The Augmented Geometrically Spaced Transform: Applications of the Single Channel Frequency Estimator

by

Jonathan Michael Feldman

B.Ed. University of Western Ontario (2004)
B.Sc. McGill University (1996)

Submitted to the Program in Media Arts and Sciences
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 2021

Author .....
Program in Media Arts and Sciences
School of Architecture and Planning
February 2021

Certified by .....
Joseph A. Paradiso
Professor
Thesis Supervisor

Accepted by .....
Tod Machover
Academic Head
Program in Media Arts and Sciences

The Augmented Geometrically Spaced Transform:

Applications of the Single Channel Frequency Estimator

by

Jonathan Michael Feldman

Submitted to the Program in Media Arts and Sciences
School of Architecture and Planning
on February 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

The *Augmented Geometrically Spaced Transform* (AGST) is an auditory model that is based on an inversion of the acoustic piano, where the piano produces music and the transform analyses it. In contrast with the standard spectrogram, which is a complex frequency vector versus time, the AGST is based around a matrix of frequencies, known as the *AGST Frequency Matrix*, where for every frequency in the matrix, a spectral envelope is computed using a *Single Channel Frequency Estimator* (SCFE). The core invention of the thesis is the algorithm for the SCFE, which computes spectral envelopes with maximally high definition in a computationally efficient manner. A bank of SCFEs is assembled into a constant Q transform, known as a *Geometrically Spaced Transform* (GST). The GST can be used to visualize harmonics inside of musical notes, or audio in general, in a constant Q fashion. It is then shown that the AGST is a good front-end model for computational pitch perception. For example, it can be used to solve an important problem in auditory perception, the case of the missing fundamental. The entire thesis is framed in the context of building artificially intelligent music systems, including synthetic listeners (machines that listen in the way that people do), and synthetic performers (machines that allow for interactive music performance).

Thesis Supervisor: Joseph A. Paradiso
Title: Professor

Acknowledgments

I would like to thank many people for their encouragement and support in writing this thesis. First and foremost, I would like to thank my advisor Joe Paradiso, for supporting my research direction and helping me return to the lab in 2020 to complete my master's thesis.. I would then like to thank my original thesis advisor, Barry Vercoe, for inviting me to join the *Machine Listening Group* at the Media Lab in 1996. Both Joe and Barry are visionaries in the world of audio, and I truly had no idea how much I would learn from both of them.

I would really like to thank my thesis readers, Bill Gardner and Spencer Russell. They have both offered valuable guidance, insight, and support toward the completion of this thesis. I would also like to thank everyone in the *Responsive Environments Group* for making me feel welcome this term.

In my 4th year at McGill University, I discovered the field of *auditory scene analysis*, through the work of Dan Ellis. In a very coincidental way, I also discovered that the father of *auditory scene analysis*, Dr. Albert Bregman, was in the psychology department at McGill. Along with a number of other students who were studying computer music with Bruce Pennycook in the music department, I took Dr. Bregman's course. The course was fascinating, I had never taken a psychology course before and Dr. Bregman's work on *auditory scene analysis* amazed me. I was very lucky to have had a number of insightful and interesting conversations with him, and I enjoyed his course very much. I thank him with much respect because now I realize that he followed a scientific process in research related to auditory perception.

I would then like to give a special thanks to Dan Ellis. It was Dan's work on *computational auditory scene analysis* that I so admired that led me to apply to the Media Lab. I am also indebted to Eric Scheirer, who was my office mate, whose work I also studied as a 4th year undergrad at McGill. While I was a student, I had many fruitful conversations with everyone in the group, including Paris Smaragdis, Michael Casey, Bill Gardner, Keith Martin, Adam Lindsay, and Joe Pompei, and they are all inspirations to my work. And a very warm and special thanks to Judy Brown, who

invented the Constant Q Transform. She was an associate of the *Machine Listening Group* while I was a student and I had no idea that in the end, my research would be based on her work.

I would like to thank all of my professors while I was a graduate student, including Michael Bove, Rosalind Picard, Michael Jordan, Neil Gershenfeld, Whitman Richards, Sherry Turkle, Mitch Resnick, and James Ward. A special thanks goes to Kay Shelemay, as I was lucky to be able to cross-register for a course in ethnomusicology at Harvard. It was this class that rounded out my Media Lab experience and balanced out my education between arts and sciences.

Another very important mentor of mine was Marvin Minsky. I was very curious about AI, and I was lucky to have a number of amazing conversations with him about AI and music. He was the person who warned me that trying to solve problems in AI and music are very hard. He was right, they are hard, but it has been a fun journey applying everything I know about math and music and ultimately developing a deeper understanding of artificial intelligence.

Yet another very important mentor of mine is Bob Chidlaw. From 1998-2000, I took a break in my studies at the Media Lab and worked at *Kurzweil Music Systems* where I conducted research on reverb algorithms, and he was the chief scientist there. Bob gave me an opportunity to get hands on experience in the music technology industry, and was the person who got me interested in the field of audio engineering.

I would like to thank a number of people of other Media Lab students who were great friends while I was at MIT: Ben Vigoda, Kathi Blocher, Jocelyn Scheirer, Arjan Schutte, and Jon Dakks. I would also like to mention my close friend Aaron Lipman, and my other housemates Russell Epstein and Alex Holcombe.

I would like to thank all of my other teachers that I have had throughout my lifetime, who are like a converging Taylor series to me. They include my professors at McGill University, my high school teachers at Westdale High School, my elementary school teachers at the Hamilton Hebrew Academy, and the professors at both Western University where I studied education and at the University of Toronto where I studied music performance. Every teacher I have had in my life has had a huge impact

on me, as I always loved school and loved learning. It is truly a dream to look back and think of everyone who has taught me something, especially about math and music. One person I must mention is Russ Weil, who was the director of the music department at Westdale High. He gave me the opportunity to play in the Westdale jazz band, and ultimately the Hamilton All-Star Jazz Band. While I was classically trained in music from a very young age, my musical development really bIossomed when I fell in love with jazz. I've had many great piano teachers along with way outside of school, including Honey Burkle, Donna Barnes, Roland Packer, Bart Nameth, Tilden Webb, Yelena Neplokova, David Zoffer, Jerry Bergonzi, David Braid, and Gary Williamson. That said, I really do now love all of the disciplines and am grateful for all of the courses I have taken throughout my lifetime, and all of the wonderful teachers who taught them. All of that said, I would also like to mention a fantastic music teacher that I had when I was a student at U of T. Instead of strictly following the requirements in jazz performance, I took two music theory courses on classical music, that were taught by Mark Sallmen. He is the guy who dissected classical music theory for me and taught me advanced classical music ear training. Every now and again he would reference a pop or rock song, but for the most part he explained music in the tradition of Mozart, Beethoven, Haydn, Brahms, Tchaikovsky, and other great classical composers.

Finally, I would like to thank my family for all of their support and encouragement, including my father David, my mom Ilona, my sister Kayla, and Mark. Kayla is my closest, one and only sibling. Her work aesthetic and skills as a mom are an inspiration. My mom Ilona has been supportive of me in every way of my academic endeavors throughout my life, and for that I am truly grateful. Finally, all of my other aunts, uncles, and cousins, my brother-in-law, niece, and nephew, and my step-siblings and their families have been there for me as well, and I am truly blessed to have such a wonderful inclusive family.

This masters thesis has been examined by a Committee of the
Department of Media Arts and Sciences as follows:

Professor Joseph A. Paradiso .....
Chairman, Thesis Committee
Professor of Media Arts and Sciences

Dr. William G. Gardner .....
Thesis Supervisor
Ph.D., President, Wave Arts, Inc.

Dr. Spencer Russell .....
Member, Thesis Committee
Ph.D., Media Arts and Sciences

Contents

List of Figures

Chapter 1

Introduction

The *Augmented Geometrically Spaced Transform* is a K x M x N tensor that is a novel representation of a signal. There are K frequency channels, M harmonics, and N samples that are computed using *Single Channel Frequency Estimators*, or SCFEs. On the K axis it computes frequency envelopes and on the N axis it computes spectral envelopes. The M axis models a second order differential equation that computes harmonics. In the most general sense, the SCFE is a signal analysis tool that has a centre frequency f_k and a setting for its Q. For the purpose of this thesis, I let $Q = 16.81...$ since this corresponds to semi-tone spacing in music. But in practicality, it can be tuned to any desired centre frequency f_k and can be used with any Q you choose. In the case of audio, with infinitely high oversampling, there is infinite time resolution at lower frequencies that lie in the frequency range of human hearing. It's applications lie in the area of auditory signal analysis, including music, speech, bird songs, and other auditory-based recordings. It, however, can also be used to analyse sound that lies above the range of human hearing. For example, with high enough sample rates, the system could be used to analyse various bands of signals such as microwaves, x-rays, and ultrasound. This thesis is a scientific-based history of the discovery of this signal representation.

The applications are numerous and I hope that people will find this to be a useful tool in research. I have tried to show how it useful for analysing musical instrument samples, such as the piano and the electric guitar. I assume a basic understanding of

music, such as music theory in the tradition of Western musicology where the tonic, sub-dominant, and dominant chords play an essential role. Just like Darwin observed the Galapagos Islands and created a theory of evolution, I have observed my own musical development and my knowledge of both piano and guitar-based music that has influenced my musical knowledge and experience. While the experience of music is subjective, here I propose that it can be analysed using computational methods.

One important idea that I propose is that auditory perception is an evolutionary process. People have been learning to listen to sounds for thousands of years, beginning perhaps with the weather, drumbeats that were used in tribal practices, drumbeats that were used for communication, and musical instruments that were tuned in various ways. Concerning music, the ear has *learned* to listen to music, including harmony. The psychological elements of auditory perception, including the perception of pitch, beats, and timbre, evolved over time. What the ear is able to listen to and enjoys listening to then continues to evolve through the creation of new musical instruments and new methods of playing them.

I suggest that there is a clear link between music, technology, and culture. Based on this observation and the theory developed here, I suggest that there is an emerging field of study known as *computational ethnomusicology*. It is the study of how computational methods can be used in anthropological research, especially as it relates to *music in culture*. By the end of the thesis, I will show how the *Augmented GST* can be used as an analysis tool to study music from any world culture. I will also suggest, however, that it may be biased towards music that stems from Western Civilization, where for many centuries the piano has been the dominant compositional tool. I will also suggest that the ear has certain innate abilities to analyse the sound that enters the auditory system, but its interpretation and enjoyment is culturally subjective. The idea then, of *computational ethnomusicology*, is to have a computational framework that can be used to study music, and I would also suggest that the model could be expanded, for instance, to be useful for listening to the music of India, where quarter-tone spacing is highly relevant to their form of classical music. The thesis then is quite open ended because now that I have written it, it is clear

that it leads to further work in physics, affective computing, artificial intelligence, computational modelling, robotics, and other disciplines including music theory and ethnomusicology.

Chapter 2

A Brief History and Background

The idea of *synthetic performance* was pioneered at M.I.T. in the 1980s, where Barry Vercoe built a music accompaniment system for a live flutist [**Vercoe1984a**] [**Vercoe1984b**]. The flute had to be augmented with optical sensors to aid the real-time pitch detection algorithm that was used to drive the system. Vercoe described computer-based accompaniment systems as systems that could listen, learn, and perform with human musicians. This system motivates the need for real-time pitch detection algorithms. The work was also pursued at other music research institutes such as IRCAM where Pierre Boulez employed real-time computer interpretation of audio in enhanced live performances with his Ensemble Intercontemporain.

There are a number of algorithms that go together to build a machine that *listens*. These include pitch detection, beat tracking, tempo tracking, et al, potentially including instrument identification and source separation. These auditory listening processes need to happen in real-time. Many algorithms that were developed by the *Machine Listening Group* at the M.I.T. Media Lab and others in the 1990s, including [**Slaney1990**], [**Scheirer1997**], [**Martin1998**], and [**Smaragdis2001**] are foundational to building real-time sophisticated synthetic performance systems. What is needed is a comprehensive auditory model that acts as a real-time front end processor and is an integrated framework for computational auditory perception.

One potential application of such a model, as it applies to building synthetic performers, is *interactive music performance*, systems where humans and computers can

interact and perform music together. A dream system would be a system that augments the experience of not just jamming and performing, but also aspects of learning to play music, such as practicing technique and ear training. In the end, it would be fun to play interactive musical games as Ben Vigoda describes in *Musical Games: A Guide for Group Improvisation*, also known as *Games for Song* [**Vigoda2005**]. Playing these games are an example of meaningful musical group experiences. In order for synthetic musicians to truly be successful, the boundary between humans and machines would need to be blurred to a point where these machines pass the Turing test for creating music as art [**Turing1950**].

Other practical applications of having an accurate perceptual computational model of audition includes building more sophisticated hearing aids, music production systems, and other AI-based sound and music processing tools.

In this thesis, I propose a computational model of pitch perception that is based on a matrix of frequency estimators that compute spectral envelopes called the *Augmented Geometrically Spaced Transform.* I propose a constant Q frequency analyser, called the *Geometrically Spaced Transform* (GST), as a computational alternative to typical front-end auditory modelling and processing such as the Patterson and Holdsworth auditory model [**Patterson1995**]. I then show that the GST can be augmented by computing a tensor based on what I call the AGST Frequency Matrix. This forms the basis of a computational model of pitch perception, and can be used as a real-time system that computes pitch and, for instance, solves an important problem in pitch perception, the case of the missing fundamental. I then show that the inverse SCFE can be used to synthesize monophonic audio and the inverse GST can be used to synthesize polyphonic audio.

In the original title of this thesis I used the term *Online Frequency Estimation.* I envisioned a system where frequency estimates are computed as every new sample enters the system. With this in mind, I had the idea that every frequency estimate could be computed using a linear update rule, and this would give the maximum time-frequency resolution possible. In Neil Gershenfeld's *The Nature of Mathematical Modelling* class, I made the observation that the Discrete Fourier Matrix was N-

periodic. I envisioned spectral signal processing as having the Fourier matrix roll like a cylinder across the input samples. This is atypical in traditional discrete-time signal processing where spectral estimates are computed on blocks of input samples. However, the approach taken here is that the auditory system operates computationally in an online fashion and is high resolution, thus giving it high fidelity. Using this approach, and following the seminal papers written by Judith C. Brown [**Brown1990**] [**Brown1992**], I was able to derive an algorithm for online machine listening that is constant Q and wavelet-like where frequency estimates are more localized in time as frequency increases towards the top end of the hearing spectrum.

I now frame the thesis in a larger context, and explain how it might be related to the field of Affective Computing [**Picard1995**]. I suggest that auditory perception is an evolutionary process, and that the perception of auditory elements in music such as pitch, beats, and timbre has changed over time. One goal of this may be to enhance the emotional experience of the listening process. For example, the more a person listens to music that is based on the rules of classical harmony and that is tuned based on the equal-tempered scale, the better it is able to enjoy a peaceful, harmonious listening experience [**Haignere2013**]. By harmonious, I mean an experience that produces emotions that excite the corresponding emotional patterns in the brain. I would suggest, for example, that Mozart played in major keys is more peaceful to listen to than the blues, rock music, or heavy metal. Music in general plays with the emotions in the brain, which is exemplified, for example, by the solo piano improvisations of Keith Jarrett. He has synthesized classical and jazz in a harmonious way that tells a story, that travels through the emotional experience of living life by using intricate, harmonic-based chord progressions. But I think from an emotional point of view, one way of ensuring a happy experience is to listen to music that is based on the chord progression of the tonic travelling to the sub-dominant travelling to the dominant and cadencing back to the tonic. This can be found for numerous examples in classical European music, traditional African music, rock n roll music such as Twist and Shout by The Beatles, ska music such as A Message To You Rudy by The Specials, reggae music such as Stir It Up by Bob Marley and the

Wailers, and electronic music such as Autobahn by Kraftwerk. What I mean to say is that every note and every chord in a musical sequence evokes emotion, and triggers corresponding emotion patterns in the brain that are a response to harmony. The emotional experience is a direct response to the harmonic experience.

I also believe that it is important to keep in mind that music has an anthropological significance. I am reminded of the thought that *music creates culture* and the idea that the music that people listen to triggers an emotional response that influences human thought, feeling, behaviour, and experience. For example, every generation in American culture since the advent of the blues in the early 20th century has given rise to dance, joy, sorrow, culture, and sociology. As music continues to evolve, society unfolds in new and myriad ways, as seen, for example, since the beginning of the 20th century where both musical and cultural development accelerated greatly, spurred by advances in recording, communications, and media technology. Culture and history continue to unfold in the 21^{rst} century as music, both new and old, is consumed and experienced, where new culture emerges that is touched upon by the music that people listen to. These musical and cultural forces interact to create a diverse fabric of the human experience. It turns out that even cultural aspects of music performance, such as instrument choice, modes, and tunings, can be analysed with computational methods using a model such as the *Augmented Geometrically Spaced Transform*.

Chapter 3

Time-Frequency Analysis

3.1 Introduction

Time-Frequency Analysis is often used to compute a *Spectrogram* that displays features of a signal with time along the x-axis and frequency along the y-axis. Typical applications include music, sonar, radar and speech. The most common tool used to compute a Spectrogram is the *Short-Time Fourier Transform* (STFT), which uses the *Fast Fourier Transform* (FFT) to compute a frequency vector for each windowed block of the input signal. In the case of music and speech, where the contents of the signal contain pitched information, a logarithmic scale is typically used to display frequency. For music and speech analysis, the problem is that there is too little information available at low frequencies and too much information at high frequencies. See figure (3-1). The result is that the Q of each frequency channel is wider at low frequencies and narrower at high frequencies.

$$Q = \frac{f_k}{\Delta f_k} \tag{3.1}$$

The Constant Q Transform (CQT) was proposed to solve this problem [**Brown1990**]. See figure (3-2). Brown devises a 24 band per octave filter bank, which has quarter-tone spacing, and can resolve the fundamental frequencies of adjacent musical notes with a semitone spacing. Brown states that "the resolution should be geometrically

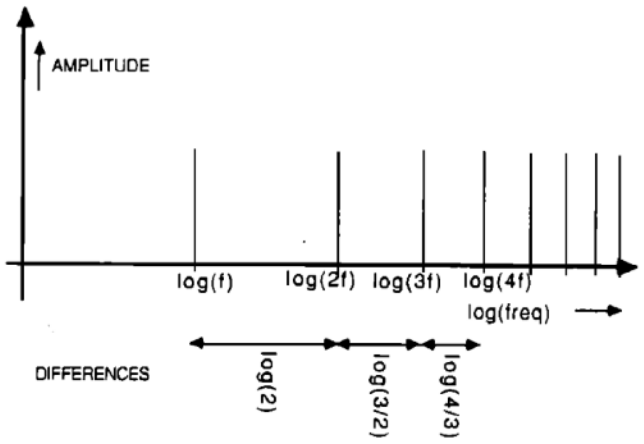

FIG. 1. Pattern of Fourier transform of harmonic frequency components plotted against log(frequency).

Figure 3-1: Pattern of Fourier Transform harmonic frequency components plotted against log(frequency)

related to the frequency e.g., 3% of the frequency in order to distinguish between frequencies with semitone (6%) spacing." What is desired is the ratio of centre frequency to bandwidth should be constant; that is, the Q factor of each frequency channel is constant.

I propose an alternative to this approach, which I call the *Geometrically Spaced Transform* (GST), which uses a bank of *Single Channel Frequency Estimators* (SCFEs) that are tunable to any desired center frequency. I tune the center frequencies to the fundamental frequencies of the notes of the piano, which are semi-tone spaced, and with the Q set as in equation (3.2):

$$Q = \frac{1}{2^{\frac{1}{12}} - 1} \tag{3.2}$$

In this thesis, the algorithm operates off-line. The algorithm is designed however to operate in an online fashion, where each successive filter output is computed using a linear update rule based on the previous filter output as each sample enters the system. Also note that here I suggest that semi-tone spacing is sufficient to resolve

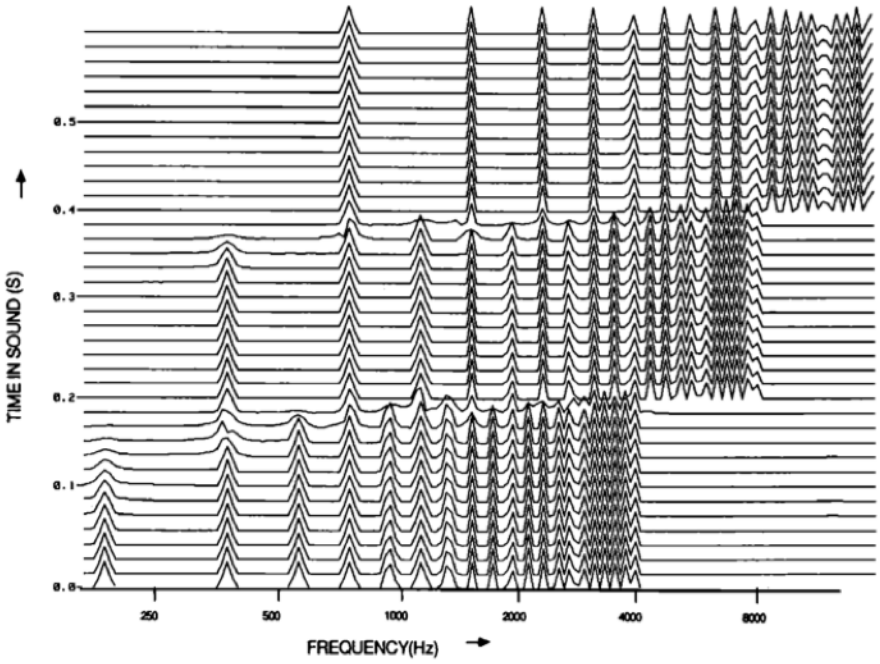

FIG. 2. Constant Q transform of three complex sounds with fundamentals G_3 (196 Hz), G_4 (392 Hz), and G_5 (784 Hz), and each having 20 harmonics with equal amplitude.

Figure 3-2: Brown's Constant Q Transform example

individual pitches and that quarter-tone spacing is not necessary as Brown suggests in [**Brown1990**]. See figure(3-3) where the GST mimics Brown's figure.

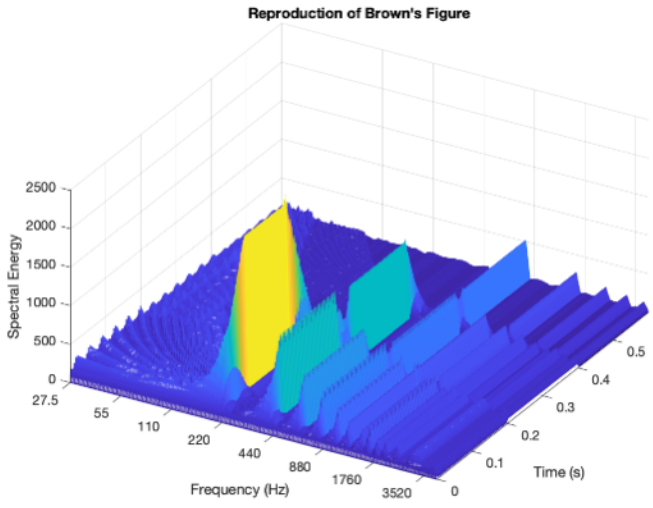

Figure 3-3: GST analysis that mimics Brown's example

Each SCFE is fast. It requires one real multiply, two real additions, two complex multiplies, and two complex additions to process each time domain sample. The

SCFE can be inverted to recover the original audio with perfect signal reconstruction to recover the audio input. In this thesis, I apply a matrix of SCFEs, known as the *Augmented GST*, and suggest that it can be used as a general auditory model for computational audition. Here, I focus on the CQS as the front end of a pitch detection system. In the end, I also show that the inverse SCFE algorithm can be modified to be used as a monophonic synthesizer, and the inverse GST algorithm can be modified to be used as a polyphonic synthesizer.

I further describe the *Single Channel Frequency Estimator*. Given a real-valued signal input, the algorithm estimates spectral energy at a given center frequency with a fixed Q. A typical use of this in music signal analysis is that the Q is fixed to resolve semi-tones of notes that are produced by typical Western musical instruments ($Q \approx 17$) [**Mathieu1997**]. A bank of SCFEs can be put together into a transform that mimics Brown's *Constant Q Transform* (CQT), which I call the *Geometrically Spaced Transform*.

I show a history of how I came to develop the SFCE algorithm, which begins with an observation about the periodic nature of the *Discrete Fourier Matrix*. I develop what I call the *Online Fourier Transform*, which is similar in principle to a *Short-Time Fourier Transform* with a step size of one sample. I then show how instead of computing this in matrix form, an algorithm can be developed by keeping track of phase as a phasor circulates around the complex unit circle. This leads to the formulation of the *Geometrically Spaced Transform*, where instead of computing linearly spaced frequency components, the computed frequency components are geometrically spaced, akin to the fundamental frequencies of the piano. The GST then is extended to formulate the *Augmented GST*, where an additional dimension is added to the GST to compute harmonics. It is this signal representation that is proposed as a computational auditory model and when it comes to computing pitch, it for example solves the problem of the missing fundamental [**Moore1994**].

3.2 Constant Q Transform History

The *Constant Q Transform* (CQT), which has geometrically spaced frequency channels, was originally proposed by Brown in 1990 [**Brown1990**]. A more efficient implementation of the CQT was proposed in 1992 [**Brown1992**], which efficiently implements the following equation:

$$X[k] = \sum_{n=0}^{N_k-1} w[n,k]x[n]e^{-j\omega_k n} \tag{3.3}$$

While the CQT can be used for analysis purposes, a goal is to be able to invert the transform so that it can be used for re-synthesis. An approximate of an inverse transform was described by FitzGerald, Cranitch, and Cychowski in 2006 [**Fitzgerald2006**]. In 2010, an improvement on the inverse transform was proposed by Schorbuber and Klapuri who reported a signal-to-noise ratio of 55 dB for a re-synthesized input signal by analysing 12 bins per octave [**Schorbuber2010**].

A *modified constant Q spectrogram* was proposed by [**Ingle2011**], whose algorithm "computes the N_k-long DFT for each frequency, f_k, of interest and then picking out the Q^{th} DFT coefficient. The equation is as follows:

$$X[k] = \frac{1}{N_k} \sum_{n=0}^{N_{k-1}} w[k,n]x[n]e^{\frac{-j2\pi Qn}{N_k}} \tag{3.4}$$

The proposed *Online Fourier Transform* is similar to the *Sliding DFT* (SDFT) [**Jacobsen2003**]. The main difference between the two algorithms is that the SDFT uses a circular buffer whereas the OFT uses a circular matrix, which I call the Fourier cylinder. The result is that while the magnitude spectra are equivalent, the phase spectra are different. In the case of the SDFT, Jacobsen's research lead to a constant Q transform, which is described in *Sliding with a Constant Q* [**Bradford2008**]. Here, the OFT, via the *phase update method*, leads to the GST. The value of the GST compared to existing work is in its simplicity, compared with [**Brown1992**], [**Bradford2008**], and [**Velasco2011**].

3.3 Complex Exponentials

Complex exponentials are the basic unit that is used in harmonic analysis and can be found in both physics and electrical engineering. The complex exponential can be derived by solving the following differential equation:

$$\frac{dy}{d\theta} - jy = 0 \tag{3.5}$$

or

$$\frac{dy}{d\theta} = jy \tag{3.6}$$

The solution to (3.6) is the complex sinusoid

$$y = cos\theta + jsin\theta \tag{3.7}$$

Here note that

$$\frac{dy}{d\theta} = -sin\theta + jcos\theta \tag{3.8}$$

and

$$jy = -sin\theta + jcos\theta \tag{3.9}$$

So

$$\frac{dy}{d\theta} = jy \tag{3.10}$$

and

$$y = cos\theta + jsin\theta \tag{3.11}$$

is the solution.

I now examine the Taylor series for $y = sin\theta$ and $y = cos\theta$ and show that when I put them together according to

$$y = cos\theta + jsin\theta \tag{3.12}$$

I get the Taylor series for $y = e^{j\theta}$.

$$cos\theta = 1 - \frac{1}{2!}\theta^2 + \frac{1}{4!}\theta^4 - \frac{1}{6!}\theta^6 + \frac{1}{8!}\theta^8 - ... \tag{3.13}$$

$$sin\theta = \theta - \frac{1}{3!}\theta^3 + \frac{1}{5!}\theta^5 - \frac{1}{7!}\theta^7 + ... \tag{3.14}$$

So

$$[1 - \frac{1}{2!}\theta^2 + \frac{1}{4!}\theta^4 - \frac{1}{6!}\theta^6 + ...] + j[\theta - \frac{1}{3!}\theta^3 + \frac{1}{5!}\theta^5 - \frac{1}{7!}\theta^7 + ...] \tag{3.15}$$

$$1 + j\theta - \frac{1}{2!}\theta^2 - j\frac{1}{3!}\theta^3 + \frac{1}{4!}\theta^4 + j\frac{1}{5!}\theta^5 - \frac{1}{6!}\theta^6 - j\frac{1}{7!}\theta^7 + \frac{1}{8!}\theta^8 + ... \tag{3.16}$$

$$= e^{j\theta} \tag{3.17}$$

So

$$e^{j\theta} = cos\theta + jsin\theta \tag{3.18}$$

which is known as Euler's Formula [**Strang2014**]

In this thesis, I begin by examining the *Discrete Fourier Matrix*, which is a matrix of complex exponentials. I use these complex exponentials to derive the *Single Channel Frequency Estimator*, which forms the basis for the proposed *Geometrically Spaced Transform* and *Constant Q Spectrogram.*

3.4 Discrete Fourier Matrix

The *Discrete Fourier Matrix* is a square matrix that is defined as follows:

$$F = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & w & w^2 & w^3 & \dots & w^{N-1} \\ 1 & w^2 & w^4 & w^6 & \dots & w^{2(N-1)} \\ 1 & w^3 & w^6 & w^9 & \dots & w^{3(N-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{2(N-1)} & w^{3(N-1)} & w^{6(N-1)} & \dots & w^{(N-1)(N-1)} \end{bmatrix} \tag{3.19}$$

where $w = e^{-j2\pi/N}$ [**Strang1998**]

A *Discrete Fourier Transform*, calculated as a matrix multiplication, is as follows:

$$X = Fx \tag{3.20}$$

Typically, however, the *Fast Fourier Transform* algorithm is employed instead of using matrix multiplication. An FFT efficiently computes such transformations by factorizing the DFT matrix into a product of sparse, mostly zero, factors. This reduces the computational cost from $O(N^2)$ to $O(NlogN)$ [**VanLoan1992**]

3.5 Short-Time Fourier Transform

Given that I am trying to compute a new frequency estimate for every sample that enters the system, I examine the *Short-Time Fourier Transform* with a step size of one sample. Computing it this way ensures that the time resolution of the spectral output is maximal. Typically, with the STFT, a window is used, but here I show that for this algorithm windowing is not necessary. The *Discrete Fourier Matrix* is used at every step to calculate the *Discrete Fourier Transform* of the windowed input samples at any time t, and this is computed quickly using the *Fast Fourier Transform* algorithm [**Cooley1965**], which takes advantage of the structure of the *Discrete Fourier Matrix* to reduce the computation from a computational cost of $O(N^2)$ to $O(NlogN)$ using a recursive divide-and-conquer butterfly algorithm.

The standard method of then computing the *Short-Time Fourier Transform* according to [**Smith2011**] is:

$$X_m(\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n - mR)e^{-j\omega n} \tag{3.21}$$

I simplify the math by showing how the DFT matrix can be used to compute an STFT in the N = 4 case. Note that everyone uses the FFT to compute an STFT, but here I am doing it with matrix multiplication because it informs the work to come on the *Online Fourier Transform*.

$$F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \tag{3.22}$$

$$\vec{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \dots \end{bmatrix} \tag{3.23}$$

The output of the STFT produces a spectrogram as follows:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & \dots \\ x_2 & x_3 & x_4 & x_5 & \dots \\ x_3 & x_4 & x_5 & x_6 & \dots \\ x_4 & x_5 & x_6 & x_7 & \dots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \dots \\ \vec{f}_4 & \vec{f}_5 & \vec{f}_6 & \vec{f}_7 & \dots \\ \vdots & \vdots & \vdots & \vdots & \dots \end{bmatrix} \tag{3.24}$$

Note that in the STFT representation, the harmonics of the musical audio signal do not necessarily coincide with the centre frequencies of the bands that are being analysed. This is the motivation behind designing a geometrically spaced transform whose analysis frequencies match those of the typical frequencies inside of a harmonic-based sound. See figures (3-4) and (3-5) as examples of STFT-based analysis of a piano note. Note that with the STFT, it is typical to use a window on the input samples before taking the FFT, which reduces spectral leakage between adjacent frequency channels.

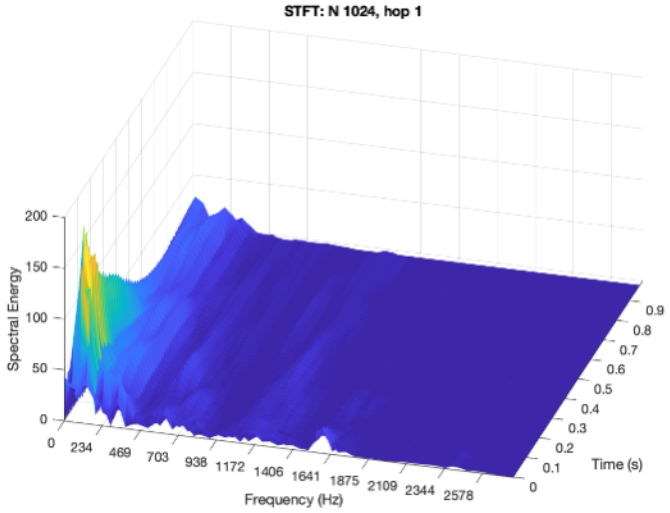

Figure 3-4: STFT analysis of piano note A110, N = 1024, hop = 1, $f_s = 48$ kHz

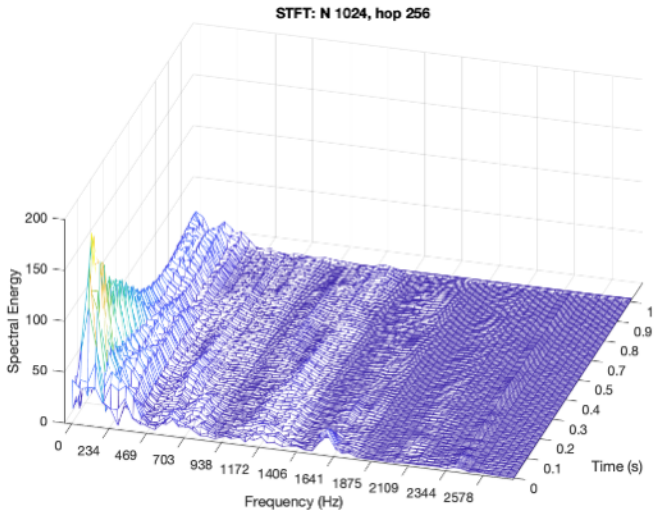

Figure 3-5: STFT analysis of piano note A110, N = 1024, hop = 256, $f_s = 48$ kHz

Chapter 4

Online Fourier Transform

4.1 Introduction

In this chapter, I discuss the *Online Fourier Transform* (OFT). I modify the STFT by making the observation that the DFT matrix is N-periodic. I derive a linear update rule to compute the filterbank output at each time step as a function of the previous timestep. In the end, I will have a time-frequency matrix, or *spectrogram*, where each frequency bin in a frequency estimate has been integrated over a *time smear* of N samples, but where otherwise, the representation has the maximum time resolution possible. I can show graphically, without a proof, that the magnitude of the OFT spectrogram is equal to the magnitude of the STFT spectrogram. The phase of the OFT differs however from the phase of the STFT because the OFT is demodulated. It appears that for spectral channels that exactly detect a sinusoid whose frequency is matched with the center frequency of its listening channel, the phase is a constant. Figure (4-1) shows how I envision the Fourier matrix rolling like a cylinder across the input samples. The mathematics in this section is similar to the *Sliding DFT* [**Jacobsen2003**] (shown below).

I now show the calculation of the OFT. Again, I simplify the math by showing the case for N = 4, i.e. where the DFT matrix is of size 4x4:

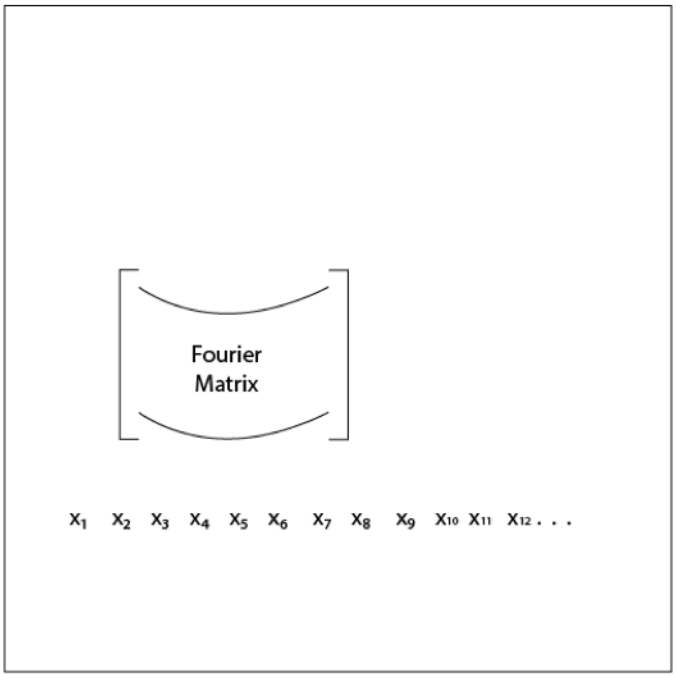

Figure 4-1: A Fourier cylinder rolling across the input samples

$$F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{4.1}$$

Recall that a *Discrete Fourier Transform* is calculated as a matrix multiplication as $X = Fx$. Analysed in another way, the *Discrete Fourier Transform* is the vector projection of the input signal onto each row of the *Discrete Fourier Matrix*:

$$\vec{p_1} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$
$$\vec{p_2} = \begin{bmatrix} 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{4.2}$$

etc.

Using the idea of the *Fourier Cylinder*, each row of F can be seen as a complex wave (i.e. a complex exponential) and each row can be expanded cyclically:

$$\vec{p_1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \ldots \end{bmatrix} \tag{4.3}$$

$$\vec{p_2} = \begin{bmatrix} 1 & -j & -1 & j & 1 & -j & -1 & j \ldots \end{bmatrix} \tag{4.4}$$

$$\vec{p_3} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \ldots \end{bmatrix} \tag{4.5}$$

$$\vec{p_4} = \begin{bmatrix} 1 & j & -1 & -j & 1 & j & -1 & -j \ldots \end{bmatrix} \tag{4.6}$$

4.2 Forward Algorithm

The DFT matrix is viewed like a cylinder moving across a real time input vector $\vec{x}$. The notation is as follows. I use the matrix F_4 to calculate a frequency estimate at time t = 4 with input samples x_1, x_2, x_3, and x_4. Then at t = 5, I use F_5 with input samples x_2, x_3, x_4 and x_5, etc. Note that here k is a whole number and $N = 4$.

$$F_{4+kN} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} F_{5+kN} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -j & -1 & j & 1 \\ -1 & 1 & -1 & 1 \\ j & -1 & -j & 1 \end{bmatrix} \tag{4.7}$$

$$F_{6+kN} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & j & 1 & -j \\ 1 & -1 & 1 & -1 \\ -1 & -j & 1 & j \end{bmatrix} F_{7+kN} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ j & 1 & -j & -1 \\ -1 & 1 & 1 & 1 \\ -j & 1 & j & -1 \end{bmatrix} \tag{4.8}$$

$$\vec{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \dots \end{bmatrix} \tag{4.9}$$

To compute the next frequency estimate, the idea is given by:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \dots \\ 1 & -j & -1 & j & 1 & -j \dots \\ 1 & -1 & 1 & -1 & 1 & -1 \dots \\ 1 & j & -1 & -j & 1 & j \dots \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \cdots \\ \vec{f_4} & \vec{f_5} & \vec{f_6} & \cdots \\ \vdots & \vdots & \vdots & \cdots \end{bmatrix} \tag{4.10}$$

where

$$\vec{f_4} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ -j \\ -1 \\ j \end{bmatrix} x_2 + \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} x_3 + \begin{bmatrix} 1 \\ j \\ -1 \\ -j \end{bmatrix} x_4 \tag{4.11}$$

$$\vec{f_5} = \begin{bmatrix} 1 \\ -j \\ -1 \\ j \end{bmatrix} x_2 + \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} x_3 + \begin{bmatrix} 1 \\ j \\ -1 \\ -j \end{bmatrix} x_4 + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} x_5 \tag{4.12}$$

$$= \vec{f_4} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} x_5 - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} x_1 \tag{4.13}$$

$$= \vec{f_4} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} (x_5 - x_1) \tag{4.14}$$

$$= \vec{f_4} + F_{:,1}(x_5 - x_1) \tag{4.15}$$

So in general:

$$\boxed{\vec{f_t} = \vec{f_{t-1}} + F_{:,t\%N}(x_t - x_{t-N})} \tag{4.16}$$

where the Fourier matrix F is expressed with Matlab indexing and is of size NxN and t%N is actually mod(t,N) if t <N and N if t = N. In the next section I use this equation to generate the Fourier coefficients of $F_{:,t\%N}$ on the fly. This method leads to the GST. Note that the computational cost of a new frequency estimate in O(N) whereas the cost of computing an FFT is O(NlogN). This is of comparable computational complexity to the Sliding DFT [**Jacobsen2003**]:

$$S_{k,t} = S_{k,t-1}e^{j2\pi k/N} + x_t - x_{t-N} \tag{4.17}$$

Note that I am circulating the columns of the *Discrete Fourier Matrix* and whereas in the *Sliding DFT*, Jacobsen is using a fixed DFT matrix and a circular buffer for the samples according to the DFT circular shift property. Equations (4.16) and (4.17) differ by a modulation, according to the following equation:

$$f_{k,t} = e^{\frac{-j2\pi kt}{N}} S_{k,t} \tag{4.18}$$

4.2.1 Analysis

I begin by analysing a signal that contains a sine wave at 1000 Hz, with a sample rate of 8000 Hz. I set N = 8 so that the frequency channels are tuned to multiples of 1000 Hz, which results in 8 computed frequency channels. See figure(4-2).

In the magnitude response, the 1000 Hz band and corresponding imaginary 7000 Hz band are constant. Note that in the phase response, the band at 1000 Hz is constant at -$\pi/2$ and the band at 7000 Hz is constant at $\pi/2$.

Because I am interested in analysing audio signals that contain music, I test the algorithm on a piano note whose fundamental frequency is 110 Hz. Here, the sample rate is 48 kHz. I display the first 64 channels. See figure (4-3).

Note that the harmonics of the piano notes are not well resolved in the magnitude response, with the low frequency information muddled together in the bottom few rows of the spectrogram. Note also, that in the phase response, there are horizontal bands that appear as features of the image. It appears that there is harmonic structure of the input signal that is being displayed by phase response.

4.3 Phase Update Method

In the previous section, I used the *Discrete Fourier Matrix* and let it roll like a cylinder across the input samples. I use one column of the matrix to compute each new frequency estimate. Here, instead of using the DFT matrix, I generate the Fourier coefficients on the fly. One advantage here is that I am able to derive a method to invert the transform. I can consider each frequency channel separately.

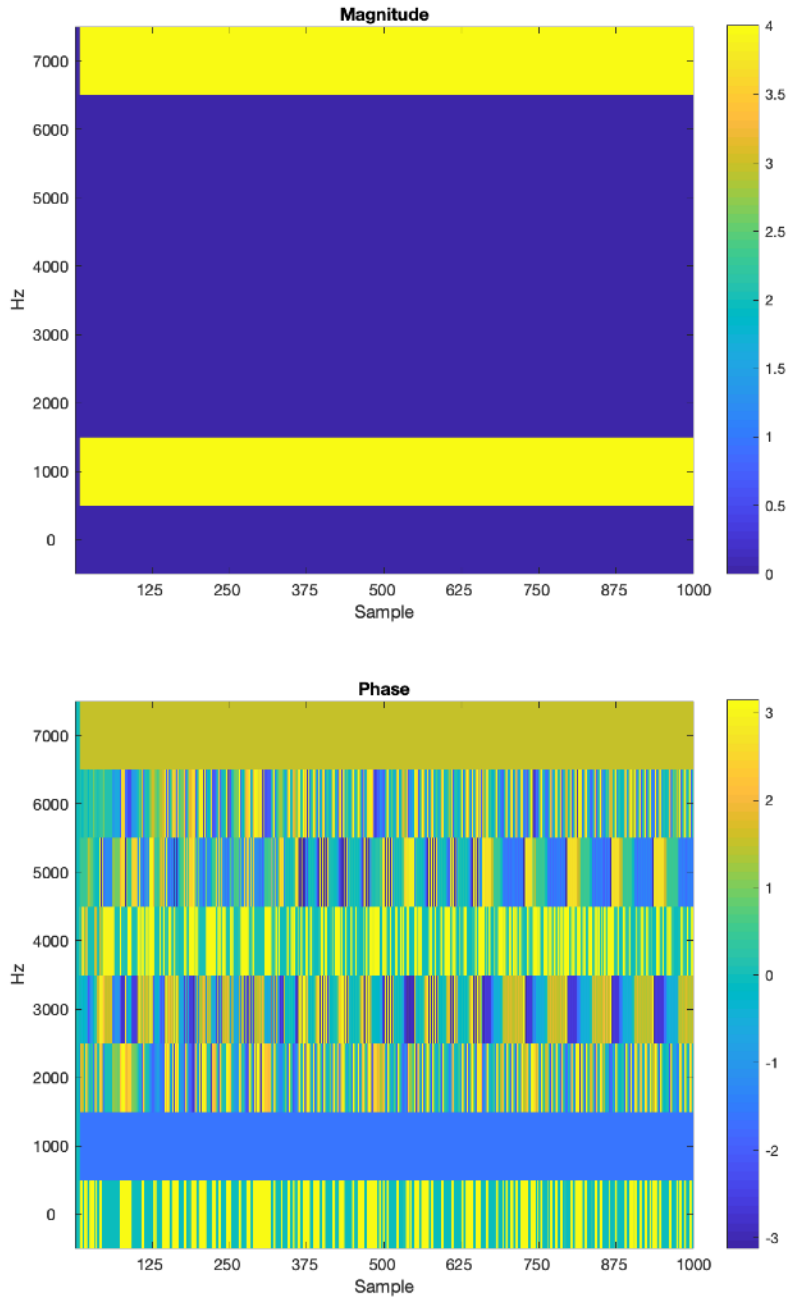

Figure 4-2: OFT Magnitude and Phase analysis of a 1 kHz sine tone

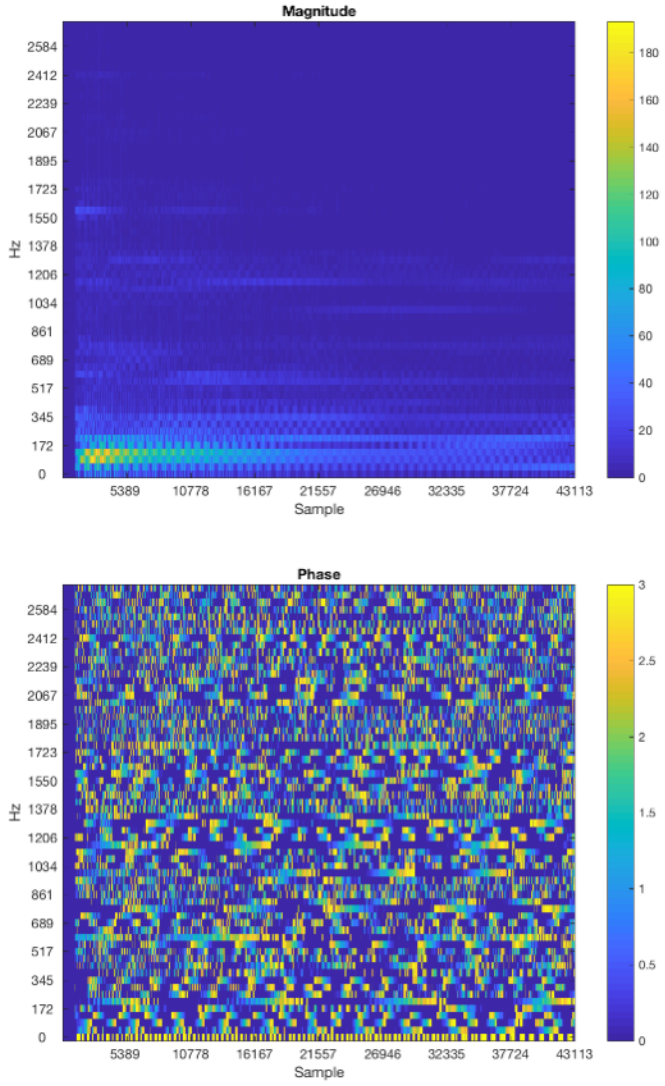

Figure 4-3: OFT analysis of audio input of piano note A2

4.3.1 Forward Algorithm

I define $\mathbf{x}$ to be the input signal, f_s to be the sample rate, $\mathbf{N}$ to be the window length (integration constant), and $\mathbf{M}$ to be the number of frequency channels starting at DC. I consider frequency channel $\mathbf{k}$ at time $\mathbf{t}$.

I start with $\theta_k = 0$ and define $\Delta\theta_k = -2\pi(\frac{k-1}{N})$

Initialization

I start with $f_{k,1} = e^{j0}x_1 = x_1$ and then for t = 2:N

$$\theta_{k,t} = \theta_{k,t} + \Delta\theta_k \tag{4.19}$$

$$f_{k,t} = f_{k,t} + e^{j\theta_{k,t}}x_t \tag{4.20}$$

I now have frequency estimates for the first N samples including the first complete N-point frequency estimate at time N $\vec{f_N}$.

Runtime

Starting with sample t = N+1:

$$\theta_{k,t} = \theta_{k,t} + \Delta\theta_k \tag{4.21}$$

$$f_{k,t} = f_{k,t-1} + e^{j\theta_{k,t}}(x_t - x_{t-N}) \tag{4.22}$$

See figure (4-4) for an example of using the OFT with the phase update method.

4.3.2 Inverse Algorithm

I use the buffered first N samples to begin the output signal.

Then starting at time t = N+1:

I start with

$$f_{k,t} = f_{k,t-1} + e^{j\theta_{k,t}}(x_t - x_{t-N}) \tag{4.23}$$

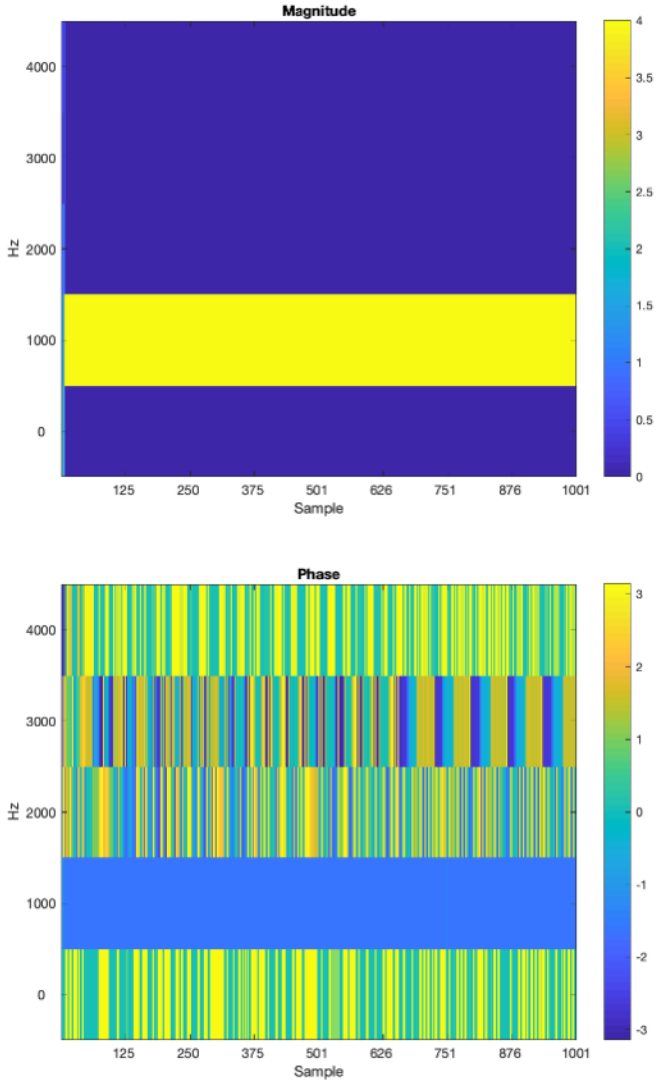

Figure 4-4: OFT phase update algorithm magnitude and phase analysis of a 1 kHz sine tone

Expressed in vector form:

$$\vec{f_t} = \vec{f_{t-1}} + e^{j\vec{\theta_t}}(x_t - x_{t-N}) \tag{4.24}$$

$$\vec{f_t} = \vec{f_{t-1}} + e^{j\vec{\theta_t}}x_t - e^{j\vec{\theta_t}}x_{t-N} \tag{4.25}$$

$$\vec{f_t} - \vec{f_{t-1}} + e^{j\vec{\theta_t}}x_{t-N} = e^{j\vec{\theta_t}}x_t \tag{4.26}$$

I let

$$\vec{g_t} = \vec{f_t} - \vec{f_{t-1}} + e^{j\vec{\theta_t}}x_{t-N} \tag{4.27}$$

so I have

$$\vec{g_t} = e^{j\vec{\theta_t}}x_t \tag{4.28}$$

Solving for x_t

$$e^{j\vec{\theta_t}}x_t = \vec{g_t} \tag{4.29}$$

$$\frac{1}{K}e^{j\vec{\theta_t}^{*T}}e^{j\vec{\theta_t}}x_t = \frac{1}{K}e^{j\vec{\theta_t}^{*T}}\vec{g_t} \tag{4.30}$$

$$\boxed{x_t = \frac{1}{K}e^{j\vec{\theta_t}^{*T}}(\vec{f_t} - \vec{f_{t-1}} + e^{j\vec{\theta_t}}x_{t-N})} \tag{4.31}$$

where K is the number of frequency channels being computed.

Chapter 5

Single Channel Frequency Estimation

5.1 Introduction

An SCFE is a high definition recursive digital filter. Each SCFE has a center frequency f, a Q, and an integration length N that determines the number of samples over which a frequency estimate is summed. Note that the concept of the SCFE is loosely related to *Goertzl's Algorithm* as it also analyses one selectable frequency component from a discrete signal [**Goertzel1958**].

5.2 Forward Algorithm

Define the center frequency of a frequency channel to be f. Then define the phase update values as

$$\Delta\theta = -2\pi(\frac{f_k}{f_s}) \tag{5.1}$$

For a given frequency channel f, set the Q to be as desired and then set the integration time for each channel as

$$N = \lfloor \frac{Q * f_s}{f_k} \rceil \tag{5.2}$$

I initialize $\theta = 0$ and $f_1 = e^{j\theta} x_1 = x_1$.

For each successive sample starting with t = 2:

$$\theta_t = \theta_{t-1} + \Delta\theta \tag{5.3}$$

$$\phi_t = \theta_{t-1} - \Delta\theta N \tag{5.4}$$

$$f_t = f_{t-1} + e^{j\theta_t} x_t - e^{j\phi_t} x_{t-N} \tag{5.5}$$

where x_p is defined for $p \geq 1$ and is otherwise zero. One thing to note here is that the phase accumulators θ_t and ϕ_t should be calculated with mod 2π so that the phase doesn't grow indefinitely.

According to this algorithm, phase updates are computed and stored in θ_t and ϕ_t. According to (5.5), each new frequency estimate is computed by summing the previous frequency estimate with a new modulated sample onto the front of the block of N samples and then demodulated off of the end of the block of N samples.

The algorithm can be summarized by the following equation that combines all three steps into one:

$$\boxed{f_t = f_{t-1} + e^{jt\Delta\theta} x_t - e^{j(t-N)\Delta\theta} x_{t-N}} \tag{5.6}$$

5.2.1 Analysis

Because the system is time-varying and not LTI, the frequency response of the system cannot be computed by taking the FFT of the impulse response. One way of

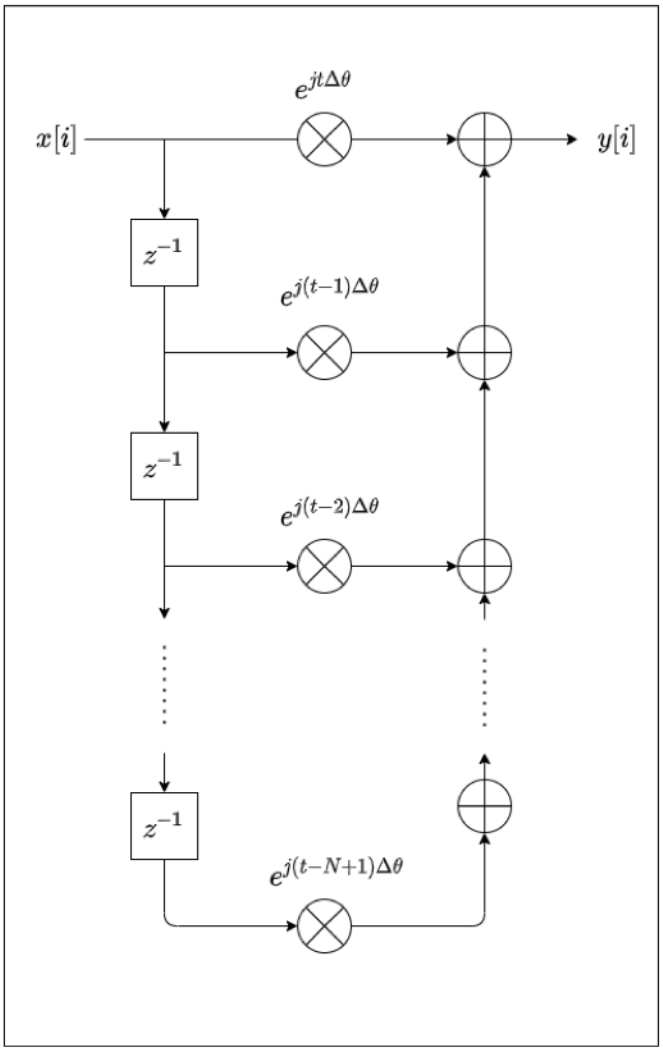

Figure 5-1: FIR filterbank which describes the SCFE rectangular summation

computing the frequency response of the system is to take a snapshot of the SCFE at a frozen time $t = i$, which is described by the FIR filterbank in figure (5-1). When $t = i$, the impulse response is a unit rectangle of length N. When you freeze time, the SCFE becomes an FIR system whose impulse response is a complex exponential. The frequency response plots have been generated by computing the 16k point DFT of the N-point complex exponential, padding with zeros.

Constant Q SCFEs

Here, I plot the frequency response of SCFEs for $f_k = 0.5$, 1, 2, 4, 8, and 16 kHz. A constant semitone Q is maintained. See figures (5-2) through (5-7).

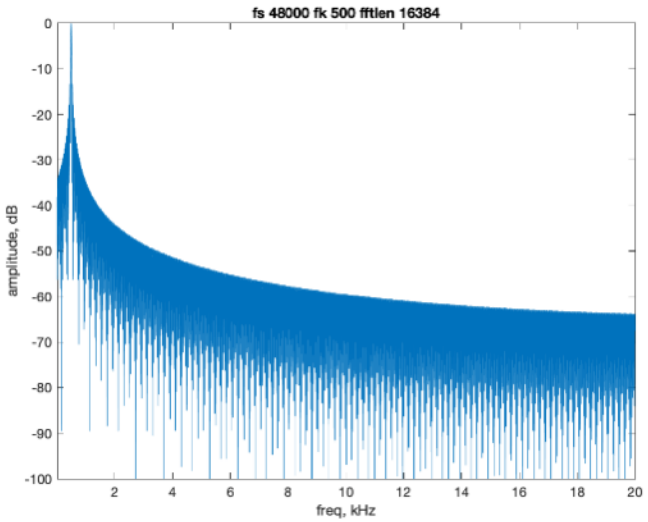

Figure 5-2: Constant Q SCFE frequency response for $f_k = 0.5$ kHz

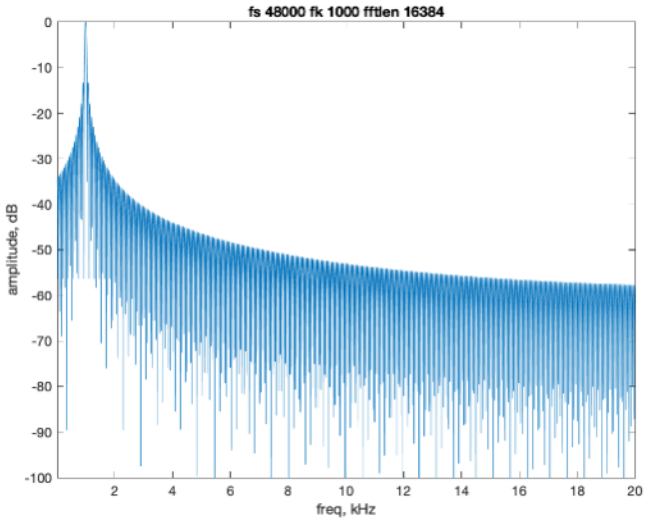

Figure 5-3: Constant Q SCFE frequency response for $f_k = 1$ kHz

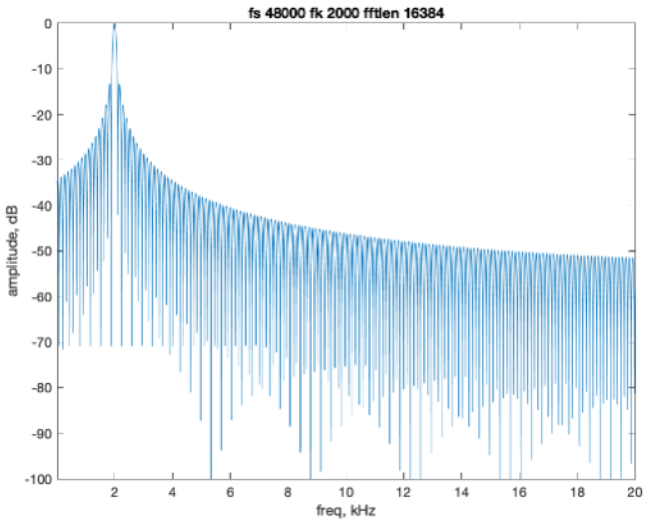

Figure 5-4: Constant Q SCFE frequency response for $f_k = 2$ kHz

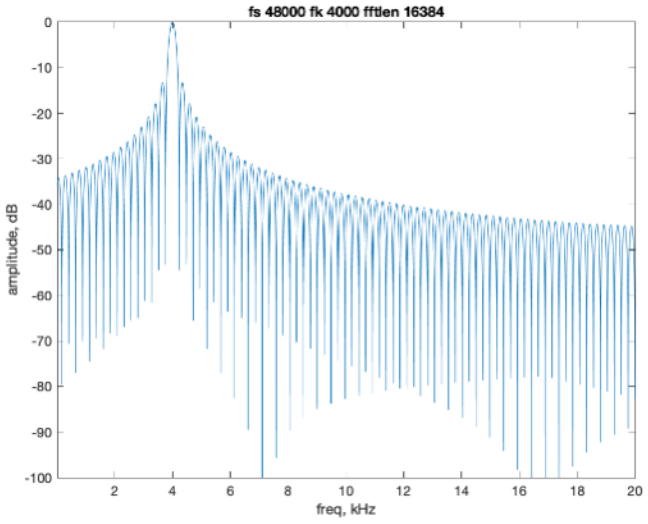

Figure 5-5: Constant Q SCFE frequency response for $f_k = 4$ kHz

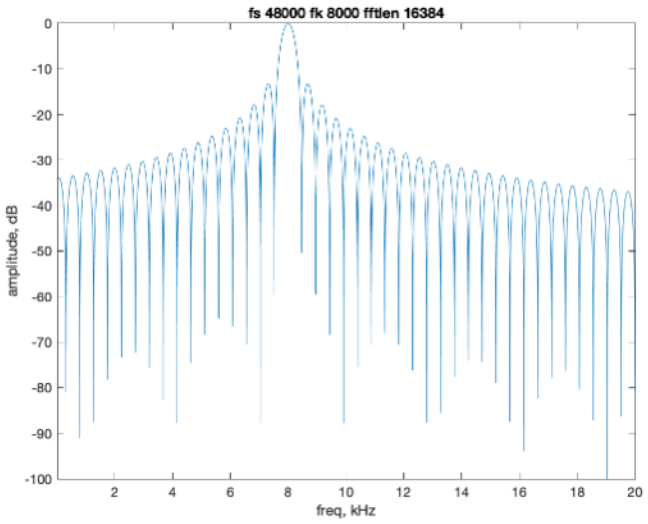

Figure 5-6: Constant Q SCFE frequency response for $f_k = 8$ kHz

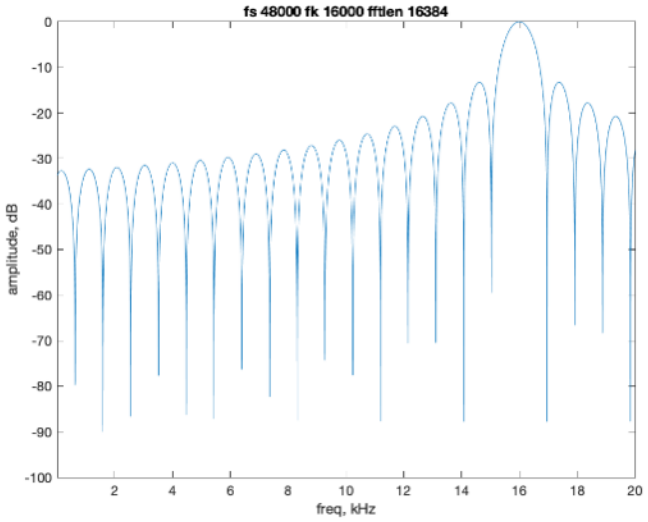

Figure 5-7: Constant Q SCFE frequency response for $f_k = 16$ kHz

A Generalized SCFE

It is possible to increase the frequency selectivity of a general SCFE at any center frequency by adjusting the integration length N as desired. This shows the general trade-off between time and frequency, where as N is longer, the filter is more frequency selective and as N is shorter, the filter is increasingly wider. Here I analyse $f_k = 5$

kHz, for $N = 125, 250, 500, 1000, 2000$, and 4000 samples, where the sample rate is 48 kHz. See figures (5-8) through (5-13).

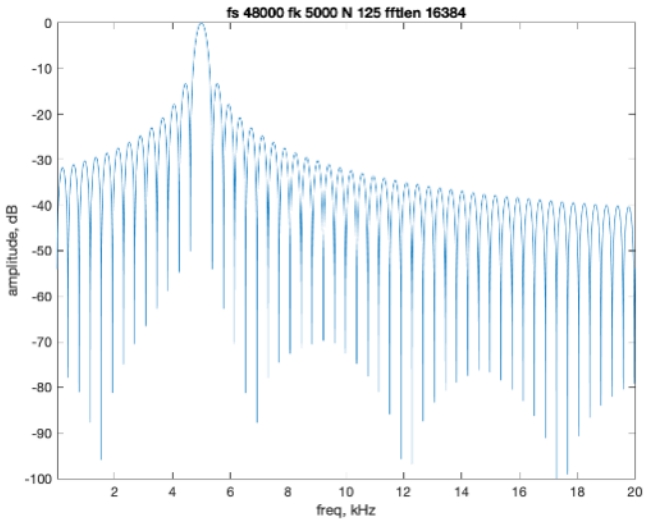

Figure 5-8: SCFE frequency response at 5 kHz, N = 125 samples

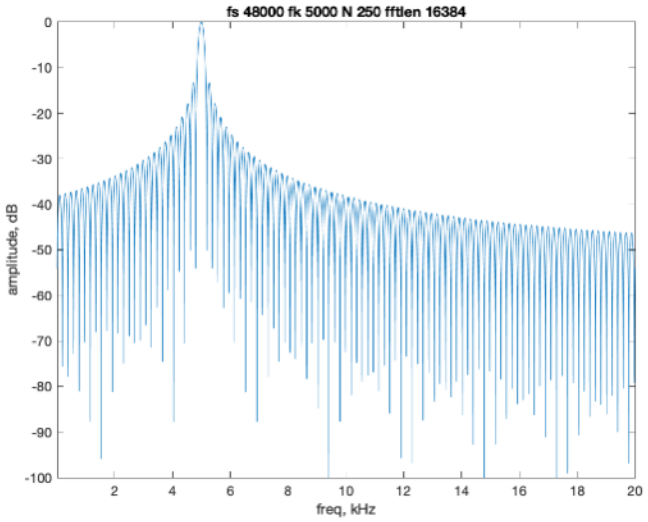

Figure 5-9: SCFE frequency response at 5 kHz, N = 250 samples

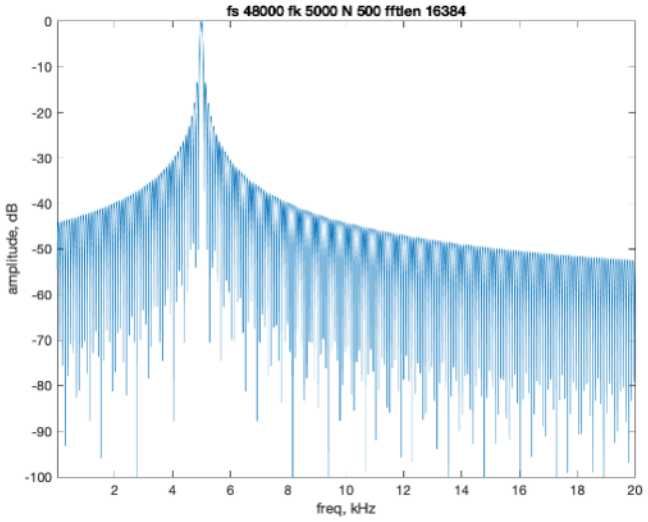

Figure 5-10: SCFE frequency response at 5 kHz, N = 500 samples

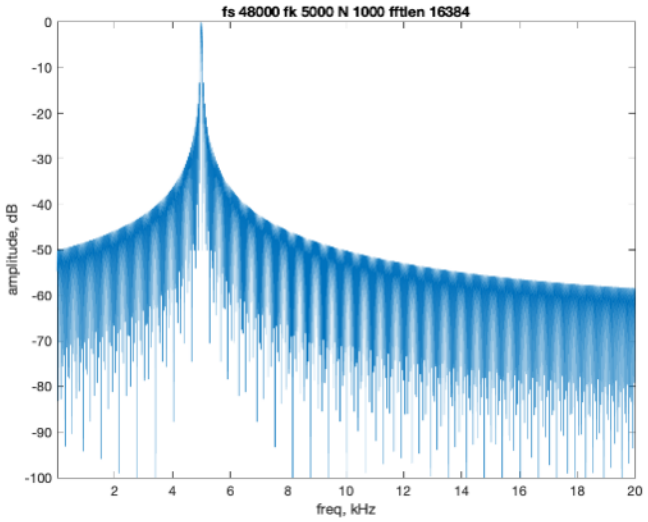

Figure 5-11: SCFE frequency response at 5 kHz, N = 1000 samples

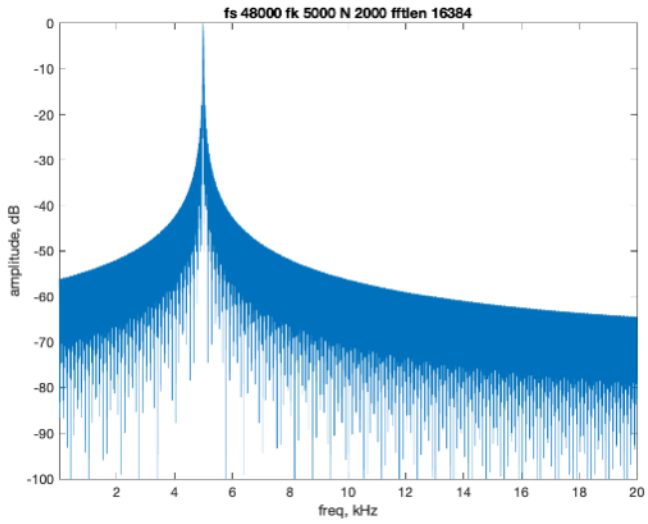

Figure 5-12: SCFE frequency response at 5 kHz, N = 2000 samples

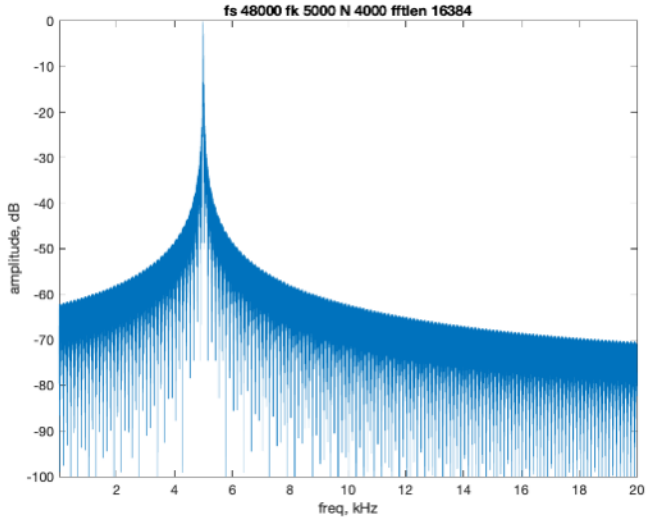

Figure 5-13: SCFE frequency response at 5 kHz, N = 4000 samples

5.2.2 Alternate Analysis

The following section suggests an alternate z-transform analysis to the SCFE system. I re-frame (5.6) as a traditional discrete time difference equation.

$$y_t = y_{t-1} + e^{jt\Delta\theta}x_t - e^{j(t-N)\Delta\theta}x_{t-N} \tag{5.7}$$

This difference equation has the following z-transform:

$$H(z,t) = e^{jt\Delta\theta}\frac{(1 - e^{-jN\Delta\theta}z^{-N})}{(1 - z^{-1})} \tag{5.8}$$

I simplify this by letting $\alpha = e^{-jN\Delta\theta}$ so the z-transform simplifies to

$$H(z,t) = e^{jt\Delta\theta}\frac{(1 - \alpha z^{-N})}{(1 - z^{-1})} \tag{5.9}$$

The z-transform shows that the system has one pole and N zeros with complex coefficients.

5.2.3 Examples

It is possible to analyse just the harmonics of a given musical note by tuning SCFEs to the harmonics of a given fundamental frequency. See figures (5-14) through (5-17). Note that this kind of analysis is equivalent to one column in the M direction of the *Augmented GST*.

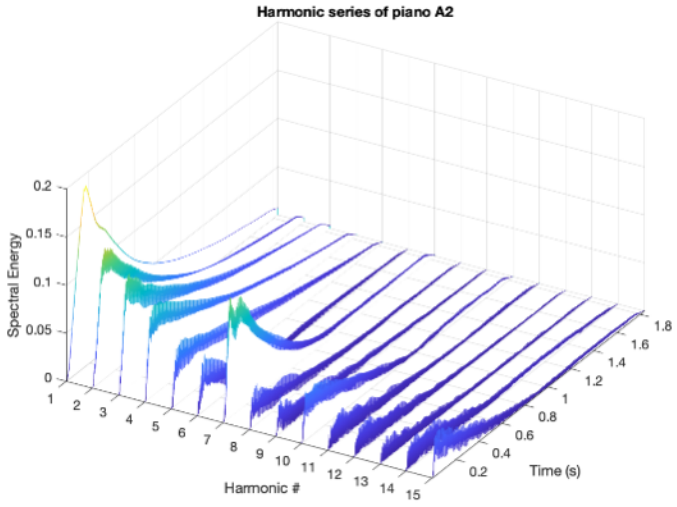

Figure 5-14: Harmonic analysis of A2 on the Piano

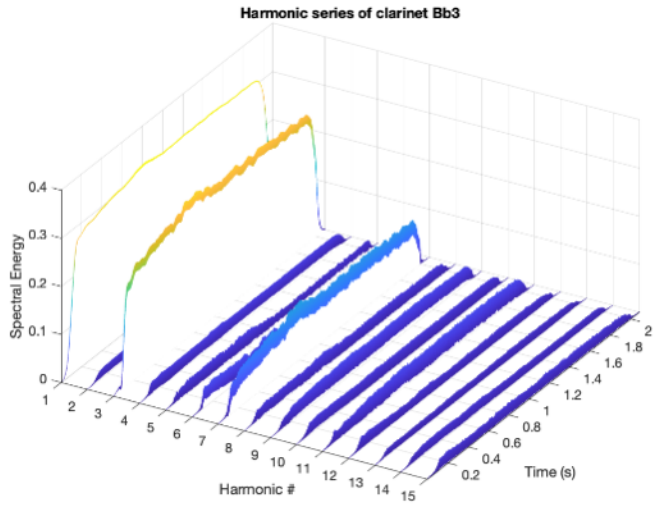

Figure 5-15: Harmonic analysis of Bb3 on the clarinet

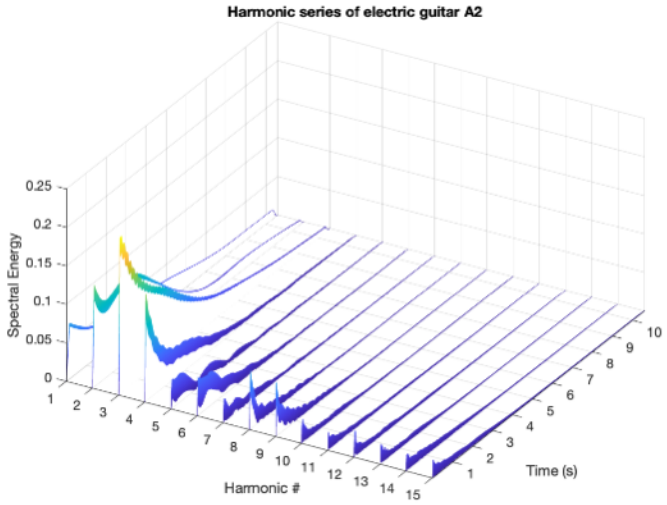

Figure 5-16: Harmonic analysis of A2 on the electric guitar

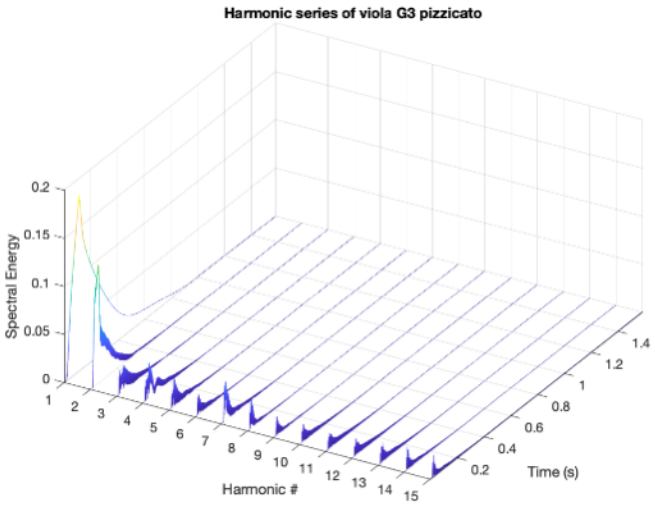

Figure 5-17: Harmonic analysis of G3 on the viola played pizzicato

5.3 Inverse Algorithm

The inverse SCFE algorithm will invert a single channel of analysed audio back into the entire original signal with perfect reconstruction. In Appendix A, it is shown how to modify this algorithms to be used as a monophonic synthesizer. The derivation is straight forward:

$$f_t = f_{t-1} + e^{j\theta_t} x_t - e^{j\phi_t} x_{t-N} \tag{5.10}$$

$$f_t = f_{t-1} + e^{j\theta_t} x_t - e^{j\vec{\phi_t}} x_{t-N} \tag{5.11}$$

$$e^{j\theta_t} x_t = f_t - f_{t-1} + e^{j\phi_t} x_{t-N} \tag{5.12}$$

$$e^{j\theta_t{}^*} e^{j\theta_t} x_t = e^{j\theta_t{}^*} (f_t - f_{t-1} + e^{j\phi_t} x_{t-N}) \tag{5.13}$$

$$\boxed{x_t = e^{-j\theta_t}(f_t - f_{t-1} + e^{j\phi_t}x_{t-N})} \tag{5.14}$$

The algorithm is then as follows:

Initialize at t = 1: $f_1 = x_1$ and $\theta_1 = 0$.

$$\Delta\theta = -2\pi(\frac{f_k}{f_s}) \tag{5.15}$$

Starting at t = 2:

$$\theta_t = \theta_{t-1} + \Delta\theta \tag{5.16}$$

$$\phi_t = \theta_{t-1} - \Delta\theta N \tag{5.17}$$

$$\boxed{x_t = e^{-j\theta_t}(f_t - f_{t-1} + e^{j\phi_t}x_{t-N})} \tag{5.18}$$

where x_p is valid for all $p \geq 1$ and is zero otherwise.

Chapter 6

Geometrically Spaced Transform

6.1 Introduction

The *Geometrically Spaced Transform* (GST) consists of a bank of *Single Channel Frequency Estimators* (SCFEs) whose center frequencies are spaced $f_{k+1} = 2^{\frac{1}{12}} f_k$ apart to model the fundamental frequencies of the notes of the piano. Each channel has a summation constant N_k that is set to ensure that the channel maintains its constant Q property. Because N_k gets shorter with increasing frequency, the GST is like a wavelet transform; however, there aren't fewer frequency estimates at higher frequencies than at lower frequecies. Rather, the number of samples used to create the high frequency estimates is fewer than that of lower frequencies. The resulting matrix is rectangular. This is similar to sampling a *Continuous Wavelet Transform* that is logarithmically-spaced in frequency and linearly-spaced in time. A GST is the first slice of an *Augmented GST* which is representative of harmonic number one, or the fundamental frequencies of the piano notes.

6.2 Forward Algorithm

Note that for the previous part of the thesis, the algorithms and graphs used a constant N. From here forward, note that N is now set to make the algorithm constant Q and is denoted N_k for the k^{th} frequency channel.

I define **x** to be the input signal, f_s to be the sample rate, N_k to be the window length of the k^{th} frequency channel, **B** to be the base band frequency, and **M** to be the number of frequency channels starting at **B** Hz.

I compute the center frequencies of each frequency channel based on **B** and a division of 12 musical notes in an octave:

$$f_k = B * 2^{\frac{k-1}{12}} \tag{6.1}$$

and then the phase update values accordingly:

$$\Delta\theta_k = -2\pi(\frac{f_k}{f_s}) \tag{6.2}$$

For a given frequency channel f_k, I set

$$Q = \frac{1}{2^{\frac{1}{12}} - 1} \tag{6.3}$$

$$N_k = \lfloor \frac{Qf_s}{f_k} \rceil \tag{6.4}$$

as in [**Brown1990**].

At t = 1, $\theta_{k,1} = 0$ and $f_{k,1} = e^{j\theta_k}x_1 = x_1$

Consider frequency channel **k** at time **t**.

Now for each successive sample starting with t = 2:

$$\theta_{k,t} = \theta_{k,t-1} + \Delta\theta_k \tag{6.5}$$

$$\phi_{k,t} = \theta_{k,t} - \Delta\theta_k N_k \tag{6.6}$$

$$f_{k,t} = f_{k,t-1} + e^{j\theta_k}x_t - e^{j\phi_k}x_{t-N_k} \tag{6.7}$$

where x_p is defined for $p \geq 1$ and is zero otherwise. This algorithm can be

summarized as follows, with this time t beginning at t = 0:

$$\boxed{f_{k,t} = f_{k,t-1} + e^{jt\Delta\theta_k} x_t - e^{j(t-N_k)\Delta\theta_k} x_{t-N_k}} \tag{6.8}$$

The algorithm is summarized in the signal flow diagram shown in figure(6-1), which includes a normalization factor of $\frac{1}{N_k}$. The GST block lengths that make the algorithm constant Q are shown in figure (6-2). The summation that happens according to the SCFE linear update rule happens over longer windows of samples for lower frequencies and shorter windows for higher frequencies.

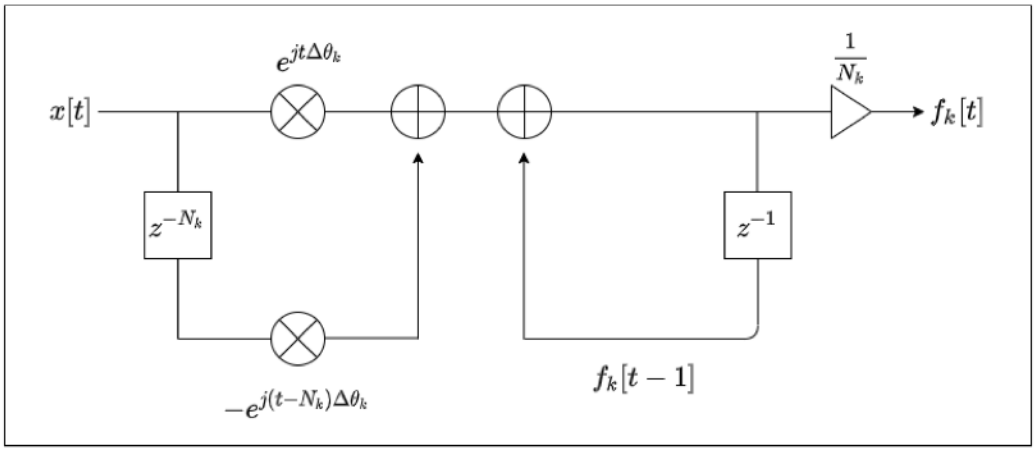

Figure 6-1: Signal flow diagram for GST algorithm

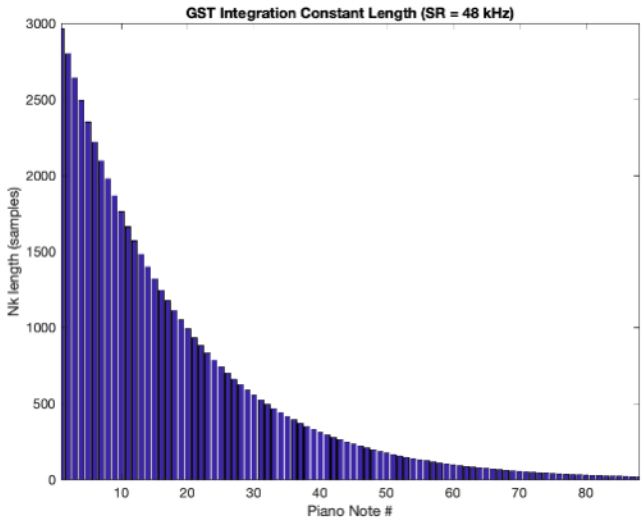

Figure 6-2: Block lengths for GST algorithm

The frequency response of the GST is shown for every note A on the piano in figure (6-3) to -60 dB, and for the entire GST filterbank to -12 dB in figure (6-4).

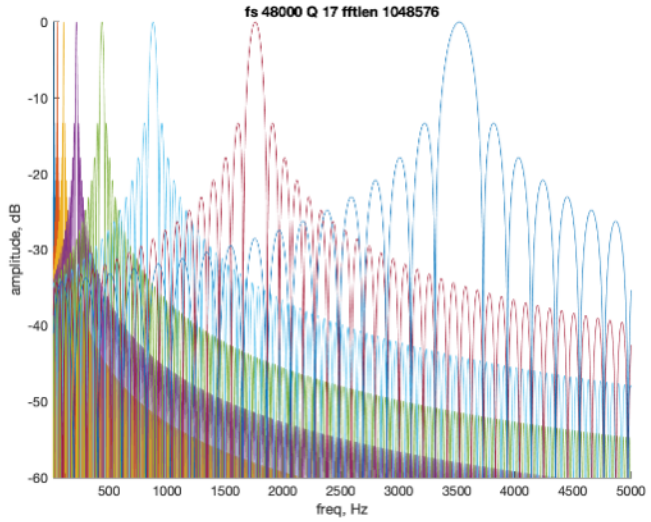

Figure 6-3: Magnitude response of the GST filters for every note A on the piano

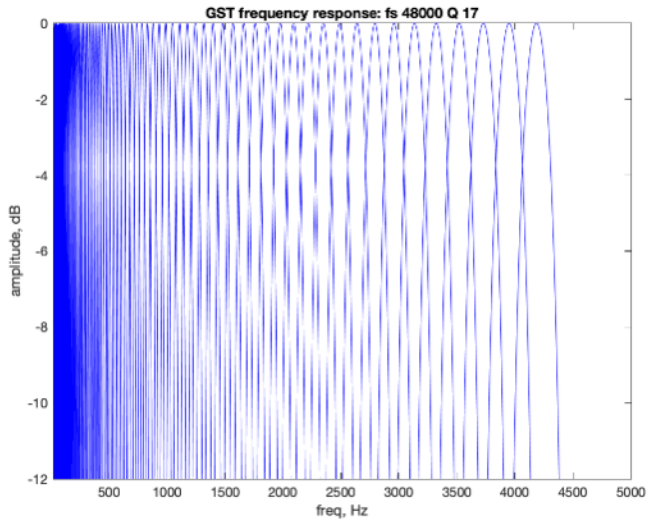

Figure 6-4: Magnitude response of the 88 filters of the GST filterbank

Note that the GST filterbank can be compared to the well known Gammatone filterbank, which is a popular auditory filter model. See figure (6-5).

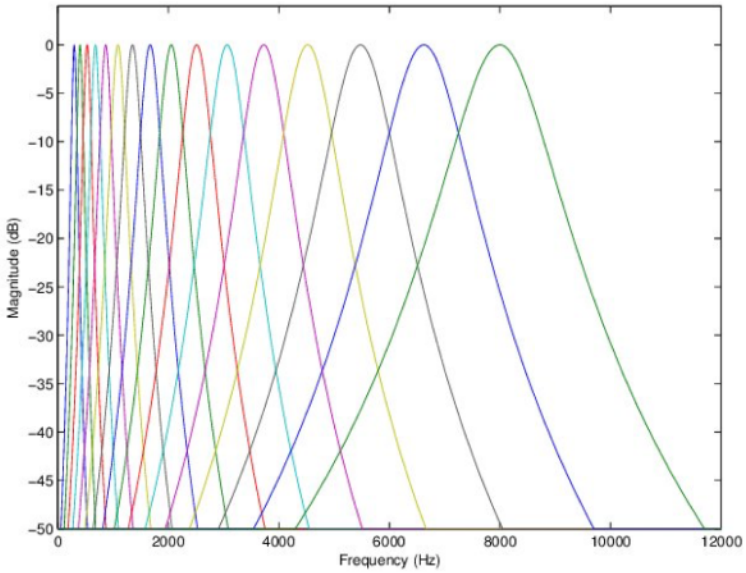

Figure 6-5: Magnitude responses of 16 gammatone filters in the frequency range 300-8000 Hz

6.3 Calculation of Q

An exact semi-tone Q, according to equation (6.3), would have a value of

$$Q = 16.817153745\ldots \tag{6.9}$$

The Q of each frequency channel in the GST can be calculated as follows:

$$Q = \frac{\Delta\theta_k N_k}{-2\pi} \tag{6.10}$$

Note that in equation (6.10), because the calculation for N_k is rounded off, the Q for the set of 88 frequency channels in the GST is very close to constant, but is not perfect. See figure (6-6). For the GST, the maximum error in the Q for the implemented filterbank is 0.0252. Note the the scale of the y axis in figure (6-6) goes from 16.78 to 16.84, where the desired Q is approximately 16.81. The Q is closer to constant at lower frequencies and the error in the Q is largest at higher frequencies.

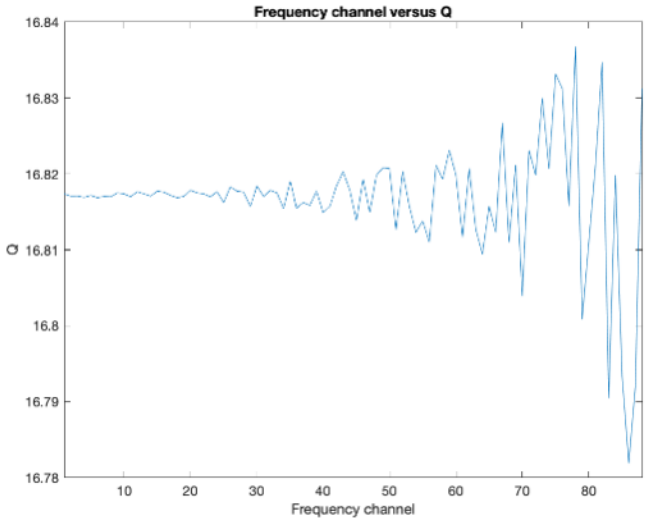

Figure 6-6: Q of each GST channel

6.4 Examples

I present a number of graphical examples of the GST, including an analysis of the piano in figures (6-7) through (6-9), electric guitar in figure (6-10), bass trombone in figure (6-11), and tambourine in figure (6-12).

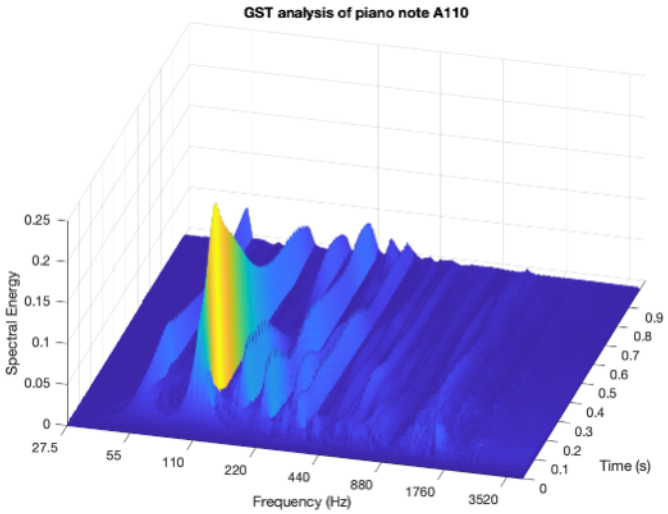

Figure 6-7: GST analysis of piano note A110

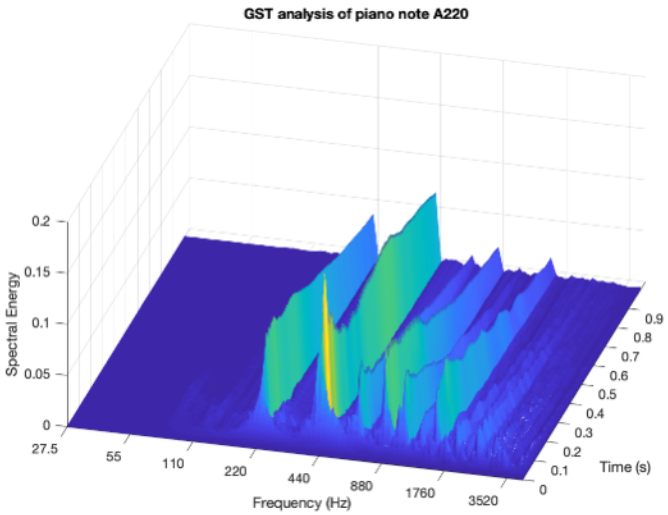

Figure 6-8: GST analysis of piano note A220

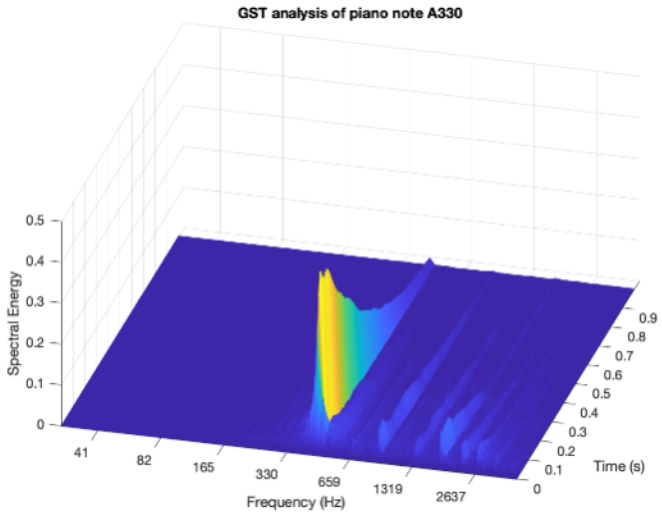

Figure 6-9: GST analysis of piano note A330

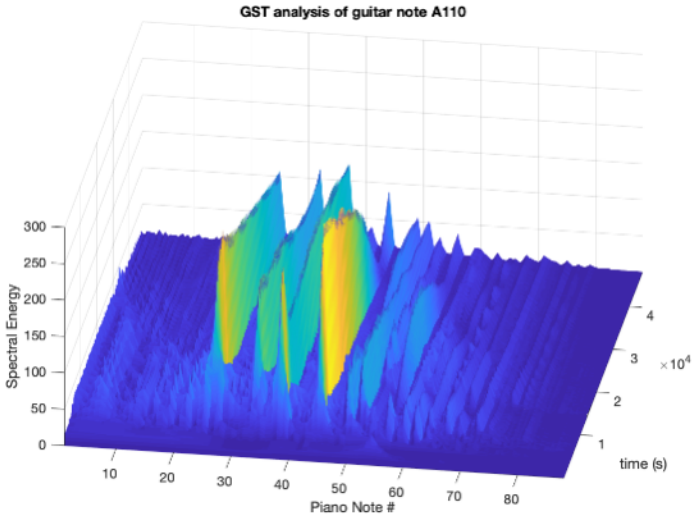

Figure 6-10: GST analysis of guitar note A110

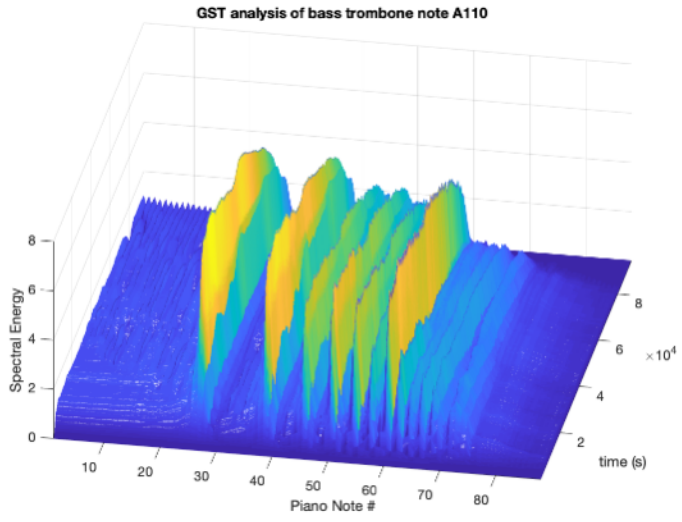

Figure 6-11: GST analysis of bass trombone note A110

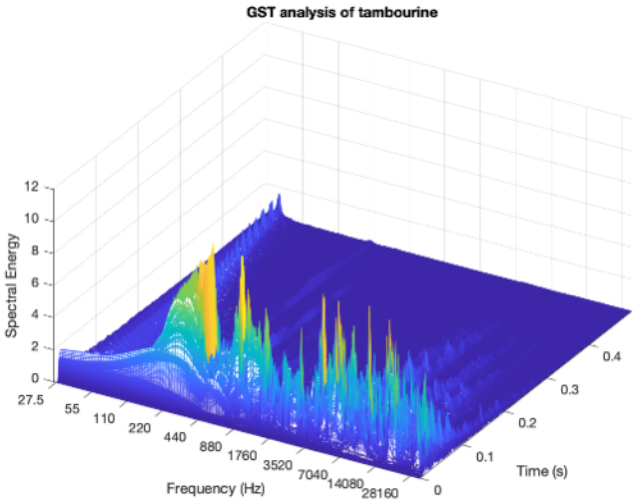

Figure 6-12: GST analysis of a tambourine slap

6.5 Inverse Algorithm

In Appendix A, it is shown how the inverse GST algorithm can be modified to be a polyphonic synthesizer. The derivation is similar to the inverse SCFE, except that here it is expressed in vector form with K SCFE channels.

$$\boxed{x_t = \frac{1}{K} e^{j\vec{\theta_{t,k}}^{*T}} (\vec{f_t} - \vec{f_{t-1,k}} + e^{j\vec{\phi_{t,k}}} x_{t-N_k})} \tag{6.11}$$

Chapter 7

Computational Pitch Perception

7.1 Augmented Geometrically Spaced Transform

The *Augmented Geometrically Spaced Transform* is a matrix of SCFEs that are computed for each frequency of the *Augmented GST Frequency Matrix* (AGST Frequency Matrix). Each entry in the *AGST Frequency Matrix* contains a center frequency for which an SCFE is computed. Note that for the AGST, non-normalized SCFEs are used. The matrix contains the fundamental frequencies of the piano along the K (fundamental frequency) axis, and then harmonics of these fundamental frequencies along the M (harmonics) axis. See figure (7-1). In comparison, figure (7-2) shows the subset of frequencies in the *AGST Frequency Matrix* that lie from 20 Hz to 20 kHz. I make the comparison to this figure because in the AGST model, SCFEs are computed for frequencies that lie from 3.4375 Hz to 83720 Hz, which includes frequencies that lie outside the normal range of human hearing which is from 20 Hz to 20 kHz. Note that because the highest computed frequency is 83720 Hz, the sample rate of the system would have to be at least this value to avoid aliasing in the *Augmented GST*. Typically in audio this would be 96 kHz.

The *Augmented GST* can be thought of as a movie, where the K x M frequency analysis unfolds over time t. Each snapshot in time is a low level representation of pitch, where the x axis includes the notes of the piano keyboard, but extended by three octaves below the piano in order to be able to represent the pitch from the lowest

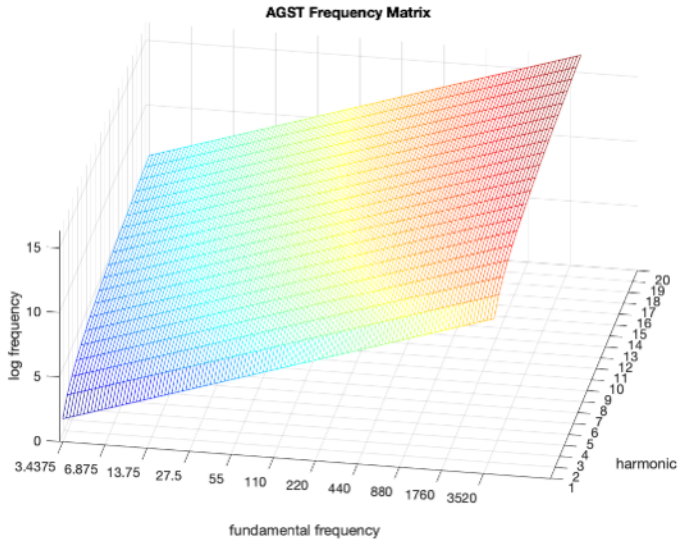

Figure 7-1: AGST Frequency Matrix that extends from 3 Hz to 83720 Hz

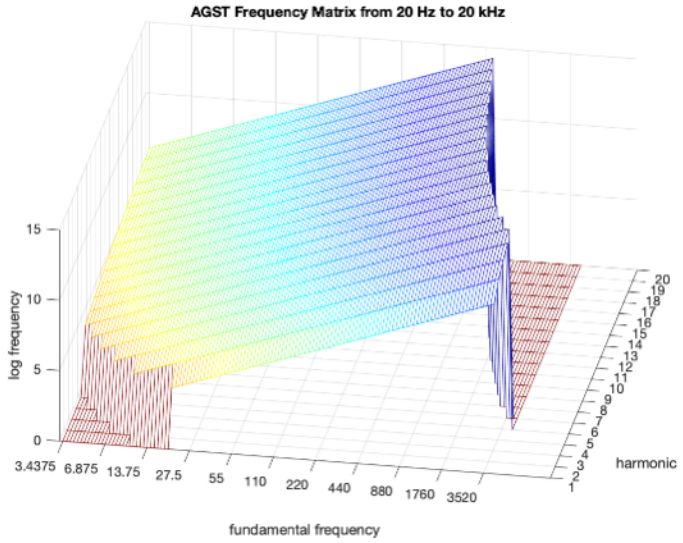

Figure 7-2: AGST Frequency Matrix that shows which frequencies lie from 20 Hz to 20 kHz

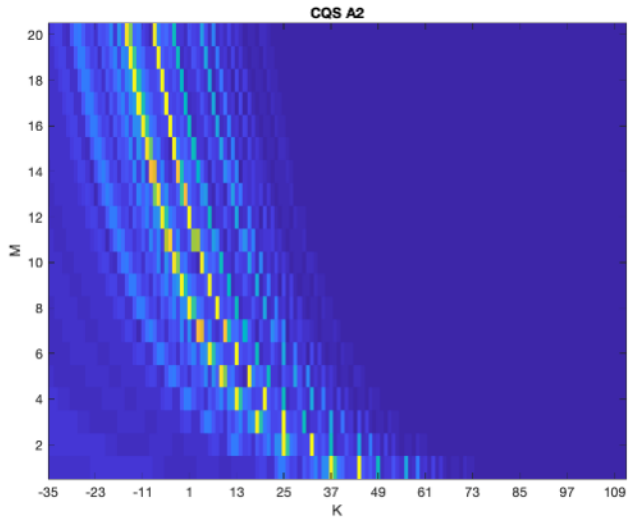

Figure 7-3: Snapshot of AGST of piano note A2

note (A0 with a fundamental frequency of 27.5 Hz) to the highest note (C8 with a fundamental frequency of 4186 Hz). For examples of the AGST taken as a snapshot at time t = 0.1 seconds into a number of piano samples, see figures (7-3) through (7-6). Note that piano note 37 corresponds to the note A2. Piano note 41 corresponds to C#3. Piano note 44 corresponds to E3. And piano note 47 corresponds to G3.

The system is designed for the purposes of listening to Western music that is tuned to an equal tempered scale based around the note A440, which has a fundamental frequency of 440 Hz. Note here that I use the term *augmented* because the range of frequencies computed are not just the fundamental frequencies of piano notes but also frequencies that lie both below and above this frequency range. While it is said that the frequency range of listening is from 20 Hz to 20 kHz, I would suggest that the frequencies below 20 Hz and above 20 kHz are not perceived but are still computed by a model such as this one.

7.2 The Summed AGST

In the *Summed AGST*, spectral energy is summed across the harmonics to produce a single spectral energy estimate that is representative of all of the spectral energy

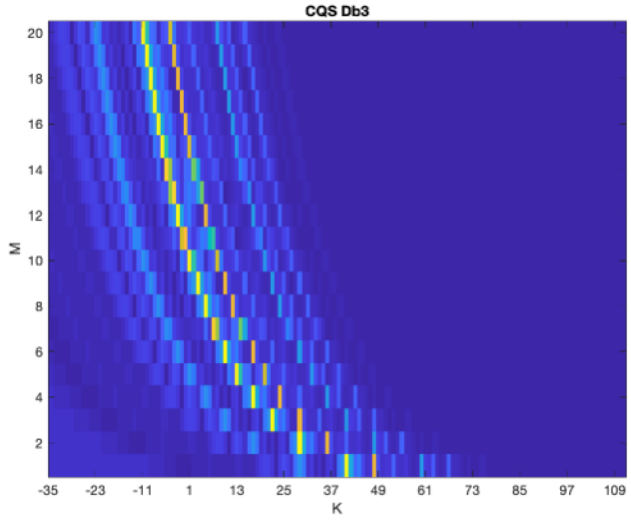

Figure 7-4: Snapshot of AGST of piano note C#3

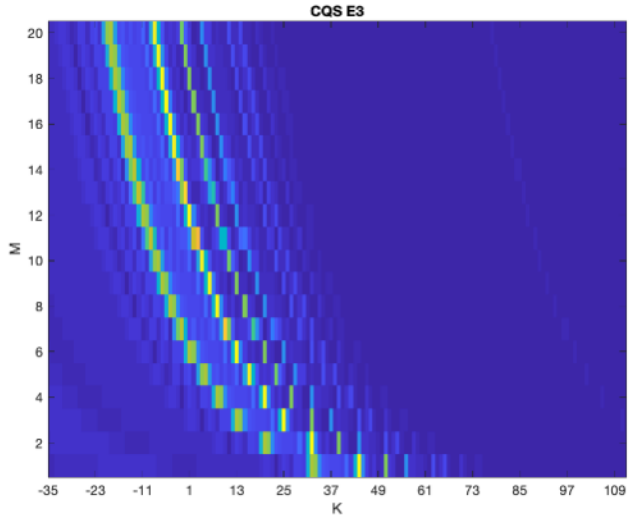

Figure 7-5: Snapshot of AGST of piano note E3

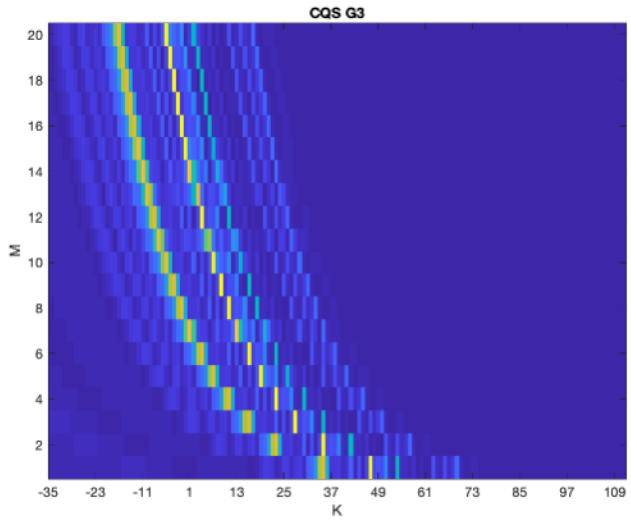

Figure 7-6: Snapshot of AGST of piano note G3

that is produced by one harmonic series. See figure (7-9). What appears to be revealed is a subharmonic series, or undertone series, that is related to the fundamental frequency according to the following relationship: $(f, \frac{f}{2}, \frac{f}{3}, \frac{f}{4}, \ldots)$. These harmonics are not actually produced sonically by acoustic instruments, but they are revealed here computationally through the *Summed AGST*. In section (7.4), it is shown that it is possible to implement a simple monophonic pitch tracking algorithm using this representation. It still needs to be shown that the *Summed AGST* is a useful representation for perception of musical chords. Note that listening to a musical chord may be akin to listening to a *chimera*, as described in [**Bregman1990**], and that the relative strength of subharmonics may be a good input representation to a pattern classifier.

7.3 Onset Detection

The onset represents the time at which a musical note is first perceived, and detecting the actual onset time of a music note has been much debated with a myriad of ways to compute it [**Sandler2005**]. Here, I proposed a representation of a spectral envelope that could be used for onset detection, which is the summation over the spectral

envelopes of the subharmonic series produced by the *Summed AGST*. See figure (7-7).

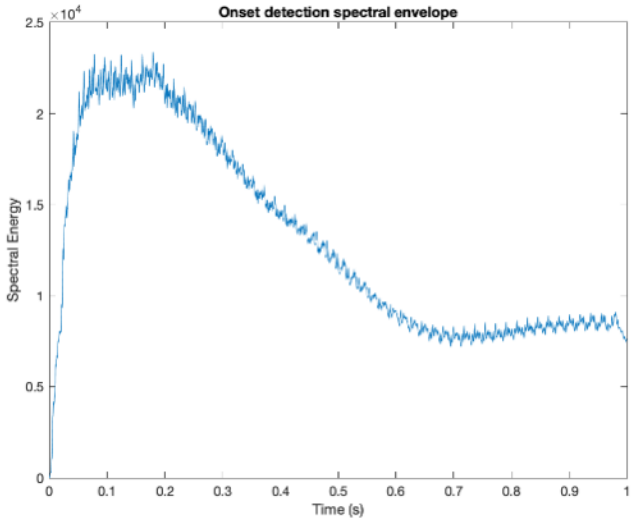

Figure 7-7: Spectral envelope produced by the sum of the subharmonics

7.4 Pitch Perception

7.4.1 Pitch Tracking

An algorithm is proposed for pitch estimation that is based on the perceptual representation that is revealed by the *Summed AGST*. See figure (7-8). First, the signal is transformed by the AGST. Then, the *Summed AGST* is computed by summing the harmonics of the AGST. The *Summed AGST* reveals a subharmonic series for the input signal, which I propose as a perceptual representation of pitch. Using this representation, an algorithm is applied to produce pitch estimates. Here, I use a simple peak picking algorithm that computes the highest subharmonic in the subharmonic series, which corresponds to the fundamental frequency of the input note. A better approach might be to integrate information across the subharmonic series to compute pitch. For examples of analysing monophonic signals, see (7-9), (7-10), (7-11), and (7-12).

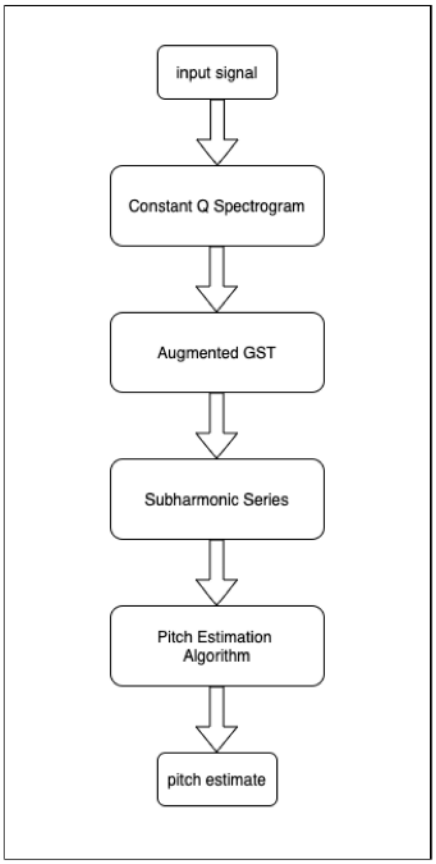

Figure 7-8: Algorithm for pitch estimation

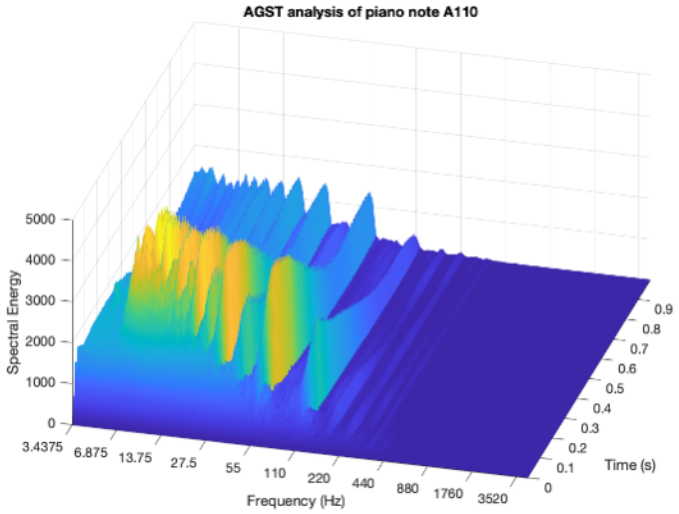

Figure 7-9: Summed AGST for piano note A110

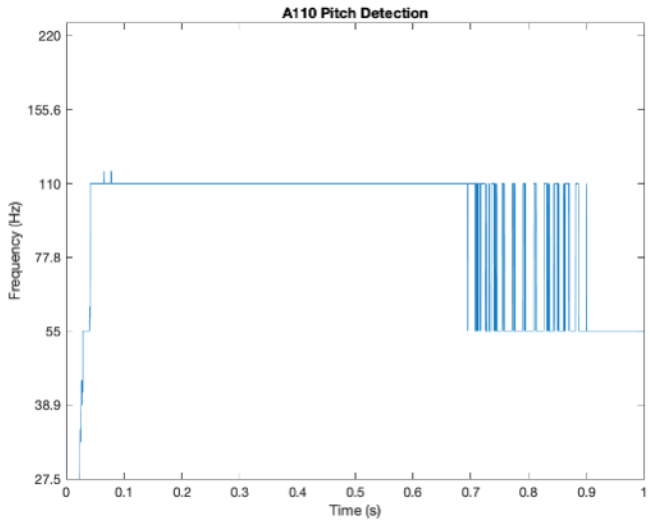

Figure 7-10: Pitch estimation algorithm output for piano note A110

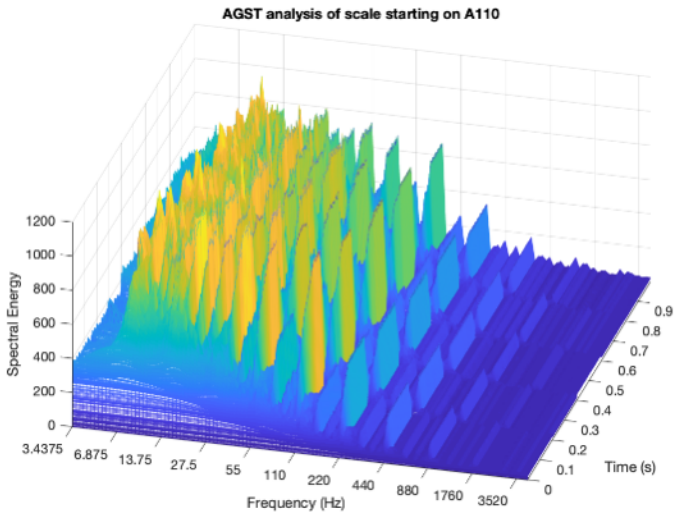

Figure 7-11: Summed AGST for major scale starting on A110

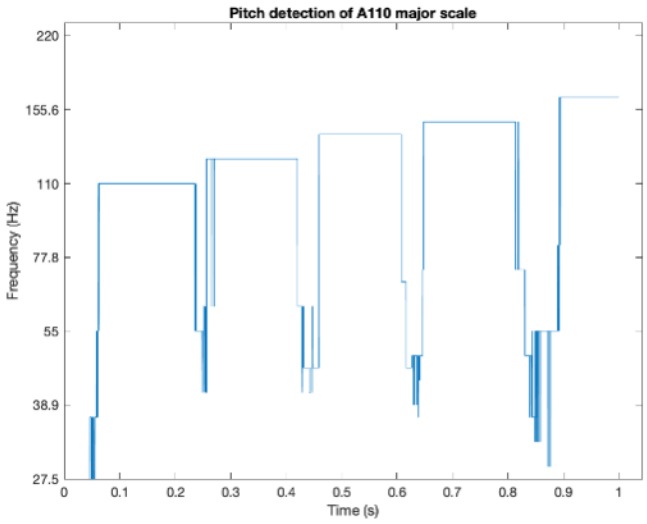

Figure 7-12: Pitch estimation algorithm output for major scale starting on A110

7.4.2 The Case of the Missing Fundamental

A pitch is said to have a missing fundamental when its harmonics suggest a fundamental frequency that is not physically present in the sound itself. The brain perceives the pitch by the periodic relationship between the higher harmonics. That is, a number of different tones that each contain variations of periodically related harmonics may be perceived as the same pitch even if the fundamental frequency is missing.

One way to know that the AGST is a perceptual model is that it solves such problems related to pitch perception [**Moore1994**]. With this in mind, I investigate the *Summed AGST* for a synthesized sum of sine waves with a harmonic series based on the fundamental frequency of 196 Hz, but with the 196 Hz fundamental frequency missing. See figure (7-13). I process this input with the AGST, and sum over the harmonics to compute the *Summed AGST*. I find that the *Summed AGST* actually computes the the fundamental frequency that is missing. See figure (7-14). This is possible to show this because the harmonic series of G196 is represented multiple times in the *AGST Frequency Matrix*, where for instance 98 Hz has a harmonic of 196 Hz, and 49 Hz has harmonics of both 98 Hz and 196 Hz.

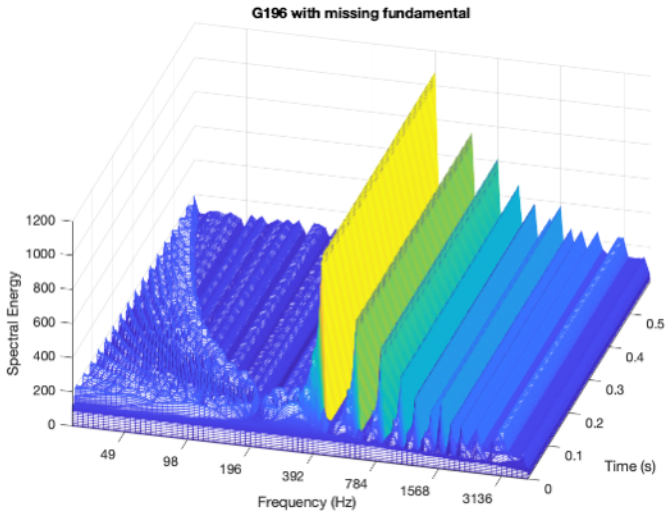

Figure 7-13: GST analysis of G196 synthesized with a missing fundamental

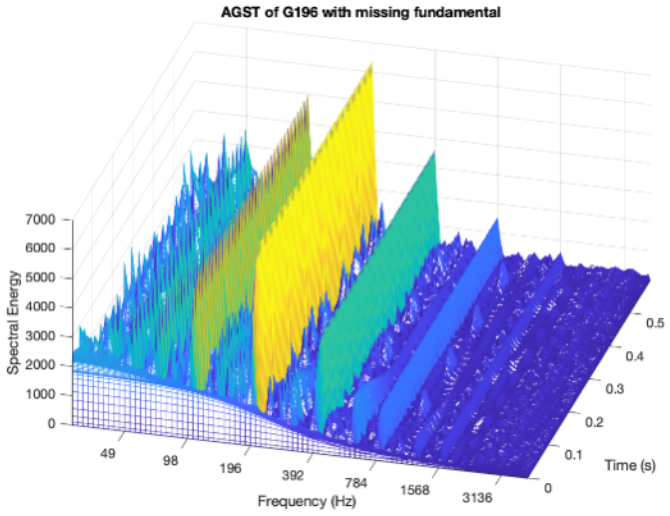

Figure 7-14: Summed AGST of G196 with missing fundamental with pitch perceived as 196 Hz

The following is another example of a synthesized sound for G196, this time with the lowest 4 harmonics missing. See figure (7-15). The *Summed AGST* still computes the missing fundamental. See figure (7-16).

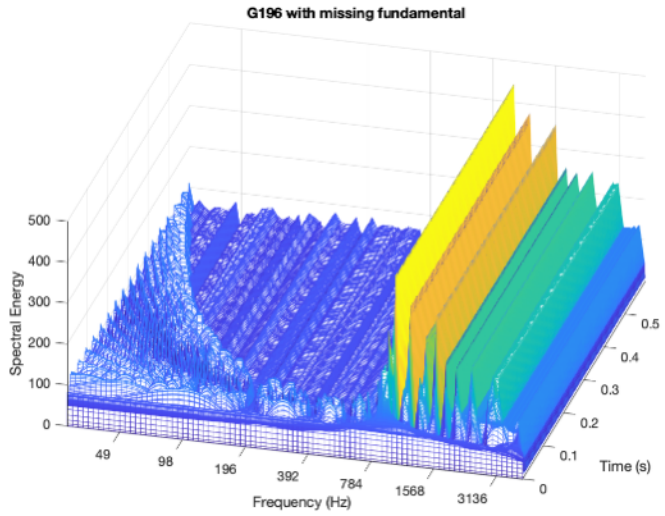

Figure 7-15: GST analysis of G196 synthesized with a missing lowest 4 harmonics

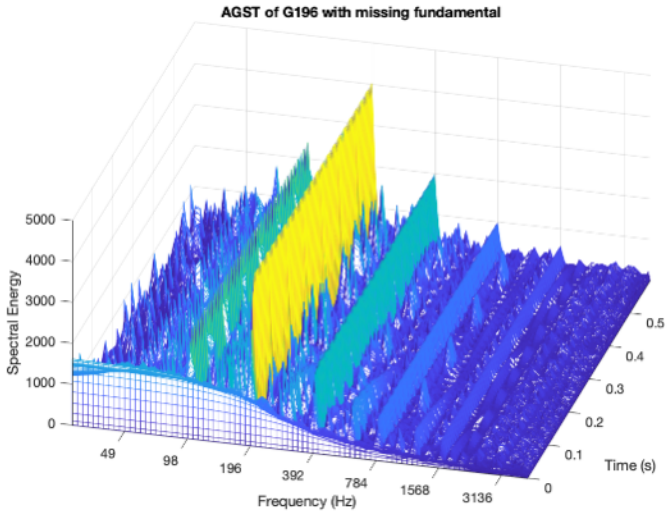

Figure 7-16: Summed AGST of G196 with missing lowest 4 harmonics with pitch perceived as 196 Hz

7.5 Analysis and Future Work

The *Summed AGST* is a useful as a perceptual respresentation of pitch. The idea is that instead of computing pitch based on the GST, the *Summed AGST* is used instead

because it integrates spectral information across the harmonics that are produced by any given note and produces a subharmonic series which is well known in acoustics. In the proposed model, pitch is extracted from this subharmonic series. Also note that the pitch onset, which is a perceptual property that needs to be very exact, is proposed to be extracted from the spectral envelope formed from the sum across the subharmonic series.

Here, I return to the concept of Western harmony. The most common compositional tool that is used to create music in the West is the piano. The emotional experience of listening to Western music is related to a response to $f(k, m, t)$, which is the *Augmented GST* but thought of as a synthesizer. The AGST is an inversion of the physical system that produces music on the piano because it is a machine listening analysis model. That is, the system that listens to it is related to the system that produces it. Because it is a physical system, properties like pitch perception arise from the physics where a perceptual model is an extension built upon a foundation of a front-end signal processing model such as the *Augmented GST*. The *Summed AGST* is then an example of a first level of perception where a perceptual representation appears by summing across the M harmonics of the AGST.

Chapter 8

Conclusions

The original goal was to build a system where frequency estimates are updated for every sample that enters the system with the aim of computing a time-frequency matrix with maximum time resolution. A method for accomplishing this is to use the STFT with a step size of one sample. This, however, leads to computing an FFT at every time step (i.e. with every new sample). To process N samples, the cost of the STFT, if used in this way, is $O(NMlogM)$ where M is the frame size, which corresponds to $O(N^2logN)$ when N = M. I observe the periodic nature of the *Discrete Fourier Matrix* and derive a linear update rule for computing a per sample time-frequency representation that is similar in spirit to an STFT [**Smith2011**] and a Sliding DFT [**Jacobsen2003**], which I call the *Online Fourier Transform*. I then derive a method for computing this transform using phase updates. This in turn led to a method for computing an online *Geometrically Spaced Transform*, which is constant Q. The average cost of processing N_k samples using the GST is $O(KN_k)$, where K is the number of frequency channels being estimated.

From the GST, I extract the *Single Channel Frequency Estimator*, which is the algorithm which produces each row of the GST. I initially approach the problem of pitch detection as follows. I build a matrix of K x M SCFE estimators. For example, there is a bank of M estimators for each harmonic series of each of the K notes along the frequency axis. I then sum the spectral energy of the harmonics to produces a spectral graph that has K estimates. This is called the *Summed AGST*. Analysis

of a pitched audio sample using the *Summed AGST* displays the subharmonics of the input sample. I propose that this subharmonic series is used to compute pitch, and I show a simple method where this is possible using peak picking. Finally, I theorize that through careful mathematical modelling and analysis, it can be shown that perception is just another part of computational processing, including auditory scene analysis. This approach is possible if the correct input representation is derived carefully. In the end, I refine the model and let $K = 160$, which corresponds to the 8 octaves of the piano, 3 octaves below the piano, and 3 octaves above the piano. The extended three octaves that lie below the range of the piano are necessary in order to be able to compute the AGST of the lowest note on the piano, with fundamental frequency of 27.5 Hz, and the three octaves above the piano are necessary in order to be able to compute the GST of the highest notes of the piano.

The *Augmented Geometrically Spaced Transform* is used here as a front end for computational pitch perception, but it may be possible to use it as a general front-end auditory model. One extension of this work would be to use it as a front end for musical instrument identification, as in [**Martin1998**]. Also, [**Scheirer1997**] computes spectral envelopes as the front end of his beat tracking system, which is also what the AGST already does. So it may be possible to use the *Constant Q Spectrogram* as a general front-end auditory processing framework for real-time machine listening systems.

The auditory system evolved over time. The auditory system may have learned to compute pitch, beats, timbre, and other perceptual properties as an evolutionary process. It is interesting, for example, to compare the auditory processing of humans to the auditory processing of other animals. With regards to sound and music, what do animals perceive? [**Walker2019**] In terms of anthropology, how does the auditory processing of one culture compare to another? It may be that for people who have evolved to listen to music in the Western European tradition, auditory perception operates one way and that for people in Africa, India, Bali, and other cultures, their auditory perception is different [**McDermott2019**]. This may explain, for example, why an auditory system that knows how to listen to classical music finds it challenging

to learn to listen to an Indian raga, or Balinese gamelan, and that it takes time to learn how to perceive and enjoy music from other cultures. Clearly, auditory perception is complex and no one model can explain the perception of music in every culture. In fact, it may be that everyone's perception of music is slightly different, depending on what music that person has listened to in their lifetime. What is inherited and what is learned is an open question, this is a nature versus nurture kind of argument. These thoughts may be part of a field of study that may be called *computational ethnomusicology*, where modelling of auditory perception varies from culture to culture [**Kippen1992a**] [**Kippen1992b**] [**Kippen1994**].

The scope of this thesis is broad and specific at the same time. The model proposed here is based around music that is composed on the piano, which has been a dominant instrument for music composition since its inception around the year 1700. Its range of pitches run from 27.5 Hz to 4186 Hz. The history of keyboard-based music composition goes back in time much further, to include music composed on the harpsichord and clavichord. Music in the West has changed dramatically in the 20th century, as music transitioned from classical music to new styles like blues, jazz, rock n roll, reggae, atonal music, and a myriad of other styles and art forms. Also, the emergence of the electric guitar as a dominant compositional tool for many styles has changed music as well. The human perceptual system is challenged when it encounters new music that it doesn't quite know how to listen to. There is excitement, for example, when a person listens to something for the first time. The music gets filtered through the auditory perceptual system, sent to the emotional system for processing, and then registers in the brain as it is reconciled by memory. The thought then is that auditory perception will continue to evolve, and is both an individual and cultural process. As mentioned at various points in the text, the transform and frameworks introduced in this thesis not only aspire to provide new insight into analysis of audio signals and human perception of sound, but also promise to open new capabilities in digital musical instruments.

Appendix A

Synthesis

A.1 Monophonic Synthesizer

The *SCFE Synthesizer* (iSCFE) is a constant Q synthesis algorithm that is based on inverting the algorithm for an SCFE. There is a mapping from complex valued functions to real valued functions that produce monophonic audio, according the equation (A.1).

$$\boxed{x_t = e^{-j\theta_t}(s_t - s_{t-1} + e^{j\phi_t} x_{t-N})} \tag{A.1}$$

where s_t is the input signal. The algorithm is explained by the signal flow diagram in figure (A-1).

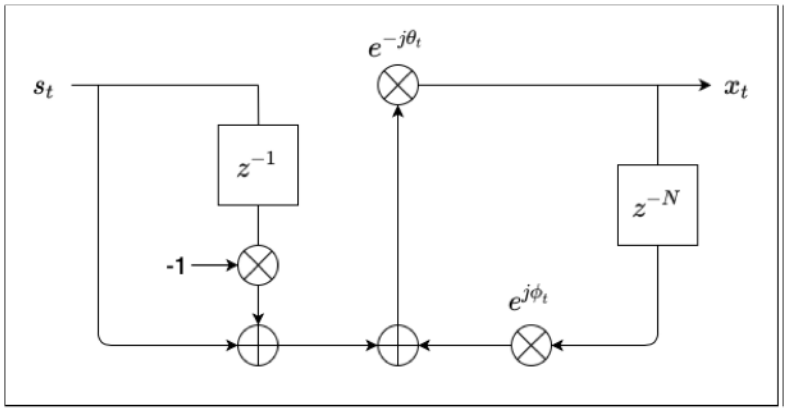

Figure A-1: Signal flow diagram for SCFE synthesizer

There are two main parameters that can be adjusted for synthesis. The first is a pitch parameter, and the second is the integration time N, which is set automatically during analysis to make the SCFE analyser constant Q but can be set arbitrarily here for the iSCFE.

I outline three methods for feeding the synthesizer input:

1. Let the input signal **s** be a real-valued function. For example, it is easy to feed the system functions like linear ramps, quadratic functions, or any arbitrary synthetic real-valued function.

2. Let the input signal s be a complex-valued function in the first $s = xe^{j\theta}$, where **x** is a synthetic real-valued function for the magnitude, and θ is a synthetic real-valued function for the phase

3. Generate a complex-valued input function s by taking an FFT, or a series of FFTs, of a real valued function. Here, the real-valued function could be an audio sample. See (A-4) and (A-5).

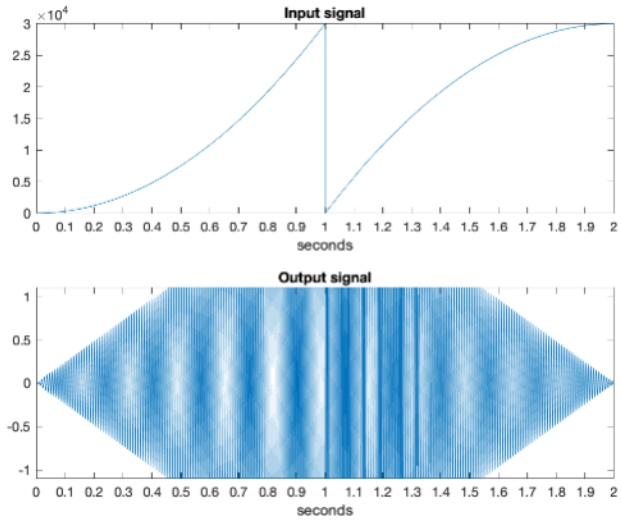

Figure A-2: SCFE synthesizer i/o with quadratic based input function

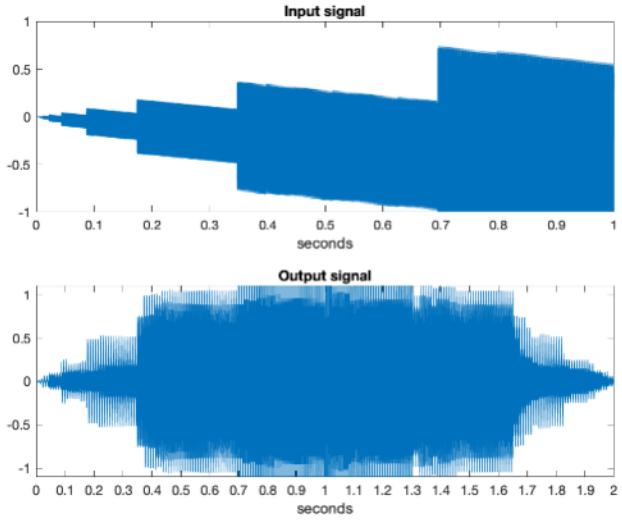

Figure A-3: SCFE synthesizer i/o with unusual real input

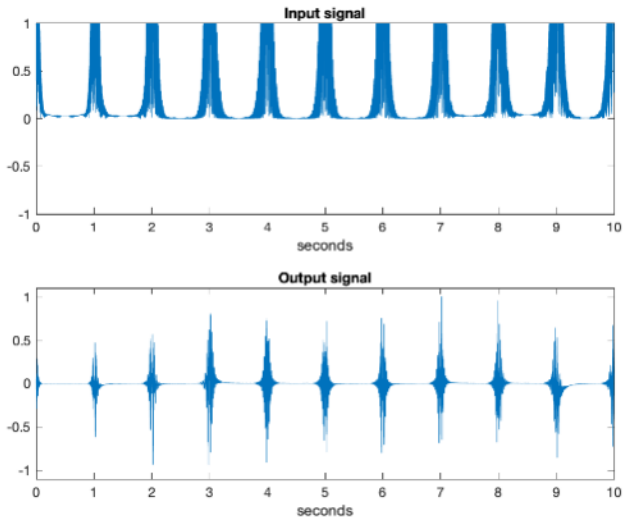

Figure A-4: SCFE synthesizer with processed trombone sample using successive FFTs

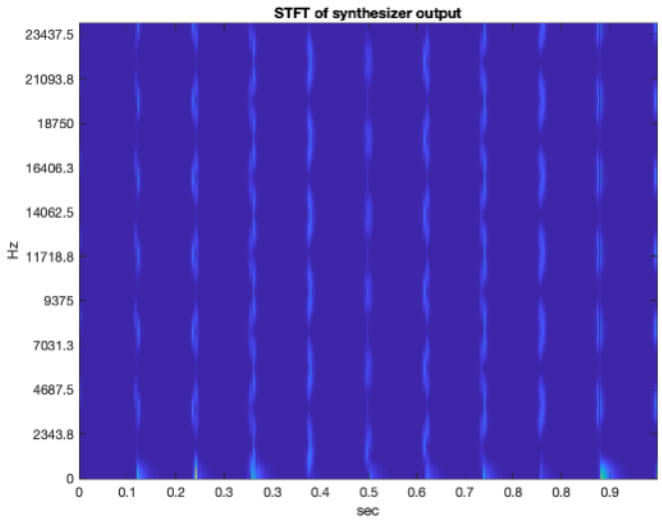

Figure A-5: Spectrogram of synthesizer output

A.2 Polyphonic Synthesizer

It turns out that the inverse GST algorithm can be modified to be used as a polyphonic synthesizer, as in equation (A.2). See figure (A-6) as an example.

$$\boxed{x_t = \frac{1}{K} e^{j\vec{\theta_t}^{*T}} (\vec{s_t} - \vec{s_{t-1}} + e^{j\vec{\phi_t}} x_{t-N})} \tag{A.2}$$

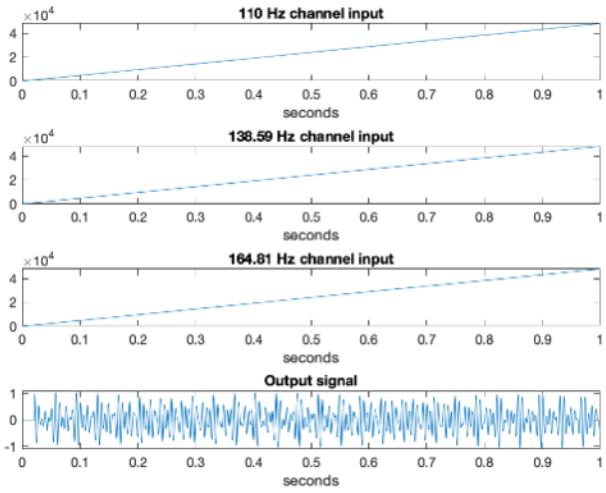

Figure A-6: Polysynth output of a bank of 3 linear ramp input functions with frequencies 110 Hz, 138.59 Hz, and 164.81 Hz (A, C#, E)

Appendix B

Signal Reconstruction

It is possible to reconstruct the original signal of a *Constant Q Spectrogram* analysis using additive synthesis [**Roads1996**]. The original audio in the reconstructed signal bears a strong resemblance to the original but with some distortion. The algorithm is as follows. For each frequency in the *AGST Frequency Matrix*, generate a sine wave and modulate the signal by the magnitude response of the corresponding SCFE analysis channel. Then sum the modulated signals together. Note that a reconstruction with 48 bands per octave gives good results. Note that there is an interesting, flange-like effect at the beginning of the reconstructed signal, probably because the bank of filters being used for the signal reconstruction resembles a comb filter.

```
1
2 %
3 % Constant Q Spectrogram Additive Synthesizer
4 %
5
6 seconds = 30;
7 div = 48;
8
9 K = 88;
10 M = 20;
```

```
11
12 % generate the AGST Frequency Matrix
13
14 for p = 1:K*div/12
15     for q = 1:M
16         F(p,q) = 27.5 * 2^((p-1)/div) * q;
17     end
18 end
19
20 % load the input signal into x here
21
22 ...
23
24 % sample code for synthesis
25
26 out = zeros(1,fs*seconds);
27
28 for k = 1:K*div/12
29     for m = 1:M
30         y = SCFE(x,fs,F(k,m),div);
31         s = sineWAVE(F(k,m),fs*seconds,fs);
32         z = s .* abs(y);
33         out = out + z;
34     end
35 end
36
37 out = out ./ max(out);
```

Appendix C

On Music and Technology

The following section contains the original historical-based introduction to the thesis:

We are at a point in music history where humans extensively augment performance of musical instruments with electronics to achieve new sonic textures and landscapes. The history of the development of the guitar began with amplification, and then included pedals to achieve effects such as distortion, delay, chorus, flange, and a vast array of other means of altering the sound of the instrument. The first polyphonic synthesizer can be attributed to Hammond, who created the Novachord and debuted it at the New York World's Fair in 1939. Although tone-wheel technology first appeared at the turn of the last century in Thaddeus Cahill's Telharmonium [**Weidenaar1995**], the Hammond model B-3 organ appeared in 1954. In the late 1960s, the Fender Rhodes was invented and became an influencial keyboard in music history. For example, Miles Davis bought one for Herbie Hancock, and this can be attributed to the beginning of jazz-rock fusion. A popular effect that was used in conjunction with this instrument was the ring modulator, based on discoveries in radio technology and analog signal processing. In 1969, Barry Vercoe composed the computer-generated piece *Synthesism* at M.I.T. [**Vercoe2014**], and later went on to develop the music synthesis language *Csound*. In the 1970s, the first analog synthesizers were developed, including a massive and intricate system built by Joe Paradiso, an electrical engineer and physicist at M.I.T., that generates avante-garde

synthesized music [**Paradiso2012**]. In the commercial space, pioneers like Buchla, Moog and Oberheim sent the first analog keyboards into the marketplace. The progressive rock era was born, as these keyboards were used as the driving compositional force behind bands like Genesis, Supertramp, Emerson Lake & Palmer, and others. In jazz, acoustic musicians went electric with the appearance of jazz supergroups like Herbie Hancock's HeadHunters, Joe Zawinul's Weather Report, and Chick Corea's Return to Forever. One of the first pioneering electronic bands appeared in 1974, Kraftwerk, who used synthesizers, vocoders, and other electronic means of creating music based on advancements in computer music. Thus, music technology was the driving force that gave rise to new sounds and new ways of music composition and expression.

There is a synergy between technology and art. With the appearance of the Yamaha DX-7 in 1983, which was the first widely adopted digital synthesizer (and was based on FM synthesis), a new era of synth driven pop was born. (Note that the first commercial digital keyboard/synthesizer was the Allen Digital Organ from 1971, based around simple wavetable oscillators). Artists such as A-ha, Kenny Loggins, Kool & the Gang, Whitney Houston, Chicago, Phil Collins, Luther Vandross, and Billy Ocean used it. Many of the rock pioneers from the 1970s such as David Bowie, Peter Gabriel and Stevie Wonder, and jazz visionaries like Herbie Hancock (who formed the Rockit band) and Chick Corea (who formed the Elektric Band) continued to re-invent themselves in the 1980s. Pat Metheny, in his Pat Metheny Group, began using a synth guitar which was accompanied by keyboard and synthesizer wizard Lyle Mays. Also, a notable Canadian band, Rush, experimented extensively with synthesizers in the 1980s and used them to redefine their sound. Two notable examples, as the use of synths, guitars, and electronics continued into the 1990s, include the massively successful rock band Radiohead, and the highly experimental avante-garde klezmer fusion band created by John Zorn, Electric Masada. The latter includes, for example, an artist, Ikue Mori, who contributes electronic sounds from her laptop in the context of a live, conducted, improvising ensemble. This software is digital and software-based, a progression from the hardware-based electronics of the 1970s.

Today, while most gear is digital, musicians continue to use technology that is based both on analog and digital signal processing, in a wide range of styles including hip hop, rock, pop, electronic, dance music, and a myriad of other styles.

Bibliography

[Bradford2008] R. Bradford, J. Ffitch, and R. Dobson (2008) *Sliding With A Constant Q*, Proc. of the 11th Conference on Digital Audio Effects (Espoo, Finland)

[Bregman1990] A. Bregman (1990) *Auditory Scene Analysis*, MIT Press

[Brown1990] J. C. Brown (1990) *Calculation of a constant Q spectral transform*, J. Acoust. Soc. Am. vol. 89, no. 1, pp. 425-434

[Brown1992] J. C. Brown (1992) *An efficient algorithm for the calculation of a constant Q transform*, J. Acoust. Soc. Am. vol. 92, no. 5, pp. 2698-2701

[Cooley1965] Cooley, James W. and Tukey, John W. (1965) *An algorithm for the machine calculation of complex Fourier series*, Math. Comput. 19 (90): 297–301

[Ellis1996] D. P. W. Ellis (1996) *Prediction-driven Computational auditory scene analysis*, M.I.T. Media Lab, 1996

[Fitzgerald2006] D. Fitzgerald, M. Cranitch, and M. T. Cychowski (2006) *Towards an inverse constant Q transform*, 120th Audio Engineering Society Convention (Paris, France)

[Gershenfeld1999] N. Gershenfeld (1999) *The Nature of Mathematical Modelling*, Cambridge University Press

[Goertzel1958] G. Goertzel (1958) *An Algorithm for the Evaluation of Finite Trigonometric Series*, The American Mathematical Monthly, vol. 65, no. 1, 1958, pp. 34–35

[Handel1993] S. Handel (1993) *Listening: An Introduction to the Perception of Auditory Events*, MIT Press

[Haignere2013] S. Norman-Haignere, N. Kanwisher, & J. H. McDermott (2013) *Cortical Pitch Regions in Humans Respond Primarily to Resolved Harmonics and Are Located in Specific Tonotopic Regions of Anterior Auditory Cortex*, Journal of Neuroscience, 33(50), 19451–19469

[Jacobsen2003] E. Jacobsen and R. Lyons (2003) *The Sliding DFT*, IEEE Signal Processing Magazine, Vol. 20, Issue 2

[Ingle2011] A. Ingle (2011) *The Modified Constant Q Spectrogram And It's Applications To Phase Vocoding*, Master's Thesis, Univesity of Wisconsin-Madison

[Izmirli1999] Ozgur Izmirli (1999) *A Hierarchical Constant Q Transform for Partial Tracking in Musical Signals*, Proceedings of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx99), NTNU, Trondheim, Norway, 1999

[Kippen1992a] J. Kippen and B. Bel (1992) *Modelling music with grammars: formal language representation in the Bol Processor*

[Kippen1992b] J. Kippen (1992) *Music and the computer: Some anthropological considerations*, Journal of New Music Research. 21. 257-262

[Kippen1994] J. Kippen and B. Bel (1994) *Computers, Composition, and the Challenge of "New Music" in Modern India*, Leonardo Music Journal. 4

[Martin1998] K. D. Martin & Y. E. Kim (1998) *Music instrument identification: A pattern-recognition approach*, Presented at the 146th meeting of the Acoustical Society of America, Oct. 13, 1998

[Mathieu1997] W. Mathieu (1997) *The Harmonic Experience*, Inner Traditions

[McDermott2019] N. Jacoby, E. A. Undurraga, M. J. McPherson, J. Valdes, T. Ossandon, J. H. McDermott (2019) *Universal and Non-universal Features of Musical*

Pitch Perception Revealed by Singing, Current Biology, Vol 29 Issue 29, p. 3229-3243

[Merriam1964] A. P. Merriam (1964) *The Anthropology of Music*, Northwestern University Press

[Moore1994] B. C. J. Moore (1994) *An Introduction to the Psychology of Hearing*, 3rd ed., Academic Press, London

[Murakami2016] T. Murakami and Y. Ishida (2016) *Generalizing Sliding Discrete Fourier Transform*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences p. 338-345

[Paradiso2012] J. A. Paradiso (2012) *A Massive Patchable Modular Synthesizer*, http:// paradiso.media.mit.edu/synth.html

[Patterson1995] R. D. Patterson, M. H. Allerhand, C. Giguere (1995) *Time-domain modeling of peripheral and auditory images*, Advances in speech, hearing, and language processing Vol. 3, JAI Press, London

[Pavan2017] S. Pavan (2017) *On linear periodically time varying LPTV systems with modulated inputs, and their application to smoothing filters*, 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD

[Picard1995] R. W. Picard (1995) *Affective Computing*, M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 321

[Rumelhart] Rumelhart, Hinton, Williams (1986) *Learning representations by back-propagating errors*, Nature. 323 (6088): 533–536

[Roads1996] C. Roads (1996) *The Computer Music Tutorial*, M.I.T. Press

[Sandler2005] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, M. B. Sandler (2005) *A Tutorial on Onset Detection in Music Signals*, IEEE Transactions on Speech and Audio Processing 13(5), pp 1035–1047

[Scheirer1997] E. D. Scheirer (1997) *Tempo and beat analysis of acoustic music signals*, J. Acoust. Soc. Am., Vol. 103, No. 1, January 1998, pp. 588-601

[Schorbuber2010] C. Schorbuber and A. Klapuri (2010) *Constant-Q Transform Toolbox for Music Processing*

[Slaney1990] M. Slaney and R. F. Lyon (1990) *A perceptual pitch detector* International Conference on Acoustics, Speech, and Signal Processing, Albuquerque, NM, USA, 1990, pp. 357-360 vol.1

[Smaragdis2009] P. Smaragdis (20009) *Relative Pitch Tracking of Multiple Arbitrary Sounds*, J. Acoust. Soc. Am.

[Smaragdis2001] P. Smaragdis (2001) *Redundancy reduction for computational audition, a unifying approach*, M.I.T. Ph.D. thesis

[Smith2011] J. O. Smith III (2011) *Spectral Audio Signal Processing*, W3K Publishing

[Strang1998] G. Strang (1988) *Linear Algebra and Its Applications, 3rd Edition*

[Strang2014] G. Strang (2014) *Differential Equations and Linear Algebra*

[Turing1950] A. Turing (1950) *Computing Machinery and Intelligence*, Mind, 59 (236): 433–60.

[VanLoan1992] C. Van Loan (1992) *Computational Frameworks for the Fast Fourier Transform*, Society for Industrial and Applied Mathematics

[Velasco2011] G. A. Velasco, N. Holighaus, M. Dorfler, T. Grill (2011) *Constructing an Invertible Constant-Q Transform with Nonstationary Gabor Frames*, Proc. of the 14th Intl. Conference on Digital Audio Effects (Paris, France)

[Vercoe1984a] B. Vercoe (1984) *The Synthetic Performer in the Context of Live Performance*, Proceedings of the International Computer Music Conference (Paris, France)

[Vercoe1984b] B. Vercoe (1984) *The Synthetic Performer*, https://www.youtube.com/watch? v=vOYky8MmrEU

[Vercoe2014] B. Vercoe (2014) *Barry Vercoe - Synthesism (1969)*, https://www.youtube.com/ watch?xv=TB7kGqQXw9Y

[Vigoda2005] B. W. Vigoda (2005) *Musical Games: A Guide for Group Improvisation*, unpublished, First Edition

[Walker2019] K. M. M. Walker, R. Gonzalez, J. K. Kang, J. H. McDermott, A. J. King (2019) *Across-species differences in pitch perception are consistent with differences in cochlear filtering*, eLife 2019;8:e41626 DOI: 10.7554/eLife.41626

[Weidenaar1995] R. Weidenaar (1995) *Magic Music from the Telharmonium*, The Scarecrow Press