelpan signal in signal sent to the speakers. I recommend sending the dry steelpan audio only to the front speakers to reflect the actual physical spatialization of the steelpan.

8.1.2 Ambient Reverb Chain

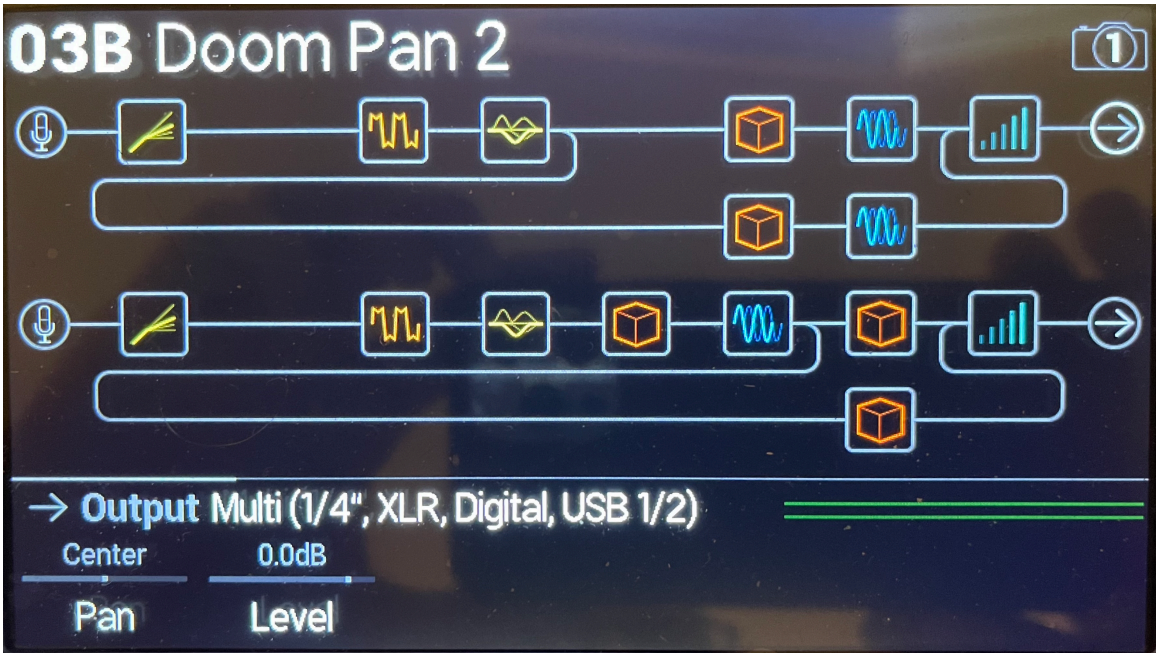


Figure 8.2: *Steelpanopticon* ambient reverb signal chain.

Unlike the fuzz distortion chain, the ambient reverb chain features two parallel signal chains—one routed to the front speakers and one routed to the back speakers.

The effects blocks of the first signal chain are:

LA Studio Comp → Kinky Boost → Simple EQ → Split A/B → Searchlights → Ampeg Liquifier → Mixer → Volume Pedal → USB 1/2.

This chain routes the live steelpan audio through a compressor, a boost effect, and an equalizer before the split. After the split, the signal is processed in parallel through two Searchlights and Ampeg Liquifier blocks and then recombined at the Mixer block. The Volume Pedal block is mapped to an expression pedal to control the effects level. This is then output to USB channels 1 and 2.

The effects blocks of the second signal chain are:

LA Studio Comp → Kinky Boost → Simple EQ → Hall → Gray Flanger → Split A/B → Dynamic Plate → Mixer → Volume Pedal → USB 3/4.

This chain routes the live steelpan audio through a compressor, a boost, and an equalizer before the Hall reverb block—an algorithmic hall-style reverb. The output

of the Hall is then passed to the Gray Flanger, which is based on the MXR 117 Flanger pedal. The signal is then split and processed in parallel through two Dynamic Plate reverb blocks. It is then recombined at the Mixer block before going through the Volume Pedal block and output to USB channels 3 and 4. This Volume Pedal is controlled by the same expression pedal as the Volume Pedal on the other chain.

8.2 Analysis

Steelpanopticon begins without subtlety: its first notes are a D5 chord on the steelpan, with dual pitch shifters dropping it by one and two octaves, and all of that pass through a reverb for some sustain and then through two different fuzz distortion pedals. The resulting sound is abrasive and shocking—not at all what one would expect from a steelpan.

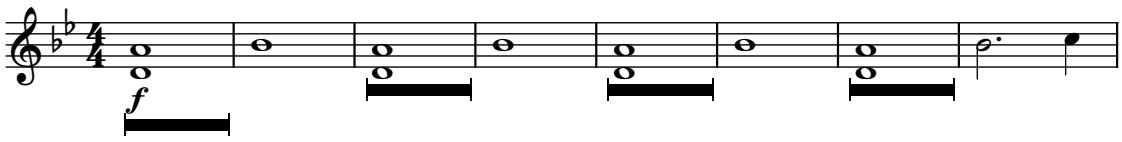


Figure 8.3: Opening line of *Steelpanopticon*.

The second note is a single B♭ a minor sixth above the root of the D5 (shown in Fig. 8.3). For this note, the performer adjusts the Line 6 Helix's expression pedal which fades the pitch shifters to the background. The black lines under the staff in Fig. 8.3 indicates when the expression pedal should be in toe position, when the pitch shifters are mixed into the signal. This emphasizes the high frequencies of the distorted signal to maximize the contrast between the two figures: low and high. This repeats two more times: low and high, low and high. On the fourth round, the performer plays a C on beat four after the B♭. This gives the sensation that it is resolving to a D, but the next sounding D is three octaves below the D that it would resolve to. These figures build a tension that wants to, but does not quite resolve.

An elaboration of this pattern unfolds in Section A shown in Fig. 8.4. Now the performers plays in parallel fifths, changing chords more frequently. In performance, I adjust the decay time of the Dynamic Plate reverb from 1.2 seconds for the introduction to 0.8 seconds starting at Section A. In the Introduction, the notes are much more sparse and the increased reverb time enhances the sustain. For the majority of Section A, the playing is much more dense and reducing the reverb time enhances the

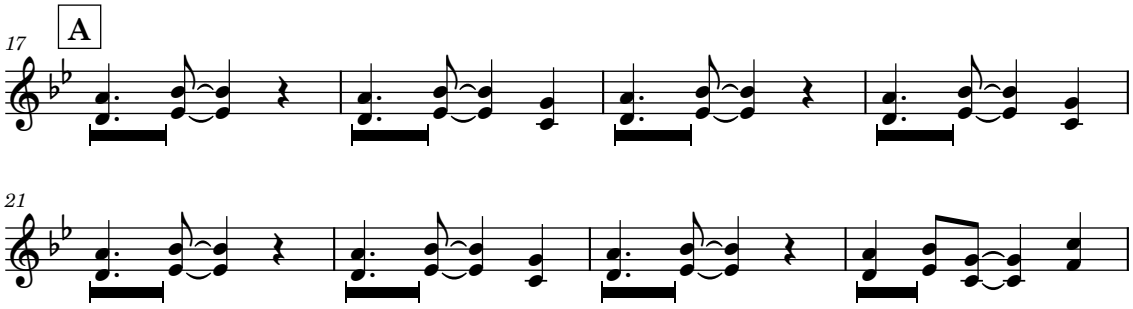


Figure 8.4: *Steelpanopticon* mm. 17-24 of Section A.

clarity. With multiple fuzz pedals active too much reverb causes the distorted signal to become unintelligible. Take, for example, m. 28 (shown in Fig. 8.5). With too much reverb that quick succession of fifths would become in distinct in the fuzz.

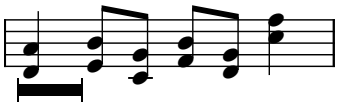


Figure 8.5: *Steelpanopticon* m. 28.

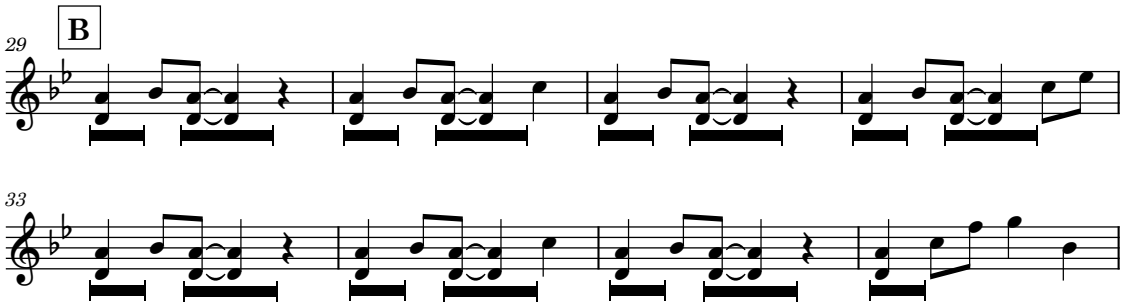


Figure 8.6: *Steelpanopticon* mm. 29-36 of Section B.

The first eight measures of Section B further elaborate on the material from Section A. This is shown in Fig. 8.6. This material features more frequent shifts between the pitch shifted D5 chords and higher voiced melody notes. The rhythm of this material gives a strong suggestion of groove—enhanced by the steelpan serving as both a bass and a melody instrument. After m. 36, the piece returns to abbreviated versions of the material from Section A and then the Introduction. This then concludes the first main portion of *Steelpanopticon*.

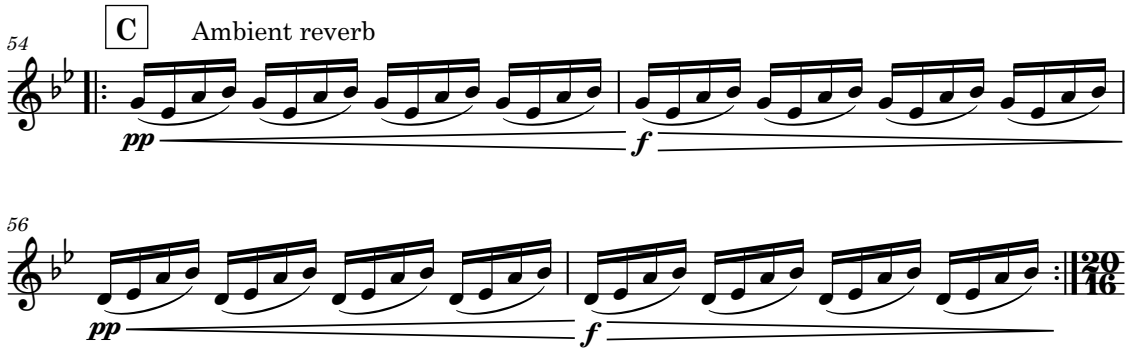


Figure 8.7: *Steelpanopticon* mm. 54-57 of Section C.

Section C introduces material that sharply contrasts with the first portion. The performer changes patches at this point to one that features modulated ambient reverb as described in Section 8.2. New melodic material in mm. 54-57 (shown in Fig. 8.7) lays the foundation for the rest of the section despite being in a different time signature. This material is vaguely in E♭lydian, but there is no functional harmony in this portion of the piece. The frequency half-steps and ambient reverb give Section C a slightly unsettled, mysterious feel.

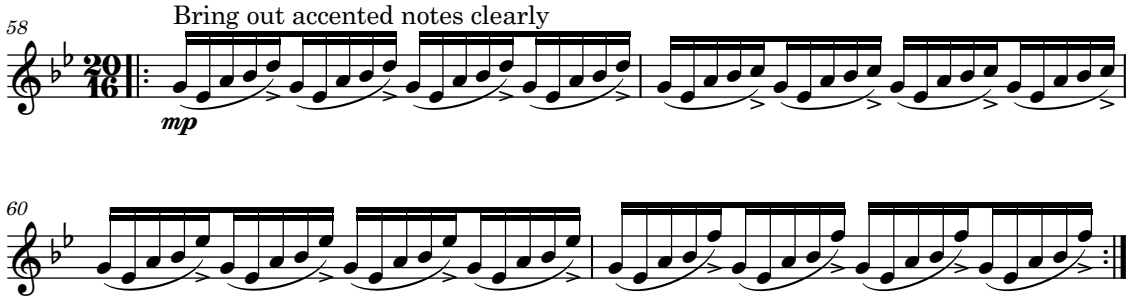


Figure 8.8: *Steelpanopticon* mm. 58-61 of Section C.

At m. 58, the time signature changes to 20/16 until the end of Section C. The 20/16 pattern is built off of the 4/4 pattern from mm. 54-55, but adds a fifth note to the repeating sequence which acts as a melody note. See mm. 58-61 in Fig. 8.8. This new pattern should be played such that the first four notes of the sequence are mezzo piano while the accented fifth note is forte.

I perform Section C with two mallets because it requires a high degree of control to control the balance of the notes. The five-note sequences require that the performer play doubles on a hand frequently. I chose to use a sticking of RLRLR for all of this

section. While this results in a few difficult jumps, it works well. The main concern in this regard is to make sure the first note of the sequence—the second of the double played by the right hand after it plays the accented fifth note—is not overplayed. One most tenor steelpans, the low G is a note that speaks well and is easy to overplay after playing an accented high note.

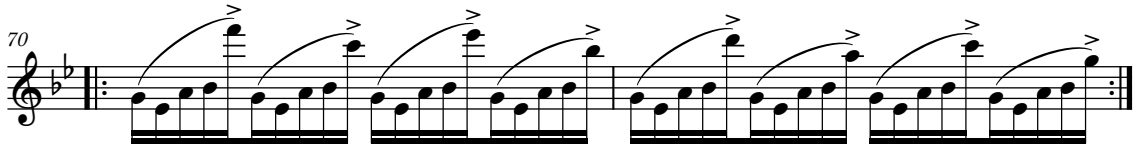


Figure 8.9: *Steelpanopticon* mm. 70-71 of Section C.

From m. 54 until m. 77, there are perpetual sixteenth notes. Several simple melodies emerge that slowly move up the steelpan's range and peak with the melody at mm. 70-71 (shown in Fig. 8.9). The melody note of m. 71 is an F6. Many tenor steelpans do not have this note. Only my custom-made tenor steelpan built by Kyle Dunleavy includes this note and that was by request. This melody moves back down the steelpan in a zig-zag motion: down a fourth, up a third, down a fourth, etc.

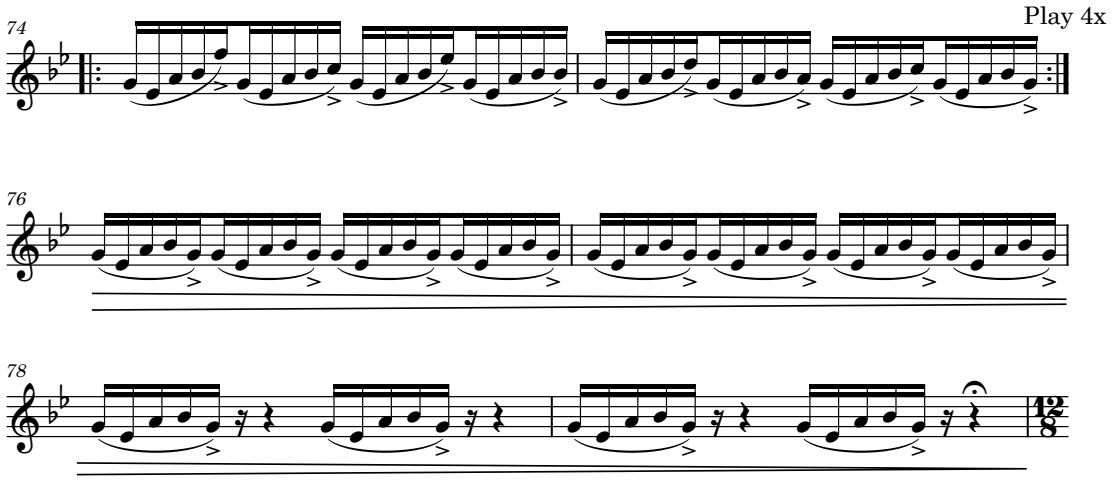


Figure 8.10: *Steelpanopticon* mm. 74-79

The melody from mm. 70-71 repeats transposed down an octave in mm. 74-75 (see Fig. 8.10). This transposed figure has interesting interactions with the repeated four-note pattern that precedes the melody notes. The last note of m. 74 is a repetition of the previous note of B♭3. The performer should strive to bring out stark

tonal difference between these notes so that the first B♭sounds like a background note and the second is clearly a melody note. This method of interpretation should continue until the end of the section so that every note that note below the B♭3 is heard as a melody note while the remainder are background notes. I accomplished this by beginning the diminuendo notated in m. 76 early. I began de-emphasizing the background notes in mm. 74-75 so that I could also play those melody notes slightly softer since it is easier to overplay them in this range on the steelpan.

This section then trails off with repetitions using a G3 as the "melody" note at mm. 76-79. The long ambient reverb tail gives the performer time to switch back to four mallets for Section D.

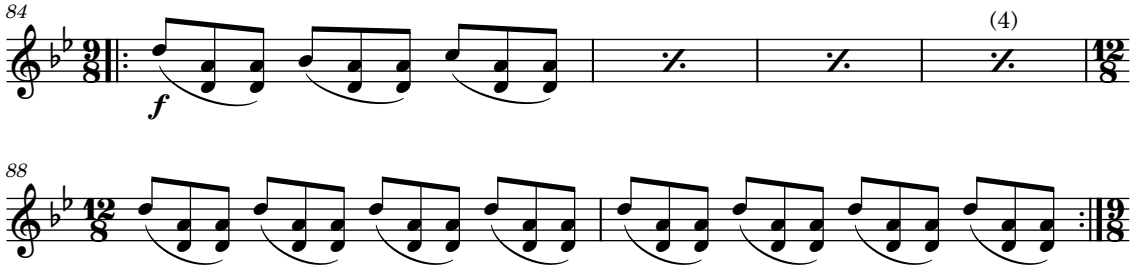


Figure 8.11: *Steelpanopticon* mm. 84-89 of Section D.

Section D requires four mallets again and features a different kind of virtuosity on the steelpan that Section C. Four mallet technique, while not unheard of, is not common on the steelpan, even in the solo literature. Section D is built from an eighth note triple figure with the left hand playing a melody note and the right have responding by striking a D5 chord twice. The player also returns to the fuzz distortion processing chain at this point.

After an introduction using only D5 notes for the melody in mm. 80-83, the first true melody of Section D is presented in mm. 84-89 (shown in Fig. 8.11). The phrasing of this section sets up several contrasts with earlier sections. First the time signature changes periodically from 9/8 to 12/8 (the "triplet" feel is derived from the time signature). Sections A and B are effectively based on eighth notes while Section C is based on groupings of five sixteenth notes. Here, the triplet feel is reinforced by a three note melody repeated across measures of 9/8 from m. 84 to m. 87. Then two measures of 12/8 are used to create space before repeating the pattern. This whole figure is repeated in mm. 90-95, but the melody notes are transposed up a fifth to A, F, and G.

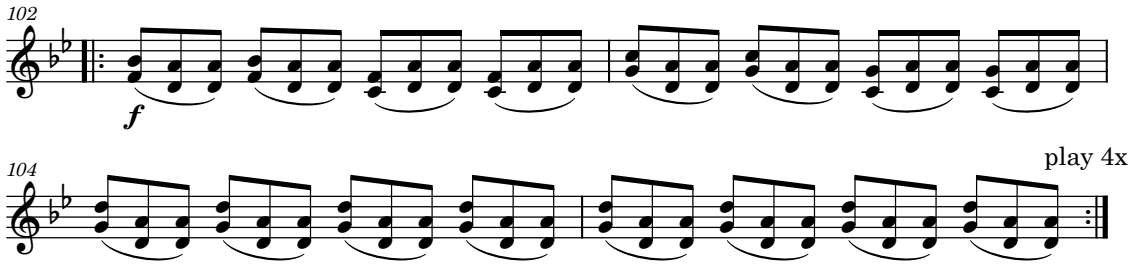


Figure 8.12: *Steelpanopticon* mm. 102-105 of Section D.

At m. 100, the left hand starts playing a G5 chord against the right hand's D5 chord as a segue to mm. 102-105 (shown in Fig. 8.12). In mm. 102-103, the left hand now hops around playing three different perfect fourth voicings before finishing with C5. This excerpt heightens the tension by featuring "melody" chords that are in the same range as the "background" D5 chords. Combined with the sustain provided by the Dynamic Plate reverb, this creates a smear-like aural texture. Section D ends by repeating m. 105 four more times with a diminuendo. The G5 in the left hand played against the D5 in the right creates a morphing Dsus texture as the section fades away.

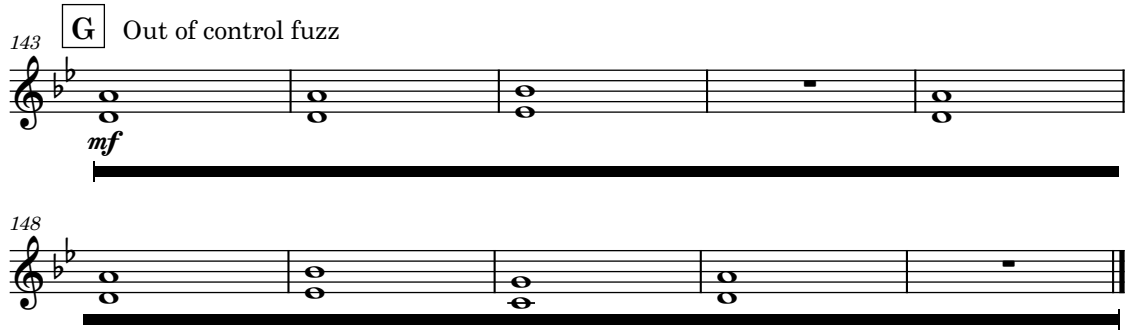


Figure 8.13: Final measures, mm. 143-152, of *Steelpanopticon* Section G.

The material from the beginning of the piece re-emerges at m. 111. Section E repeats the introduction (first 16 measures) while Section F is a repetition of Section A. Section G, shown in Fig. 8.13, ends the piece with new material based on the chords from Section A. At Section G, the Vermin Dist is activated on the Helix and the decay time of the Dynamic plate reverb is raised from 1.2 to 1.8 seconds. This has a profound effect on the tone. The sense of pitch is diminished as it transforms into a screeching distortion the sustains for significantly longer than before. Do not let the decay times of 1.2 and 1.8 seconds mislead you. Due to being processed through three

fuzz pedals, each of these final notes sustains for more than 5 seconds. Section G is not played in strict time at all. Each note is allowed to breath for as long as it needs to and the next one is struck as the sound begins to fade away. This desctruction of the steelpan's tone and timbre is meant to send the message that the steelpan can generate sounds that no one had of for it before.

Chapter 9

(De/Con)struction in Steel and Electricity

(De/Con)struction in Steel and Electricity was originally composed in 2014 for the 2014 James P. and Shirley J. O'Brien Endowment Composition Competition hosted by University of Arizona (UA) and won first prize. Originally scored for steelband and electronic percussion, it was premiered in a joint performance by the UA Steelband and CrossTalk, UA's electronic percussion ensemble. Later in 2014, it was performed and recorded by the Southern Oregon Univeristy Percussion Ensemble and released on their album, *Electric Rebel Poetry*. The original score is included in appendix A.3.1.

(De/Con)struction in Steel and Electricity is scored for a steelband and electronic percussion. The steelband follows the common instrumentation of lead pans, double tenors, double seconds, cellos, and bass pans. The electronic parts are labeled "Electric Guitar", "4-string Bass Guitar", "Synthesizer 1", "Synthesizer 2", and "Percussion". These parts are named to reflect the type of sound timbre the electronic part should have, but does not imply they should be played by those instruments or that the performer should use sample libraries of those instruments.

At its premiere, the electronic parts for *(De/Con)struction in Steel and Electricity* were performed by CrossTalk using used MIDI controllers with software synthesizers and samplers hosted in Ableton Live. The MIDI controllers included the DrumKAT, MalletKAT, and TrapKAT made by Alternate Mode as well as the Zendrum. Southern Oregon University used Alternative Mode controllers as well as a Roland HPD-15 HandSonic and Korg Wavedrum. The electronics parts were composed with these

instruments in mind, but can also be performed on different MIDI controllers, synthesizers, or other electronic instruments. The sound design is left to the performer's discretion and the instrument labels are soft suggestions of what kind of timbral space they should occupy.

The Drum Set part can be played either an acoustic or electronic instrument. Both University of Arizona and Southern Oregon University doubled the Drum Set part blending both acoustic and electronic drums. At both universities, the steelbands and electronic ensembles often rehearsed separately, but both ensembles need someone performing the Drum Set part for rhythmic coherence. When the two ensembles rehearsed or performed together, they had both drummers play together.

9.1 Analysis

The main melodic themes of *(De/Con)struction in Steel and Electricity* are derived from a four-note pattern borrowed from the song *H.* by the band Tool. Shortly after the halfway point of *H.*, the guitarist starts playing an ostinato as in Fig. 9.1. This simple pattern forms the basis for most of the melodic and harmonic content of *(De/Con)struction in Steel and Electricity* – excepting the solo section in the middle of the piece.

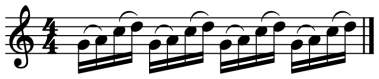


Figure 9.1: Guitar ostinato from *H.* by Tool.

(De/Con)struction in Steel and Electricity begins with the Bass Guitar playing the four-note pattern (hereafter referred to as the main theme) from Fig. 9.1 slowed down considerably to one note per beat (at a tempo of ♩ = 80 beats per minute) as shown in Fig. 9.2. This bass ostinato continues until m. 64. In mm. 5-8, the Electric Guitar part (shown in the top line of Fig. 9.2) plays four four-note figures that are permutations of the main theme. The purpose of featuring the Bass Guitar ostinato and the Electric Guitar at the beginning of the piece is to signify the importance of these themes throughout the piece. Much of the rest of the piece features similar permutations, or rearrangements, of the theme. Sometimes the permutations are transposed – usually up a musical fourth. The permutations of the notes of the main theme and their transpositions are shown in Table 9.1. Variations of these

permutations of the main theme appear repeatedly throughout the first, second, and fourth sections of the piece.

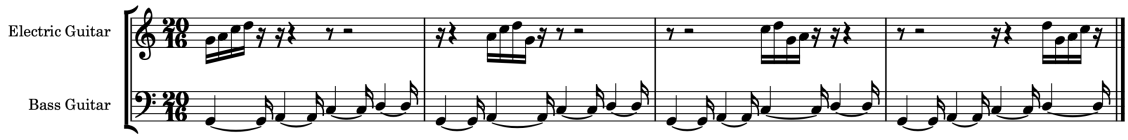


Figure 9.2: Theme permutations and bass ostinato from mm. 5-8 of *(De/Con)struction*.

Root	Perm. 1	Perm. 2	Perm. 3
G-A-C-D	A-C-D-G	C-D-G-A	D-G-A-C
C-D-F-G	D-F-G-C	F-G-C-D	G-C-D-F

Table 9.1: Permutations of the main theme at the original pitches and transposed up a fourth.

In the excerpt in Fig. 9.2, you can also see that the time signature at the beginning of *(De/Con)struction in Steel and Electricity* is 20/16. This time signature is in effect for the first half of the piece, from the beginning until m. 64. This time signature should be interpreted as having four "beats" per measure with each beat consisting of five sixteenth notes. The sixteenth notes act as quintuplet subdivisions of the beats. The notes are generally beamed in groups of five sixteenth notes to support this interpretation. The 20/16 time signature was used to avoid the need to label every beat with a quintuplet much like how a 12/8 time signature is often used to imply four beats per measure with triplet subdivisions instead of labeling every beat of 4/4 with triplets. It may not be readily apparent due to the way the rests are displayed, but in Fig. 9.2, each of the Electric Guitar's four-note figures starts on a downbeat.

In general, there are two rhythmic motifs for the 20/16 portion of the piece. The first is based on the Electric Guitar part in Fig. 9.2. The second motif constructs a rhythmic skeleton from two eighth notes and a sixteenth note per beat as in Fig. 9.3.

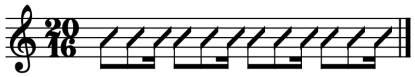


Figure 9.3: Rhythmic base for the first section of *(De/Con)struction*.

In m. 9, a sequence of chords also derived from the main theme is introduced that continues until Section D. The chord sequence played by Synthesizer 1 is (B♭, C, Em, F), but the intended underlying harmonic sequence is actually (Gm7, Am7, Cmaj7, Dm7). The root notes of this sequence are the notes from the main theme. Since the bass ostinato includes all of these notes in every measure, I decided to let the roots notes of these chords be implied by the Bass Guitar and Bass Steelpan parts while Synthesizer 1 only plays the upper three voices of the chords. This leads to interesting interactions between the bass ostinato and obscures slightly the origin of the harmonic sequence.

Section A culminates with the rhythmic pattern from Fig. 9.3 combined with the four-note melodic theme. The tension between the repeating three note rhythmic pattern and four-note melodic theme heightens the drama at this point. In mm. 25-28 (condensed to a single measure in Fig. 9.4), the Double Tenor Steelpans play this pattern with the notes of the main theme while the Tenor Steelpans play it transposed up a fourth. The Electric Guitar doubles the Double Tenor Steelpans and the Percussion part plays a rhythmic version on (unpitched) electronic toms. For mm. 27-28, every melodic instrument is playing this pattern except for the two synthesizer parts which are still providing harmony. This moment culminates with every instrument performing a crescendo and then suddenly stopping except for the bass ostinato which emerges from the decaying texture of the other instruments.

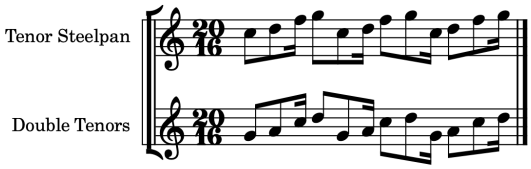


Figure 9.4: Melodic pattern at the culmination of section A of *(De/Con)struction*.

Sections B and C are also composed using variations of the four-note theme, but also introduce more involved melodies beginning at m. 37. The harmonic sequence of (Gm7, Am7, Cmaj7, Dm7) from Section A continues and provides the basis for the new melody. The first two measures of the repeating melody are based on the Am pentatonic scale. The melody then repeats transposed up a musical fourth to the Dm pentatonic scale. In m. 41, the Double Second Steelpans enter with a counter-melody. The melody and counter-melody are shown in Fig. 9.5.

Variations of this four measure melodic pattern repeat until m. 52. The Double

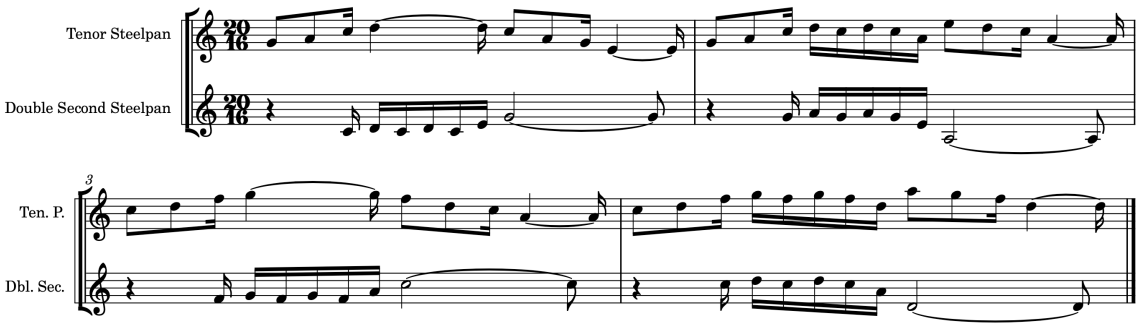


Figure 9.5: Main melody and counter-melody for Sections B and C.

Tenor Steelpans enter and double the Tenor Steelpan melody at m. 41. At m. 45, the Tenor Steelpans then transpose the four-measure pattern up a musical fourth so that they play the same melody, but based on the Dm and Gm pentatonic scales. This creates an interesting tension since the third and fourth measures of the melody feature the note B♭ as a passing tone while the harmony of the third measure is based on a Cmaj7 which features the note B♮. This tension is allowed to persist without resolution until Section D at m. 53.

Underneath the melody in mm. 41-52 is an interesting textural interjection from Synthesizer 2. For mm. 41-44, permutations of the main theme are played on the last four sixteenth notes of each measures (See the first four measures of Fig. 9.6). Since the piece should be interpreted as having four beats per measure at this point, this puts these phrases on the second through fifth subdivisions of beat four. This mirrors the rhythms of the percussion interjections that already appeared in mm. 37-40. The first interjection starts with the theme rotated once to become (A-C-D-G). The theme is similarly rotated three more times over the next three measures, ending with the original theme.

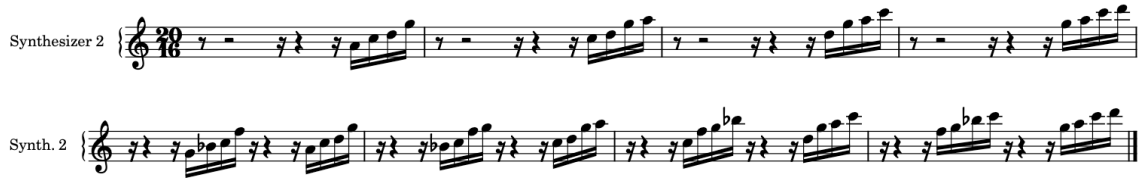


Figure 9.6: Synthesizer 2 interjections from mm. 41-48.

These interjections are further heightened in mm. 45-48 (which is repeated in mm. 49-52). In mm. 41-44, the melody consists of a sustained note on beat four of each measure. There is no rhythmic or melodic competition between the two lines.

Starting in m. 45, each measure's interjection from beat four is copied to beat two and transposed down a whole step. The new notes of the transposed theme are (F-G-B♭-C). This has a few effects due to how it interacts with melodic lines of the Steelpans. The melodic content and rhythmic phrasing clash with the counter-melody of the Double Second Steelpan which reinforces the interjective and textural nature of the Synthesizer 2 lines. A second interesting effect is that this transposition featuring B♭ reinforces the presence of B♭s in the last two measures of the melodic phrase against a harmony of Cmaj7 as discussed above. This contrasts between these interjections, the melodies, and the harmony brings the tension to a peak in mm. 49-52.

However, the melodic and harmonic tension of Section C gives way to new rhythmic tension in Section D starting at m. 53. The time signature begins to transition from 20/16 to 4/4 at this point. However, not all of the instruments switch simultaneously with patterns suggesting 20/16 (at ♩ + 𝅘𝅥𝅯 = 80 bpm), 5/4 (suggesting five beats per measure with ♩ = 100 bpm), and 4/4 (also at ♩ = 100 bpm) occurring simultaneously. As m. 65 approaches, the instruments playing 20/16 patterns gradually fade out with the bass ostinato that starts the piece being the last 20/16 pattern to disappear. At m. 65 (Section E), the time signature changes to 4/4, maintaining the sixtheenth note, and the drumset establishes a new funk groove for this portion of the piece.

The final holdovers from the previous section are Synthesizer 2 and Percussion playing the main them as sixteenth notes. They fade out at m. 68 and make room for the new bass ostinato to enter at m. 69 shown in Fig. 9.7. This fully establishes the basis for the new groove and thematic material.

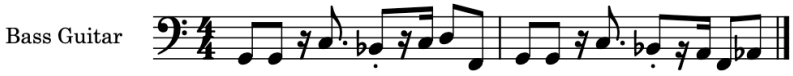


Figure 9.7: Bass ostinato for mm. 69-104.

Section F (mm. 77-96) is designed to feature a soloist while the other instruments play slowly shift background lines. The tonality is primarily G♭ minor, but the background lines feature different chromatic excursions to create a sense of tension and release over the ostinato. At mm. 85-88, the Double Second and Cello Steelpans (doubled by Synthesizer 1 and Synthesizer 2) play moving triad patterns that end with out of key tonalities on beat 4 of each measure. See Fig. 9.8. In each measure, the first three triads form B♭, Cm, and Dm chords. In the first and third measures

of the pattern, the terminal chord is Em where E♭ would be the next chord in the sequence from the key of G minor. This chord is non-functional and does not relate to the bass line either which plays the note F underneath the chord. In the second measure of the pattern, the Em is modulated down a half-step to E♭m. This chord features the note G♭ and is used to suggest a transitory change to the key of G♭major (from which the E♭m chord is borrowed). When played as part of a D or D7 chord and written as F♯, this note would be used as a leading tone to return to the root, G. However, in this situation, it is once again being used in a non-functional way to create a sense of tension. In the fourth measure of this pattern, the Double Second Steelpan plays a line that once again suggests a temporary switch to the key of G♭ major.

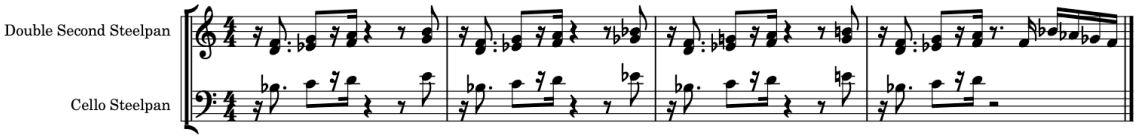


Figure 9.8: Background lines at mm. 85-88.

The tension reaches its highest in Section F at mm. 93-96. A rhythmically modified version of the background lines from mm. 85-88 are combined with the background lines from mm. 89-92 as shown in Fig. 9.9. The Double Tenor part is the same as in mm. 89-92 except that the last beat of the fourth measure is eliminated to focus attention on the Double Second part. The Double Tenor isolated (as it is in mm. 89-92) reinforces the underlying G minor tonality of the bass ostinato. The fourth beats of the first and third measures are the same and when combined with the bass line suggest a change to an F chord as a contrast to highlight the return to the tonic note of G at the beginning of the next measure. Beat four of the second measure (which is the same as beat four in m. 92), reinforces the chromatism of the bass line which uses an A♭ as a chromatic passing tone, but the Double Tenor moves chromatically from C to B♮ and then returns to B♭.

When combined with the Double Second and Cello parts as shown in Fig. 9.9, the Double Tenor part takes on a different character due to the interplay between the two parts. The third triad in all four measures of the Double Second and Cello parts is delayed by one sixteenth note to accentuate the interplay. The first two chords of the Double Second and Cello parts align with notes in the Double Tenor part since the notes of the Double Tenor melody at those times are chord tones from

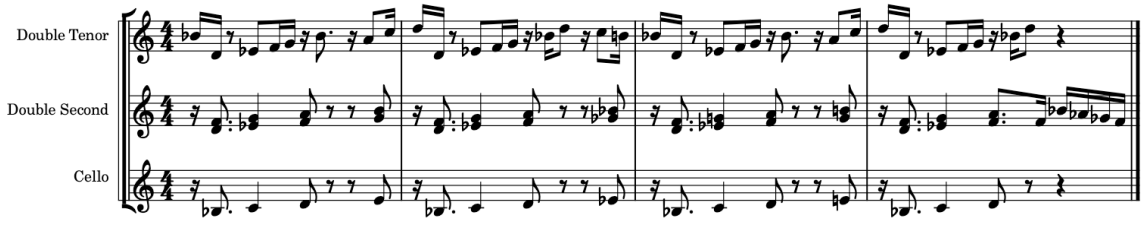


Figure 9.9: Background lines at mm. 93-96.

the Double Second and Cello parts. The third and fourth chords of those patterns, however, fall in the rhythmic holes of the Double Tenor melody since the melody notes surrounding those chords do not match the chords. This alternation creates an interesting texture and heightens the tension from the previous background figures. The Double Second and Cello parts featuring tonalities from outside of G minor at the end of each measure, but the Double Tenor melody not doing so reinforces that tension and provides the soloist with the most harmonically interesting landscape upon which to improvise. This is the end of the solo section and in m. 97 the piece transitions to new material that is once again based on the main four-note theme.

In Section G, the density of the piece reduces greatly to the Bass Steelpan and Bass Guitar continuing the bass ostinato accompanied by the Drumset. As interjections over this, the Electric Guitar plays the main four-note theme with a sixteenth note rhythm on beat four for four measures to foreshadow the re-emergence of the theme as the primary melodic material for the remainder of the piece. Starting in m. 101, the Tenor Steelpan, Double Tenor Steelpan, and Electric Guitar enter playing non-stop repetitions of the main theme. The slowly grow from pianissimo to mezzo forte over the course of four measures. The Percussion part doubles the pattern, but on unpitched drums. The Drumset also transitions from the funk pattern for the solo section to "four on the floor" pattern that emphasizes the pulse. The Bass Guitar plays the ostinato for the last time at m. 104. Section H begins at m. 105 and the four-note theme has by now prominently reappeared in various transformations being played by the Tenor, Double Tenor, Cello, and Bass Steelpans as well as Synthesizers 1 and 2.

Rehearsal mark I marks the true transition to the final of the piece. In this section, every melodic instrument except for the Bass Guitar and Bass Steelpan is playing variations on the main theme. At this point the Bass Guitar and Bass Steelpan play a 16-measure pattern twice and take over as the primary "melody" until Section J

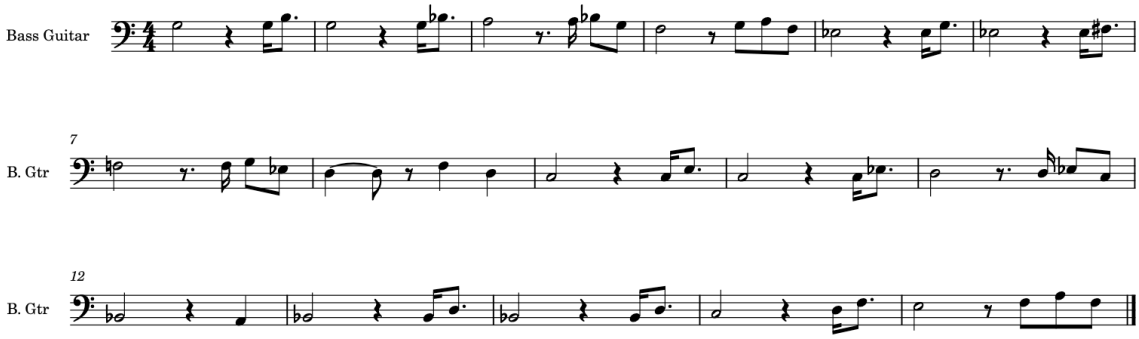


Figure 9.10: Bass melody at mm. 113-144.

(see Fig. 9.10). For mm. 113-120, the Tenor and Double Tenor Steelpans alternate the pattern with Synthesizers 1 and 2 as in Fig. 9.11. In the excerpt of the score, you see that the Synthesizer parts are labeled with 'R's and 'L's. This is meant to convey hard panning information for these notes. The intention is to create a spatial effect between the Steelpans who are either performing live onstage or panned center and the Synthesizers. At m. 117, the panning instructions for Synthesizer 2 invert so that it is always panned opposite Synthesizer 1 to create a thicker orchestration.

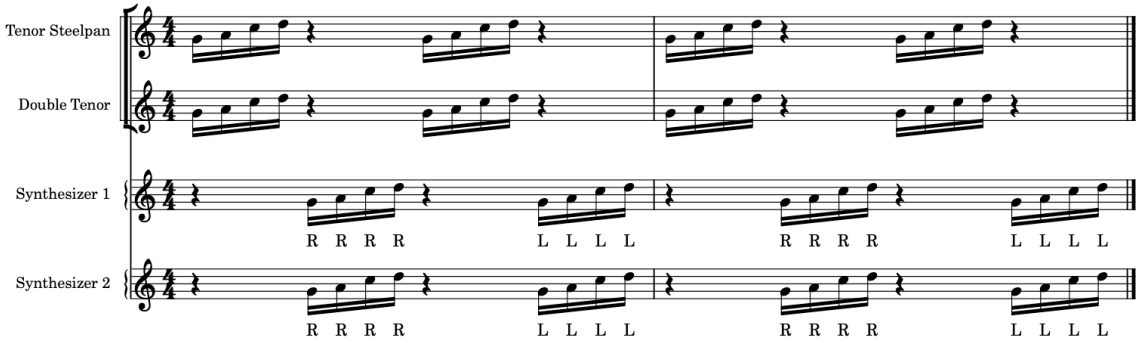


Figure 9.11: Tenor and Double Tenor Steelpans alternating with Synthesizers 1 and 2 at mm. 113-120.

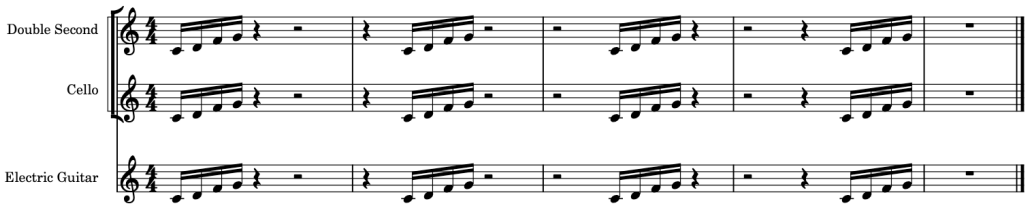


Figure 9.12: Transposed theme recurring every five beats.

Simultaneously, the Double Second Steelpan, Cello Steelpan, and Electric Guitar interject with the main theme transposed to (C-D-F-G) every five beats (see Fig. 9.12). This pattern continues for these instruments from m. 113 until the last phrase of the piece starts at m. 148. The repeating pattern is five beats long as an allusion to the 20/16 patterns at the beginning of *(De/Con)struction*. In this situation it creates a pattern that although regularly repeating a familiar theme, still creates tension since its occurrences do not align with the beginnings of other phrases.

At m. 121, halfway through Section I, the Tenor and Double Tenor Steelpans and Synthesizers 1 and 2 begin passing around the theme in a new way. In a pattern repeating every measure for four measure, the Tenor Steelpan plays the theme on beat 1, the Double Tenor on beat 2, Synthesizer 1 on beat 3, and Synthesizer 2 on beat 4. The Tenor Steelpan and Synthesizer 1 transpose the theme up a fourth. A waterfall-like effect begins to emerge and is strongly reinforced at mm. 125-8 where the pattern continues but each instrument's notes are inverted. This also further differentiates this pattern from the five beat pattern of the other Steelpans and Electric Guitar. See Fig. 9.13 for both patterns.

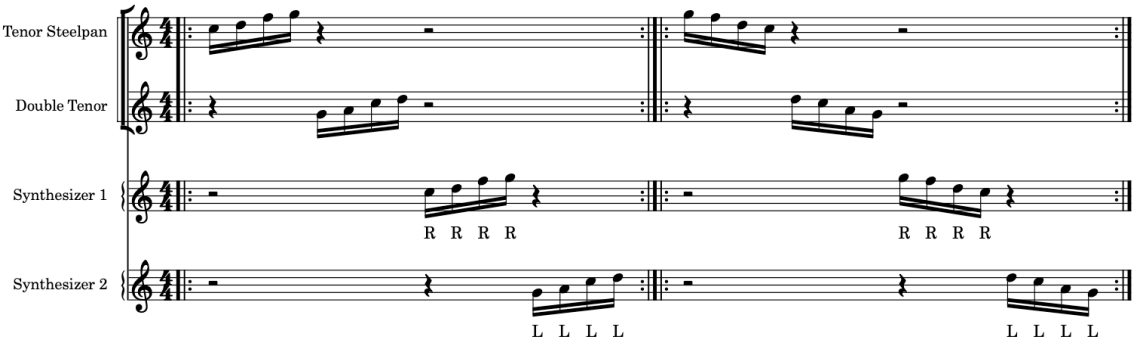


Figure 9.13: Theme passing between voices.

At Section J, the Tenor Steelpan begins playing a melody reminiscent of the melody from mm. 37-40, but adapted for the 4/4 time signature. Fig. 9.14 shows that the notes in the first and third measures of both melodies are the same. Due to the shift from a time signature of 20/16 to 4/4, it was necessary to adjust the rhythms to make the melody flow better. This changes where the point of emphasis are in the melody creating both a sense of recall and novelty. The second and fourth measures of the 4/4 melody diverge more from the original. Where they diverge, they adjust the melody line to make the notes more dramatic. In the second measure of the 20/16 melody, the high point on beat three is an E, but the analogous note in

the 4/4 melody then moves up to a G. Similarly in the fourth measure, the 20/16 melody peaks at a high A, but the 4/4 melody uses that a jumping off point to hit a high D. This also does not quite follow the pattern from the second measure of the 4/4 melody giving it a sense of spontaneity in a piece that uses a lot of precise, straightforward transformations on simple elements.

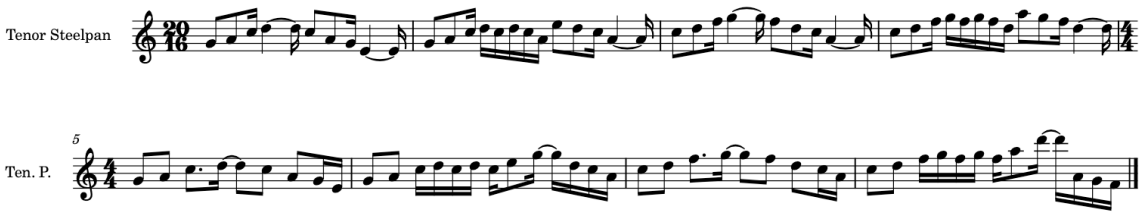


Figure 9.14: A comparison of the main melody from the beginning and end of *(De/Con)struction*.

Underneath the melody, the Double Tenor and Synthesizers 1 and 2 play the main theme in a round robin. The first four measures of Fig. 9.15, show this pattern. After four measures, the Double Tenor begins doubling the melody from the Tenor Steelpan. Synthesizer 1 and 2 then alternate playing the main theme for four measures. Four measures later Synthesizer 1 also begins playing the new melody and Synthesizer 2 plays the main theme repeatedly for four measures before switching to the new melody.

Over the course of Sections I and J, the main theme goes from being passed between four instruments to three to two and finally to one. As the number of instruments passing the main theme diminish, they switch over to doubling the new melodic line. This creates a "crossfade"-like effect between repetitions of the main theme and the new melody without any instruments performing a crescendo or diminuendo.

The final phrase of the piece, shown in Fig. 9.16 begins at m. 145. The Double Second Steelpans, Cello Steelpans, and Electric Guitar continue their repetitions of the main thematic notes until m. 148. Every other melodic instrument plays a transformation of the main theme for three bars leading based on the transformation that Synthesizer 2 plays at m. 49. This transformation rotates the theme by one note to (A-C-D-G) and then transposes down by a whole step to (G-B♭-C-F). At m. 148, all of the melodic instruments except for the Cello Steelpans, Bass Steelpans, and Bass Guitar play through all four permutations of the main theme – some at the original pitch and some transposed up a fourth. The Cello steelpan plays the transposed theme as eighth notes followed by the un-transposed theme. At m. 148,

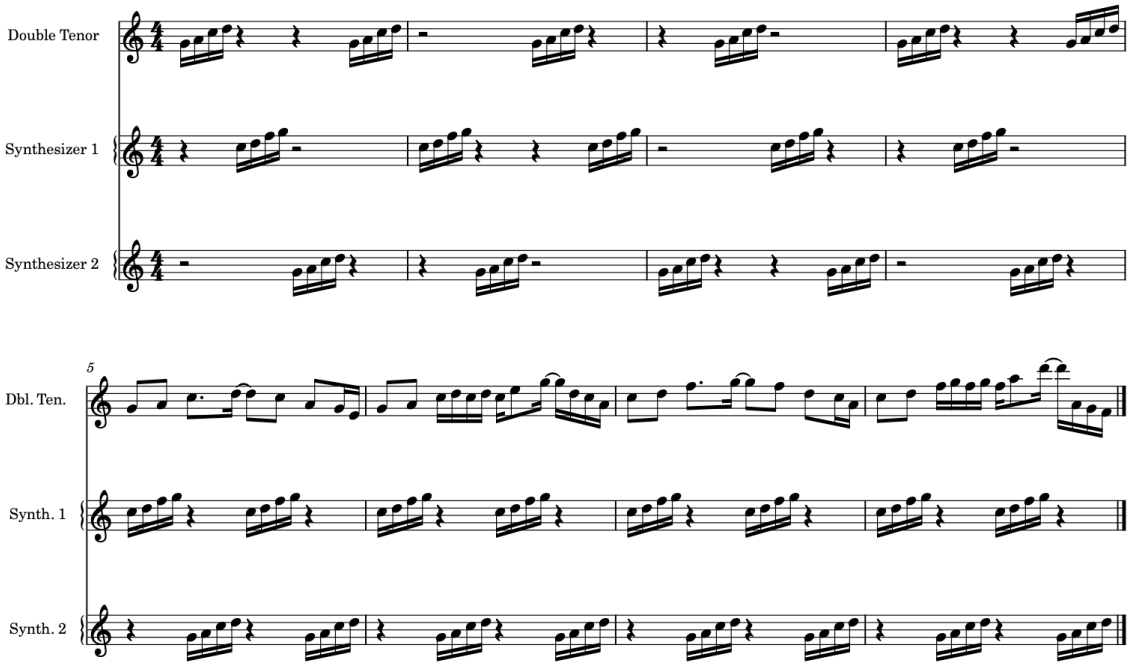


Figure 9.15: The Double Tenor and Synthesizers 1 and 2 passing the main theme in Section J.

all instruments except for the Bass Guitar and Bass Steelpans have a subito piano and then dramatically crescendo across that measure to the final note at m. 149. The Bass Steelpans and Bass Guitar play a variation on the bass line from the previous three measures designed to punctuate the end of the piece as all pitched instruments end with a G5 chord.

9.2 Solo Version

From 2014 until 2022, *(De/Con)struction in Steel and Electricity* received no performances. The instrumentation of steelband and electronic instruments is esoteric and not readily available to most schools and ensembles. In anticipation of this, the electronic parts were also composed so that they could be recorded and made into a backing track that steelbands could play along with. Since even this arrangement had difficulty in finding performances, I decided to make a version for solo steelpan and backing track. This new solo version was premiered at my PhD recital on March 26, 2022 in Phillip T. Young Recital Hall at University of Victoria [70]. The sheet music and a video recording of the premiere of the new solo version are included in

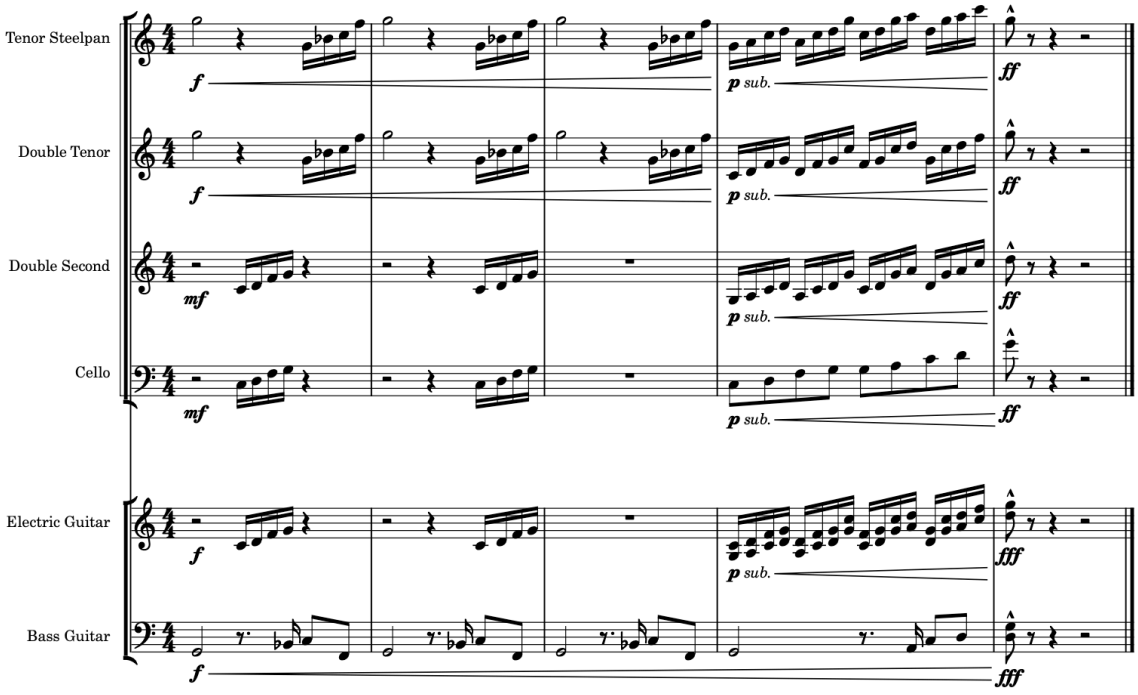


Figure 9.16: Condensed ending of *(De/Con)struction in Steel and Electricity* .

Appendix A.3.2.

While the electronic parts were already composed to work as a backing track that performers could easily follow, the Tenor Steelpan part by itself would not work well as a solo part. One reason for this is that the piece is generally composed in four-measure groupings where a new idea is layered upon previously introduced ideas. Thus when an instrument starts playing a line, it generally repeats itself many times while another instrument makes the next change. This works well in an ensemble setting, but is not ideal for a solo performer. To address this issue, the new solo part was constructed as an amalgamation of other parts instrument parts from the original score in order to maintain focus on the soloist.

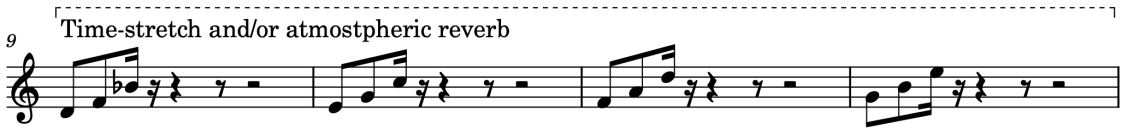


Figure 9.17: Mm. 9-12 from the solo part.

In some cases new edits were made with instructions to use new effects that were not available when I originally wrote the piece. The first instance of this occurs in

mm. 5-8. The original version of *(De/Con)struction in Steel and Electricity* included instructions for the electronic parts to pan various phrases to only the left or right speakers. When designing the solo version, I decided to use quadraphonic playback for the electronic parts and to use 3D sound effects where possible. At mm. 5-8, the Solo Steelpan part is derived from the original Electric Guitar part which included instructions to alternate the panning of each phrase and to use "spacey sound with echo and/or reverb". The new instruction in the Solo Steelpan part is to instead use "quad ping pong delay". This is a Max for Live device I built for my other composition, *Cycles*, that routes the echoes around the audience in a circular fashion.

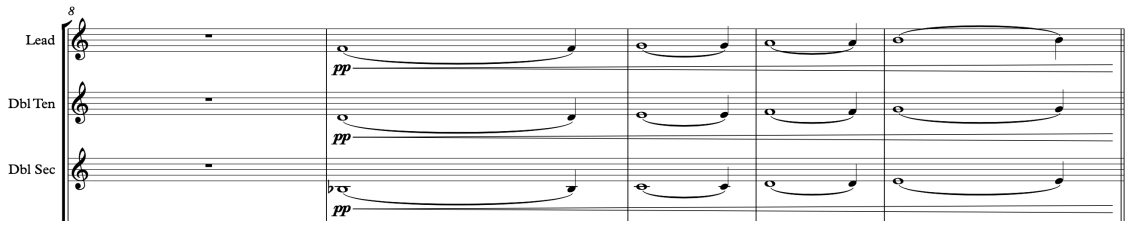


Figure 9.18: Mm. 8-12 from the original score.

Another adaption can be seen in mm. 9-12 (see Fig. 9.17). The solo part consists of arpeggiations of the harmonic sequence in first inversion while in the original score, the notes are played in rhythmic unison and the chords are in root position as in Fig. 9.18. The reasons for this change are that the root of the first chord (B♭) is out of the tenor steelpan's range (its lowest note is a C), playing three notes simultaneously requires using four mallets, and instead of rolling the notes I chose to create sustain by using the Realstretch effect. This creates a more interesting, haunting texture for the arpeggios than rolling chords that are not practical for a solo performer. The settings I desired to use with Realstretch were too demanind and caused audio glitches, however. For performance, I use a pre-rendered track with the stretched audio which alleviates the latency and computational costs of using Realstretch in performance.

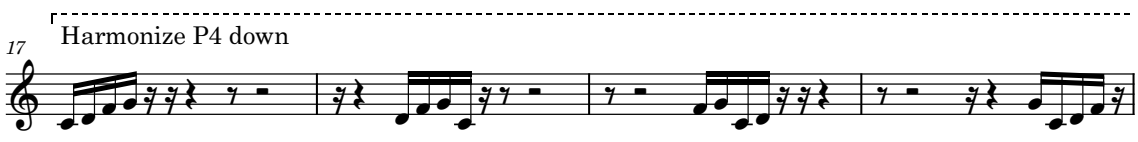


Figure 9.19: Mm. 17-20 from the Solo Steelpan part.

At mm. 17-20, the Solo Steelpan plays the upper voice of the Synthesizer 2 part

as in Fig. 9.19. The lower voice is the same phrase, transposed down a fourth. To account for this, a harmonizer effect on the Line 6 Helix is used to add a steelpan voice transposed down a fourth. The harmonization effect continues until m. 28. For mm. 21-28, the Double Tenor Steelpan doubles the Tenor Steelpan down a fourth. For the rest of the passage, the Solo Steelpan is able to perform both parts with the harmonizer. In this way, the steelpan is able to overcome its range limitation through the use of technology.

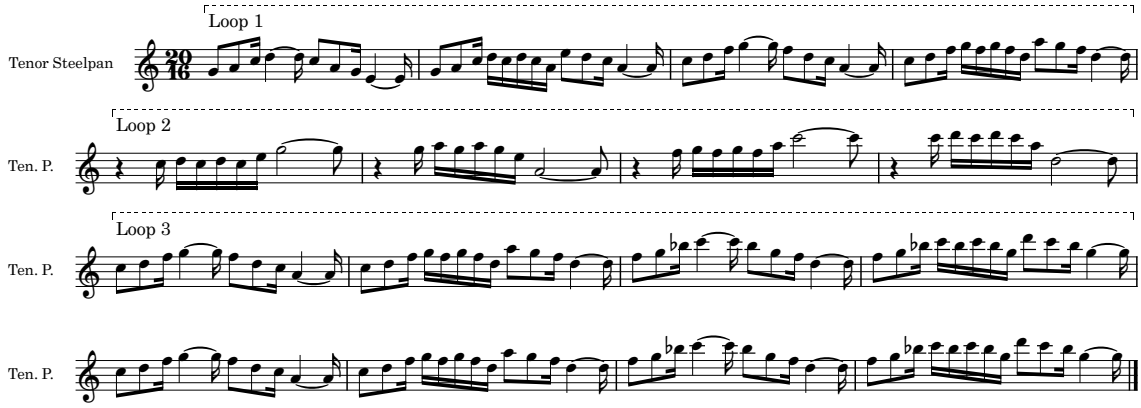


Figure 9.20: Mm. 37-52 of the Solo Steelpan part showing the beginnings and ends of the loops.

From mm. 37-48 (shown in Fig. 9.20), a four-measure looper is used to recreate the layered entrances in the original score. The looper is automated in Ableton Live so that it is aligned with the backing track and does not need to be triggered by the performer. It first records a loop of mm. 37-40. Then during mm. 41-44 it plays back the first loop while overdubbing the new material. This process repeats for mm. 45-48 – loops 1 and 2 are played back while loop 3 is recorded from the live audio. By the time we reach m. 49, the original melody (loop 1), the counter melody (loop 2), and the transposed melody (loop 3) are all playing back alongside the performer as they repeat the transposed melody. This serves to create harmonizations of the musical lines as well as thicken the texture of the piece as we reach the peak of this section of *(De/Con)struction in Steel and Electricity*.

At m. 53, the Solo Steelpan once again uses a harmonization effect. This time the steelpan is harmonized down by an octave. In the original score, the Double Tenor Steelpan doubles the Tenor Steelpan down an octave. The Solo Steelpan then proceeds to fade out over mm. 57-68 as it copies the first the Electric Guitar and then Synthesizer 2. At Section F, the Solo Steelpan improvises and/or plays the backing

lines until arriving at Section G.

At this point, the Solo Steelpan is playing the original Electric Guitar part. In the original score, that part has instructions to alternate between panning each phrase right and left. In this version, the amplified Solo Steelpan signal has instructions to be sent to the front-right, then front-left, then back-right, and finally back-left before returning to a front-center spatialization. This is all accomplished inside Ableton Live.

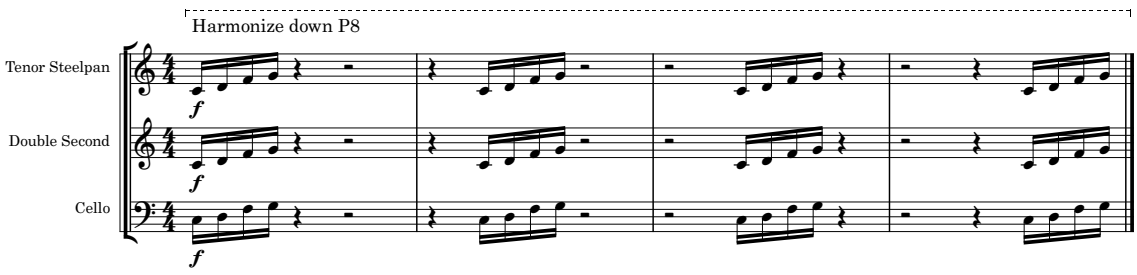


Figure 9.21: Solo Steelpan, Double Second Steelpan, and Cello Steelpan parts at mm. 109-112.

At mm. 109-112, the Solo Steelpan plays the Double Second Steelpan part. In the original score, the Cello Steelpan doubles the Double Second Steelpan down an octave. Once again, the Solo Steelpan uses a harmonizer to pitch shift down an octave to cover the Cello Steelpan part. At Section I, the Solo Steelpan part becomes a combination of the original Tenor Steelpan and Double Tenor Steelpan parts. At mm. 137-149, the Solo Steelpan part suggests to optionally harmonize the part down a perfect fourth, but I did not do this in the performance after all.

9.2.1 Helix Signal Chain

Since the steelpan parts for *(De/Con)struction in Steel and Electricity* were originally intended to be performed live on acoustic steelpans with no processing, the processing happening on the Helix is relatively minimal. At the front of the signal chain is a compressor with a fast attack to control the steelpan's transients as well as boost the sustained portions of notes. At the time of my PhD recital, I used the LA Studio Comp effect in the Helix, but have since switched to using the 3-Band Comp for its greater flexibility. That way, the compression can be dialed in for the low, mid, and high portions of the steelpan's tone individually. After the compressor, the 10-Band Graphic EQ is used to tame the high-mid frequencies a bit.

The next portion of the signal chain is a subtle chorus effect (the Ampeg Liquifier on the Helix) going into the Minotaur distortion. The Minotaur is a model of the famous Klon Centaur pedal. The gain parameter of the Minotaur is mapped to an expression pedal and kept low for the majority of the piece. During the improvisation at Section F, the gain is raised to about 50% to give the steelpan a little more boost and bite to its tone.

This then goes into US Deluxe Nrm amplifier model. This is a model of the normal channel of the Fender Deluxe Reverb amplifier. The drive on the amplifier model is kept at a middle value so that the amplifier model adds a bit of tone, but does not cause the signal to distort.

All of this is then sent to the Dynamic Plate reverb plugin. The reverb decay was set to 1.4 seconds with a mix setting of 30%. The other settings on the reverb were kept modest so that it would just a bit of ambience to the steelpan. The performance of this piece took place in a recital hall that contributes its own reverb as well. The output of the reverb was sent to inputs 1 and 2 in Ableton Live.

The second processing chain of the Helix consisted of only a compressor and the Dynamic Hall reverb. The settings for this reverb are a bit stronger. The decay was set to 3.0 seconds with a 60% wet mix. This was sent to inputs 3 and 4 in Ableton Live.

9.3 Ableton Live Session

The new backing track for *(De/Con)struction in Steel and Electricity* was generated in Ableton Live. The steelband and electronic instrument parts were made using sample libraries, software synthesizers, and other effects on MIDI tracks. There is also some live processing happening in Ableton during the performance.

The Live Session takes two stereo channels as input. The main steelpan signal from the Helix is routed to all four speakers with a 3 dB drop in level to the front speakers and 11 dB drop to the rear speakers. The secondary input signal is routed only to the rear speakers with an 11 dB drop in gain. The secondary input signal consists primarily of reverb and is meant to contrast with the primary signal and add some ambience.

There are two effects being applied to the main live input signal: dynamic equalization and quad ping pong delay. In order to control the brightness of the steelpan signal, the iZotope Neutron 3 Equalizer is used. It is set to dynamically reduce the

gain by 3.9 dB at about 1.6 kHz with a peak filter. A dynamic high shelving filter is also active with -2.6 dB of gain with a cutoff frequency of 3.5 kHz. The equalization is active throughout the piece.

The other effect applied to the live signal is a quad ping pong delay. This effect is active at mm. 5-12 and 17-20. The delay time is synced to a length of five sixteenth notes. This is also a spatialized delay effect. The first echo of an input sound is routed to the front left speaker, the second echo to the front right, the third to the back right, and the fourth to the back left. Subsequent echoes continue in this clockwise fashion around the audience.

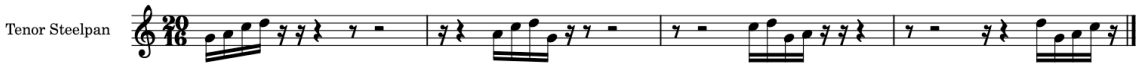


Figure 9.22: Mm. 5-8 of the solo Tenor Steelpan part to be processed with quad ping pong delay.

The spatialization of this effect pairs well with the melodic material in mm. 5-8 and 17-20. Fig. 9.22 shows mm. 5-8. The melodic material at mm 17-20 is the same except for being transposed down a perfect fifth. Since the delay time is synced to five sixteenth notes (equivalent to one beat in the 20/16 time signature), the echos for the first burst of notes will rotate through the speakers in a clockwise fashion start in the front left speaker on beat two. This means that the echos will return to the front left speaker on beat two of the next measure – at the same that the next series of notes are played. The echoes of the first phrase will progress to the front right speaker while the first echoes of the second phrase emerge from the front left speaker. As the two sets of echoes circle the audience, they will always be in neighboring speakers. This pattern will continue for the third and fourth sets of notes as well. If the feedback setting is high enough such that the echoes of the first set of notes persist, the four phrases will now circle the audience with each phrase routed to a different speaker. Two other instances of the quad ping-pong delay are used in this version of the piece, but are only used in the back track and not applied to the live steelpan signal.

Chapter 10

Conclusions and Future Work

In this document, I have presented a complete interdisciplinary sequence of research for the electroacoustic steelpan. This sequence started with acoustical research. The acoustical research was then used to inform the computational and musical research.

In Chapter 2, I showed how audio feature extract can be used to improve our understanding of the mallet-instrument system of the steelpan. I performed this analysis on a tenor steelpan using five different mallets to show that this methodology gives insight into how to make objective comparisons between the timbres of different mallets. The five mallet types included two rubber-tipped mallets, a chopstick, a cardboard tube, and a dowel rod bundle mallet. As expected, the two rubber-tipped mallets had very similar results and the two wooden mallets also had similar results. The cardboard tube sometimes correlated with the rubber-tipped mallets, other times with the wooden mallets, and periodically was on its own.

These results, while expected, still only tell a part of the story and should be extended. This study was focused exclusively on the tenor steelpan. Not only that, but it also only considers the low C tenor steelpan—there are many other variations of just the tenor steelpan. The reason for this is that that is the instrument which I have been performing on and have the most access to. The first step in extending the knowledge presented here would be to apply the acoustical studies to other types of steelpans. Notes on lower voiced steelpans have dramatically different timbral properties than the tenor steelpan. Lower pitched notes require different mallet designs so the results from Chapter 2 will not necessarily extend downwards to other instruments.

SASS-E, the Steelpan Analysis Sample Set for Evaluation, and Steelpan-Pitch, a steelpan-specific pitch detection system were presented in Chapter 3. SASS-E is a

new steelpan audio dataset featuring over 13,000 audio samples from three different tenor steelpans. Steelpan-Pitch applied SASS-E in the development of a new machine learning-based pitch detection system for steelpan audio. This system improved on pre-existing systems by lowering the minimum latency and improving accuracy. Steelpan-Pitch outperformed both PYIN and CREPE by over 20 percentage points and still performed well on novel steelpan audio.

In terms of future work, the next step would be to add samples of these instruments to the SASS-E dataset. This data would then be available to other researchers and enable other people's research on steelpans. Steelpan-Pitch is currently trained only on tenor steelpan audio, but in order to be comprehensive it should be extended to the entire steelpan family. Furthermore, adding more instances of tenor steelpans to the training data should improve accuracy and improve its generalization to new instances of steelpans.

Another area in which Steelpan-Pitch could be extended would be to add polyphonic pitch detection. In training, this is not complicated. Steelpan-Pitch already uses Gaussian-blurred targets rather than one-hot vectors. When given a polyphonic input, the training target can be extended to have two Gaussian-blurred "hotspots" in the target vector. This should work well for training. Inference is where this becomes a challenge. The post-processing of the neural network output would need to be modified. A significant amount of logic is likely necessary in order to accurately determine the number of voices and pitches for polyphonic output. Or, the final stages of the system may need to be replaced with an entirely new system. Designing a polyphonic, low-latency, audio-to-MIDI pitch detector for the steelpan would be a significant extension to the capabilities of Steelpan-Pitch.

Chapter 4 presented a modification to the circular buffer that allows for the implementation of live, real-time time-stretching—something that had previously never been done before. This was demonstrated through the open source AU/VST audio plugin, Realstretch, that won third prize in the 2020 AES Student Competition: MATLAB Plugin. The utility of this audio effect has also been demonstrated through its incorporation into many recordings and performances—including my own. However, while Realstretch is a fully operational audio effect, there are some important improvements that can be made. As mentioned in Chapter 4, the computational load and latency of Realstretch need improvement. There are strategies that can be used with the Paulstretch algorithm to reduce latency while maintaining audio quality, but adding a different time-stretching algorithm will also The next evolution of Re-

alstretch, however, should be to implement it in a guitar pedal format. While there are non-trivial technical challenges in doing this, it is meant for live performance and this would be an format for Realstretch.

In 2021, my next entry, HarmonEQ, won first prize in the AES Student Competition: MATLAB Plugin. HarmonEQ is a harmonic equalizer—an equalizer that removes the familiar frequency controls of the audio equalizer and instead utilizes 'root note' and 'chord type' controls. This change in the control scheme allows for the more important contribution from HarmonEQ: the pairing of chord detection with equalization. HarmonEQ has an automatic mode where it will analyze the incoming audio signal, perform chord detection, and automatically update its 'root note' and 'chord type' parameters to match the incoming audio. This enables new modes of interaction and processing in audio mixing that are either impractical or impossible on traditional equalizers. Despite these interesting new features, due to the nature of its development, HarmonEQ is still incomplete. It does not have all of the standard features of an equalizer and its graphical user interface could use significant improvement. Another avenue for improvement is the chord detection system. Since that is a major feature of HarmonEQ, any improvement in that algorithm—both in terms of speed and accuracy—would be good for HarmonEQ. Another piece of future related work is the playable version that takes the concept of HarmonEQ and turns it into a cross between an equalizer and a synthesizer.

In Chapter 6, I presented an overview of the electroacoustic steelpan repertoire, digital musical instruments, and my approach to using the steelpan in electroacoustic performance. Not counting the works of Gregory Boyd and Yoshio Machida or the remixes on Darren Dyke's album, there were about 10 total electroacoustic steelpan pieces in existence. In this document, I presented three of my own new compositions, *Crushed Atmos*, *Steelpanopticon*, and *(De/Con)struction in Steel and Electricity*. However, I have continued to compose and commission new works since the PhD recital. At present, I have now composed eight solo pieces for electroacoustic tenor steelpan and commissioned one for the double tenor steelpan from composer Matthew Burtner. I have almost doubled the size of the electroacoustic steelpan repertoire. As I continue to work, I am exploring new ways of augmenting the steelpan. Recently I have been experimenting with sensors to measure swinging motion and tilting of the steelpan. I have also been experimenting with placing the piezo pickup on the skirt of the steelpan. This changes the sound of the steelpan drastically. As I continue to composer new elecroacoustic works for steelpan, I will be incorporating and building

upon these new ideas to fully realize the augmented steelpan.

Bibliography

[1] Anthony Achong. The Steelpan as a System of Non-Linear Mode-Localized Oscillators, I: Theory, Simulations, Experiments and Bifurcations. *Journal of Sound and Vibration*, 197(4):471–487, 1996.

[2] Anthony Achong. The Steelpan as a System of Non-Linear Mode-Localized Oscillators, Part III: The Inverse Problem-Parameter Estimation. *Journal of Sound and Vibration*, 212(4):623–635, 1998.

[3] Anthony Achong. Non-linear Analysis of Compressively/thermally Stressed Elastic Shell Structures on the Steelpan and the Underlying Theory of the Tuning Process | Elsevier Enhanced Reader. *Journal of Sound and Vibration*, 222(4):597–620, 1999.

[4] Anthony Achong and K. A. Sinanan-Singh. The Steelpan as a System of Non-Linear Mode-Localized Oscillators, Part II: Coupled Sub-Systems, Simulations and Experiments. *Journal of Sound and Vibration*, 203(4):547–561, 1997.

[5] Andy Akiho. *21*. Self-published, 2008.

[6] Andy Akiho. *No One To Know One*. Innova Recordings #801, New Haven, CT, 2011.

[7] Javier Alvarez. *Así El Acero*. Self-published, 1988.

[8] Javier Alvarez. *Papalotl - Transformations Exóticas - The Music of Javier Alvarez*. Saydisc Records, 1992.

[9] Javier Alvarez. *Ofrande*. Self-published, 2000.

[10] Javier Alvarez. *Mantis Walk in Metal Space*. Self-published, 2001.

[11] Bruce Bartlett. Choosing the Right Microphone by Understanding Design Tradeoffs. *Journal of the Audio Engineering Society*, 35(11):924–944, November 1987.

[12] Mark Berry. *Mare Tranquillitatis*. Tapspace, 2009.

[13] Rachel M. Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan P. Bello. Deep Salience Representations for F0 Estimation in Polyphonic Music. *International Society for Music Information Retrieval*, pages 63–70, 2017.

[14] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez Gutiérrez, Sankalp Gulati, Perfecto Herrera Boyer, Oscar Mayor, Gerard Roma Trepat, Justin Salamon, José Ricardo Zapata González, and Xavier Serra. Essentia: An Audio Analysis Library for Music Information Retrieval. In *Transactions of the International Society for Music Information Retrieval*, pages 493–498, 2013.

[15] Gregory Boyd. *Beating on a Drum*. Self-published, 2016.

[16] Gregory Boyd. *Transformation*. Self-published, 2016.

[17] Robert Bristow-Johnson. Cookbook Formulae for Audio Equalizer Biquad Coefficients. *https://webaudio.github.io/Audio-EQ-Cookbook/audio-eq-cookbook.html*.

[18] Gary Bromham, David Moffat, Mathieu Barthet, Anne Danielsen, and György Fazekas. The Impact of Audio Effects Processing on the Perception of Brightness and Warmth. In *Proceedings of the 14th International Audio Mostly Conference: A Journey in Sound*, pages 183–190, Nottingham United Kingdom, September 2019. ACM.

[19] Petur Bryde and L. Mahadevan. Localization in Musical Steelpans, December 2022. arXiv:2212.14465 [cond-mat].

[20] Matthew Burtner. The Metasaxophone: Concept, Implementation, and Mapping Strategies for a New Computer Music Instrument. *Organised Sound*, 7(2):201–213, August 2002. Cambridge University Press.

[21] 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, September 2008.

[22] Mario A Carvajal. *Augkit: an Augmented Drum Set System Designed for Live Performance*. PhD thesis, Florida International University, 2019.

[23] Taemin Cho, Ron J. Weiss, and Juan P. Bello. Exploring Common Variations in State of the Art Chord Recognition Systems. In *Proceedings of Sound and Music Computing Conference*, 2010.

[24] Knud Bank Christensen. A Generalization of the Biquadratic Parametric Equalizer. *Audio Engineering Society Convention 115*, page 19, 2003.

[25] Perry Cook. Principles for Designing Computer Music Controllers. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 1–4, April 2001.

[26] Brian Copeland. Pickup Methods for the Electro-Acoustic Steelpan. *West Indian Journal of Engineering*, 18(2):41–48, 1996.

[27] Brian Copeland, Andrew Morrison, and Thomas D. Rossing. Sound Radiation from Caribbean Steelpans. *The Journal of the Acoustical Society of America*, 117(1):375–383, 2005.

[28] Brian R. Copeland, Marcel Byron, Earle Philip, and Keith Maynard. Apparatus for Percussive Harmonic Musical Synthesis Utilizing MIDI Technology, November 2011. United States patent US12438980.

[29] Salmon Cupid. Electronic Synthesized Steelpan Drum, April 2006. United States paent US7030305B1.

[30] Alain de Cheveigné and Hideki Kawahara. YIN, a Fundamental Frequency Estimator for Speech and Music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, April 2002. Acoustical Society of America.

[31] Ron Dennis. A Preliminary Investigation of the Manufacture and Performance of a Tenor Steel Pan. *West Indian Journal of Engineering*, 3(1):32–71, 1971.

[32] Charlie DeVane and Gabriele Bunkheila. Automatically Generating VST Plugins from MATLAB Code. *Engineering Brief*, 238, 2016.

[33] J. Dubnowski, R. Schafer, and L. Rabiner. Real-Time Digital Hardware Pitch Detector. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(1):2–8, February 1976.

[34] Darren Dyke. *Resonance: Steel Pan in the 21st Century*. Quiet Design Records/Alas Seis Music, 2007.

[35] Stefano Fasciani and Jackson Goode. 20 NIMEs: Twenty Years of New Interfaces for Musical Expression. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, June 2021.

[36] Aníbal J. S. Ferreira. An Odd-DFT Based Approach to Time-Scale Expansion of Audio Signals. *IEEE Transactions on Speech and Audio Processing*, 7(4):441–453, July 1999.

[37] E. Ferreyra, J. G. Maldonado, L. E. Murr, S. Pappu, E. A. Trillo, C. Kennedy, M. Posada, J. De Alba, R. Chitre, and D. P. Russell. Materials Science and Metallurgy of the Caribbean Steel Drum. Part II: Heat Ttreatment, Microstructures, Hardness Profiles and Tuning Effects. *Journal of Materials Science*, 34(5):981–996, March 1999.

[38] J. L. Flanagan and R. M. Golden. Phase Vocoder. *The Bell System Technical Journal*, 45(9):1493–1509, November 1966.

[39] Emma Frid. Accessible Digital Musical Instruments—A Review of Musical Interfaces in Inclusive Music Practice. *Multimodal Technologies and Interaction*, 3(3):57, September 2019.

[40] Shinya Fujii, Masaya Hirashima, Kazutoshi Kudo, Tatsuyuki Ohtsuki, Yoshihiko Nakamura, and Shingo Oda. Synchronization Error of Drum Kit Playing with a Metronome at Different Tempi by Professional Drummers. *Music Perception*, 28(5):491–503, June 2011.

[41] Derek A. Gay. Finite Element Modelling of Steelpan Acoustics. *The Journal of the Acoustical Society of America*, 123(5):3799–3799, May 2008.

[42] Christian Hamon, Eric Mouline, and Francis Charpentier. A Diphone Synthesis System Based on Time-Domain Prosodic Modifications of Speech. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 238–241 vol.1, May 1989.

[43] Uwe J. Hansen, Thomas D. Rossing, Ellie Mannette, and Kaethe George. The Caribbean Steel Pan: Tuning and Mode Studies. *MRS Bulletin*, 20(3):44–46, March 1995.

[44] Christopher A Harte and Mark B Sandler. Automatic Chord Identification Using a Quantised Chromagram. *Audio Engineering Society*, 2005.

[45] Scott H. Hawley and Andrew C. Morrison. ConvNets for Counting: Object Detection of Transient Phenomena in Steelpan Drums. *arXiv:2102.00632 [physics]*, January 2021.

[46] Eric J. Humphrey and Juan P. Bello. Rethinking Automatic Chord Recognition with Convolutional Neural Networks. In *Proceedings of the International Conference on Machine Learning and Applications*, volume 2, pages 357–362, December 2012.

[47] Eric J. Humphrey, Taemin Cho, and Juan P. Bello. Learning a robust Tonnetz-space transform for automatic chord recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 453–456, March 2012.

[48] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, June 2015.

[49] Robert H. Jack, Tony Stockman, and Andrew McPherson. Effect of Latency on Performer Interaction and Subjective Quality Assessment of a Digital Musical Instrument. In *Proceedings of the Audio Mostly 2016*, AM '16, pages 116–123, New York, NY, USA, October 2016.

[50] Leonardo Jenkins, Wyatt Page, and Shawn Trail. An Easily Removable, Wireless Optical Sensing System (EROSS) for the trumpet. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 352–357, 2013.

[51] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. CREPE: A Convolutional Representation for Pitch Estimation. *arXiv:1802.06182 [cs, eess, stat]*, February 2018. arXiv: 1802.06182.

[52] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. arXiv: 1412.6980.

[53] Filip Korzeniowski and Gerhard Widmer. Feature Learning for Chord Recognition: The Deep Chroma Extractor. *arXiv:1612.05065 [cs]*, December 2016. arXiv: 1612.05065.

[54] Arthur Kreiger. *Meeting Places*. Albany Records, 2003.

[55] Ulf Kronman and Erik V. Jansson. Acoustics of the Steelpan: Tone Generation and Tuning - an Introductory Study of Methods for Measurements. *Speech Transmission Laboratory. Quarterly Progress and Status Report*, 29(4):17, 1988.

[56] Nelson Posse Lago and Fabio Kon. The Quest for Low Latency. In *Proceedings of the International Computer Music Conference*, 2004.

[57] Jean Laroche and Mark Dolson. Phase-Vocoder: About This Phasiness Business. In *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*, October 1997.

[58] Jean Laroche and Mark Dolson. Improved Phase Vocoder Time-Scale Modification of Audio. *IEEE Transactions on Speech and Audio Processing*, 7(3):323–332, May 1999.

[59] Kyogu Lee. Automatic Chord Recognition from Audio Using Enhanced Pitch Class Profile. In *Proceedings of the International Computer Music Conference*, 2006.

[60] Jorn Loviscach. Graphical Control of a Parametric Equalizer. *Audio Engineering Society Convention 124*, 2008.

[61] Wenfang Ma, Ying Hu, and Hao Huang. Dual Attention Network for Pitch Estimation of Monophonic Music. *Symmetry*, 13(7):1296, July 2021.

[62] Yoshio Machida. *Infinite Flowers*. Amorfon, 2004.

[63] Yoshio Machida. *Steelpan Improvisations: 2001-2008*. Amorfon, 2009.

[64] Colin Malloy. Study of Timbral Variation of Tenor Steelpan Mallets Through Spectral Analysis. In *Proceedings of Meetings on Acoustics*, volume 39, San Diego, California, 2019.

[65] Colin Malloy. An Approach for Implementing Time-Stretching as a Live Realtime Audio Effect. In *Audio Engineering Society Convention 149*, 2020.

[66] Colin Malloy. Improved Steelpan Pitch Detection Through Audio Feature Extraction and Machine Learning. *The Journal of the Acoustical Society of America*, 149(4):A122–A122, April 2021.

[67] Colin Malloy. Steelpan Fundamental Frequency Estimation Through Audio Feature Extraction and Deep Neural Networks. *The Journal of the Acoustical Society of America*, 150(4):A175–A175, October 2021.

[68] Colin Malloy. Crushed Atmos. `https://youtu.be/Cht_AvFQOn8`, March 2022. Victoria, Canada.

[69] Colin Malloy. Cycles. `https://youtu.be/OqFnRtbCoNk`, March 2022. Victoria, Canada.

[70] Colin Malloy. (De/Con)struction in Steel and Electricity. `https://youtu.be/32mMqxS_zg4`, March 2022. Victoria, Canada.

[71] Colin Malloy. Estimating Steelpan Note Class from Attack Transients. *The Journal of the Acoustical Society of America*, 152(4):A220–A220, October 2022.

[72] Colin Malloy. Oil Drum. `https://youtu.be/T1PfoXwpy7o`, March 2022. Victoria, Canada.

[73] Colin Malloy. Steelpanopticon. `https://youtu.be/CD5gZS6wGDU`, March 2022. Victoria, Canada.

[74] Colin Malloy. Timbral Effects the Paulstretch Audio Time-Stretching Algorithm. *The Journal of the Acoustical Society of America*, 151(4):A158–A158, April 2022.

[75] Colin Malloy and George Tzanetakis. Steelpan-Specific Pitch Detection: A Dataset and Deep Learning Model. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2023.

[76] Soren E. Maloney, Clement A. C. Imbert, and Aubrey G. Bryan. The Aubrapan: Revival of a Lost Invention. In *Proceedings of the International Conference on Acoustics*, pages 858–861, Rotterdam, 2009.

[77] Soren E. Maloney and R Traynor. Modal Testing of a Soprano Pan Using a 3D Laser Doppler Vibrometer. In *Proceedings of the International Conference on Acoustics*, pages 1478–1481, Rotterdam, 2009.

[78] George Massenburg. Parametric Equalization. In *Audio Engineering Society Convention 42*. Audio Engineering Society, May 1972.

[79] Matthias Mauch and Simon Dixon. PYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, May 2014. ISSN: 2379-190X.

[80] Robert McClure. *Passacaglia on a Theme by Mark Rothko*. Media Press Inc, 2009.

[81] Robert McClure. *Electroacoustic*. Self-published, 2017.

[82] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and Music Signal Analysis in Python. In *Proceedings of the 14th Python in Science Conference*, pages 18–24, Austin, Texas, 2015.

[83] Brian McNulty. Mario DeCiutiis and the Evolution of Virtual Percussion. *Percussive Notes*, 48(2), 2006.

[84] Andrew McPherson, Robert Jack, and Giulio Moro. Action-Sound Latency: Are Our Tools Fast Enough? In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 20–25, Brisbane, Australia, July 2020. NIME.

[85] Eric Moulines and Francis Charpentier. Pitch-Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones. *Speech Communication*, 9(5-6):453–467, December 1990.

[86] Fasil Muddeen and Brian Copeland. Free-Field Sound Measurements of the Caribbean Steelpan - an Application of Cepstrum Analysis. In *Proceedings of the IEEE SoutheastCon 2003*, pages 105–110, April 2003.

[87] Fasil Muddeen and Brian Copeland. Sound Radiation from Caribbean Steelpans Using Nearfield Acoustical Holography. *The Journal of the Acoustical Society of America*, 131(2):1558–1565, February 2012.

[88] Fasil Muddeen and Brian Copeland. Microphone Placement for Tenor Pan Sound Recording: New Recommendations Based on Recent Research. *West Indian Journal of Engineering*, 35(2):95–102, 2013.

[89] L. E. Murr, E. Ferreyra, J. G. Maldonado, E. A. Trillo, S. Pappu, C. Kennedy, J. De Alba, M. Posada, D. P. Russell, and J. L. White. Materials Science and Metallurgy of the Caribbean Steel Drum. Part I: Fabrication, Deformation Phenomena and Acoustic Fundamentals. *Journal of Materials Science*, 34(5):967–979, March 1999.

[90] Lawrence Murr and Larry White. Metallurgy of the Caribbean Steel Drum. *Percussive Notes*, 38(1), 2000.

[91] Lawrence E. Murr and Everaldo Ferreyra Tello. Connecting Materials Science and Music in Steel Drums: A Serendipitous Collection of Scientific, Especially Metallurgical, Principles Created Melodic Instruments from Sawed-Off Steel Barrels. *American Scientist*, 88(1):38–45, 2000.

[92] Shahan Nercessian. Neural Parametric Equalizer Matching Using Differentiable Biquads. In *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx-20)*, 2020.

[93] A. Michael Noll. Cepstrum Pitch Determination. *The Journal of the Acoustical Society of America*, 41(2):293–309, February 1967.

[94] Phillip O'Banion. *Digital Divide*. Self-published, 2016.

[95] Sophocles J. Orfanidis. High-Order Digital parametric Equalizer Design. *Journal of the Audio Engineering Society*, 53(11), 2005.

[96] Gabriela Ortiz. Magna sin. `https://www.gabrielaortiz.com/index.php?idioma=en&p=2&tipo=otros&obra=15`.

[97] Gabriela Ortiz. *Magna Sin*. Self-published, 1992.

[98] Laurent Oudre, Yves Grenier, and Cedric Fevotte. Template-Based Chord Recognition: Influence of the Chord Types. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 2009.

[99] Cléo Palacio-Quintin. The hyper-flute. In *Proceedings of the 2003 conference on New interfaces for musical expression*, NIME '03, pages 206–207, SGP, May 2003. National University of Singapore.

[100] Cléo Palacio-Quintin. Eight Years of Practice on the Hyper-Flute: Technological and Musical Perspectives. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 293–298, 2008.

[101] Jonggwon Park, Kyoyun Choi, Sungwook Jeon, Dokyun Kim, and Jonghun Park. A Bi-directional Transformer for Musical Chord Recognition. *arXiv:1907.02698 [cs, eess]*, July 2019. arXiv: 1907.02698.

[102] Geoffrey Peeters. A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project. Technical report, IRCAM, 2004.

[103] Umesh Persad, Keivi Howard, and Jorrel Bisnath. The Roll-Up Digital Steelpan. In *Proceedings of the International Conference on Emerging Trends in Engineering & Technology (IConETech-2020)*, pages 746–755. Faculty of Engineering, The University of the West Indies, St. Augustine, 2020.

[104] M. Portnoff. Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(3):243–248, June 1976.

[105] Zdenek Prusa and Nicki Holighaus. Phase Vocoder Done Right. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 976–980, Kos, Greece, August 2017. IEEE.

[106] Miller Puckette. Phase-Locked Vocoder. In *Proceedings of 1995 Workshop on Applications of Signal Processing to Audio and Accoustics*, pages 222–225, October 1995.

[107] Elio Quinton, Florabelle Spielmann, and Bob L. Sturm. Computational Ethnomusicology for Exploring Trends in Trinidad Steelband Music Through History. In *Proceedings of the 13th International Symposium on CMMR*, pages 57–67, 2017.

[108] Prajwal S. Rao, S. Khoushikh, Sriram Ravishankar, R. Advaith Ananthkrishnan, and K. Balachandra. A Comparative Study of Various Pitch Detection Algorithms. In *Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS)*, pages 1–6, October 2020.

[109] Sarah Reid, Ryan Gaston, Colin Honigman, and Ajay Kapur. Minimally Invasive Gesture Sensing Interface (MIGSI) for Trumpet. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 419–424, 2016.

[110] Joshua D. Reiss. Design of Audio Parametric Equalizer Filters Directly in the Digital Domain. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1843–1848, August 2011.

[111] Curtis Roads. Automated Granular Synthesis of Sound. *Computer Music Journal*, 2(2):61, September 1978.

[112] Myron Ross, Harry Shaffer, Andrew Cohen, Richard Freudberg, and Harold Manley. Average Magnitude Difference Function Pitch Extractor. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 22(5):353–362, October 1974.

[113] Thomas D. Rossing. *Science of Percussion Instruments*. World Scientific, Singapore, 2000.

[114] Thomas D. Rossing and N. H. Fletcher. Nonlinear Vibrations in Plates and Gongs. *The Journal of the Acoustical Society of America*, 73(1):345–351, 1983.

[115] Thomas D. Rossing, D. Scott Hampton, and Uwe J. Hansen. Music from Oil Drums: The Acoustics of the Steel Pan. *Physics Today*, 49(3):24–29, March 1996.

[116] Thomas D. Rossing and Uwe J. Hansen. Vibrational Mode Shapes in Caribbean Steelpans: Part II: Cello and Bass. *Applied Acoustics*, 65(12):1233–1247, December 2004.

[117] Salim Roucos and Alexander M. Wilgus. High Quality Time-Scale Modification for Speech. In *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 10, pages 493–496, April 1985.

[118] Luke Rowe and George Tzanetakis. Curriculum Learning for Imbalanced Classification in Large Vocabulary Automatic Chord Recognition. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 2021.

[119] Teresa Ryan, Patrick O'Malley, Aldo Glean, Joseph Vignola, and John Judge. Conformal Scanning Laser Doppler Vibrometer Measurement of Tenor Steelpan Response to Impulse Excitation. *The Journal of the Acoustical Society of America*, 132(5):3494–3501, November 2012.

[120] Kenzo Sato. Free Flexural Vibrations of an Elliptical Plate with Simply Supported Edge. *The Journal of the Acoustical Society of America*, 52(3B):919–922, September 1972.

[121] Sébastien Schiesser and Jan C. Schacher. SABRe: The Augmented Bass Clarinet. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2012.

[122] W. Andrew Schloss. Using Contemporary Technology in Live Performance: The Dilemma of the Performer. *Journal of New Music Research*, 32(3):239–242, September 2003.

[123] Michael Schutz and Scott Lipscomb. Hearing Gestures, Seeing Music: Vision Influences Perceived Tone Duration. *Perception*, 36(6):888–897, June 2007.

[124] Baljinder Sekhon. *Regeneration*. Self-published, 2005.

[125] Alexander Sheh and Daniel P. W. Ellis. Chord Segmentation and Recognition Using EM-Trained Hidden Markov Models. In *Proceedings of the 4th International Society for Music Information Retrieval Conference (ISMIR)*, 2003.

[126] Satwinder Singh, Ruili Wang, and Yuanhang Qiu. DeepF0: End-To-End Fundamental Frequency Estimation for Music and Speech Signals. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 61–65, June 2021. ISSN: 2379-190X.

[127] Julius Orion Smith. *Mathematics of the Discrete Fourier Transform (DFT): With Audio Applications*. Julius Smith, 2008. Google-Books-ID: fTOxS9huzHoC.

[128] Man Mohan Sondhi. New methods of Pitch Extraction. *IEEE Transactions on Audio and Electroacoustics*, 16(2):262–266, June 1968.

[129] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[130] Joseph Turian, Jordie Shier, Humair Raj Khan, Bhiksha Raj, Björn W. Schuller, Christian J. Steinmetz, Colin Malloy, George Tzanetakis, Gissel Velarde, Kirk McNally, Max Henry, Nicolas Pinto, Camille Noufi, Christian Clough, Dorien Herremans, Eduardo Fonseca, Jesse Engel, Justin Salamon, Philippe Esling, Pranay Manocha, Shinji Watanabe, Zeyu Jin, and Yonatan Bisk. HEAR: Holistic Evaluation of Audio Representations, May 2022. arXiv:2203.03022 [cs, eess, stat].

[131] Yushi Ueda, Yuki Uchiyama, Takuya Nishimoto, Nobutaka Ono, and Shigeki Sagayama. HMM-Based Approach for Automatic Chord Detection Using Refined Acoustic Features. In *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5518–5521, March 2010. ISSN: 2379-190X.

[132] W. Verhelst and M. Roelands. An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 554–557, April 1993.

[133] Adam Walters. *Prayer*. Self-published, 2019.

[134] Adam Walters. *Prayer*. Self-published, 2019.

[135] Lloyd Watts. DeepPitch: Wide-Range Monophonic Pitch Estimation using Deep Convolutional Neural Networks. In *Proceedings of the Interspeech Conference*, 2018.

[136] David Wessel and Matthew Wright. Problems and Prospects for Intimate Musical Control of Computers. *Computer Music Journal*, 26(3):11–22, September 2002.

[137] Duane K Wise. Concept, Design, and Implementation of a General Dynamic Parametric Equalizer. *J. Audio Eng. Soc.*, 57(1):13, 2009.

[138] Zhang Xiuli, Zhang Ruihua, and Chen Weidong. Design of Digital Parametric Equalizer Based on Second-Order Function. In *Proceedings of the 2010 International Conference on Image Analysis and Signal Processing*, pages 182–185, April 2010. ISSN: 2156-0129.

Appendix A

Scores

A.1 *Crushed Atmos*

A video recording of the premiere performance is available at `https://youtu.be/Cht_AvFQOn8` [73].

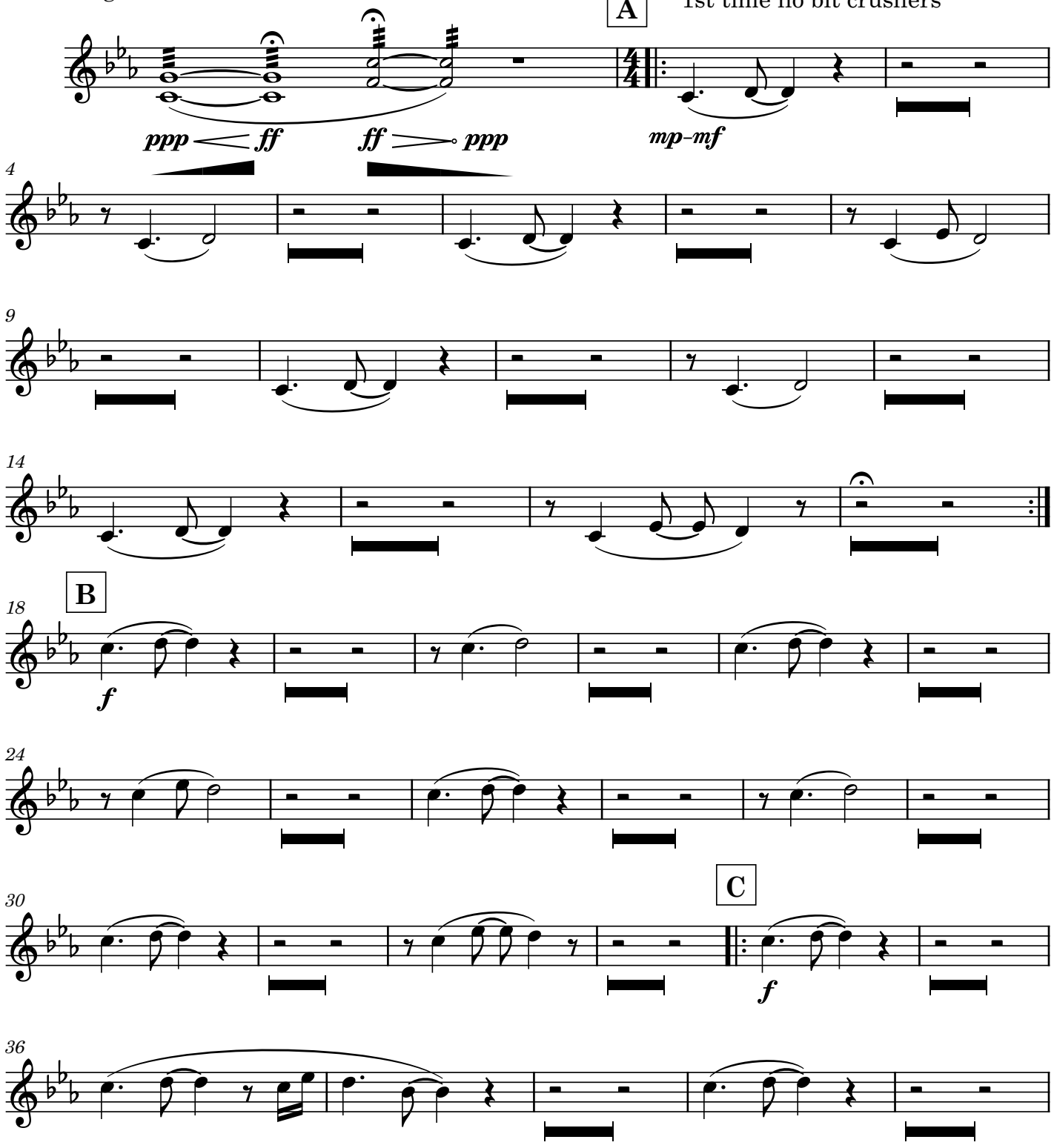
Steel Pan

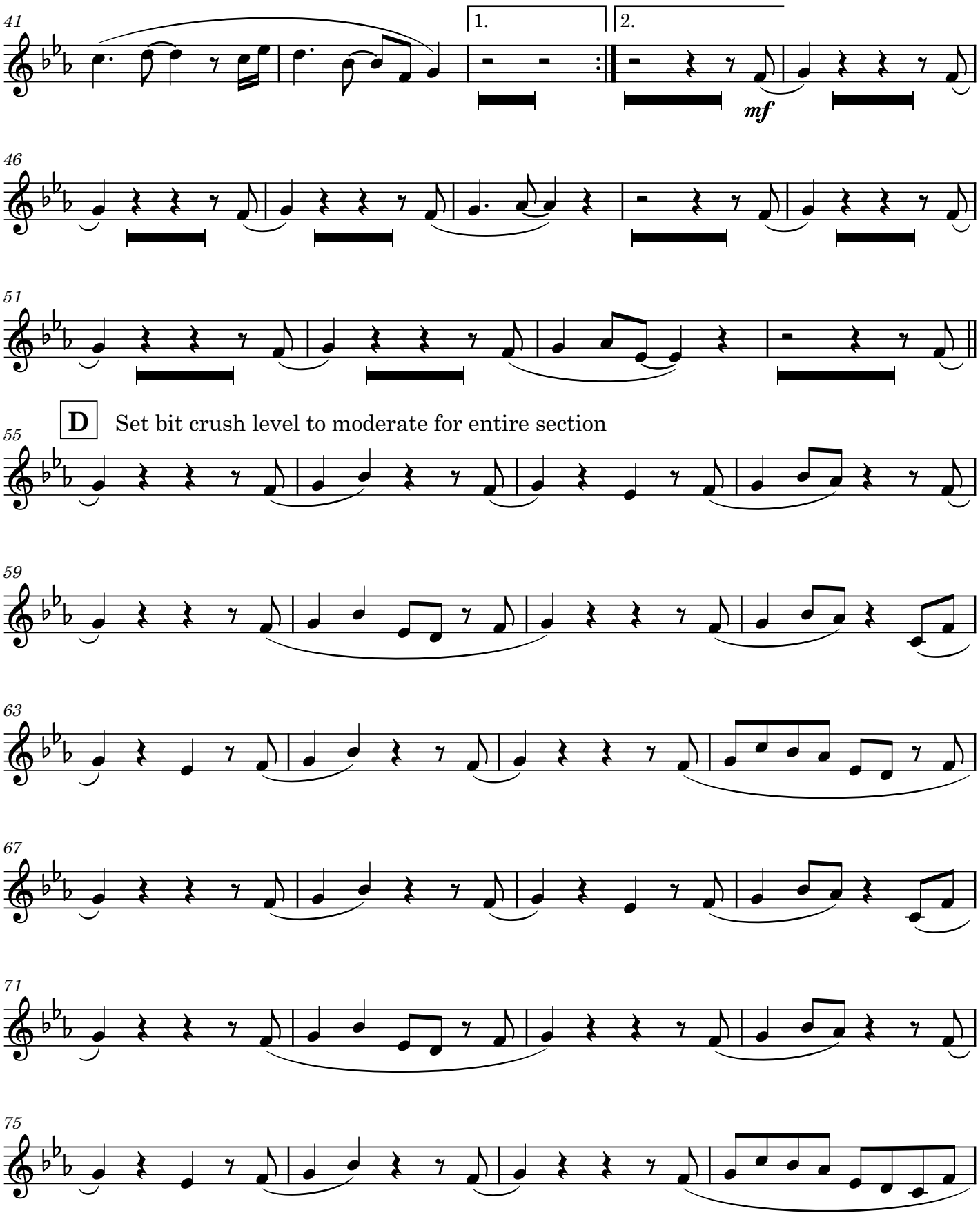
Crushed Atmos

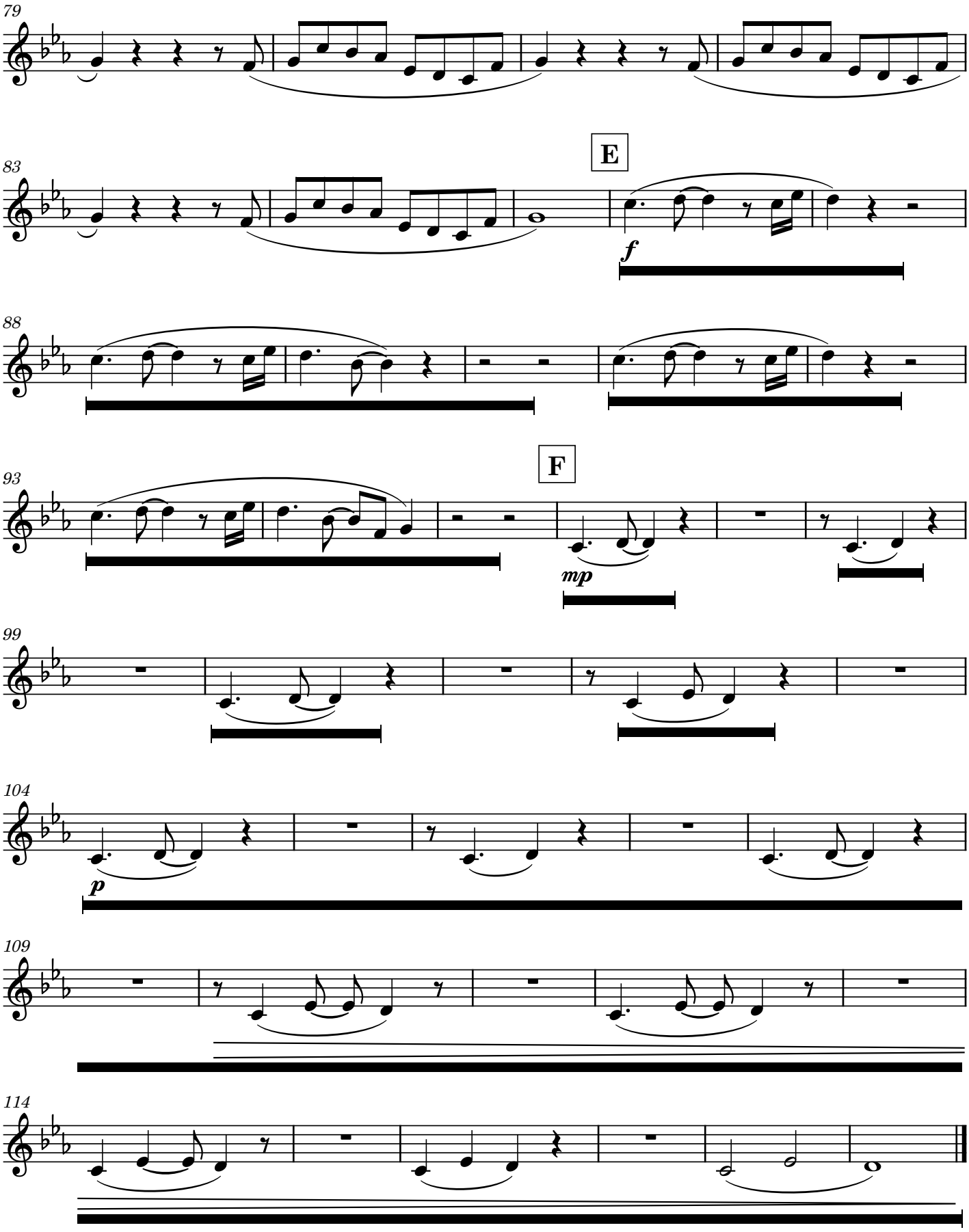
Long, slow build and release
Let this take several minutes
Ring mod on

Colin Malloy

Ring mod off
1st time no bit crushers




41
1.
2.
mf
46
51
D
Set bit crush level to moderate for entire section
55
59
63
67
71
75


79
83
E
f
88
93
F
mp
99
104
p
109
114

A.2 *Steelpanopticon*

A video recording of the premiere performance is available at `https://youtu.be/CD5gZS6wGDU` [73].

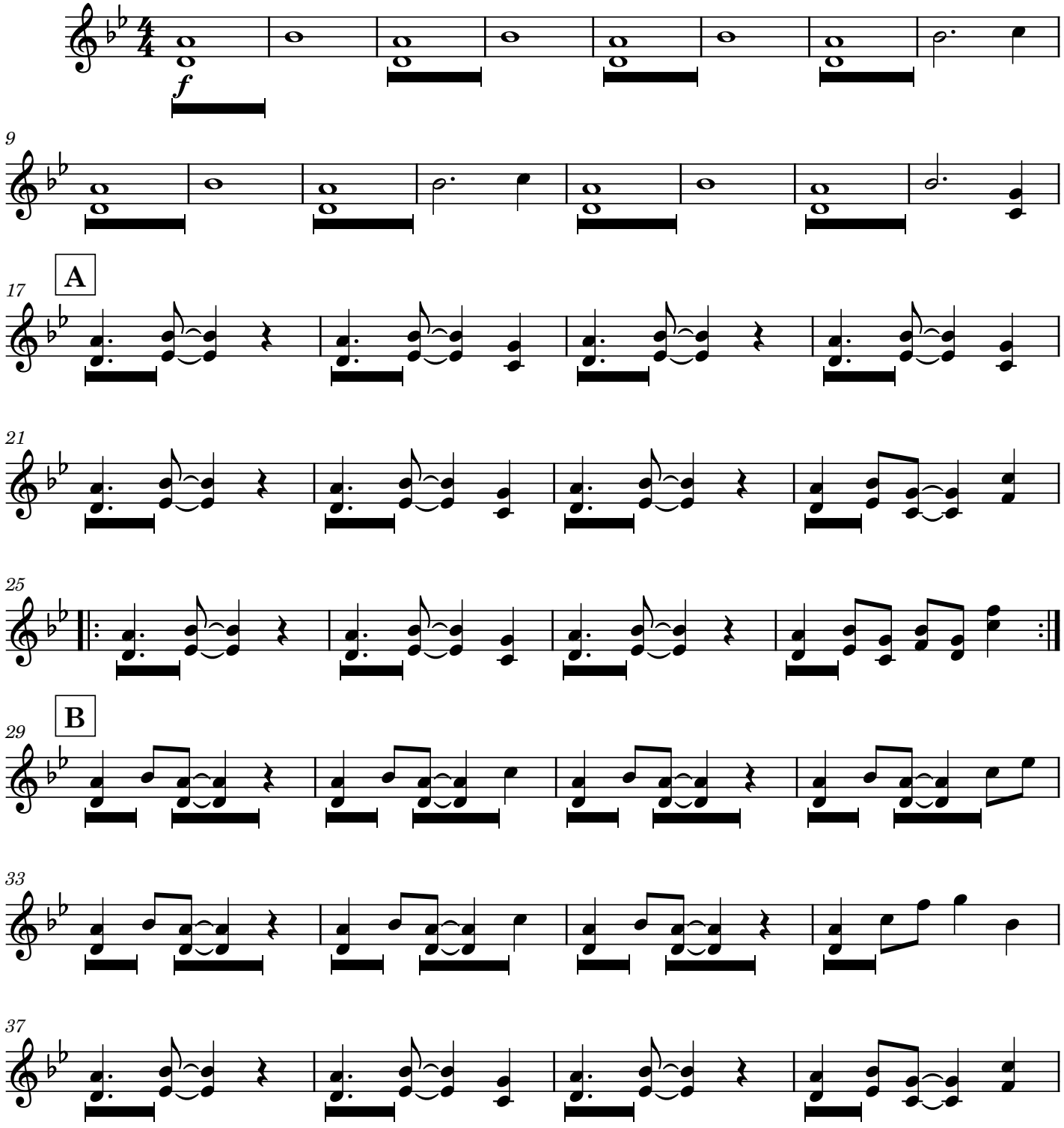
Steel Pan

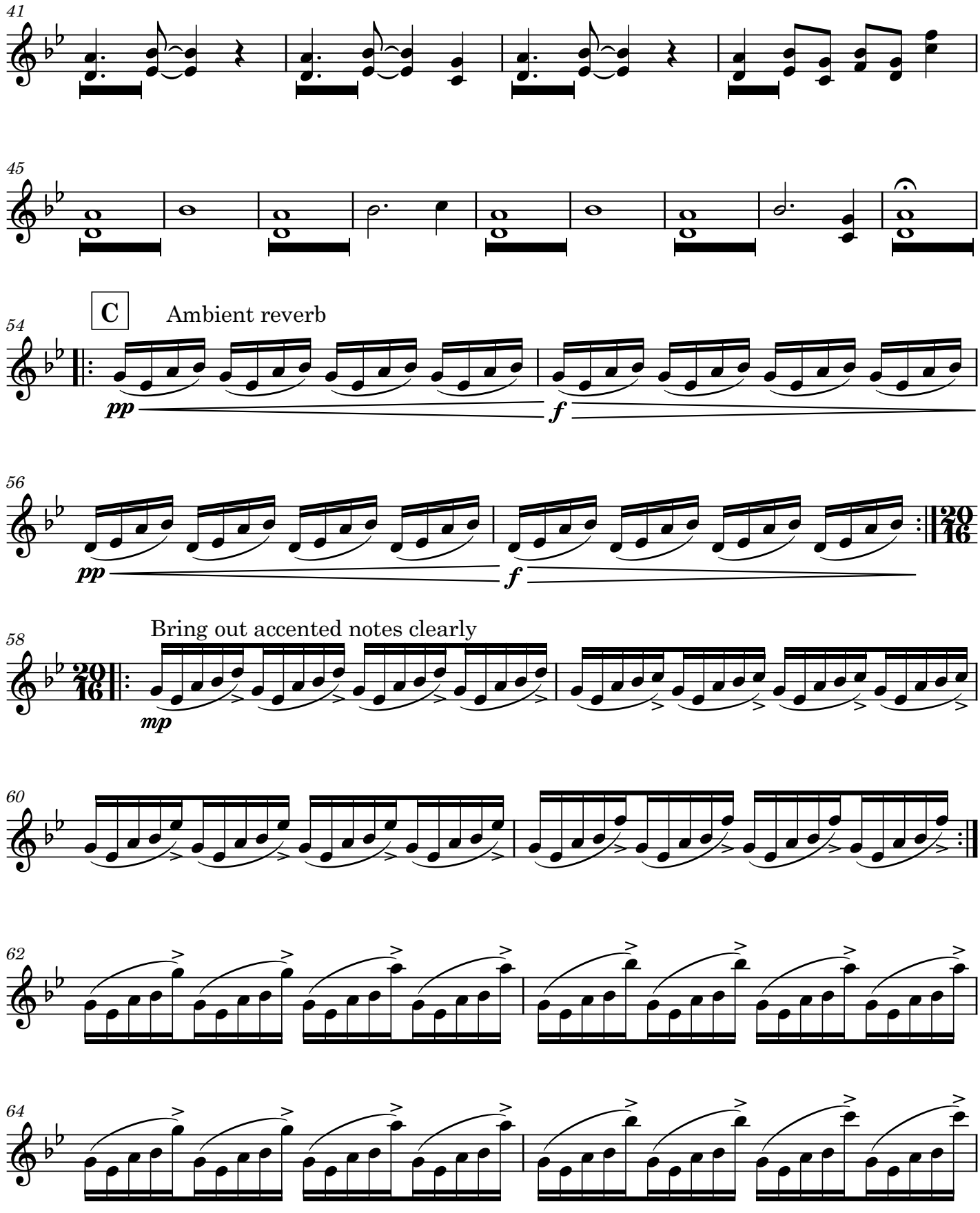
Steelpanopticon

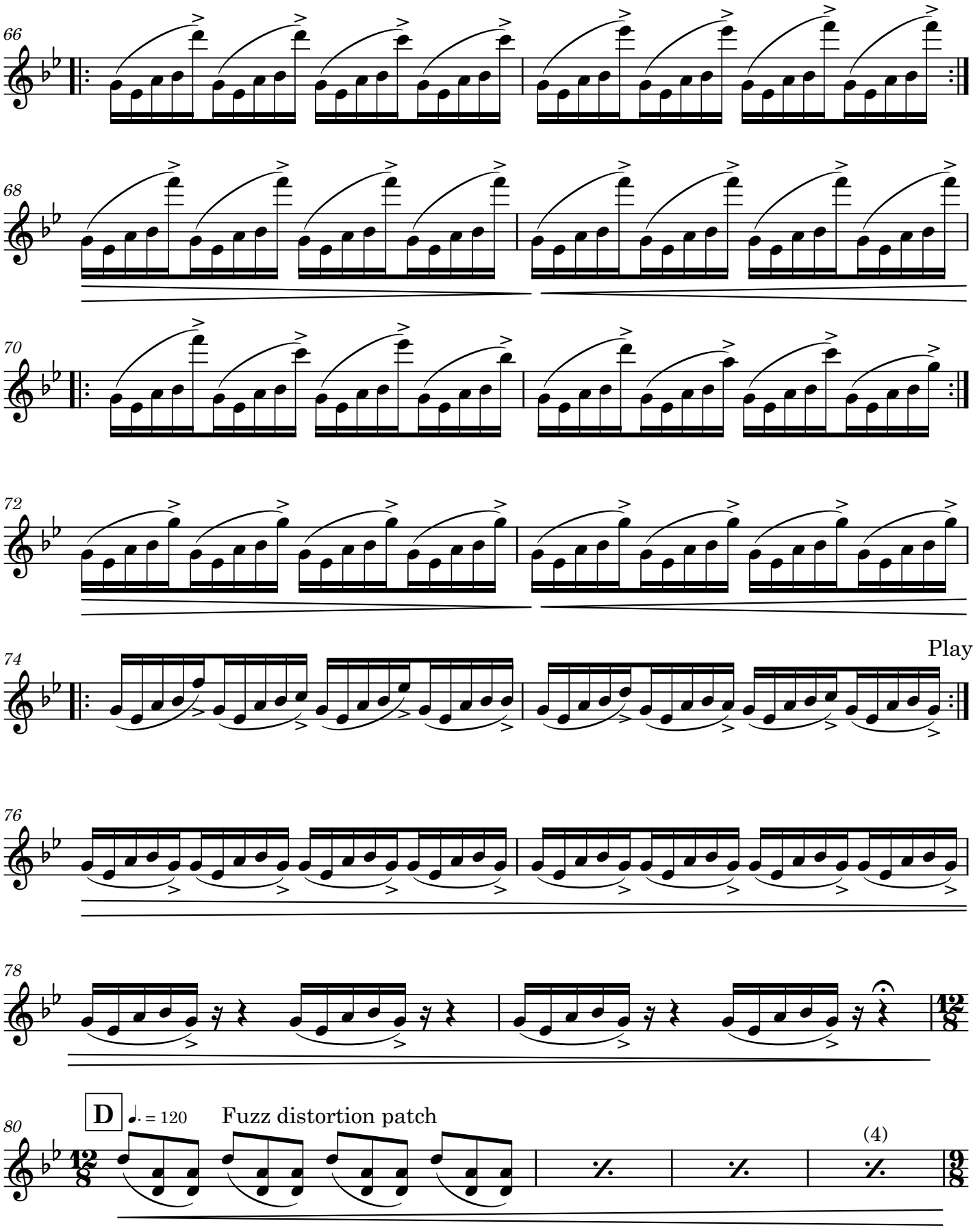
Colin Malloy

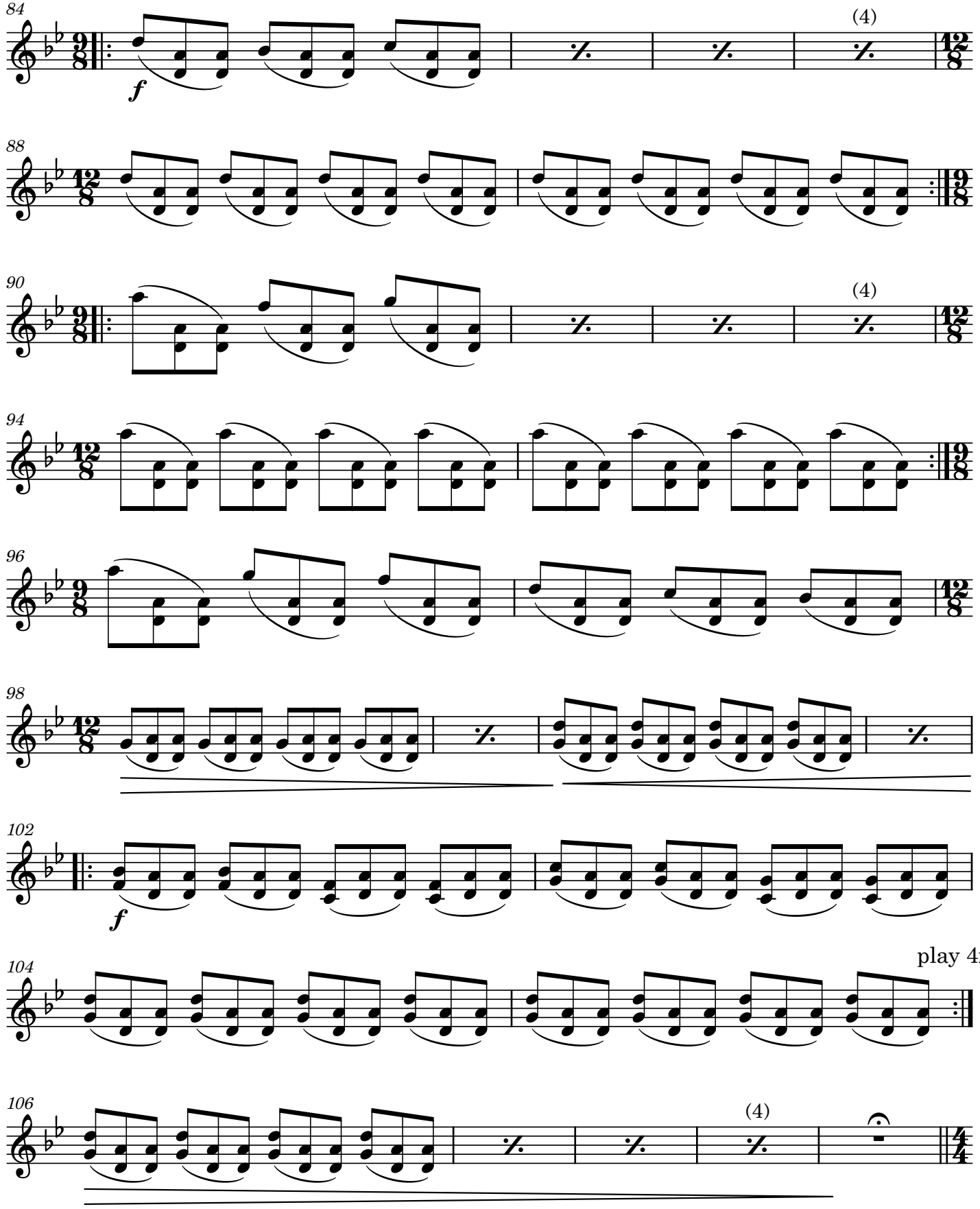
♩ = 100

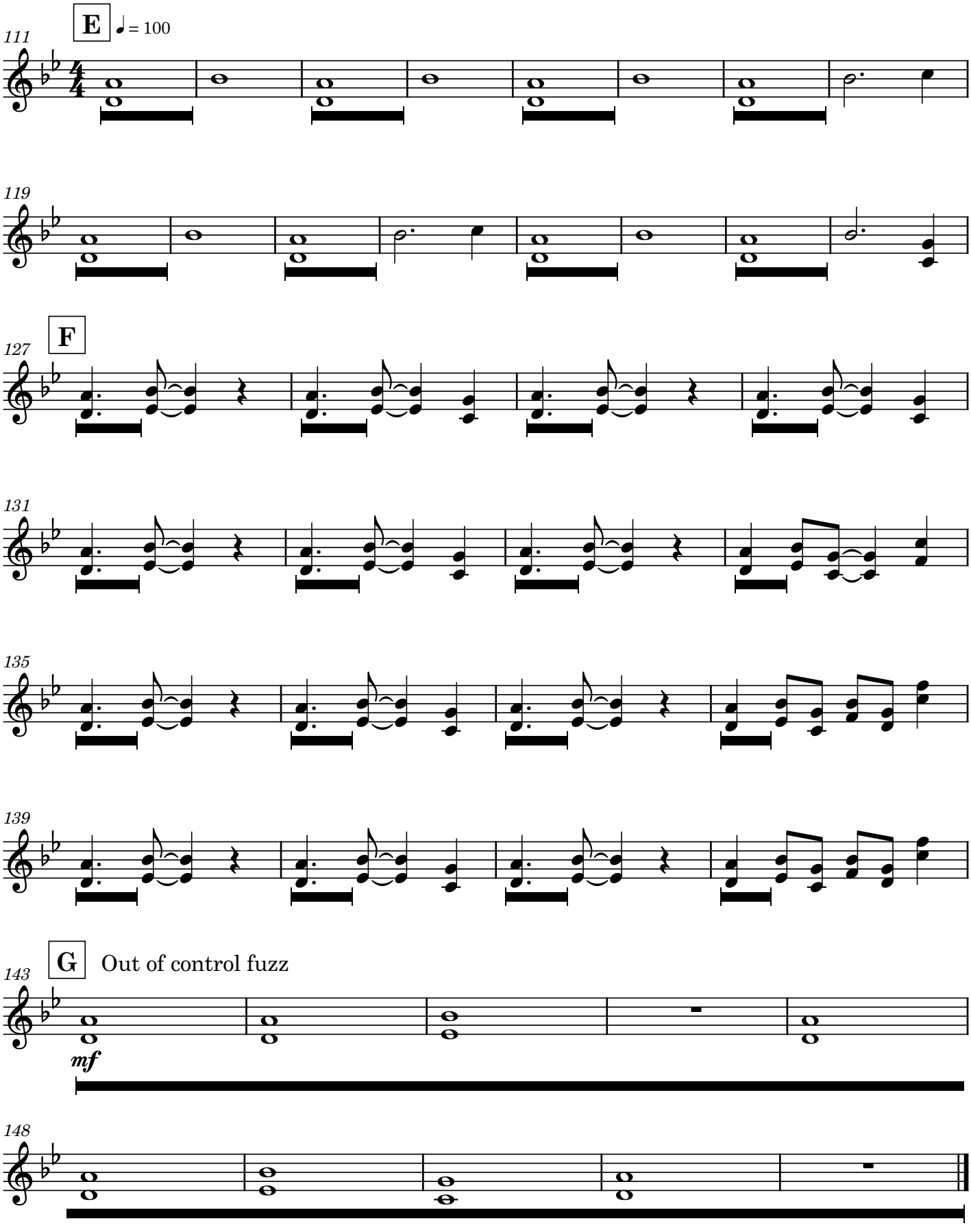
Fuzz distortion




41
45
C
Ambient reverb
54
pp
f
56
pp
f
58
Bring out accented notes clearly
mp
60
62
64


66
68
70
72
74
Play 4x
76
78
80
D
♩. = 120
Fuzz distortion patch
(4)


84
(4)
88
90
(4)
94
96
98
102
104
play 4x
106
(4)


E
♩ = 100
111
119
F
127
131
135
139
G
Out of control fuzz
143
mf
148

A.3 *(De/Con)struction in Steel and Electricity*

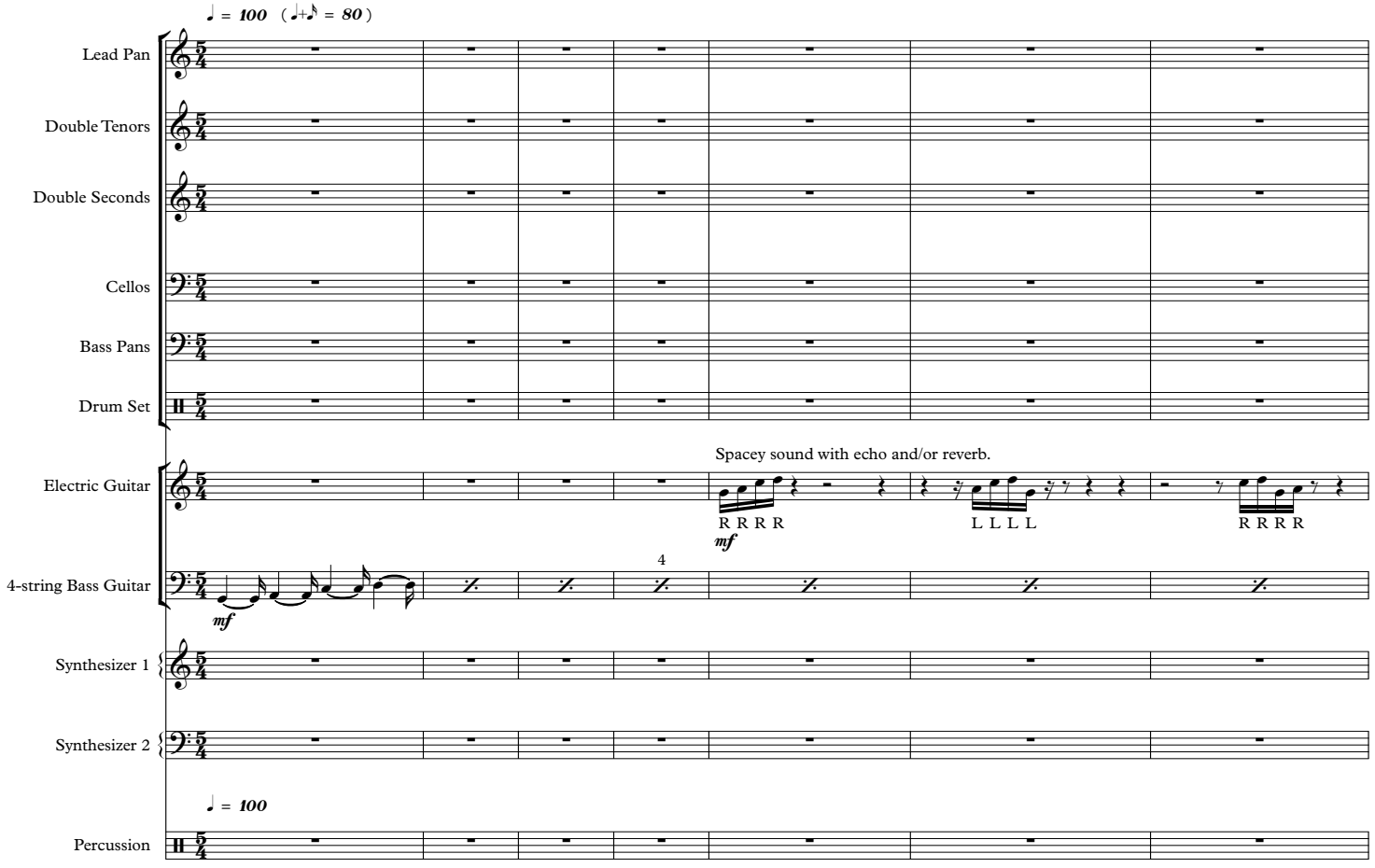
A.3.1 *(De/Con)struction in Steel and Electricity* – Original Score

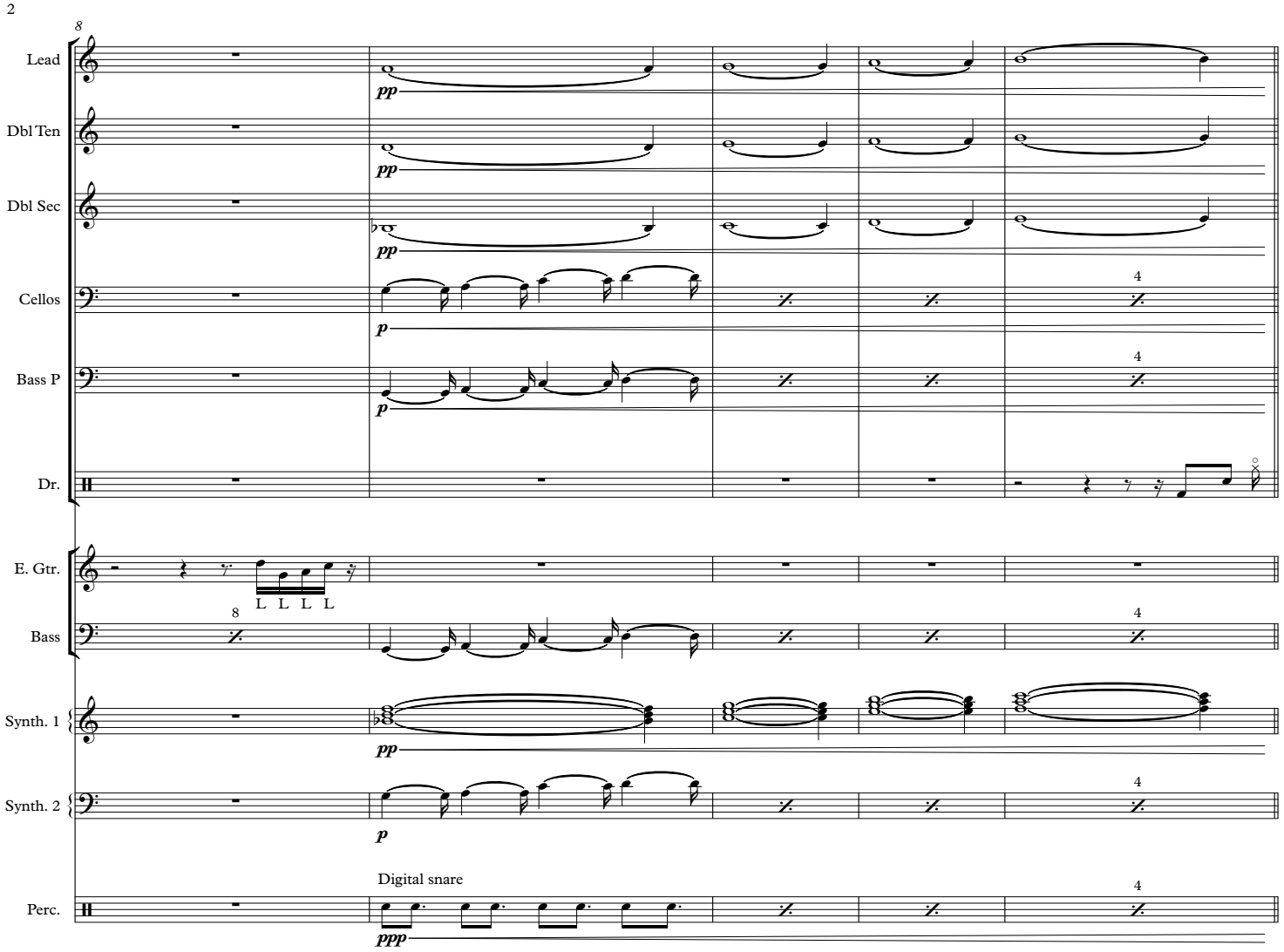
A video recording of the premiere performance of the solo version is available at `https://youtu.be/32mMqxS_zg4` [70].

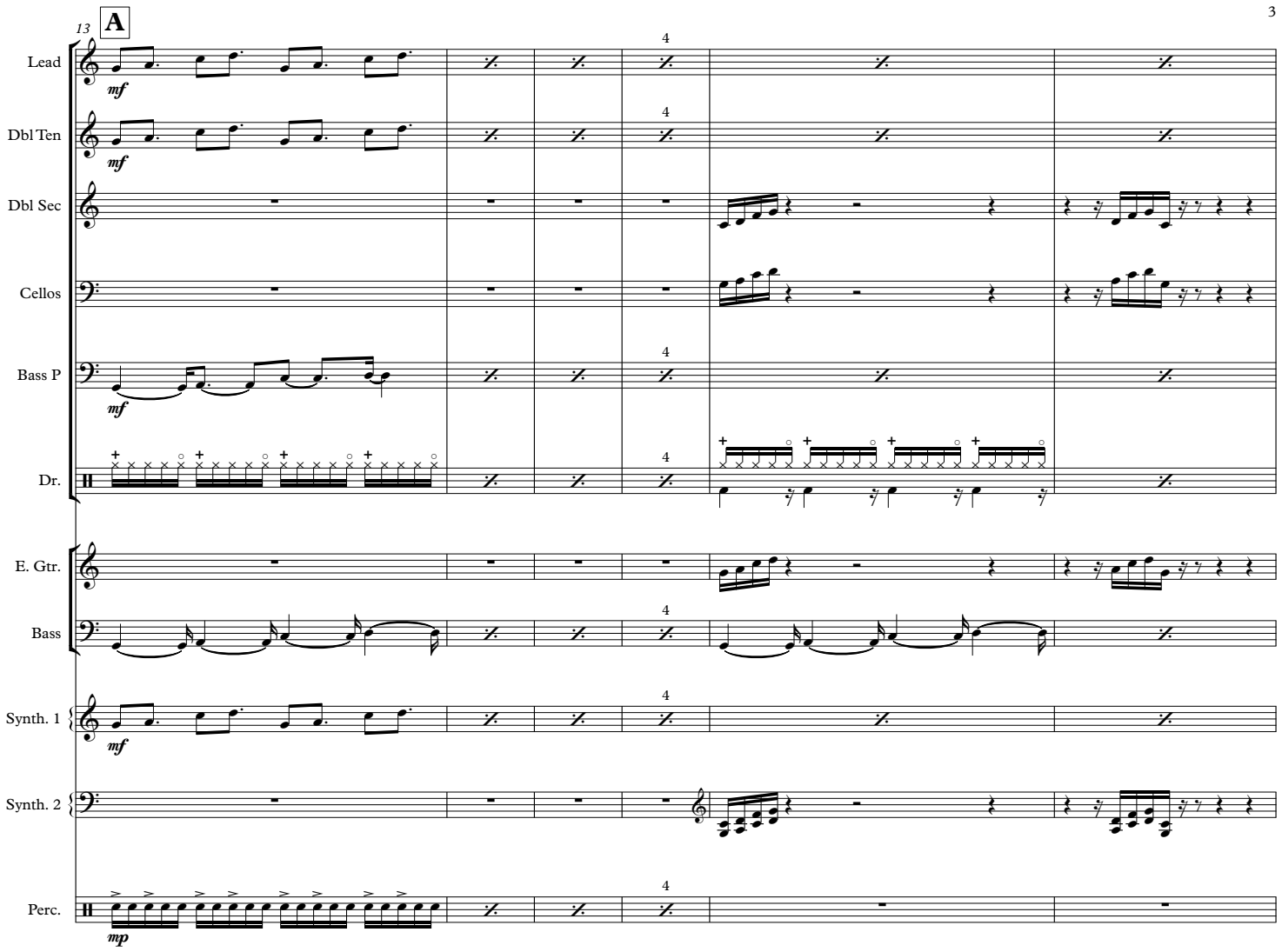
(De/Con)struction in Steel and Electricity

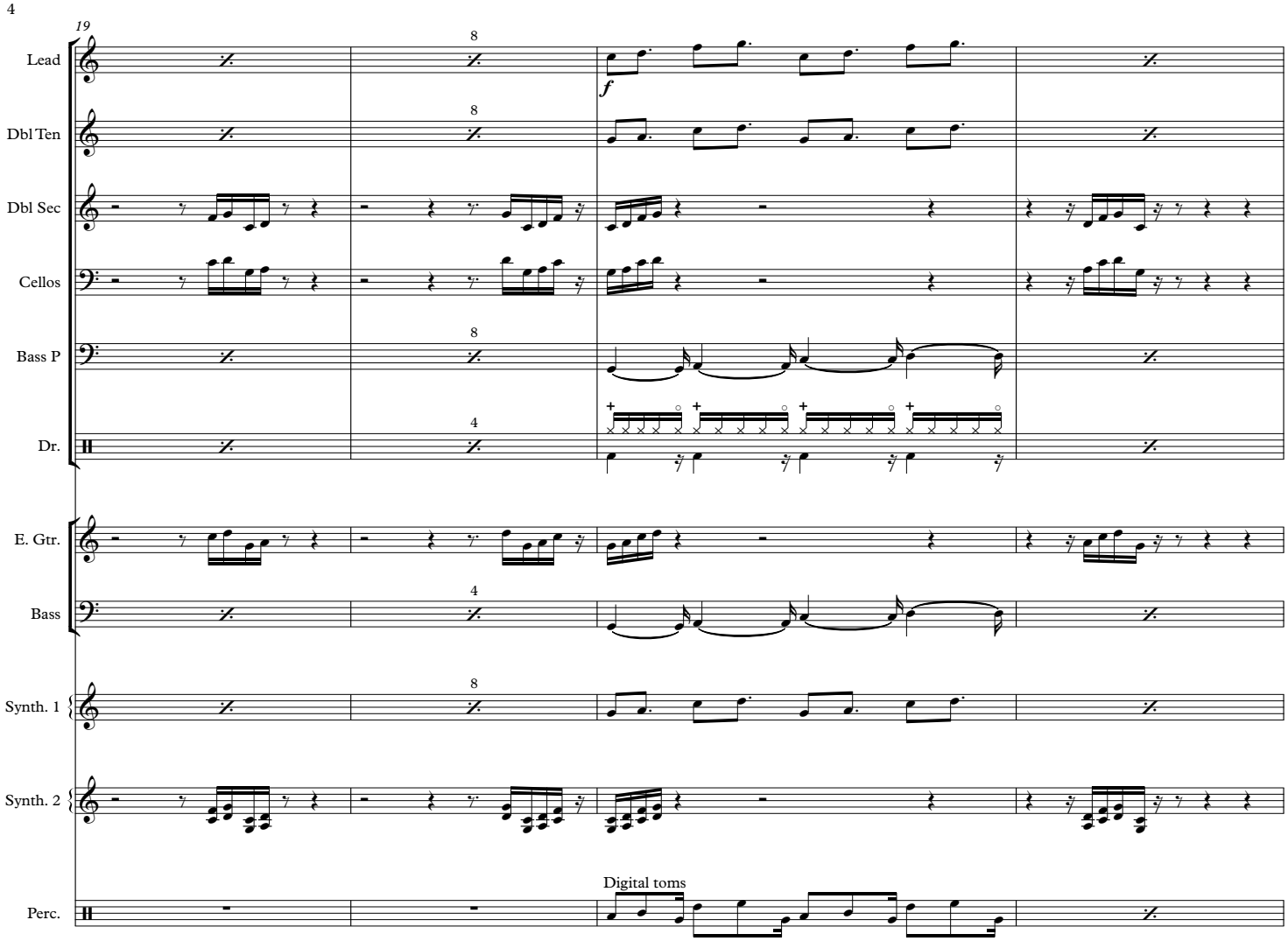
"Drops"

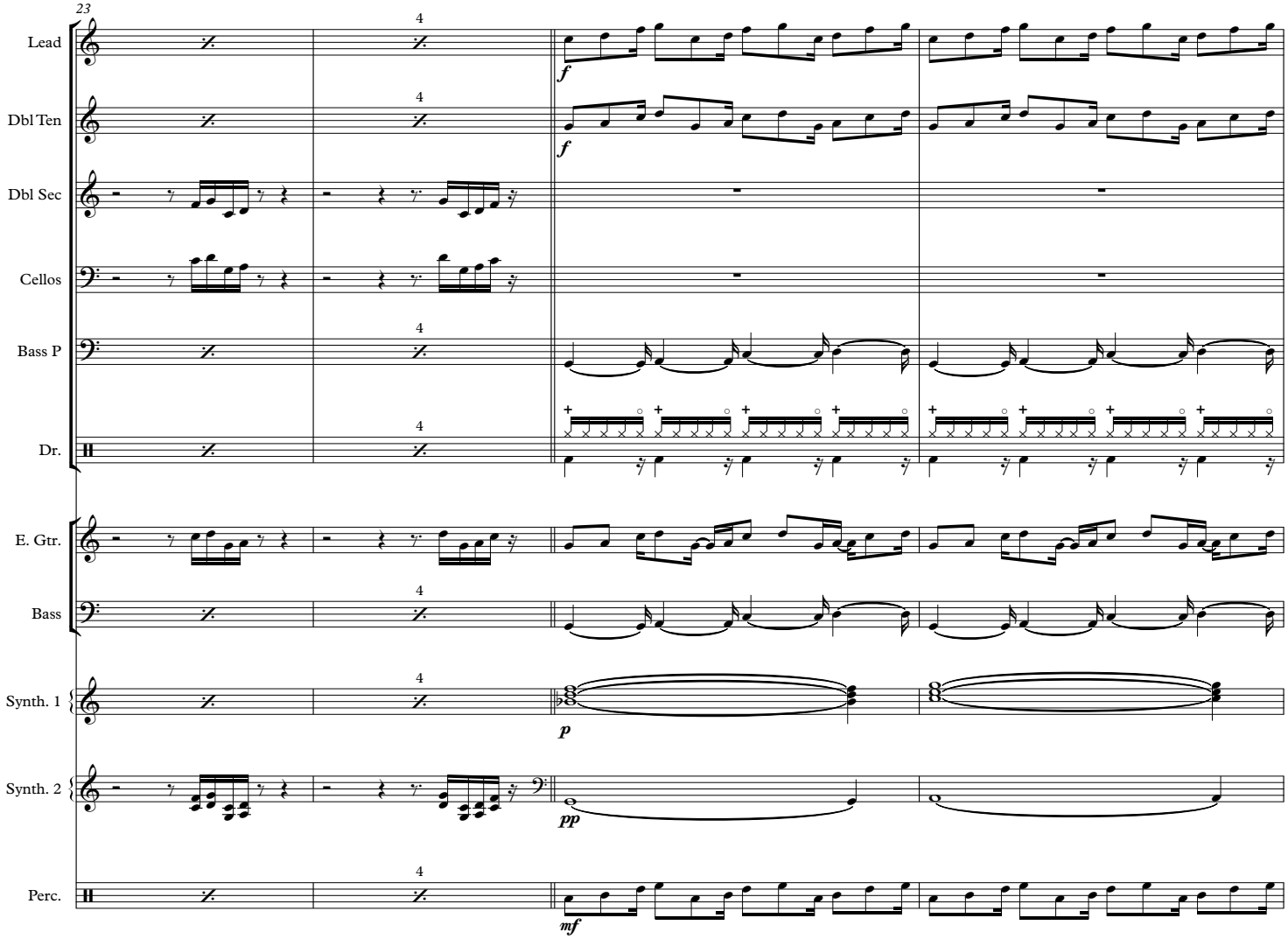
Colin Malloy

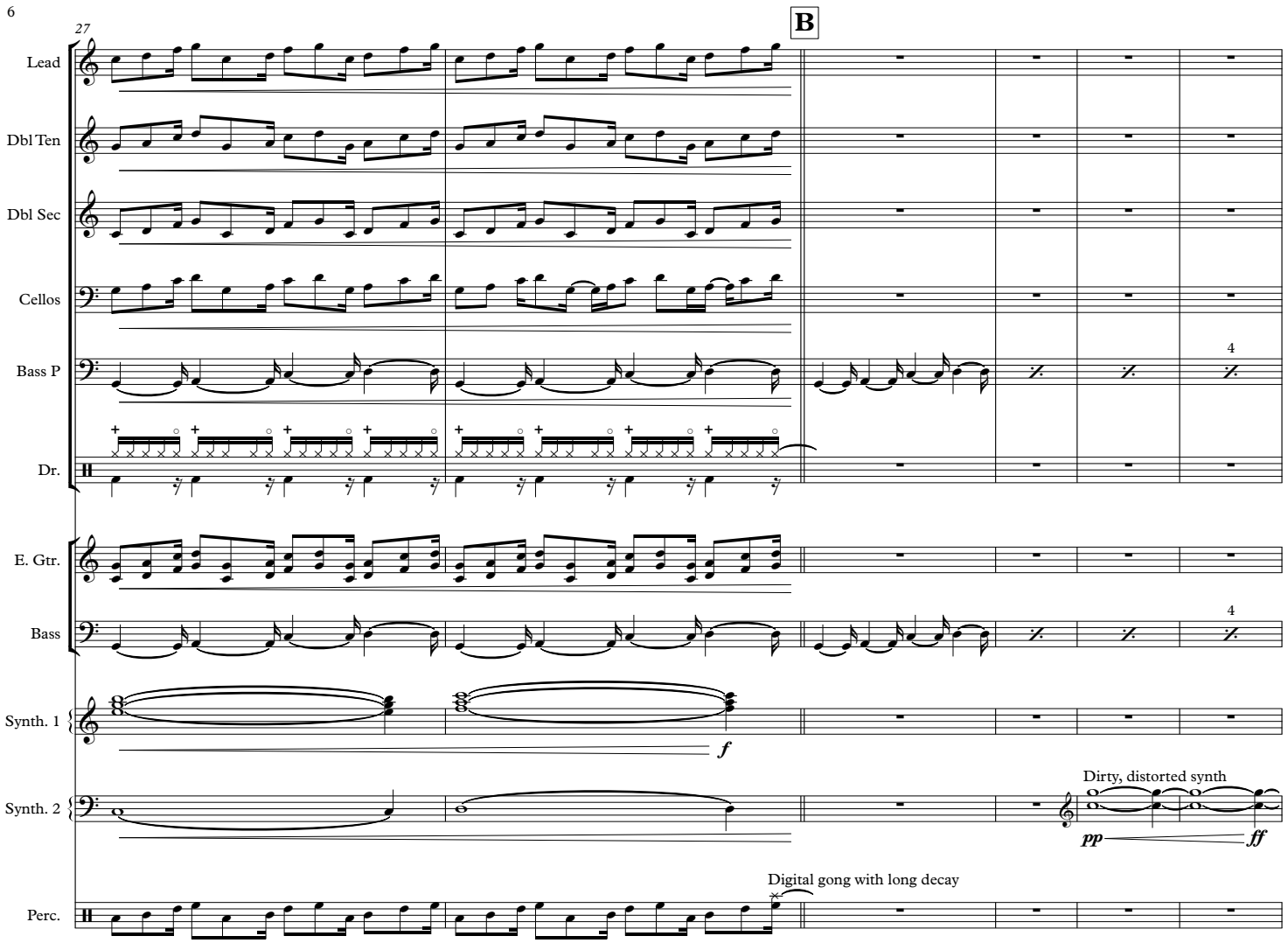


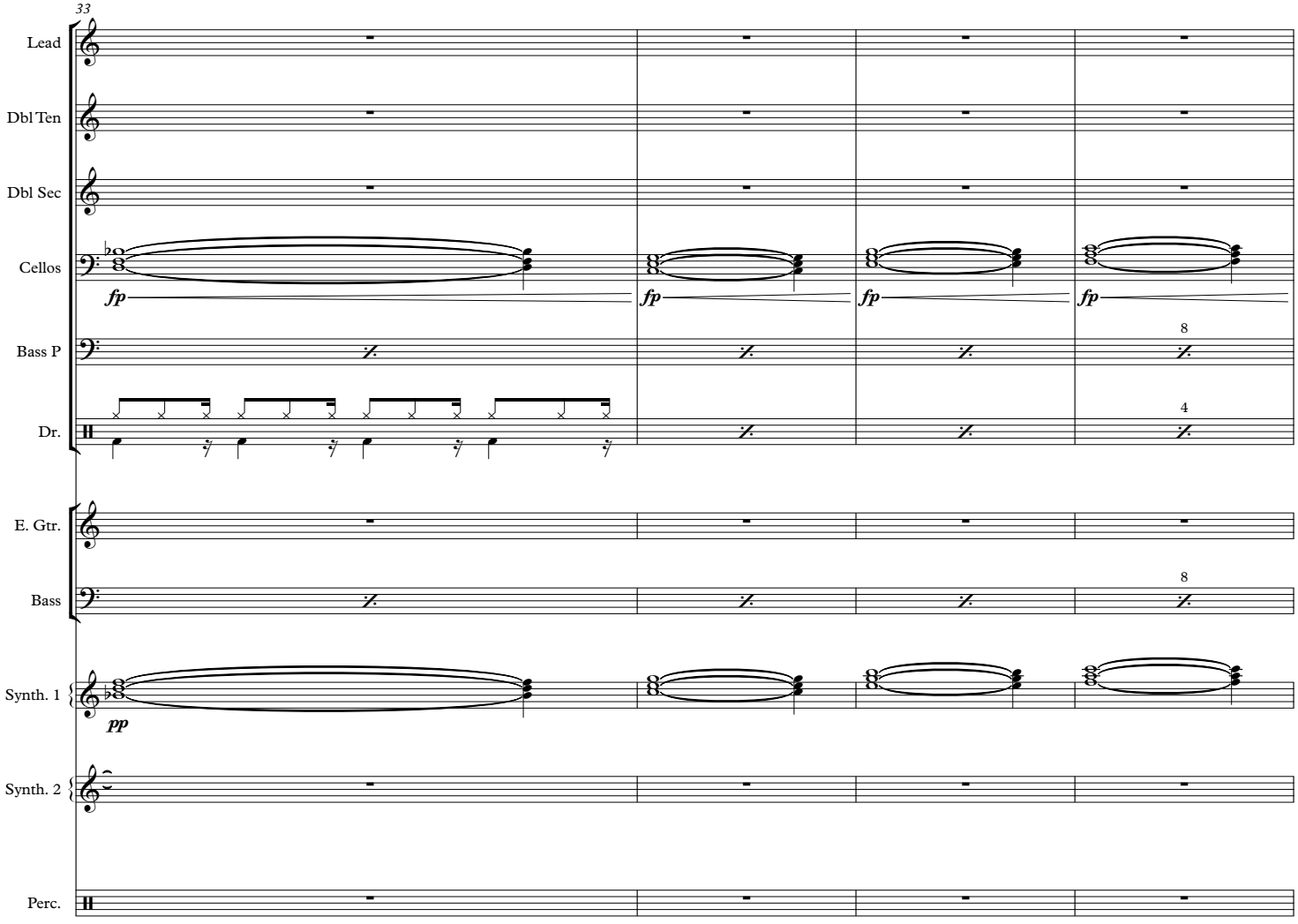

8
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
pp
p
ppp
4
8
L L L L
Digital snare

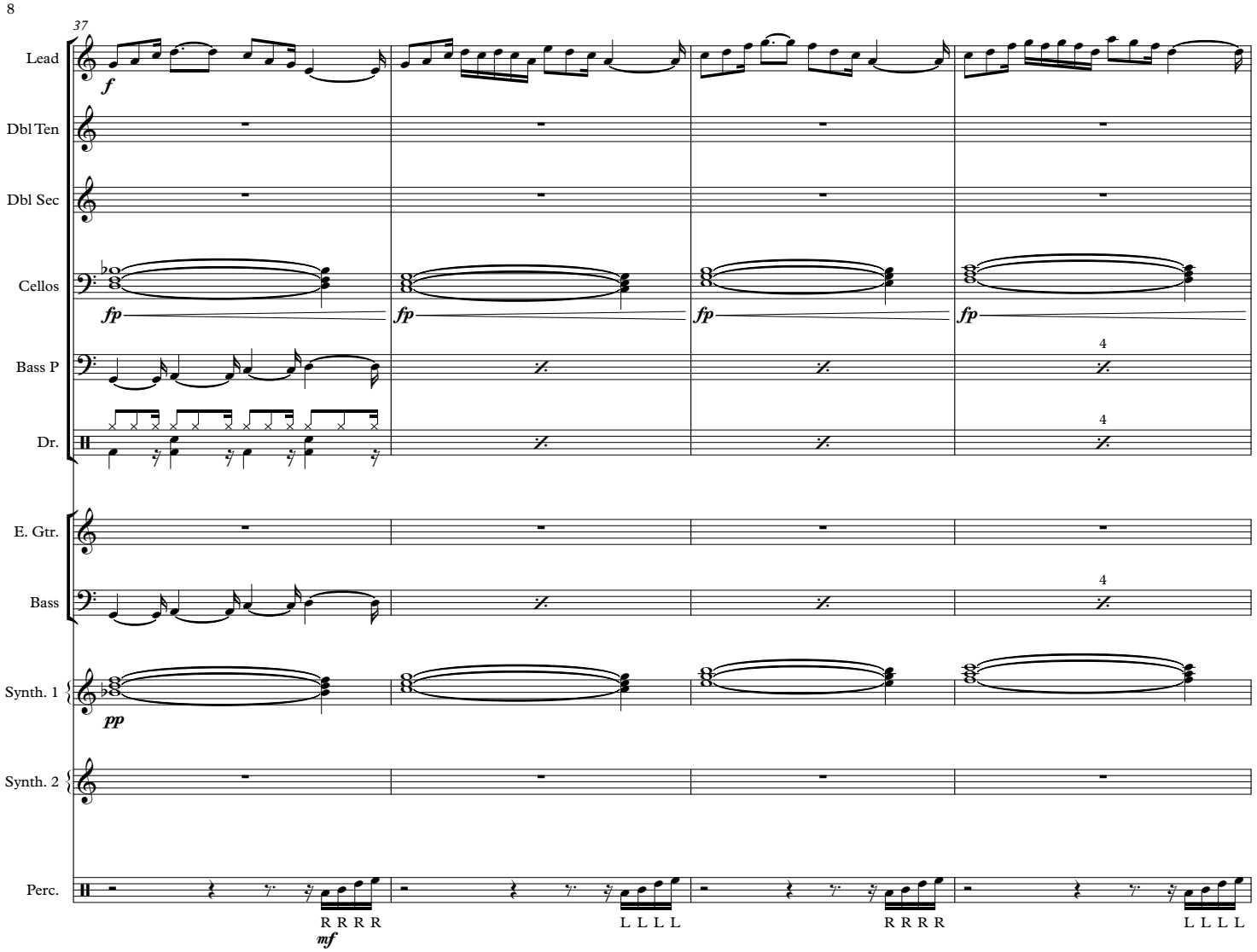

13
A
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
mf
mp
4

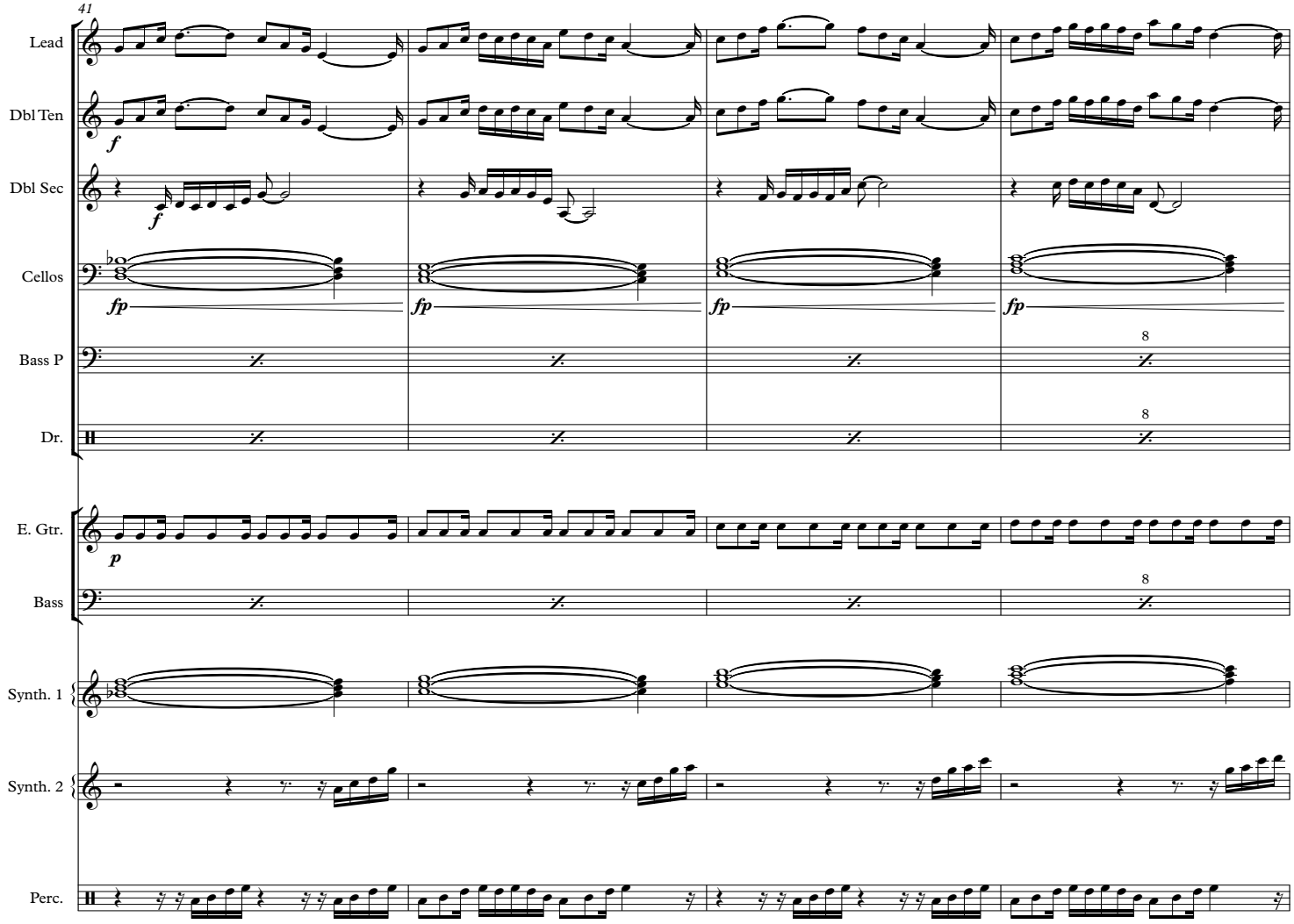

19
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
8
4
f
Digital toms

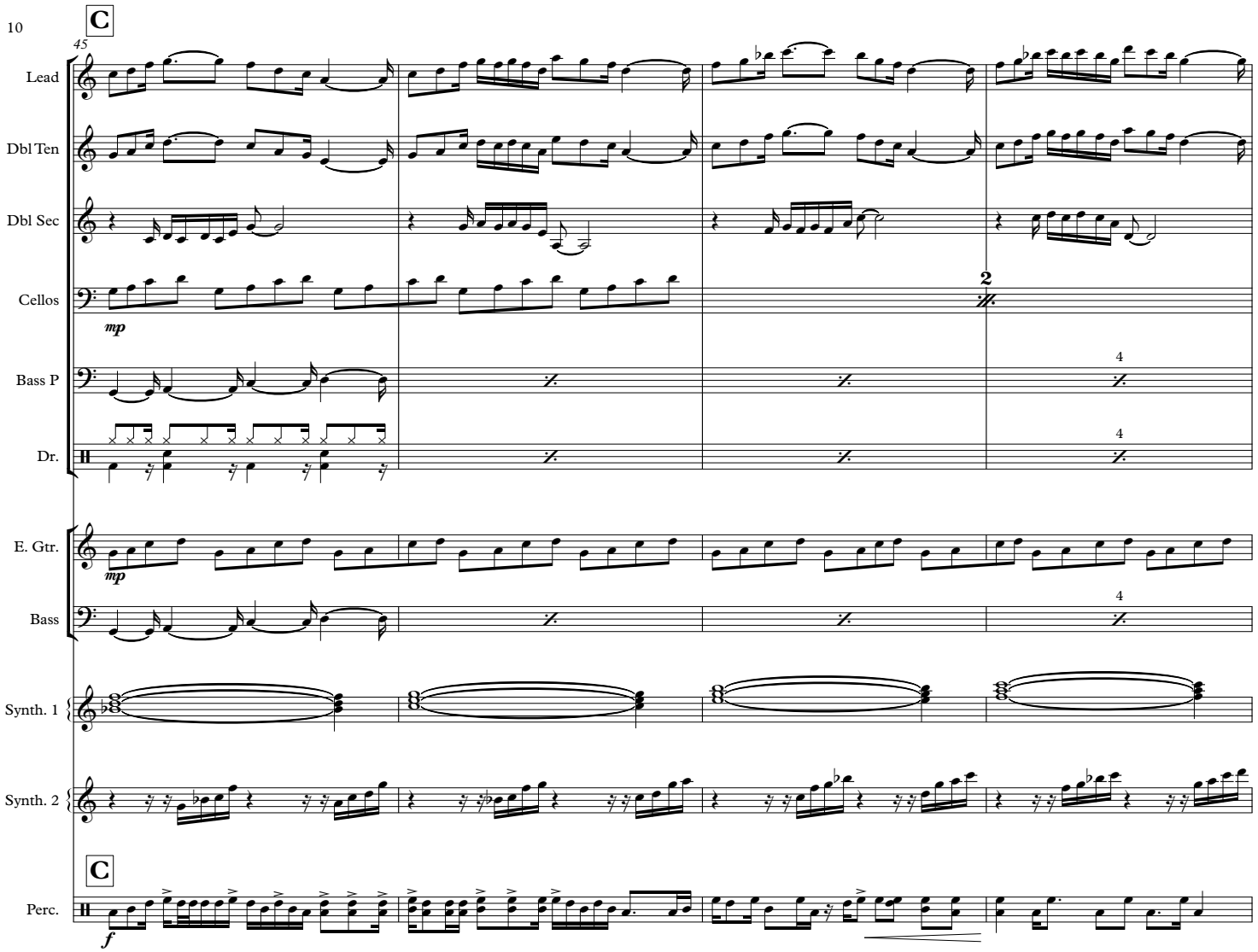

23
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
4
f
p
pp
mf

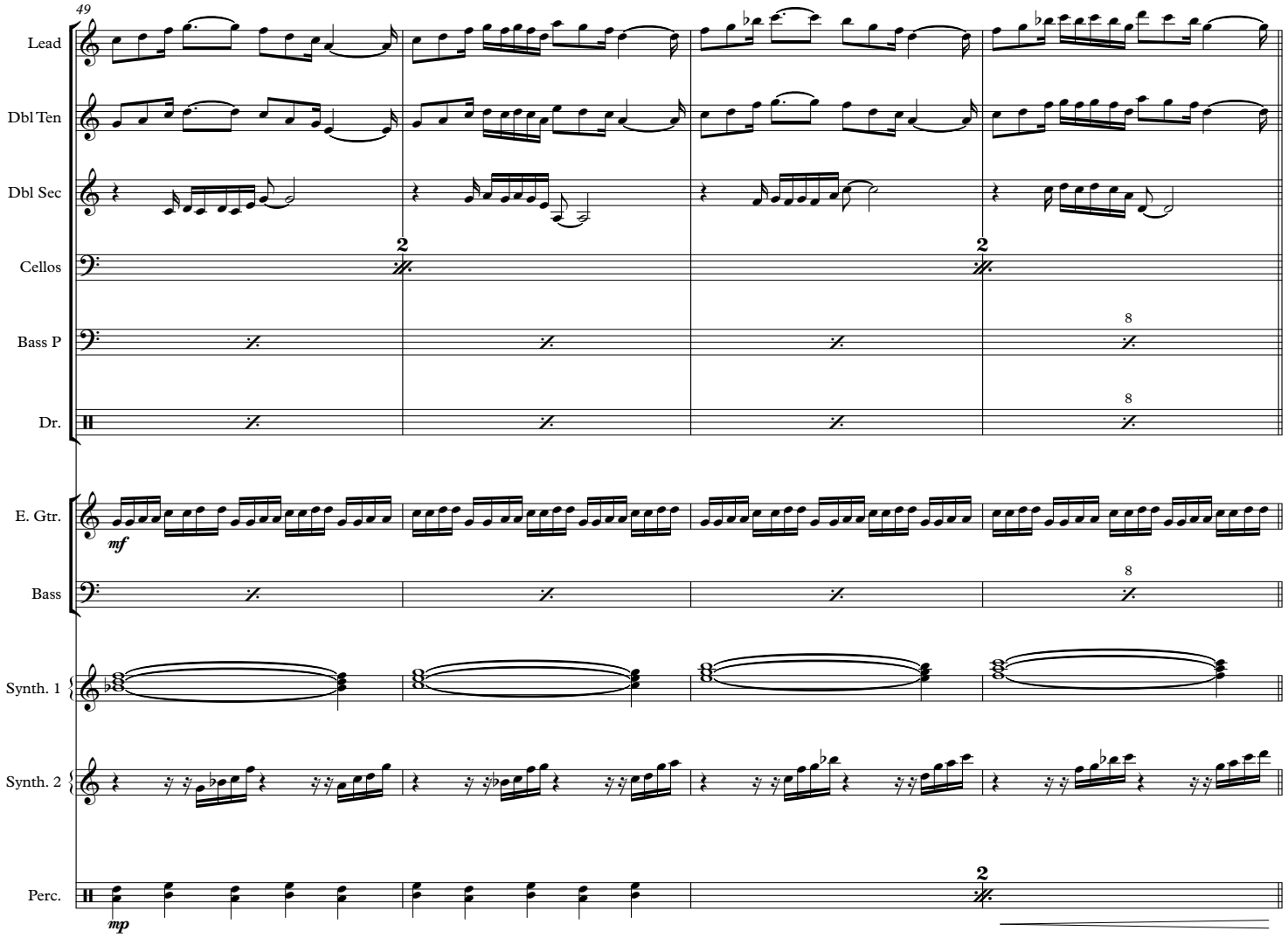

27
B
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
4
Dr.
E. Gtr.
Bass
4
Synth. 1
f
Synth. 2
Dirty, distorted synth
pp
ff
Digital gong with long decay
Perc.

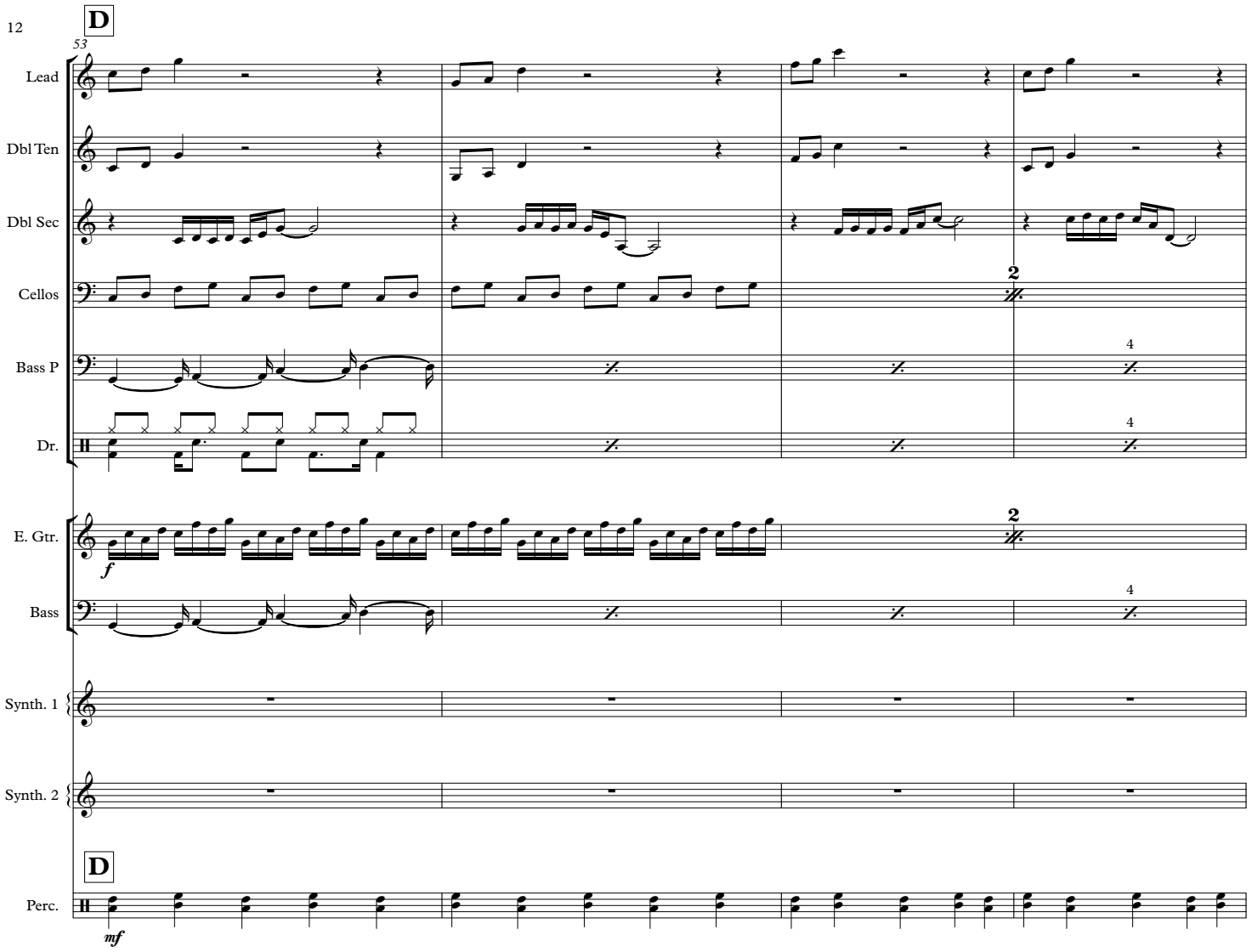

33
Lead
Dbl Ten
Dbl Sec
Cellos
fp
fp
fp
fp
Bass P
8
Dr.
4
E. Gtr.
Bass
8
Synth. 1
pp
Synth. 2
Perc.

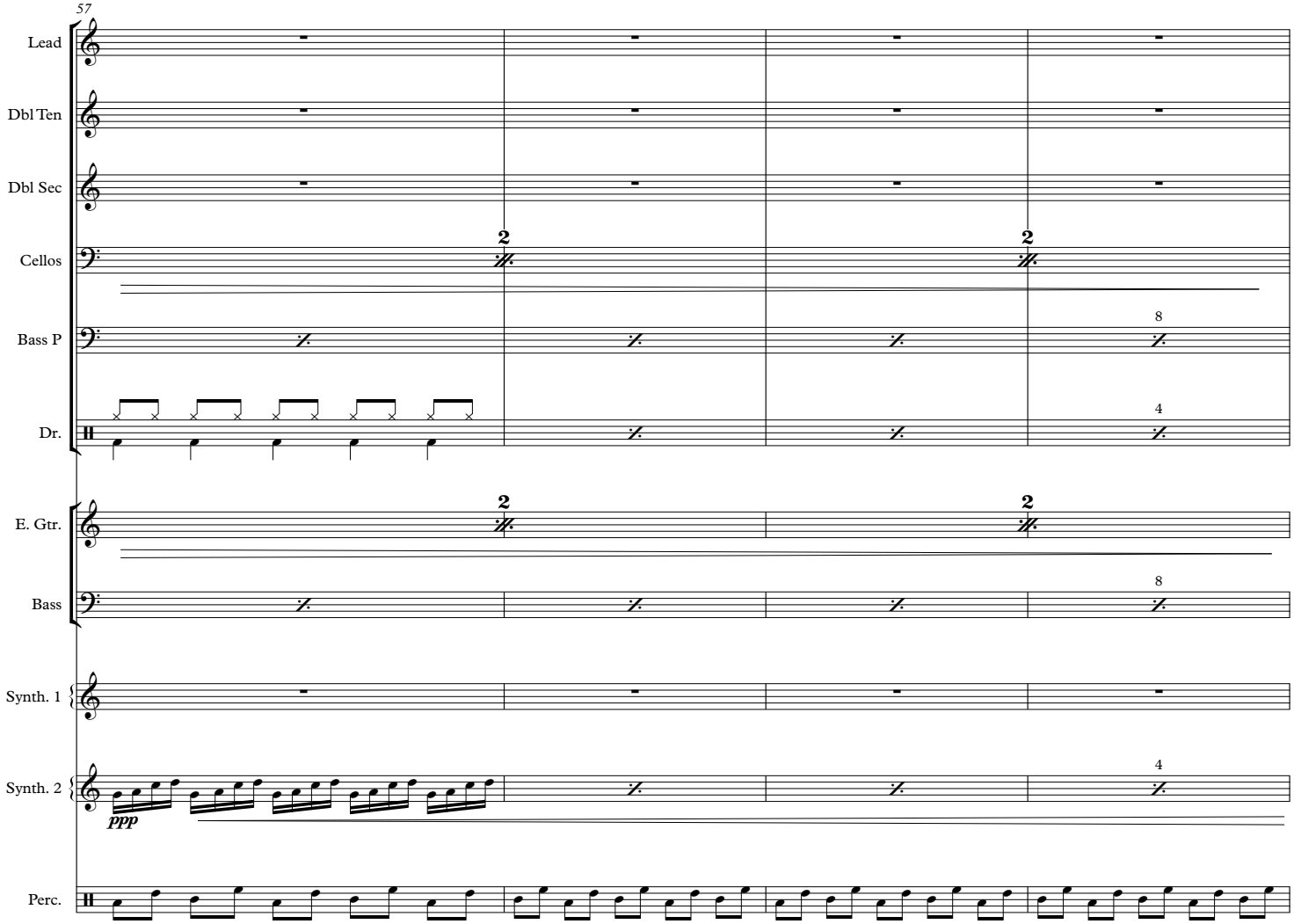

37
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
f
fp
pp
mf
4
R R R R
L L L L
R R R R
L L L L

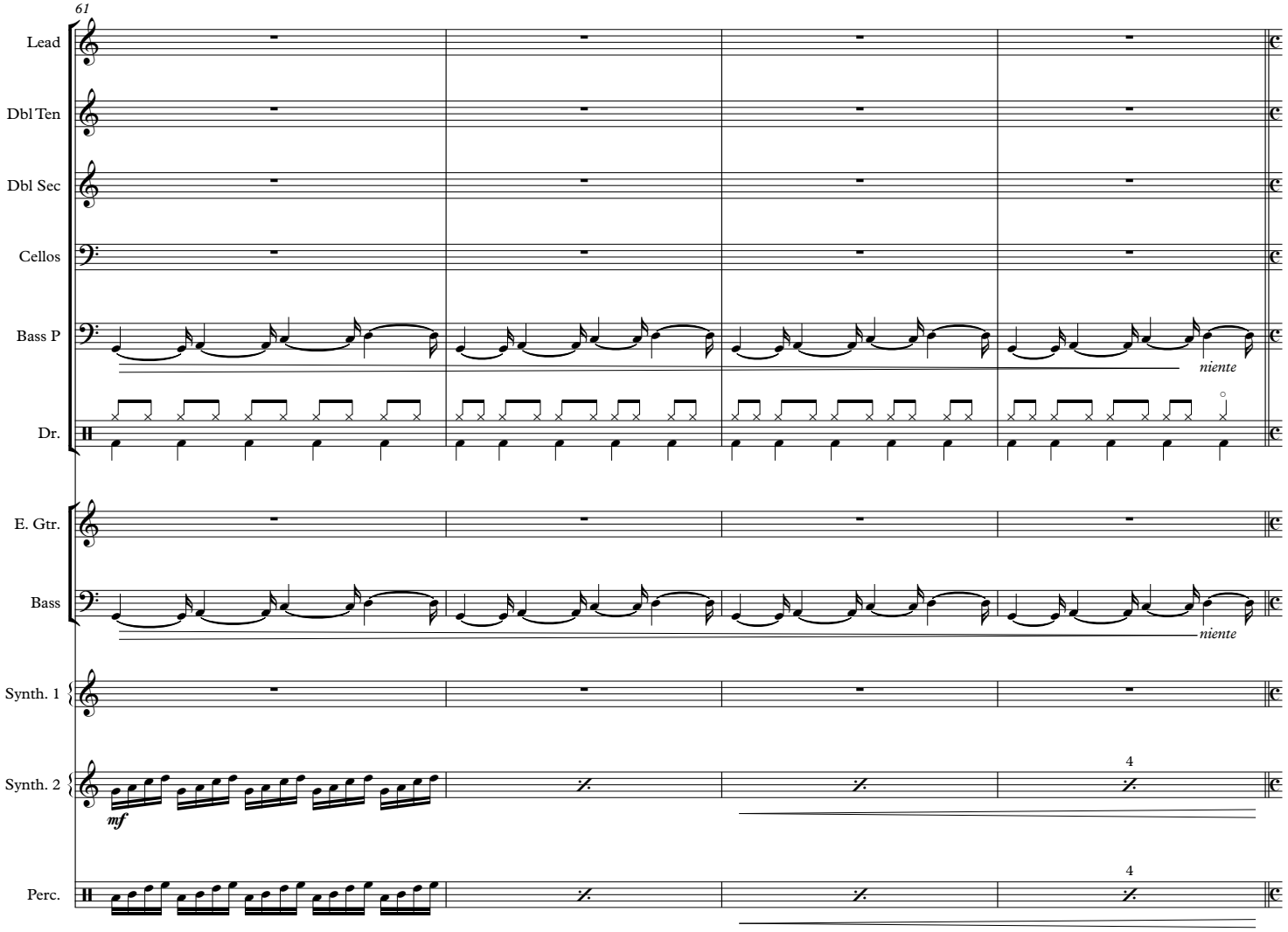

41
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
f
fp
p
8

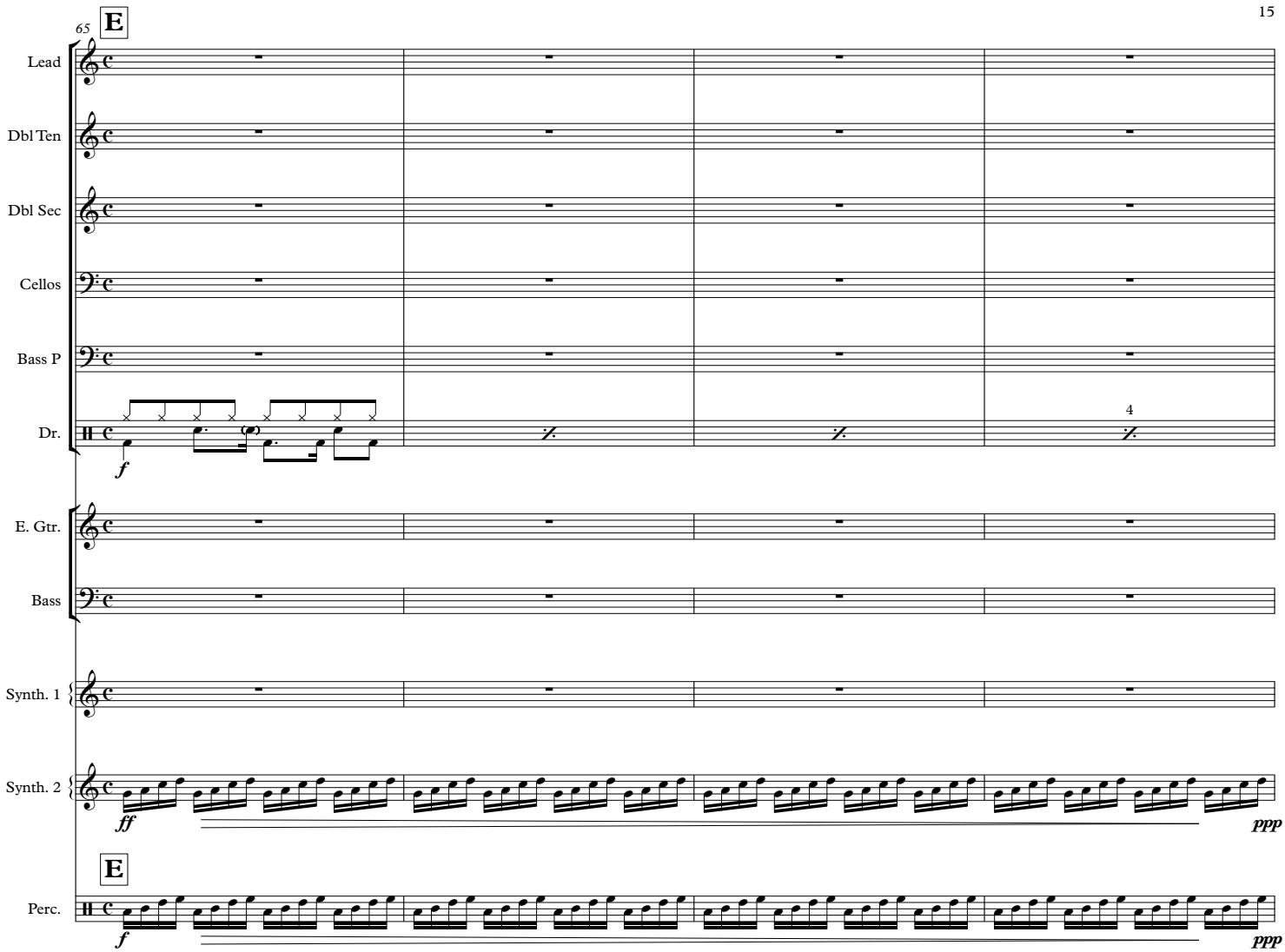

10
C
45
Lead
Dbl Ten
Dbl Sec
Cellos
mp
2
Bass P
4
Dr.
4
E. Gtr.
mp
Bass
4
Synth. 1
Synth. 2
C
Perc.
f

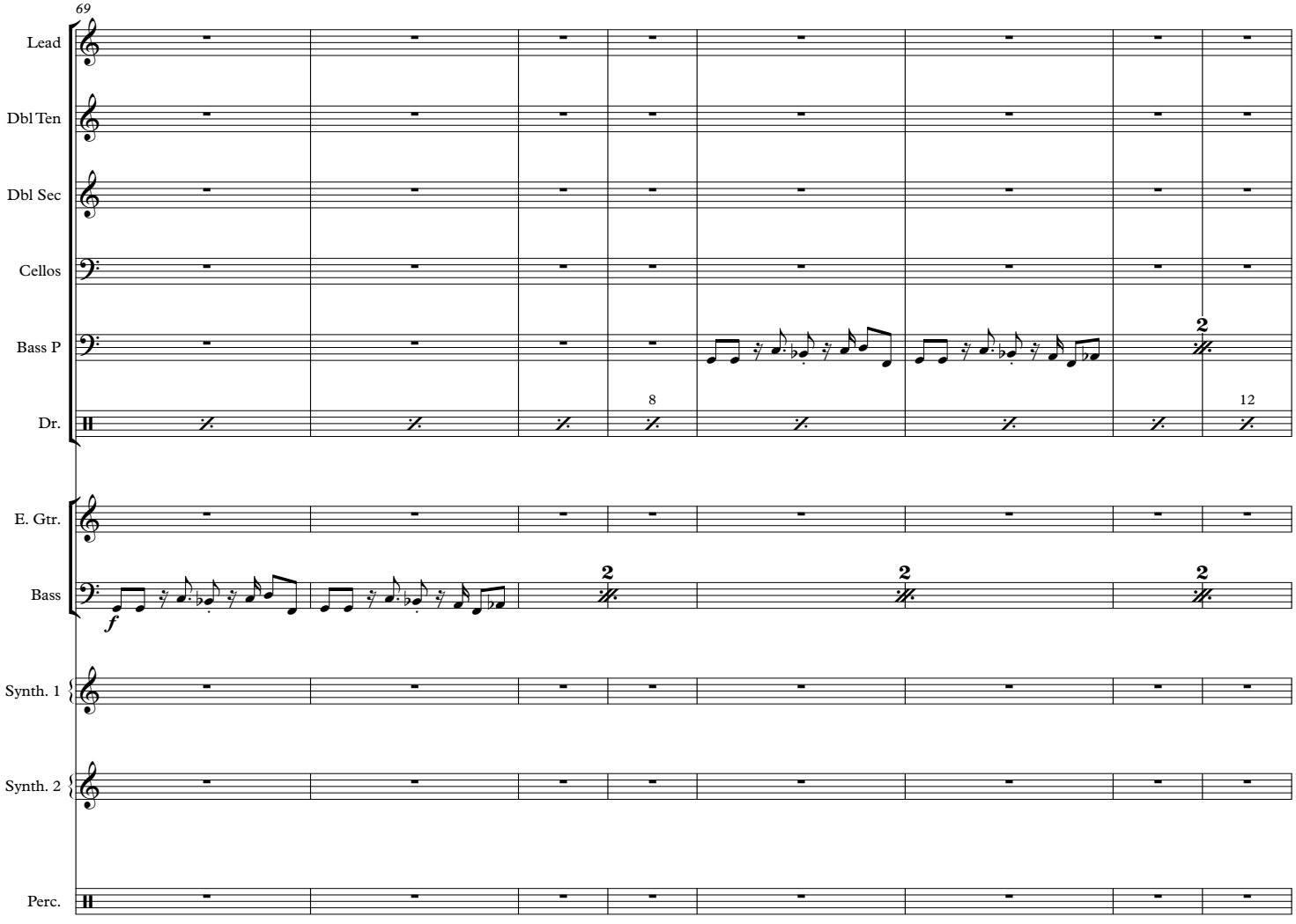

49
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
2
2
8
8
8
mf
mp

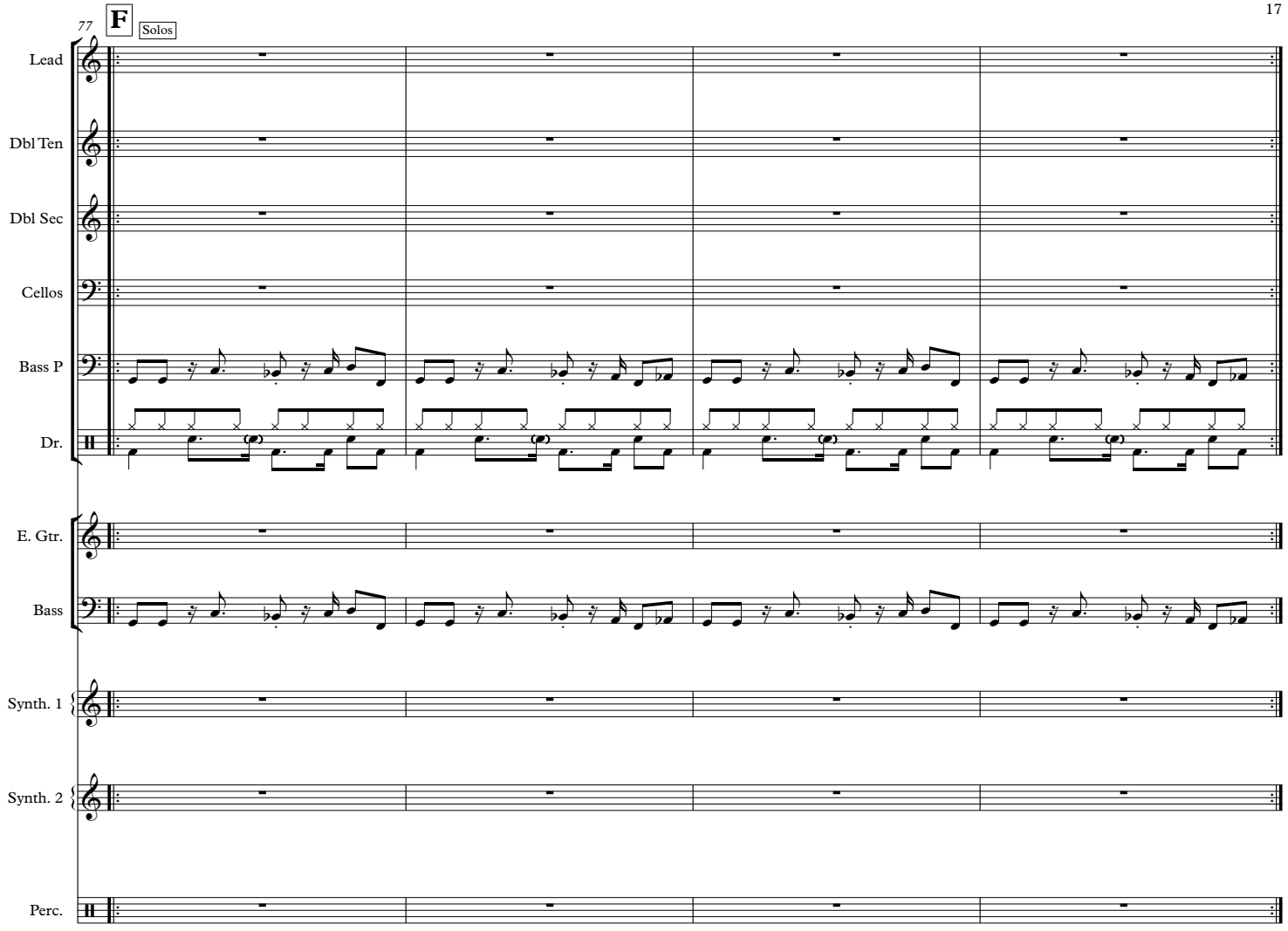

12
D
53
Lead
Dbl Ten
Dbl Sec
Cellos
2
Bass P
4
Dr.
4
E. Gtr.
f
2
Bass
4
Synth. 1
Synth. 2
D
Perc.
mf

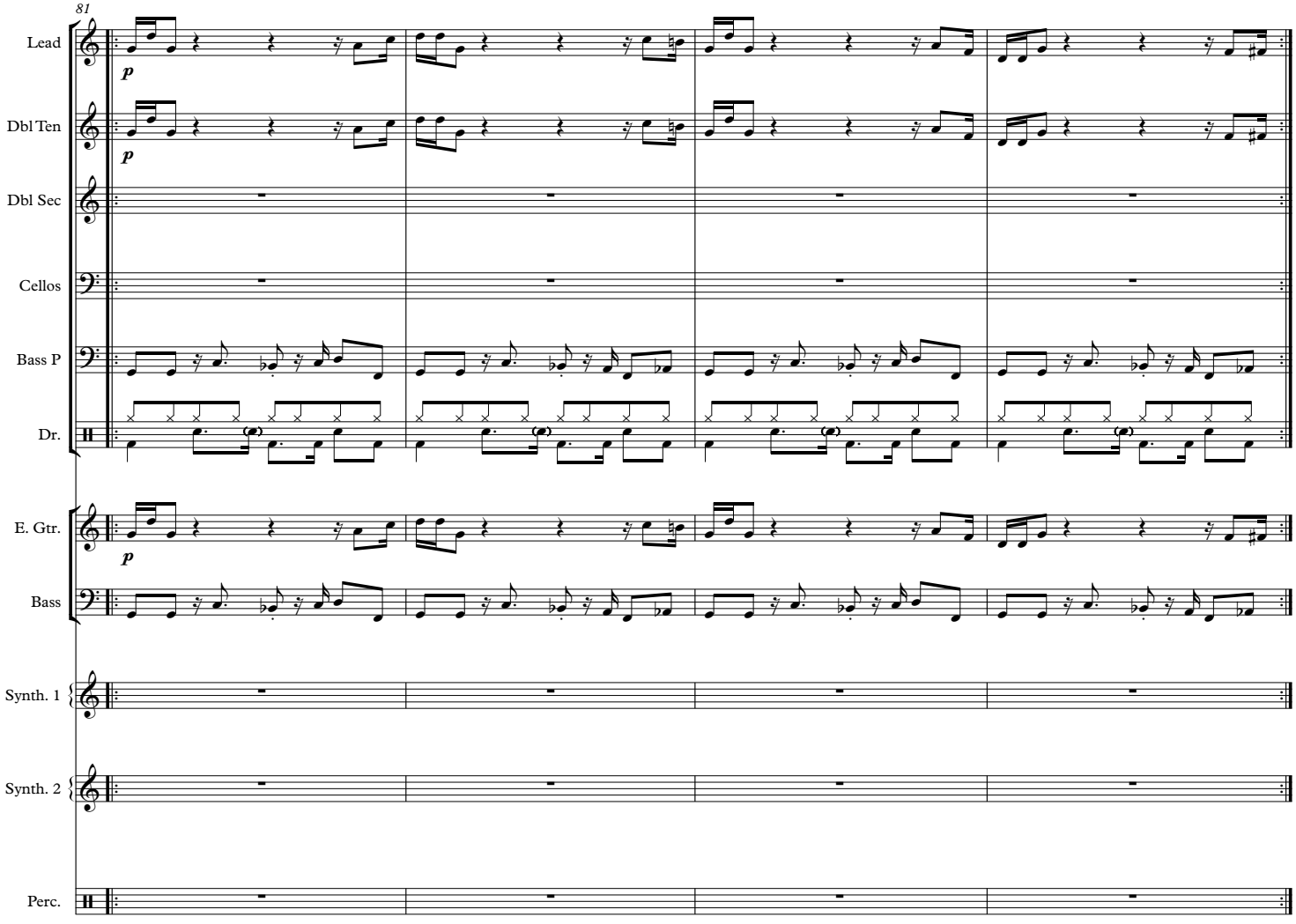

57
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
2
2
8
4
8
4
ppp

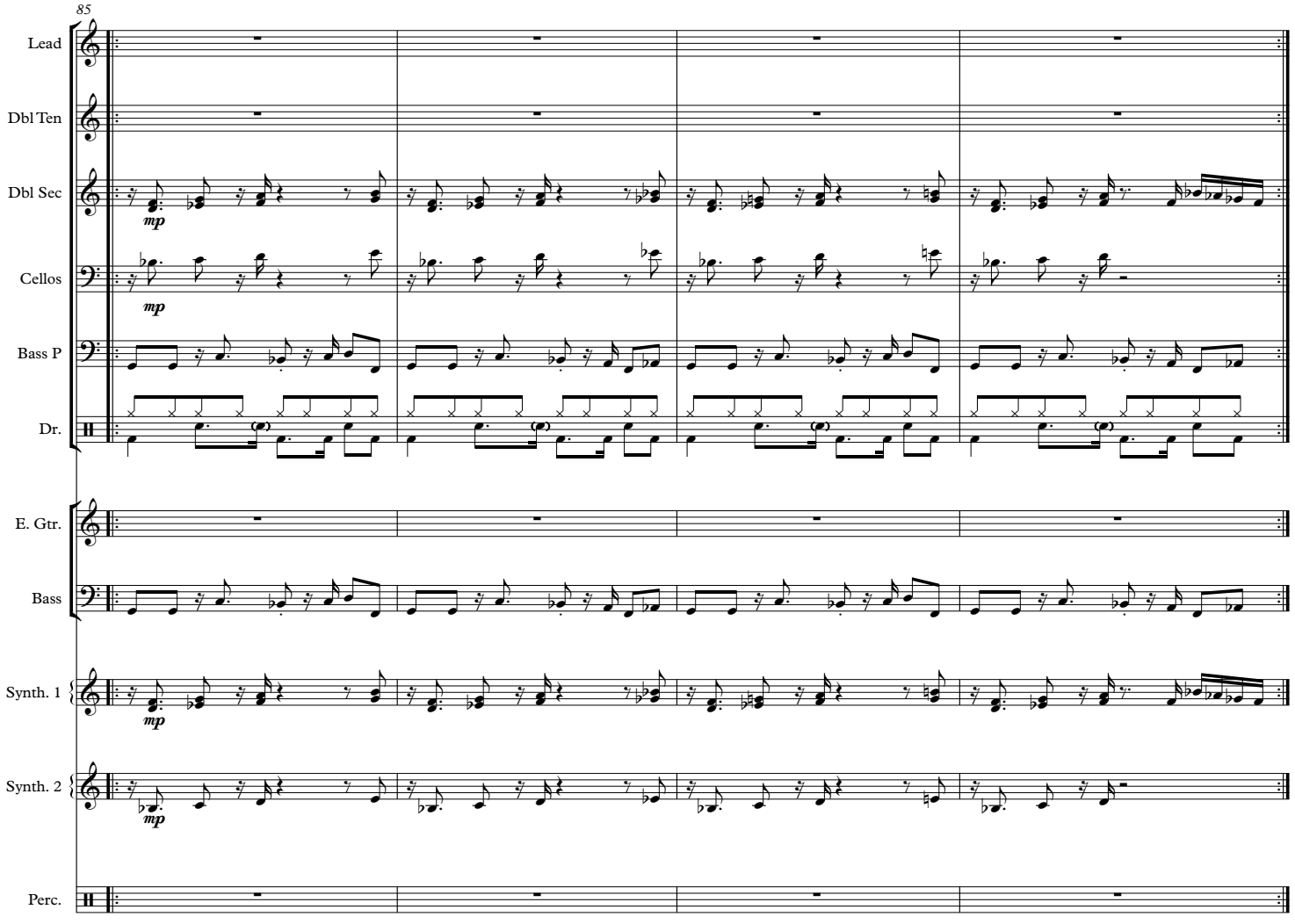

61
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
niente
Dr.
E. Gtr.
Bass
niente
Synth. 1
Synth. 2
mf
4
Perc.
4

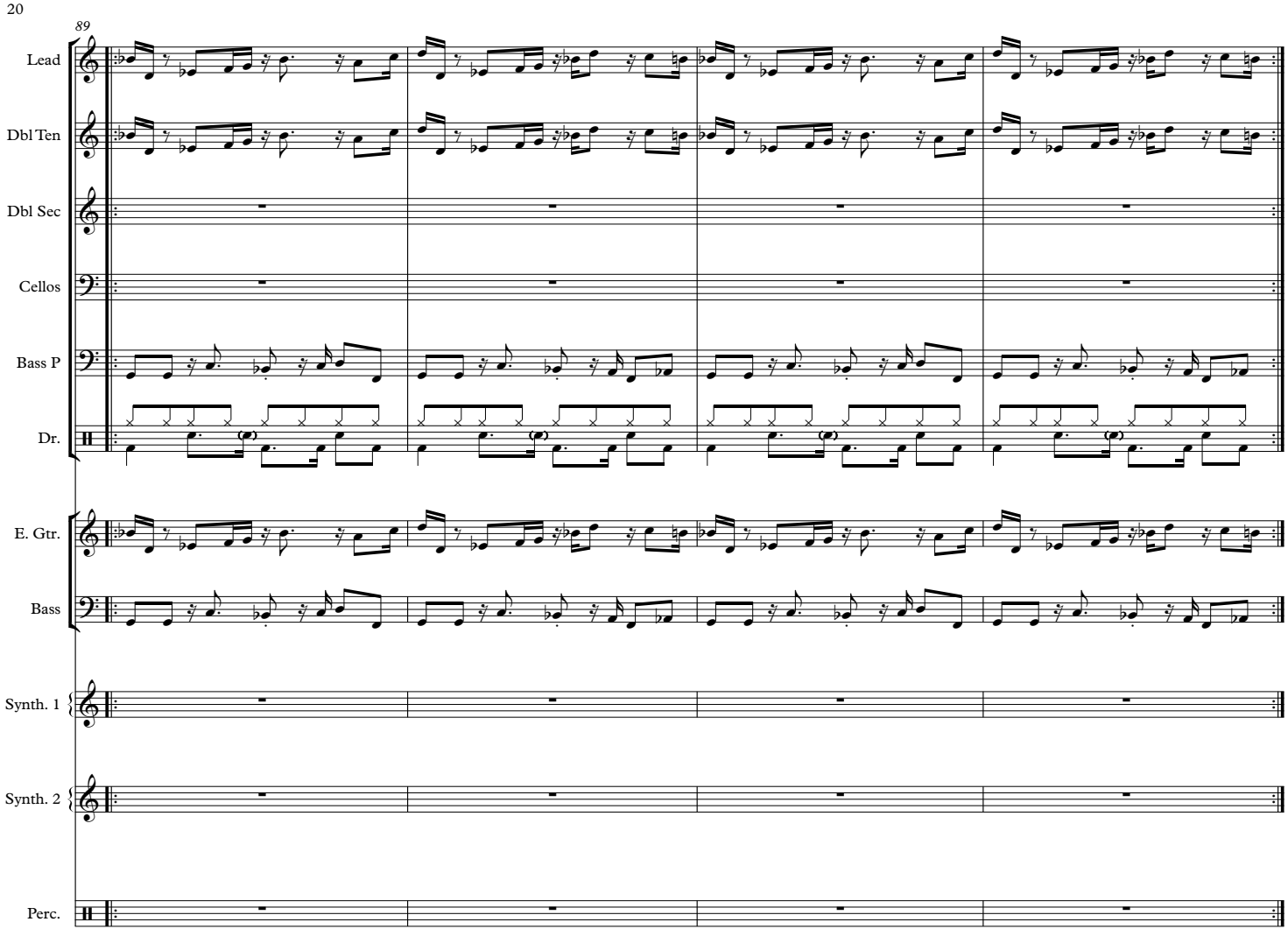

65
E
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
4
f
E. Gtr.
Bass
Synth. 1
Synth. 2
ff
ppp
E
Perc.
f
ppp

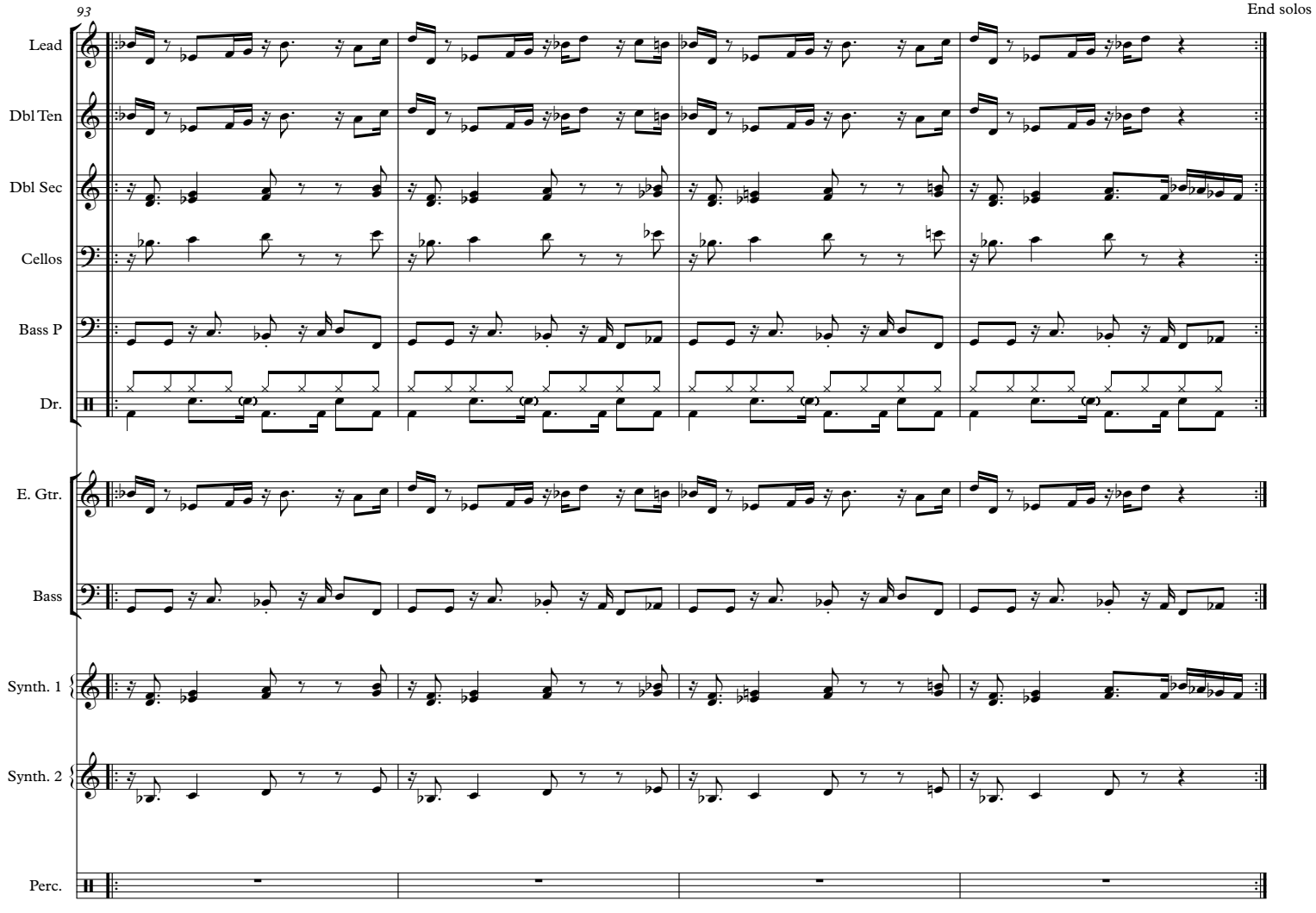

69
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
8
12
E. Gtr.
Bass
f
Synth. 1
Synth. 2
Perc.

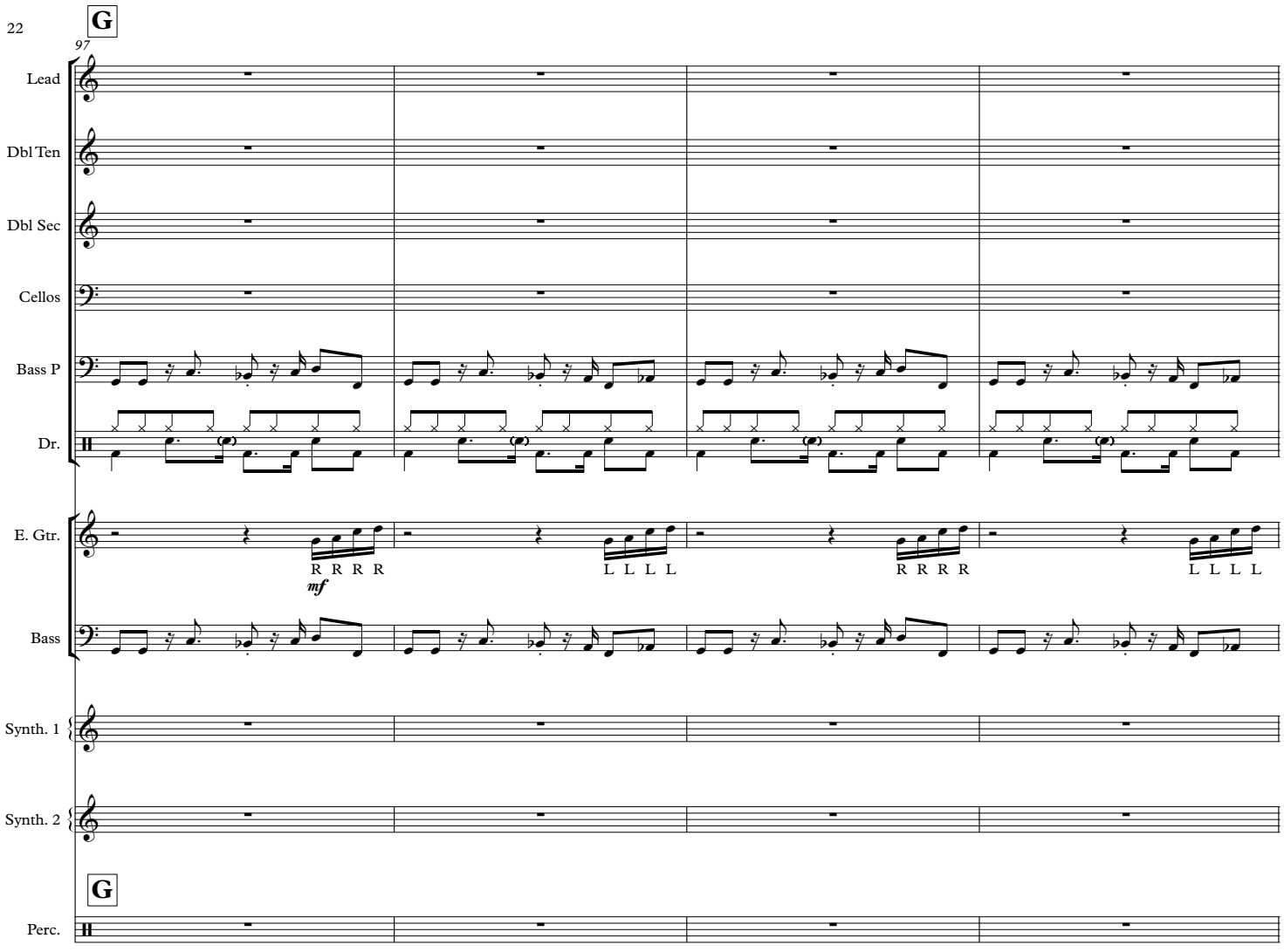

77
F
Solos
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.

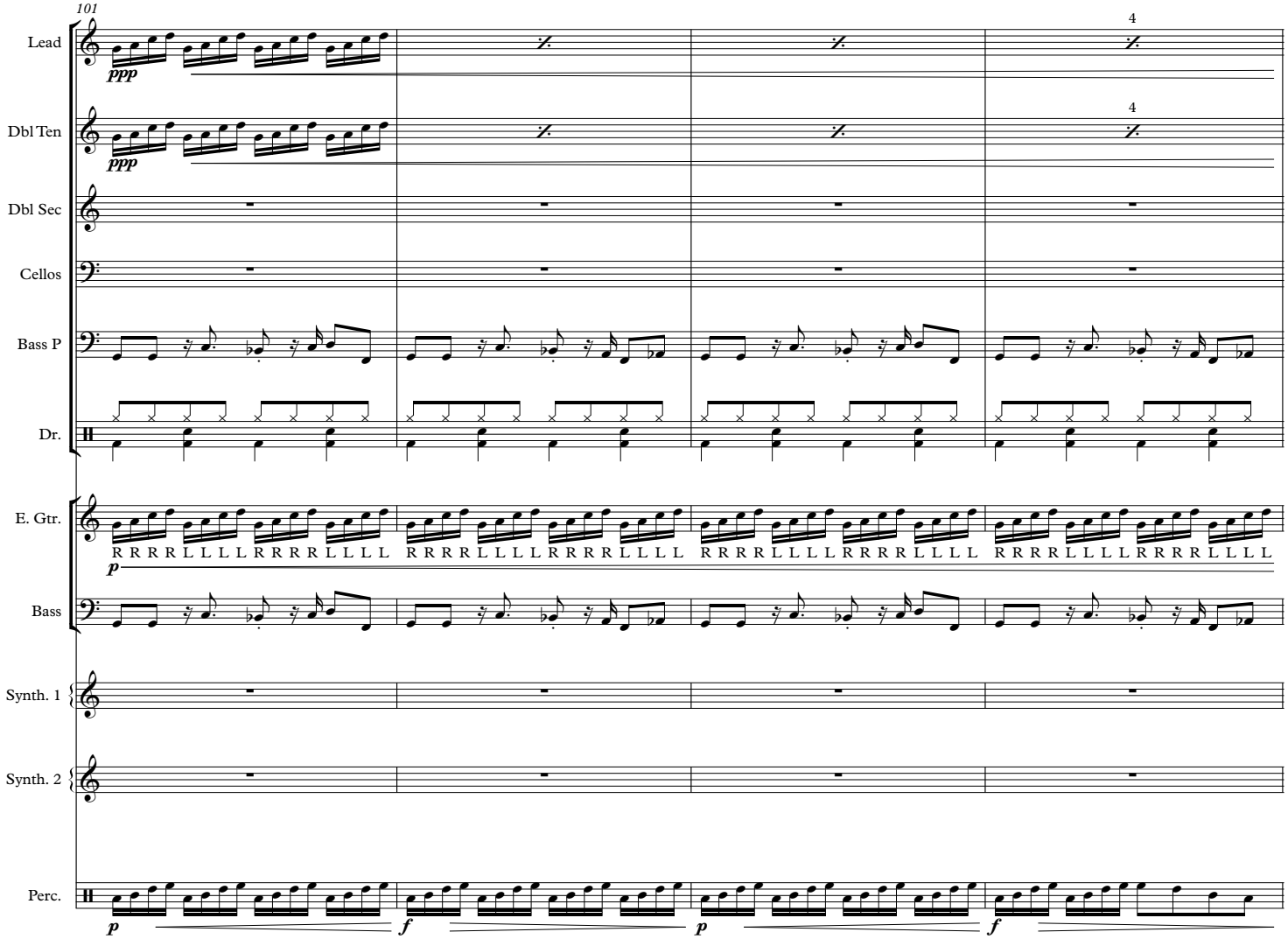

81
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
p
p
p

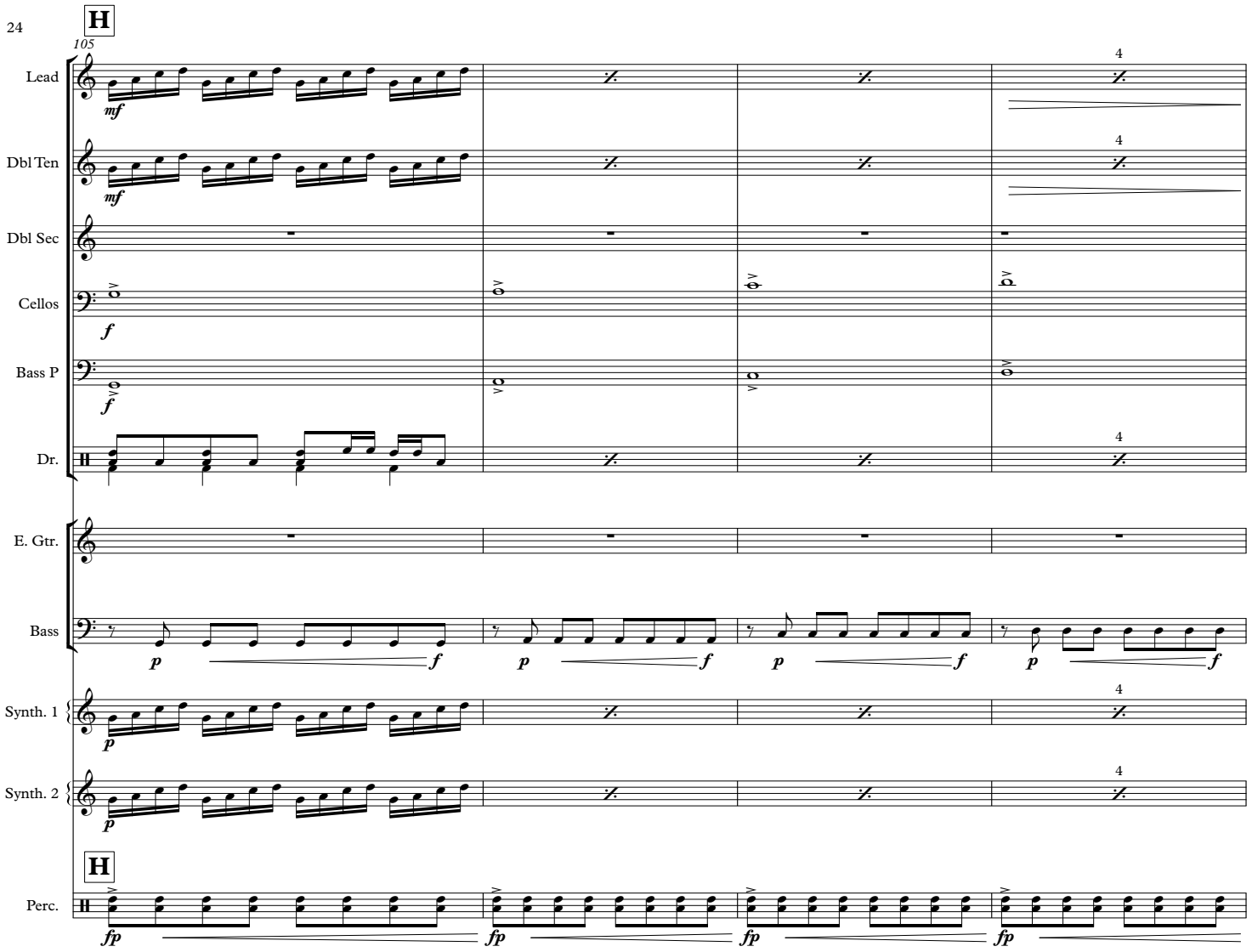

85
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
mp
mp
mp

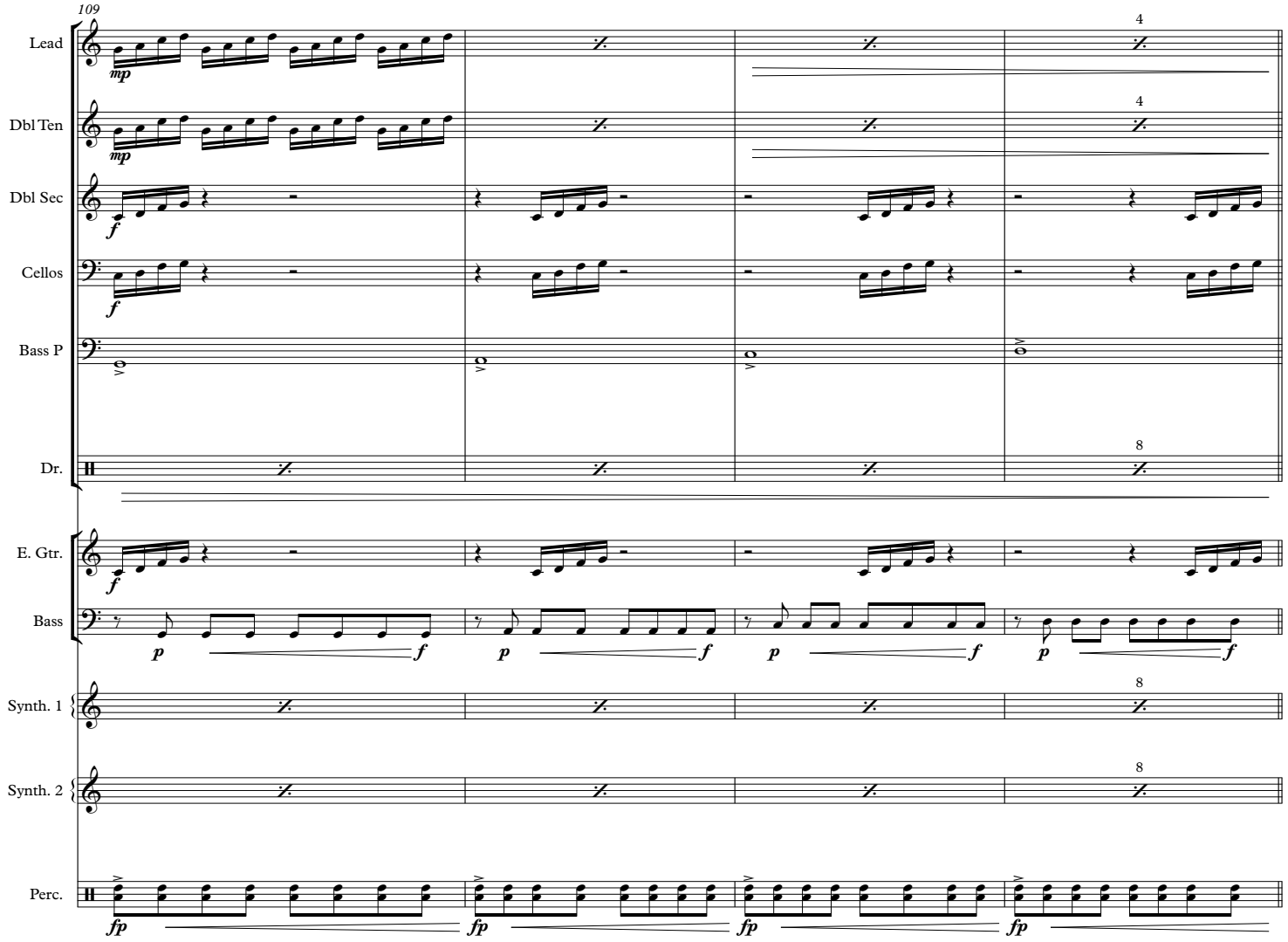

89
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.

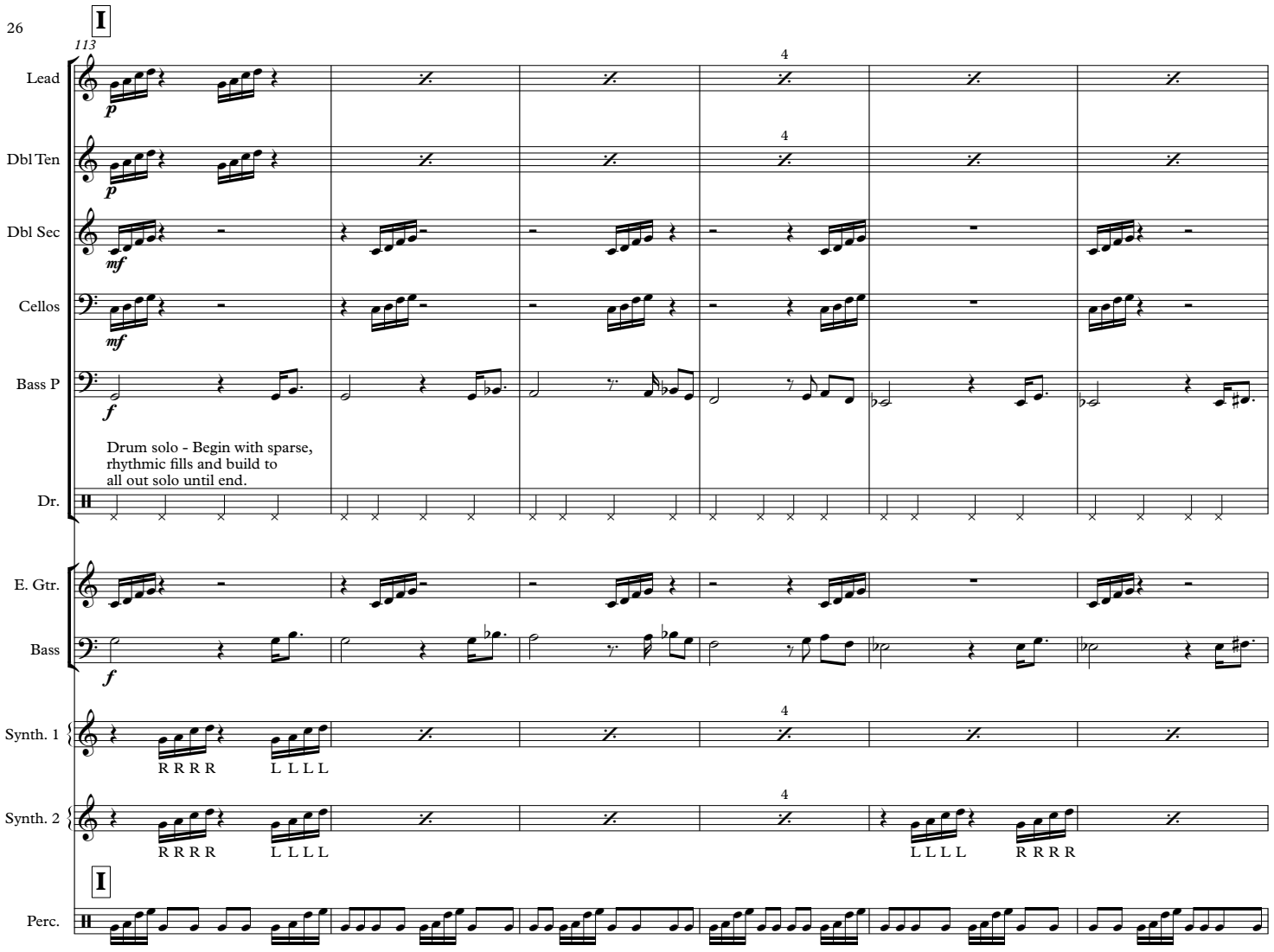

93
21
End solos
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.

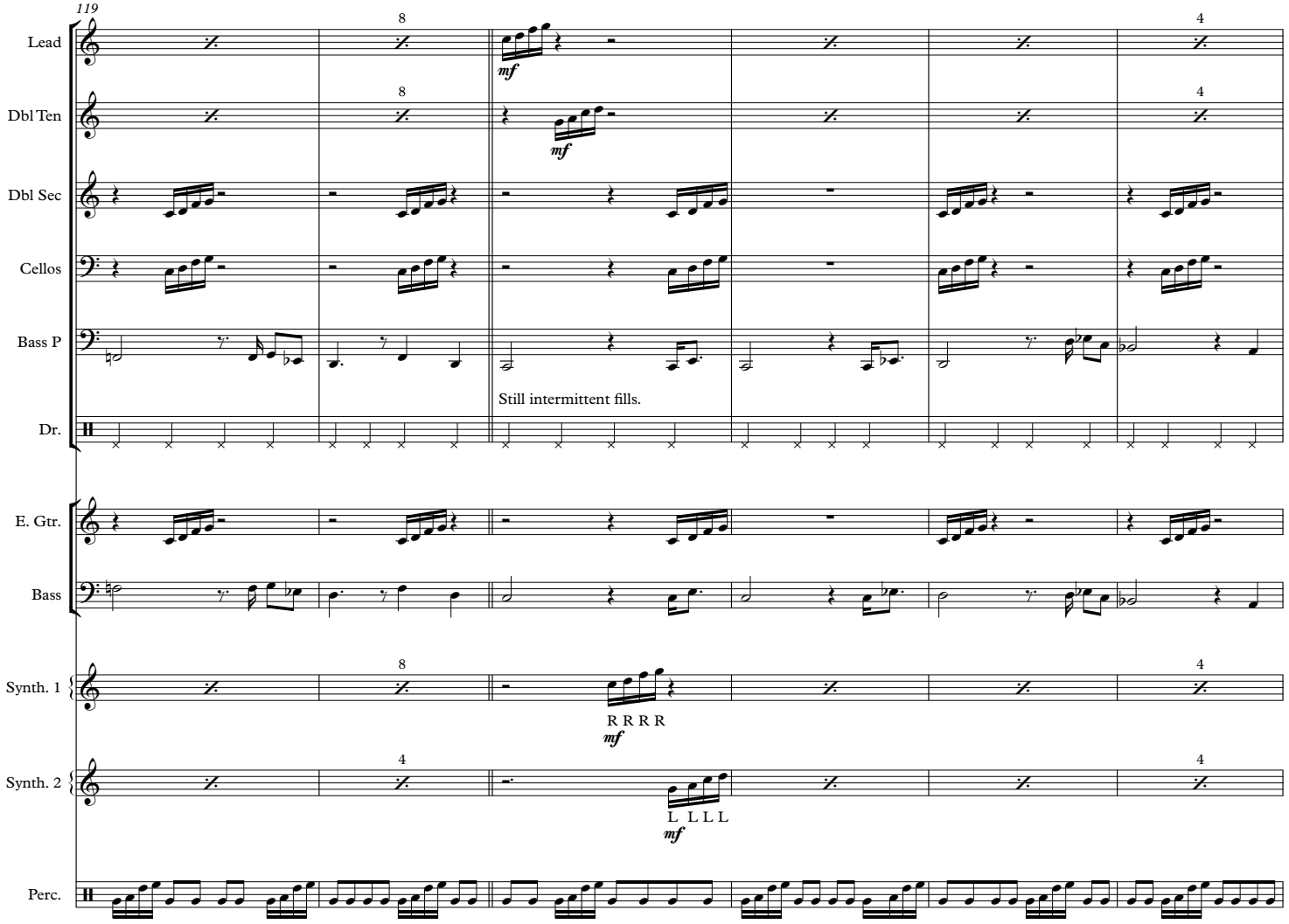

22
G
97
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
R R R R
mf
L L L L
R R R R
L L L L
Bass
Synth. 1
Synth. 2
G
Perc.

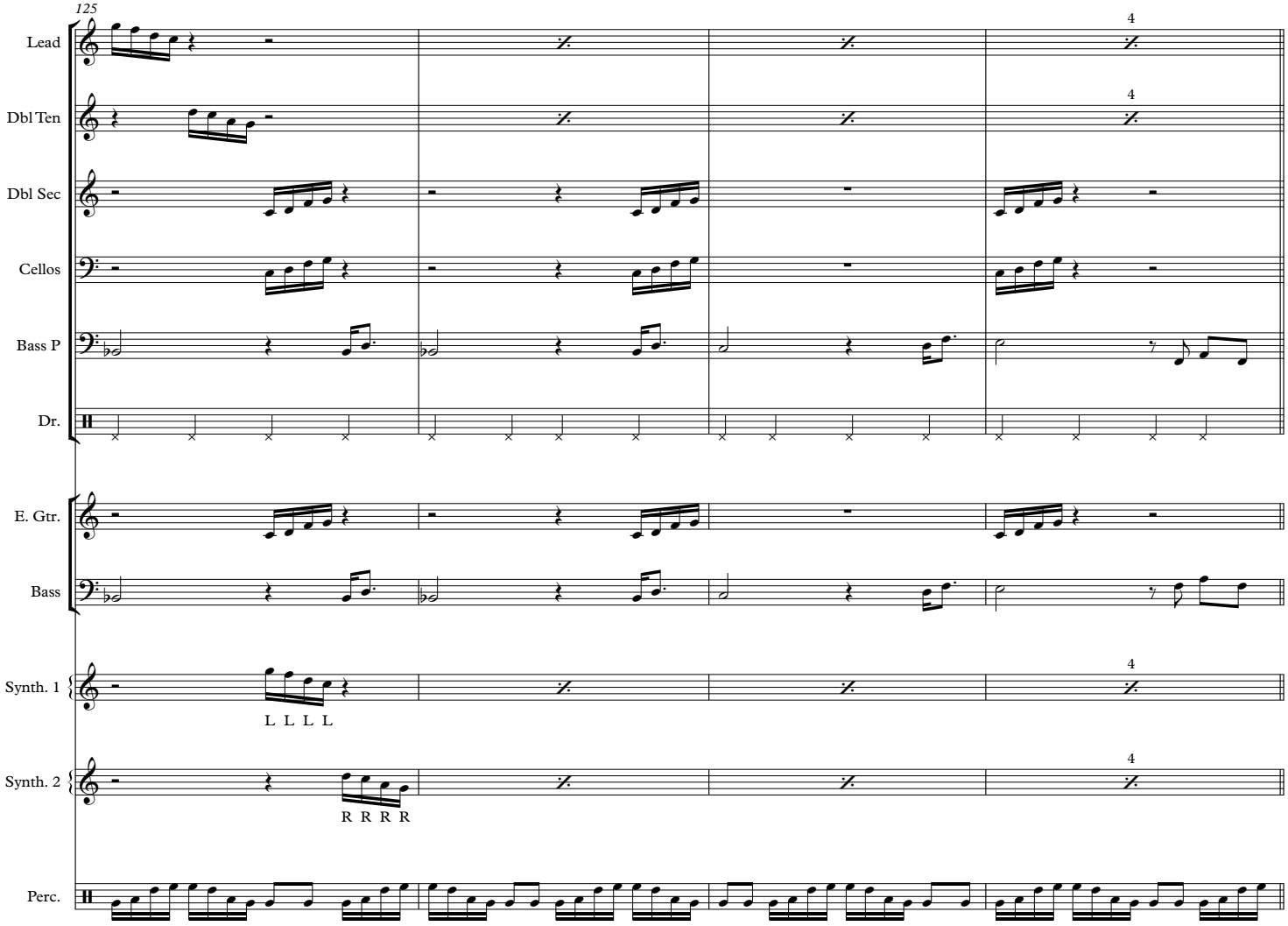

101
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
4
4
ppp
ppp
p
R R R R L L L L R R R R L L L L
f
p
f

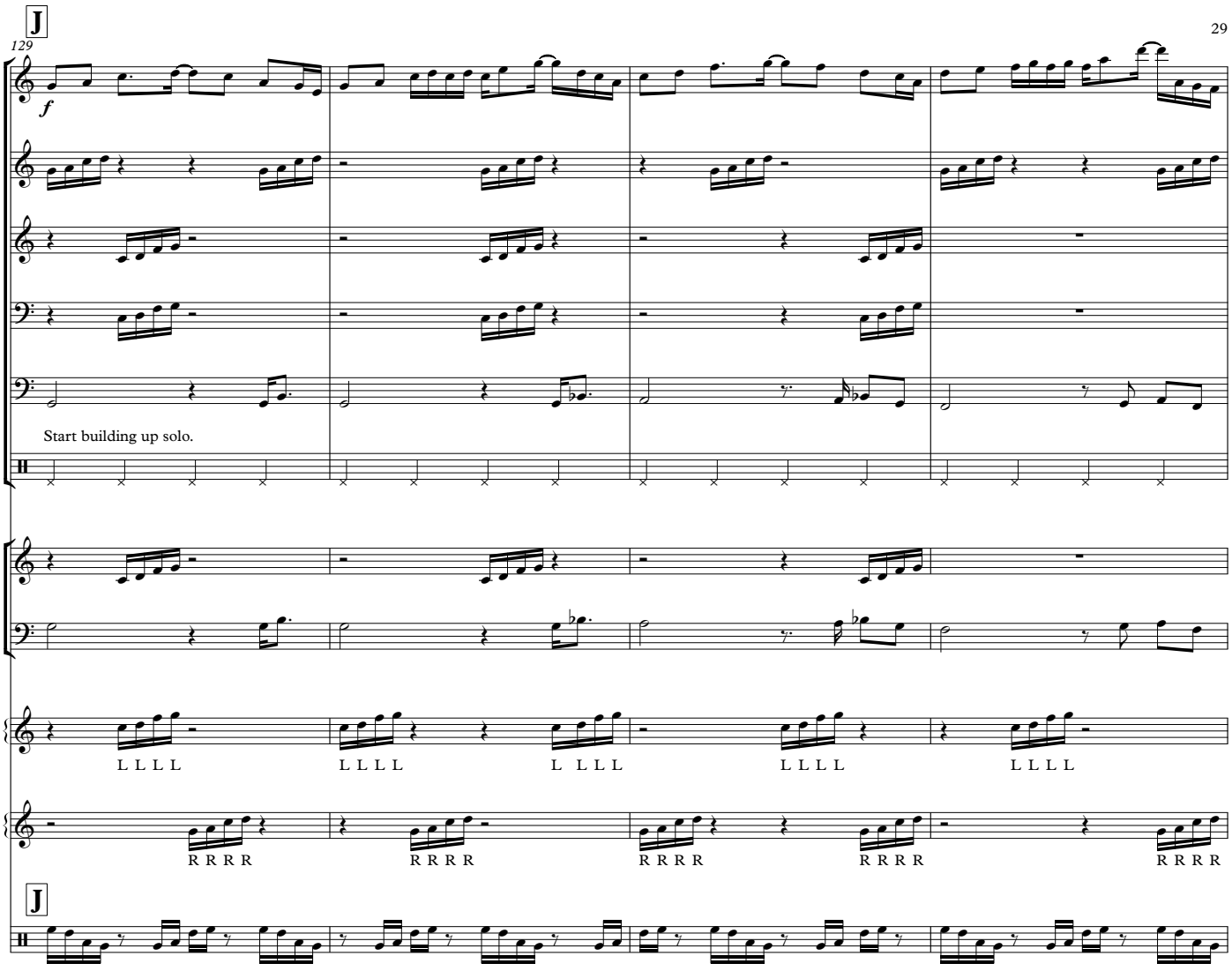

24
H
105
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
H
Perc.
mf
f
p
fp
4

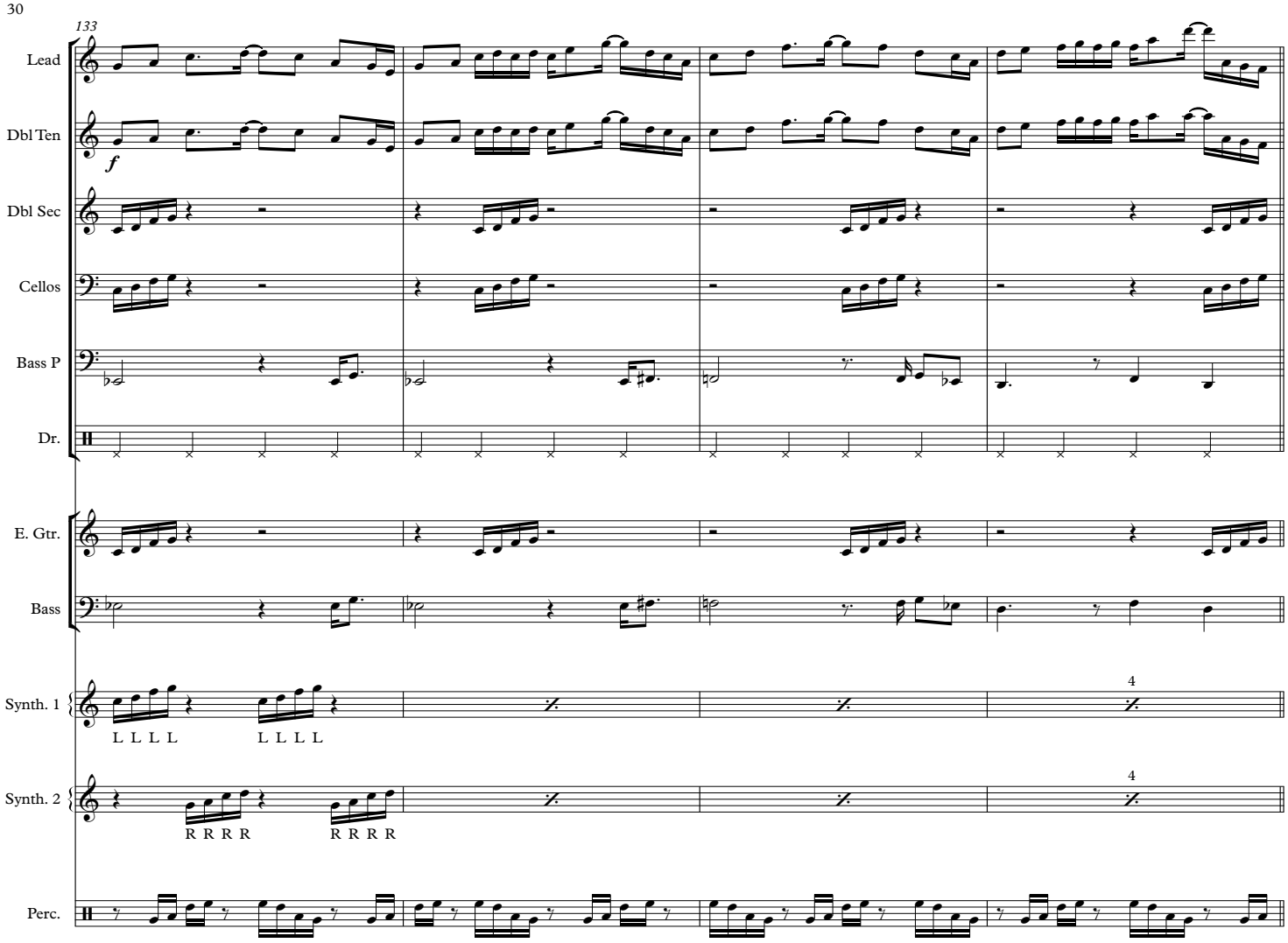

109
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
mp
f
p
fp
4
8

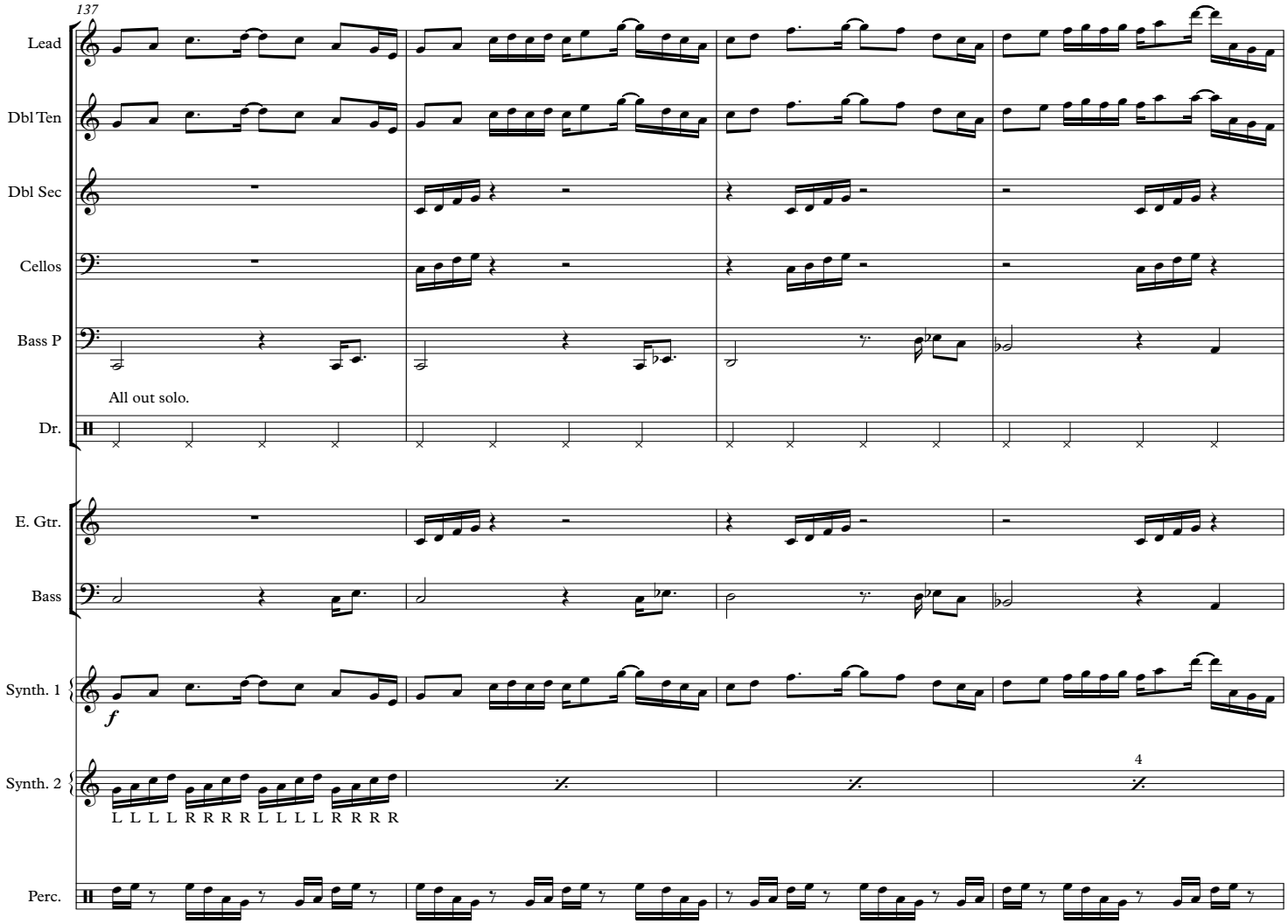

26
I
113
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Drum solo - Begin with sparse, rhythmic fills and build to all out solo until end.
Dr.
E. Gtr.
Bass
Synth. 1
R R R R
L L L L
Synth. 2
R R R R
L L L L
L L L L
R R R R
I
Perc.

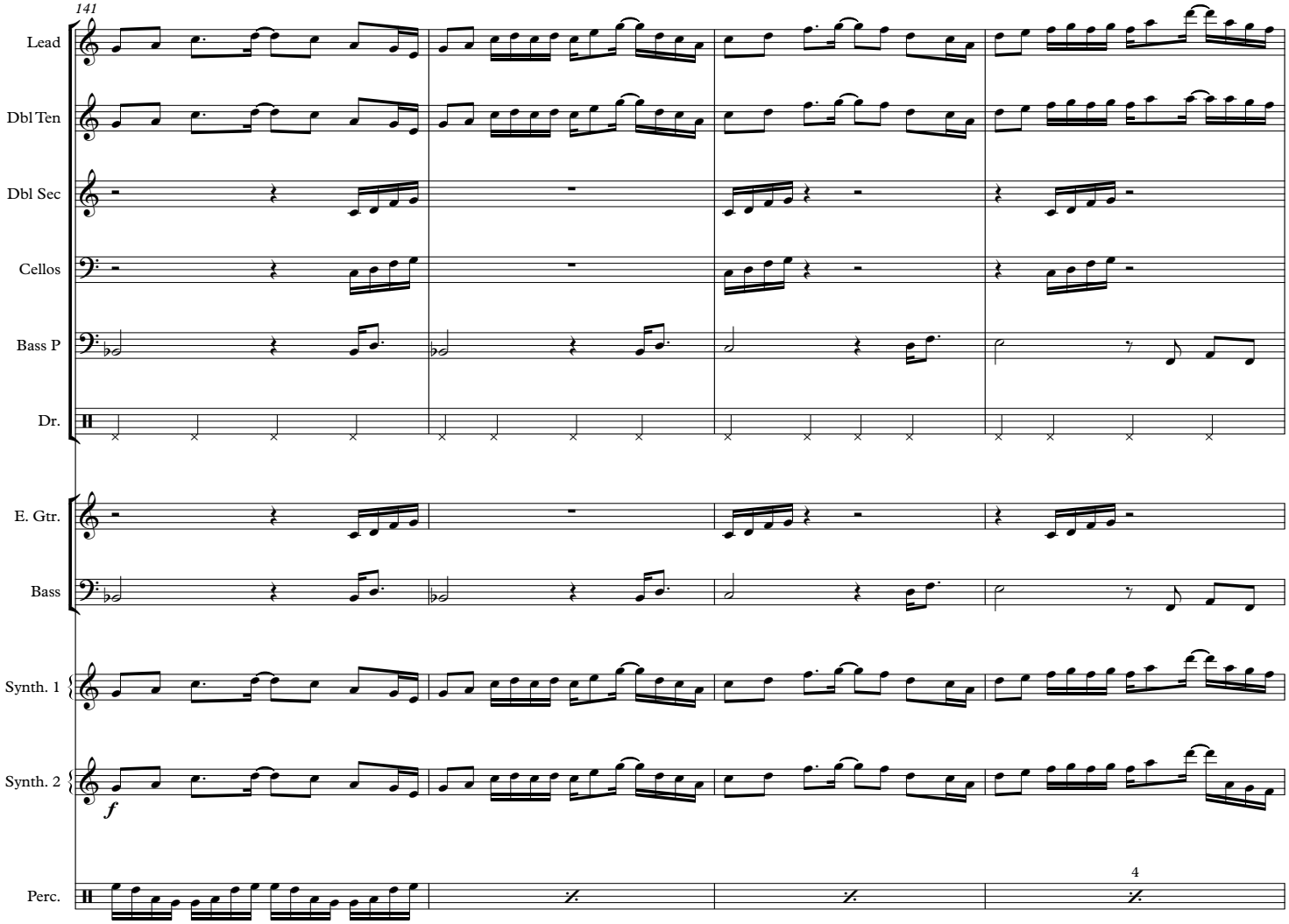

119
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
8
4
mf
Still intermittent fills.
R R R R
L L L L

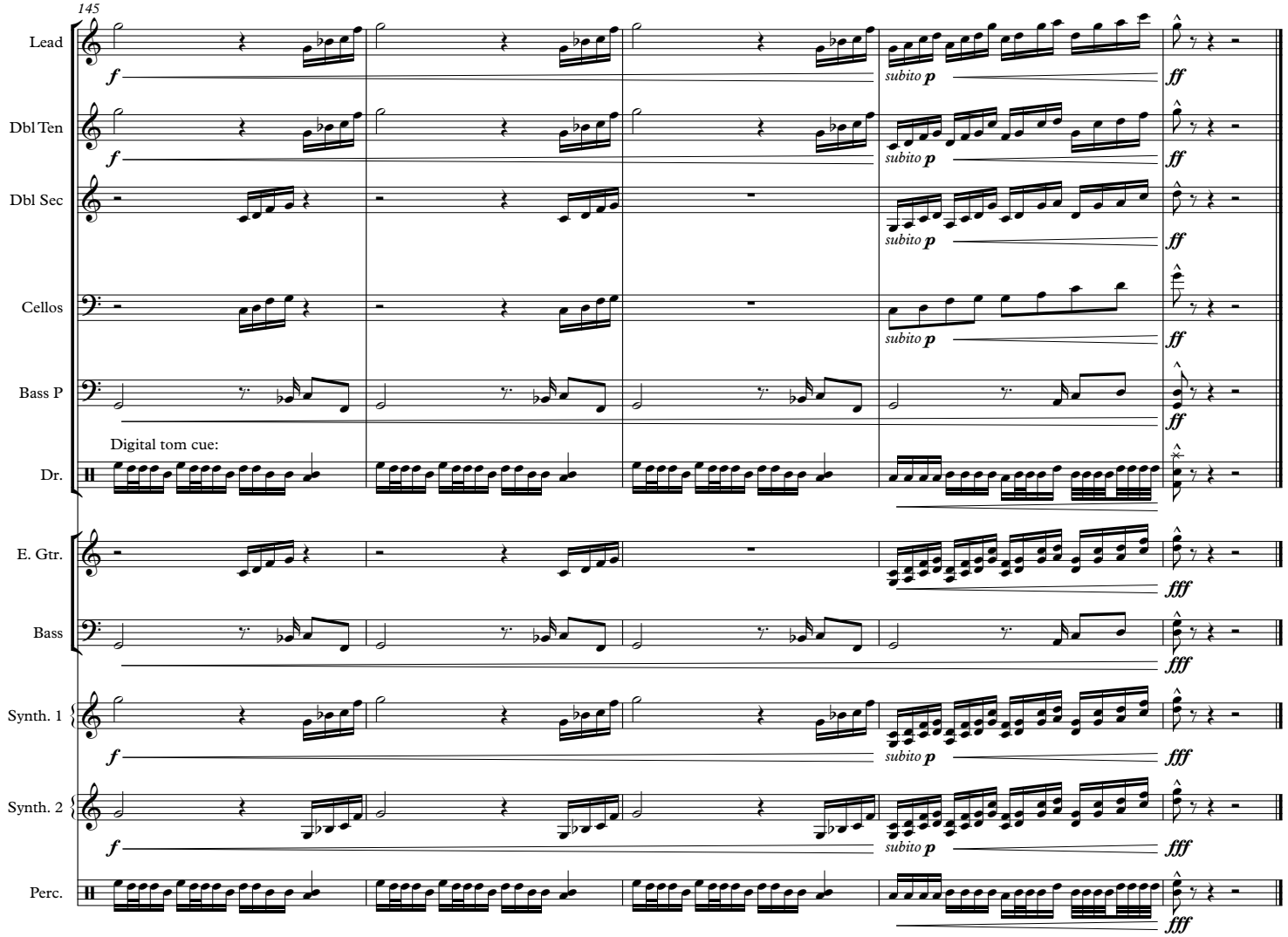

125
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
L L L L
Synth. 2
R R R R
Perc.
4


J
129
Lead
f
Dbl Ten
Dbl Sec
Cellos
Bass P
Start building up solo.
Dr.
E. Gtr.
Bass
Synth. 1
L L L L
L L L L
L L L L
L L L L
Synth. 2
R R R R
R R R R
R R R R
R R R R
J
Perc.


133
Lead
Dbl Ten
f
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
L L L L
L L L L
4
Synth. 2
R R R R
R R R R
4
Perc.


137
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
All out solo.
Dr.
E. Gtr.
Bass
Synth. 1
f
Synth. 2
4
L L L L R R R R L L L L R R R R
Perc.


141
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
f
Perc.
4


145
Lead
Dbl Ten
Dbl Sec
Cellos
Bass P
Digital tom cue:
Dr.
E. Gtr.
Bass
Synth. 1
Synth. 2
Perc.
f
subito p
ff
fff

A.3.2 *(De/Con)struction in Steel and Electricity* – Solo Tenor Steelpan

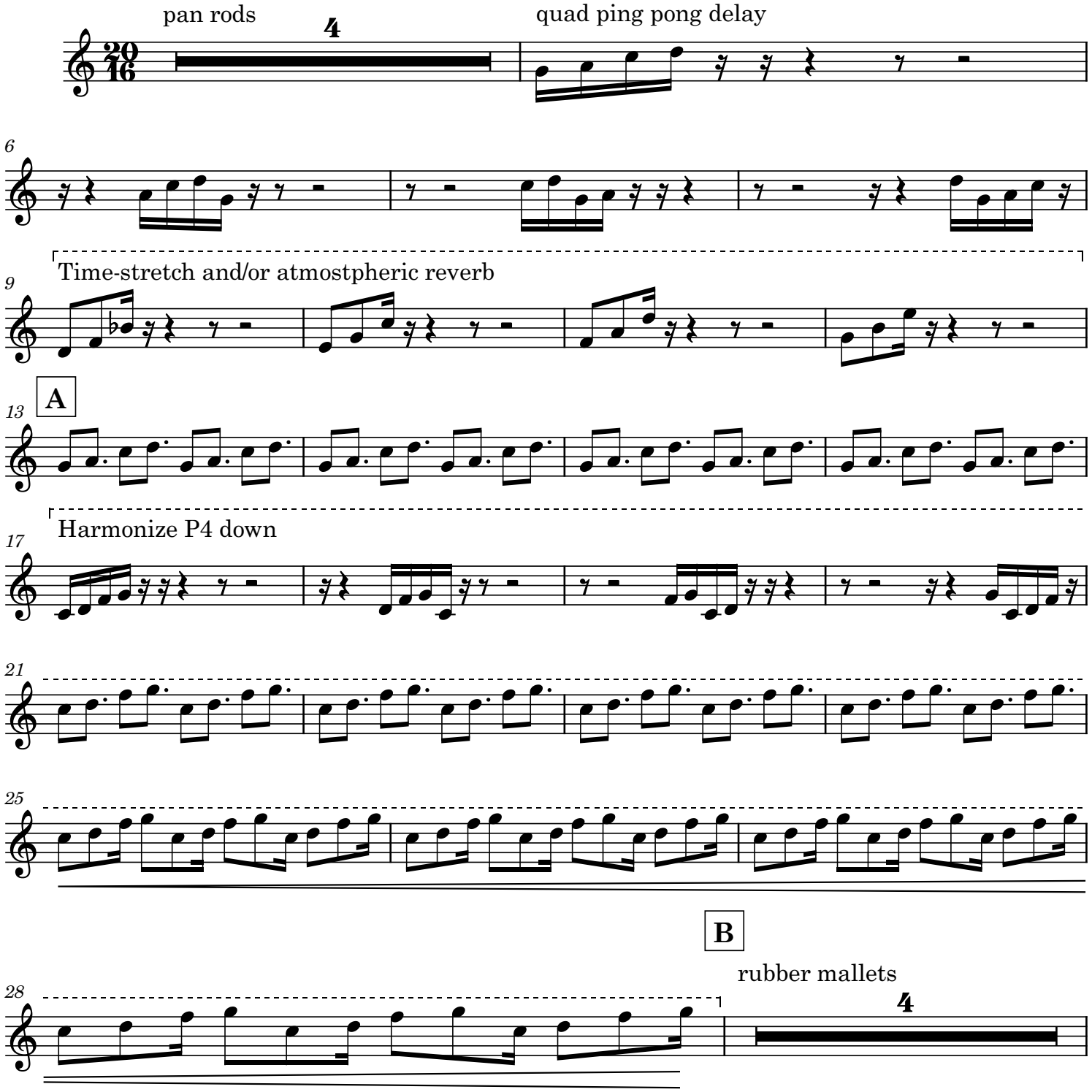
A video recording of the premiere performance of the solo version is available at `https://youtu.be/32mMqxS_zg4` [70].

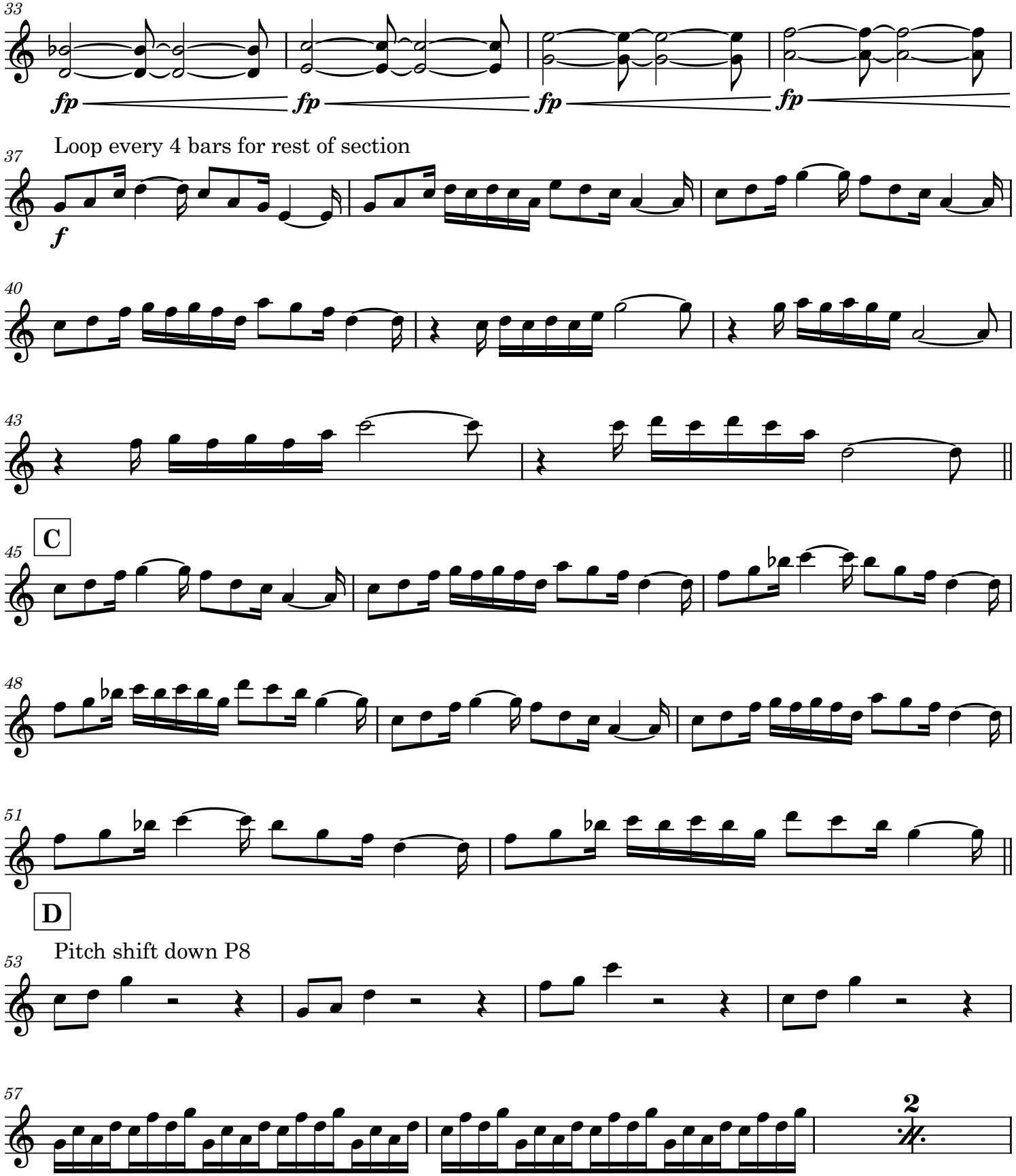
Steel Pan

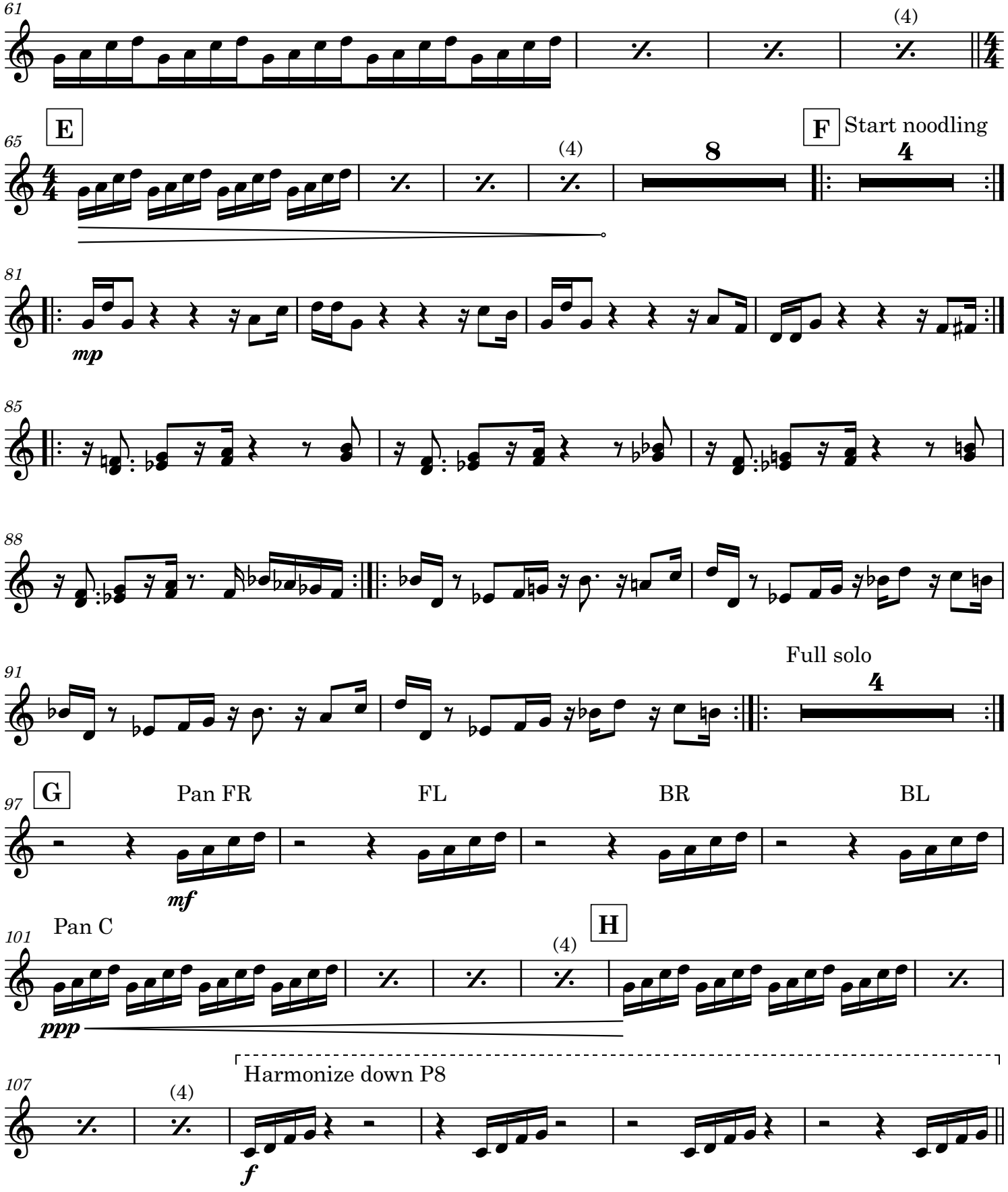
(De/Con)struction in Steel and Electricity

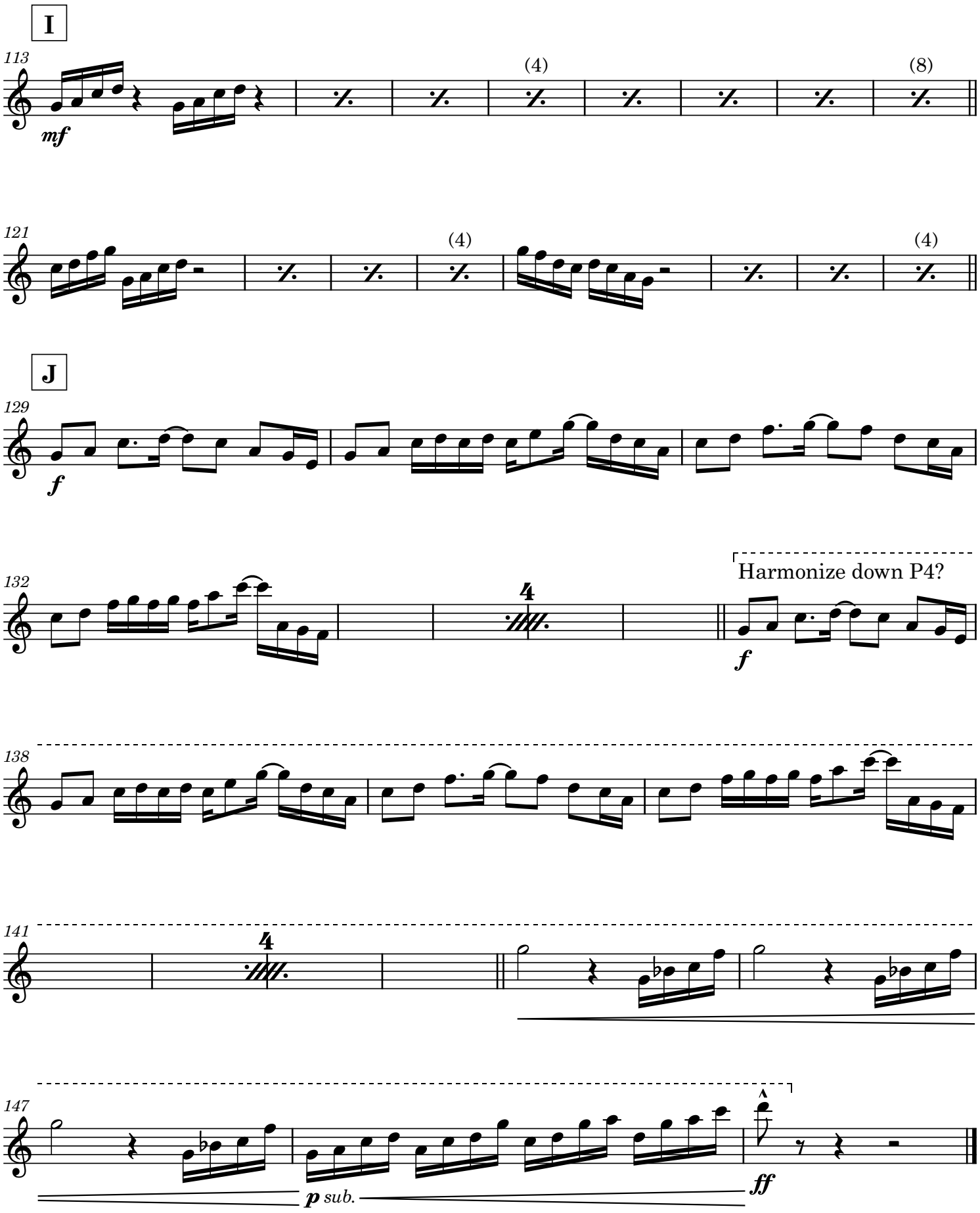
Colin Malloy

Solo Version




33
fp
fp
fp
fp
Loop every 4 bars for rest of section
37
f
40
43
C
45
48
51
D
Pitch shift down P8
53
57
2


61
(4)
E
65
(4)
8
F
Start noodling
4
81
mp
85
88
91
Full solo
4
G
97
Pan FR
FL
BR
BL
mf
Pan C
101
(4)
H
ppp
Harmonize down P8
107
(4)
f


I
113
mf
(4)
(8)
121
(4)
(4)
J
129
f
132
4
Harmonize down P4?
f
138
141
4
147
p sub.
ff