Instantaneous Harmonic Analysis and its Applications in Automatic Music Transcription

Ali Jannatpour

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctorate of Philosophy (Computer Science) at

Concordia University

Montréal, Québec, Canada

July 2013

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Ali Jannatpour**

Entitled: **Instantaneous Harmonic Analysis and its Applications in Automatic Music Transcription**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Computer Science)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. C. Chen

_____ External Examiner
Dr. M. Pawlak

_____ External to Program
Dr. Y.R. Shayan

_____ Examiner
Dr. E. Doedel

_____ Examiner
Dr. J. Rilling

_____ Thesis Co-Supervisor
Dr. A. Krzyżak

_____ Thesis Co-Supervisor
Dr. D. O'Shaughnessy

Approved by _____
Dr. V. Haarslev, Graduate Program Director

October 7, 2013

_____
Dr. C. Trueman, Interim Dean
Faculty of Engineering & Computer Science

Abstract

Instantaneous Harmonic Analysis and its Applications in Automatic Music Transcription

Ali Jannatpour, Ph.D.

Concordia University, 2013

This thesis presents a novel short-time frequency analysis algorithm, namely Instantaneous Harmonic Analysis (IHA), using a decomposition scheme based on sinusoidals. An estimate for instantaneous amplitude and phase elements of the constituent components of real-valued signals with respect to a set of reference frequencies is provided. In the context of musical audio analysis, the instantaneous amplitude is interpreted as presence of the pitch in time. The thesis examines the potential of improving the automated music analysis process by utilizing the proposed algorithm. For that reason, it targets the following two areas: Multiple Fundamental Frequency Estimation (MFFE), and note on-set/off-set detection.

The IHA algorithm uses constant-Q filtering by employing Windowed Sinc Filters (WSFs) and a novel phasor construct. An implementation of WSFs in the continuous model is used. A new relation between the Constant-Q Transform (CQT) and WSFs is presented. It is demonstrated that CQT can alternatively be implemented by applying a series of logarithmically scaled WSFs while its window function is adjusted, accordingly. The relation between the window functions is provided as well. A comparison of the proposed IHA algorithm with WSFs and CQT demonstrates that the IHA phasor construct delivers better estimates for instantaneous amplitude and phase lags of the signal components.

The thesis also extends the IHA algorithm by employing a generalized kernel function, which in nature, yields a non-orthonormal basis. The kernel function represents the timbral information and is used in the MFFE process. An effective algorithm is proposed to overcome the non-orthonormality issue of the decomposition scheme. To examine the performance improvement of the note on-set/off-set detection process, the proposed algorithm is used in the context of Automatic Music Transcription (AMT). A prototype of an audio-to-MIDI system is developed and applied on synthetic and real music signals. The results of the experiments on real and synthetic music signals are reported. Additionally, a multi -dimensional generalization of the IHA algorithm is presented. The IHA phasor construct is extended into the hyper-complex space, in order to deliver the instantaneous amplitude and multiple phase elements for each dimension.

Index Terms: Instantaneous Harmonic Analysis (IHA), Short-Time Fourier Transform (STFT), Constant-Q Transform (CQT), Windowed Sinc Filters (WSFs), Phasor, Wavelets, Short-Time Cross Correlation, Analytical Signal, Automatic Music Transcription (AMT), Audio-to-MIDI, Kernel Function, Multiple Fundamental Frequency Estimation (MFFE), Hyper-Complex Space, Quaternions, Multi-Dimensional Signal Processing.

To my parents: Hashem and Zahra Jannatpour

Acknowledgement

First and foremost, I would like to thank my advisors. My heartfelt appreciation to Adam Krzyżak for his dedication, full support, and especially, for accepting me as his student during the very last year of my doctorate. My deepest gratitude to Douglas O'Shaughnessy for his dedication, direction, invaluable feedbacks and sound advice throughout the last three years of my research. Words cannot express how thankful I am to both of them.

I am very thankful to the members of my examining committee: Miroslaw Pawlak, Yousef Shayan, Eusebius Doedel, and Juergen Rilling for their detailed feedbacks and the discussions during the defense. My special thanks to Miroslaw Pawlak, for his detailed remarks on the latest advances on the sampling theory, and to Eusebius Doedel, for his noteworthy observations on the functional analysis issues. I would like to thank Tony Kasvand for his remarkable comments during drafting my research proposal. I would also like to acknowledge Tien Bui, one of my former supervisors, for introducing me to the paper by Huang et al. [59]. The quick survey on the IF-based approaches in chapter 2 is a result of the analysis of that paper.

My sincere appreciation to Volker Haarslev and Cameron Skinner for their help and support during the final year of my research. I would also like to extend my gratitude to Theodore Stathopoulos and Halina Monkiewicz for their support throughout my studies at Concordia.

And last but not least, I would like thank my family and friends for their support, especially during the past year.

Table of Contents

List of Figures

List of Tables

List of Definitions

List of Algorithms

List of Symbols

M	Discrete time window radius
Q	Quality factor
T	Sampling period
γ	Bandwidth resolution factor
$\mathbf{j}$	The imaginary unit
n	Sample index
t	Time variable
υ	normalized frequency
ω	Angular frequency
B_k	k^{th} Band
f_{min}	Minimum audible frequency
$x(t)$	Real-valued continuous time domain signal
$x[n]$	Real-valued discrete time domain signal
$\mathrm{H}_k(t)$	The continuous IHA transform
$\mathrm{H}[k,n]$	The discrete IHA transform
$\wp_\upsilon(x[n])$	The IHA phasor construct

List of Abbreviations

ACF	Auto-Correlation Function
AMT	Automatic Music Transcription
ANSI	American National Standards Institute
CQT	Constant-Q Transform
DFT	Discrete Fourier Transform
EMD	Empirical Mode Decomposition
F0	Fundamental Frequency
FFB	Fast Filter Bank
HPS	Harmonic Product Spectrum
ICA	Independent Component Analysis
IF	Instantaneous Frequency
IHA	Instantaneous Harmonic Analysis
MFFE	Multiple Fundamental Frequency Estimation
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
MIREX	Music Information Retrieval Evaluation eXchange
NMF	Non-Negative Matrix Factorization
PLCA	Probabilistic Latent Component Analysis
QDCT	Quaternion Discrete Cosine Transform
QFT	Quaternion Fourier Transform
RGB	Red Green Blue

SFFE	Single Fundamental Frequency Estimation
STFT	Short-Time Fourier Transform
SVD	Singular Value Decomposition
TEO	Teager Energy Operator
WD	Wigner Distribution
WSF	Windowed Sinc Filter
ZC	Zero Crossing

Foreword

"In the name of Hermes, the god and the deity of the science of art[1]"

In the memory of Morteza Hannaneh (1923–1989), who believed music was not only art but science.

Throughout the thesis, references to various musical terms are made. While the reader is expected to be familiar with music terminology, a brief glossary is provided in section 1.1. Throughout chapters 1–3 a brief introduction on the harmonic analysis and music signal processing with a short survey on the state-of-the-art techniques is provided. Chapters 4–7 provide the Instantaneous Harmonic Analysis (IHA) transform along with its generalizations towards music signal processing, while chapter 8 solely focuses on music transcription. It is hoped that the fundamental contribution of the IHA transform motivates future research directions.

Ali Jannatpour,
October 2013, Montréal – Canada

[1]Quoted by Motreza Hannaneh during tutoring *Practical Manual of Harmony* by Nikolai Rimsky-Korsakov.

Chapter 1

Introduction

Harmonic analysis may be understood as a special case of functional analysis in mathematics, which concerns studying functions and their representations based on superposition of basic waves. The term harmonic originally comes from the eigenvalue problem, in which the frequencies of the waves are integer multiples of the mother wave. Fourier analysis may be considered in the context of Hilbert spaces, which provides a bridge between the harmonic analysis and functional analysis. In music theory, however, harmonic analysis is known as the study of chords and their combination in terms of producing musical effects. This thesis concerns the former definition.

While harmonic analysis dates back to Fourier's effort on analyzing the solutions of the heat and wave equations, it has been widely used in many branches of science such as mathematics, physics, and engineering, to name a few. Fourier series has been one of the earliest attempts to study periodicity, by which an arbitrary periodic function is represented

by a sum of simple wave functions [1]. The motivation of the Fourier transform comes from generalization of the Fourier series when the period of the represented function is stretched out and approaches infinity [48]. Such a definition delivers a sufficient measure for studying the frequency distribution and provides a tool for studying non-periodic functions. The periodicity itself has been studied in many branches of mathematics, which has resulted in the topics such as definition of almost periodic functions, etc. The topic is an ongoing research, especially in the generalization area [67].

Much effort has been devoted through the years to overcome the main drawback of the classical Fourier transform, with respect to the loss of time information when the signal is transferred into the frequency domain. Time-frequency analysis provides a compromise solution by studying the signal simultaneously in the time and frequency domains. It has been widely studied by many researchers in the literature. The Short-Time Fourier Transform (STFT) has been one of the earliest approaches delivering a joint time-frequency analysis [95]. It suggests applying the Discrete Fourier Transform (DFT) using a window function. Assuming the input signal is quasi-stationary, STFT provides a two-dimensional time-frequency analysis by taking the Fourier transform of the signal using a window function. Such windowing results in a tradeoff in time resolution vs. frequency resolution. Wavelets, on the other hand, provide frequency analysis at different resolutions by preserving the locality.

Wigner Distribution (WD) has been used by many researchers as a powerful time-frequency analysis tool [65]. One of the advantages of WD over STFT is using a quadratic function to avoid negativity in order to provide energy distribution. It however produces undesirable cross-terms [89]. It has been demonstrated to have a fine performance, especially

[1] Convergence of the Fourier series has been studied extensively in the literature. In general, one can consider point-wise, uniform, and L^2 convergence. It is easy to show that a Fourier series of a periodic function f in L^2 converges to f in L^2 sense. Also a Fourier series of a periodic, integrable function which is continuous at x_0 converges to $f(x_0)$. If f is periodic, continuous and differentiable on $\mathbb{R}$ with f' piecewise continuous then the Fourier series of f converges uniformly. Point-wise convergence is tricky. It was shown by Carleson [19] that Fourier series of any periodic function $f \in L^2$ converges to f point-wise, almost everywhere. Later on, Hunt [60] generalized the space to L^p for any $p \in (1, \infty)$. The main references on the topic are [8, 113].

in the analysis of non-stationary signals [58].

Alternatively, the Instantaneous Frequency (IF) is defined based on analytical signals where both amplitude and phase are represented by time-varying functions. IF is generally calculated from the analytical signal using the Hilbert transform, although other variations have been used in the literature, resulting in contradicting definitions [91]. The major difference between IF and STFT is that STFT delivers a spectral analysis in a discrete form, whereas IF provides an instantaneous measure at a frequency level. The STFT allows studying multiple frequencies within a short time, while IF delivers the instantaneous frequency of the signal in time. IF has been extensively used by many researchers. Although there are numerous applications for IF-based signal analysis, some researchers challenge the concept stating that it violates the uncertainty principle (see [51]).

The uncertainty principle states that the signal cannot simultaneously be localized in both time and frequency. It may alternatively be understood as the Gabor limit, stating that a function cannot be both time-limited and band-limited at the same time [43, 88]. Several papers have been published on the topic of the relation between the time-resolution and the bandwidth, mainly suggesting an upper limit for the product of the two. This relation plays a great role in short-time frequency analysis, especially when estimating instantaneous properties are targeted. A mathematical survey on the topic may be found in [41].

The concept of IF has provoked strong opinions among scientists with respect to the uncertainty principle [59]. Cohen has published a significant paper [28] on the topic of the product of time and frequency covariances. He states that the reason behind so much confusion on the uncertainty principle is mainly due to misinterpretation of STFT philosophy.

In frame-based analysis [6] where the signal is segmented into smaller pieces, it is

assumed that the signal is completely stationary within the frame. In case the stationary property cannot be satisfied, smaller frames may be used. However, the frames cannot be made arbitrarily small, for many different reasons, the most importantly, the uncertainty principle. Hence, a need for non-stationary signal analysis is beneficial.

One practical domain of non-stationary signal processing is the analysis of music signals. Music signals have a certain specific characteristics. For instance, they follow the pattern of musical scales which represent a discrete set of frequencies; the discrete frequencies are logarithmically spaced on the frequency axis; and, the presence of notes in time forms the melody. In this thesis, we focus on short-time frequency analysis approaches, specific to the non-stationary time-frequency characteristics of the music signals.

Several attempts have been made for choosing the best joint time-frequency resolutions based on the application [94]. For instance, in the context of tonal music analysis, considering the way that the musical scales are structured, applying a logarithmic spectral analysis is more effective than using a linear paradigm [49]. In an equal temperament system [7], the frequency ratio of the two notes within an equal interval is always constant. Therefore, the frequencies follow a logarithmic scales. Constant-Q Transform (CQT) is one of the approaches that provides a logarithmic spectral analysis with respect to the equal temperament that is used in western music [16]. It is based on the theory behind DFT, but uses a constant ratio of center frequency to resolution that is equivalent to one semitone. CQT can generally be performed by taking the Fourier transform of a windowed sequence where the window is a function of the product of time and frequency. Prior to introducing CQT by Brown, constant-Q analysis has been used in the literature. For instance, Kates [66] previously showed that the result of a constant-Q analysis using an exponentially decaying window is equivalent to the Z-transform along an outwardly going spiral in the complex Z

-plane [2].

General issues about instantaneous amplitude and phase of signals have been discussed in [91]. This thesis proposes a new model for estimating instantaneous amplitude and phase of multi-component signals which may be used as a fundamental tool in Multiple Fundamental Frequency Estimation (MFFE). MFFE, or multiple-F0 estimation, is an essential process in audio analysis by addressing the over-toning issue, which is caused by harmonic collisions from two or more simultaneous tones. A tone is a steady periodic sound, often generated by a musical instrument, that plays a musical note. It is not necessarily a pure tone. A non-pure tone may be decomposed into one pure fundamental tone and a some overtones [70].

In Single Fundamental Frequency Estimation (SFFE), a signal is assumed to consist of single Fundamental Frequency (F0) at a time. Therefore, estimating F0's is a straightforward process. In MFFE, however, the harmonic collision of the multiple overtones makes the process rather difficult. The harmonic collision is caused by the interferences of the overtones produced by multiple sources (i.e. an octave or a perfect fifth). This is very common in music where the multiple sources form consonant intervals [7].

1.1 Terminology

The thesis discusses the harmonic analysis of multi-component tonal audio signals. While the formal specification of the analysis id provided in chapters 4–6, the used terminology is

[2] Z-transform converts a discrete time-signal into a complex frequency-domain representation:

$$\mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{+\infty} x[n]z^{-n}$$

given in the following:

Multi-Component Signal: is an audio signal composed of musical tones.

Tonal Signal: is referred to an audio signal that consists of perfectly pitched tones, corresponding to the standard musical notes. Musical notes correspond to a discrete set of frequencies, each of which, designate the fundamental frequency of the note (i.e. 440 Hz for the standard concert pitch).

Transcription System: in simple words, is a system of converting an audio signal into a set of note events that can be represented in a musical notation format.

Throughout the thesis, various references to music terms are made. They are briefly explained in the following, while a comprehensive guide may be found in [46].

Accidentals: are note modifiers, which cause a pitch change that does not belong to the scale.

Cent: is a logarithmic micro-tonal unit of measure used for musical intervals, equivalent of one-hundredth of a semitone.

Chord: is a harmonic set of three or more notes that is played simultaneously.

Diatonic Scale: is an eight-note scale composed of seven notes and a repeated octave.

Harmony: is the use of simultaneous notes, usually in form of a chord. It often refers to the vertical sonority of the music.

Harmonic: is an overtone whose frequency is an integer multiple of the fundamental frequency.

Interval: is the difference between two pitches, usually on a diatonic scale.

Melody: is a rhythmic sequence of single notes. It is often regarded as horizontal, since the notes are played from left-to-right.

MIDI: Musical Instrument Digital Interface, is an industry specification for encoding, storing, synchronizing, and transmitting the musical performance, primarily based on note events.

Notation: is a system used in sheet music in order to represent a piece of music.

Note: identifies a pitched sound and is associated with a name (and sign). Notes may be considered as atoms of music that allows discretization of musical analysis.

Note Events: are the set of events identifying notes on-sets and off-sets.

Octave: is an interval between a note and its second harmonic.

Overtone: is a component of a sound with a frequency higher than the fundamental frequency.

Pitch: may be defined as the degree of highness or lowness of a tone on a frequency-related scale.

Rhythm: is the timing of musical sounds and silences.

Semitone: is the smallest interval used in Western music and is equivalent to one-twelfth of an octave. It is sometimes known as half-tone or half-step.

Scale: is a set of musical notes ordered by pitch.

Staff: is a set of five horizontal lines and four spaces that each represent a different note.

Timbre: also known as tone color or tone quality, is referred to the physical characteristics of a sound, which makes the different tonal sounds distinguishable while they have the same pitch.

Tone: is a steady periodic sound, often generated by a musical instrument, that plays a musical note[3].

[3]In some context, tone may also refer to the musical interval, equivalent to a major second. To distinguish between the two, we refer to the latter as whole-tone.

1.2 The Scope of the Thesis

This thesis primarily concerns the problems and challenges of the Instantaneous Harmonic Analysis (IHA) of audio signals, with a focus on non-stationary signal analysis techniques. The challenges include multi-pitch analysis, MFFE, harmonic collision, overtone estimation, to name a few. The thesis targets the two core areas in music transcription: MFFE and note events detection. The audio analysis in this thesis is purely tonal and therefore percussion analysis is out of the scope of this thesis.

A novel harmonic analysis algorithm, namely IHA, is provided based on enhanced Constant-Q filtering. The IHA algorithm delivers a signal decomposition scheme based on the frequency distribution of musical scales. The objective of the IHA algorithm is to provide the instantaneous amplitudes and phase lags of the signal components with respect to the discrete set of frequencies that are associated with the musical scales. It can be used as a fundamental tool for MFFE where the instantaneous amplitudes represent the presence of fundamental frequencies in time, and the instantaneous phase lags are used for overtone estimation.

The scope of the thesis is to implement and evaluate the performance of the new algorithm in the context of musical signal analysis. The performance is measured by evaluating the improvement of the note events detection process, by applying the algorithm on different sets of musical data in order to produce the note events. Note events, in simple words, are referred to as the representation of music pieces in form of note on-sets and off-sets. They are the essential part of the music and form the core of the music notation [46].

The thesis contributes to the music analysis, yet it does not aim at automating the whole music transcription process.

1.3 Thesis Contributions

The thesis delivers a novel IHA algorithm based on enhanced Constant-Q filtering [62]. An implementation of Windowed Sinc Filters (WSFs) based on the signal model in its original continuous form has been used. A new relation between the CQT and WSFs has also been provided [63]. The thesis also extends the IHA transform by employing a generalized kernel function. The algorithm contributes to an MFFE process where a post-processing overtone elimination algorithm is used for timbral analysis. A generalization in multi-dimensional space has also been provided.

1.4 Thesis Organization

The thesis is structured as follows: A background on harmonic analysis algorithms and related transformations and the motivation of the thesis is given in the next chapter. A short survey of related music signal processing techniques is provided in chapter 3. The formal specification of the IHA transform is given in chapter 4. Chapter 5 presents the generalization of the IHA transform into the multi-dimensional space. The generalization of the IHA algorithm for timbral analysis is formalized in chapter 6. The theoretical analysis of the IHA algorithm along with the relation between WSFs and CQT is presented in chapter 7. The transcription system as well as the post-processing algorithms are discussed in chapter 8. And in chapter 9 the thesis is concluded.

Chapter 2

Background and Motivation

This chapter discusses the background and motivation behind this thesis. A short survey of time-frequency analysis algorithms with an emphasis on CQT is presented. The signal decomposition scheme that is used in the thesis is discussed here.

The STFT has been one of the earliest attempts delivering a joint time-frequency analysis [95]. Huang et al. [59] provided a comprehensive survey on instantaneous frequency -based methods in particular Zero Crossing (ZC), Teager Energy Operator (TEO), and normalized Hilbert transform, to name a few. A historical review on instantaneous frequency was previously provided in [14, 15]. Rabiner and Schafer utilized the STFT in [95], which was also surveyed by Kadambe and Boudreaux-Bartels [65] along with the WD and wavelet theory. Khan et al. [68] proposed an IF estimation using fractional Fourier transform and WD.

IF has been widely used by many researchers in the literature. It is generally calculated from the analytical signal using the Hilbert transform [91]. Oliveira and Barroso used

the IF of multi-component signals in [87]. Nho and Loughlin studied IF and the average frequency crossing in [86]. An Empirical Mode Decomposition (EMD) based-method is also used by Zhang et al. in [111] for IF estimation. Arroabarren et al. in [5] have provided some methodological basis for determining the instantaneous amplitudes and frequencies.

In the context of tonal music analysis, considering the way that the musical scales are structured, applying a logarithmic spectral analysis is more effective than using a linear paradigm. CQT, originally introduced by Brown, is one of the approaches that provide such spectral analysis [16]. Brown explained how such analysis can be improved by using a spectral representation that supports the logarithmic spacing of the musical tones. She states that one of the major advantages of such representation is that the spectral components form a pattern in the frequency domain which is the same for all sounds with harmonic frequency components [16]. Therefore, using a constant ratio of center frequency to resolution, namely Q, is highly efficient in comparison with using a constant bandwidth resolution that is used in the traditional DFT-based approaches. Brown explained that one of the major advantages of such design is that the spectral components form a pattern in the frequency domain that is the same for all sounds with harmonic frequency components.

In the following sections, the formal definition of CQT is given. CQT is used in this thesis, in relation with the signal decomposition scheme. The signal decomposition scheme is subsequently discussed. The frequency quantization delivered by the decomposition scheme is explained. This chapter concludes with the motivation behind the thesis.

2.1 The Constant-Q Transform

CQT is calculated by taking the Fourier transform of a windowed sequence while the window is a function of the product of time and frequency. The constant ratio represents the quality factor and is set to the number of full cycles to be used by the temporal window. The formal definition of CQT is as follows [16].

Given $x[n]$, the input signal in discrete domain, the CQT of $x[n]$ is defined as:

$$X[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} W[k,n] \cdot x[n] \cdot \mathrm{e}^{\frac{-\mathbf{j}2\pi Qn}{N[k]}} \tag{2.1}$$

where $N[k]$ represents the temporal window length,

$$N[k] = \frac{Q}{Tf_k},$$

and Q and f_k are the quality factor and the k^{th} reference frequency, respectively. $W[k,n]$ denotes a symmetric window function, e.g., Hamming window. The above equation may also be understood as taking a normalized Fourier transform of the windowed signal by using a predefined set of digital frequencies and applying a temporal window with the exact Q number of cycles. It must be noted that $x[n]$ in (2.1) represents the windowed signal. cf. [16].

The invertibility of CQT has been a challenge for years. Although filter bank approaches are invertible in nature, the inverse transform for the CQT algorithm had not been available until recently. The first attempt of obtaining the inverse transform was made by Cranitch et al. [30]. The authors explained that the process could not formally be inverted as the matrix representation of the CQT implementation by DFT uses non-square matrices. They provided the solution using a pseudo-inverse of the transform matrix, by taking advantage of the scarcity property of the DFT, due to the fact that the time domain

signal is properly pitched. Holighaus et al. [56] recently provided a framework for calculating an invertible CQT in real-time. Their work was based on the non-stationary Gabor frames [6, 20]. Ingle and Sethares [61] also developed a least-square invertible constant-Q spectrogram for phase vocoding.

2.2 The Decomposition Scheme

Signal decomposition is a broad topic that addresses the functional relationship between an arbitrary signal and its constituent components by which the original signal can be reconstructed. Depending on the application, different decomposition paradigms may be used. While frame theory provides an orthonormal basis for signal decomposition, alternative schemes may also be used based on the application.

In tonal music analysis, where the input signal is perfectly pitched, it is desired to study signals at certain frequencies. These frequencies correspond to musical notes and the analysis examines the existence of such frequencies in short time. Hence, one may look for a decomposition scheme to transform input signals into a set of pure components that can be later on used within an MFFE process. In such a model, the reference pitches form the musical scale and are chosen according to the intonation system [7] (cf. [16], harmonic sounds in [71]). Since each tone is represented by a fundamental frequency, estimating the presence of such frequencies is the key to the note identification process [26].

In our model, a multi-component signal may be decomposed into a finite number of single-component signals, each of which represents a pure tone in the form of a quasi-sinusoidal:

$$x(t) = \sum_{k} C_k(t), \tag{2.2}$$

where

$$C_k(t) = a_k(t)\cos(\omega_k t + \Phi_k(t)), \tag{2.3}$$

represents the k^{th} component, $a_k(t)$ and $\Phi_k(t)$ represent the instantaneous amplitude and phase lag of the k^{th} component, respectively. ω_k's represent the frequencies and are logarithmically spaced on the frequency axis:

$$\omega_{k+1} = \gamma \cdot \omega_k, \quad \gamma > 1, \quad \omega_k > 0. \tag{2.4}$$

2.3 Frequency Quantization

Eq. (2.4) conveys a logarithmic quantization of the frequency axis. The frequency axis is quantized into a set of small intervals by maintaining the constant-Q whereas the reference frequencies are logarithmically centered on the corresponding intervals. Brown's quality factor Q was originally defined in the discrete domain based on the window size in the time resolution. One may redefine such a measure by using the frequency resolution instead (cf. [28]). In such definition, the time domain need not be necessarily discretized. This allows us to apply the filters in both discrete and continuous forms. Our approach is based upon a map function $\Omega : \mathbb{Z} \to \mathbb{R} \times \mathbb{R}^2$, as follows.

Given ω_0, the global reference frequency, and $\gamma > 1$, the bandwidth resolution factor, the map function Ω quantizes the frequency axis into a set of bands B_k, each of which is tagged with a reference frequency ω_k, where the reference frequency is centered in the band. In this model, the bandwidths are proportional to the frequencies. The definition of the map function is given in the following:

$$\Omega(k) = \langle \omega_k, B_k \rangle, \tag{2.5}$$

where

$$\omega_k = \gamma^k \omega_0,$$

$$|B_k| = \lambda \omega_k,$$

and $|B_k|$ denotes the bandwidth and λ is a function of γ.

A full partition of the frequency axis may be achieved by choosing non-overlapping B_k's, as in the following:

$$B_k \cap B_l = \varnothing, \quad k \neq l \quad \text{with} \quad \bigcup_k B_k = [0, \Omega] \ ,$$

where Ω denotes the bandwidth of the signal. However, in general, B_k may be chosen as

$$\left[(1 - \frac{\lambda}{2})\omega_k, (1 + \frac{\lambda}{2})\omega_k \right], \tag{2.6}$$

or

$$\left[\omega_k \frac{1}{\sqrt{\gamma}}, \omega_k \sqrt{\gamma} \right], \tag{2.7}$$

whether ω_k is linearly or logarithmicaly centered in B_k, respectively. It must be noted that in case (2.6) is used, $\lambda < 2$ must hold. Also, for small bandwidths, $\lambda \to \gamma$.

In practice, the map function Ω is bounded. Although ω_0 is chosen arbitrarily, it is commonly set to 880π, representing **A440**, the standard concert pitch, or to a minimum frequency, i.e. the minimum audible tone or the lowest frequency of the musical instruments / vocals in use. The bandwidth resolution may also be represented in cents,

$$c = 1200 \log_2 \lambda$$

an equivalent of one-hundredth of a semitone [7]. A 50-cent or a 100-cent resolution may be used whether a 12 or 24 equal temperament system is desired [32]. The two systems

are used in western and quarter-tone music, respectively [38]. The equivalent bandwidth resolution factors of such resolutions are 1.0293 and 1.0595, or in Brown's system 69.25 or 34.62, respectively. The relationship between Brown's quality factor and our bandwidth resolution factor may also be obtained by (cf. [28, 16]):

$$\frac{1}{Q} = \frac{\lambda - 1}{2}. \tag{2.8}$$

2.4 Motivation of the IHA Algorithm

The objective of our model is to find a set of pairs of time-varying functions $a_k(t)$, $\Phi_k(t)$ in (2.3), that best estimate the instantaneous amplitude and phase lag of the signal components. Our approach is based upon two constraints:

1. ω_k's are known, and
2. a_k's and Φ_k's are smooth functions such that they can be locally approximated by a constant [1].

ω_k's form the standard musical tones, and the constraints on a_k's and Φ_k's makes it possible to use a linear approach, i.e., the first derivative, for the estimation[2]. The details are given in the chapter 4.

[1] We call $a_k(t)$ and $\Phi_k(t)$ locally constant, if:

$$\begin{cases} a_k(t) \approx a_k(t \pm \Delta t), & \Delta t \to 0 \\ \Phi_k(t) \approx \Phi_k(t \pm \Delta t), & \Delta t \to 0 \end{cases}$$

[2] It must be noted that the signal is assumed to be noise-free, and hence, the estimation process does not take noise into account.

Since the input signal is real-valued, one may combine the two functions into a single time-varying complex function, such as the following phasor notation:

$$\mathrm{H}_k(t) = a_k(t)\mathrm{e}^{\mathbf{j}\Phi_k(t)} \tag{2.9}$$

where $\mathrm{H}_k(t) \in \mathbb{C}$. Therefore, given a set of reference frequencies, the multi-component real -valued input signal can be represented by a set of complex-valued time-varying functions. $x(t)$ may then be fully reconstructed using the following:

$$x(t) = \Re\{\sum_k \mathrm{H}_k(t)\mathrm{e}^{\mathbf{j}\omega_k t}\} \tag{2.10}$$

where $\Re$ denotes the real-part function. Hence, the objective of our decomposition algorithm is to estimate the complex values $\mathrm{H}_k(t)$'s with respect to the set of ω_k's representing the reference frequencies. This forms the basis of the IHA algorithm that is specified later on in chapter 4.

Chapter 3

Music Signal Processing

This chapter delivers a short survey of state-of-the-art techniques in music transcription. An overview of the music analysis techniques is given. Related works with focus on CQT are cited.

3.1 Overview

Harmonic analysis plays a significant role in audio signal processing, and in particular, in the processing of musical signals. It has been used in various areas of musical signal processing, such as classification, encoding, enhancement, registration, and automatic transcription. A piece of music may be analyzed in various aspects such as pitch and tone, mode, rhythm, chord, melody, harmony, texture, timbre, as well as form, dynamics, and articulation. Among those, melody and harmony are very important as both provide tonal information. The melody may be defined as a series of tones in succession whereas the harmony represents the vertical sonority. The timbre, also known as tone color or tone quality,

is referred to the physical characteristics of a sound, which makes the different tonal sounds distinguishable while they have the same pitch.

One of the practical examples of IHA is the analysis of musical signals in the context of polyphonic music; where the musical piece is represented by a set of tones that are played in time, so called note events. This founds the basis of the modern notation system. The term polyphony is generally referred to a texture that consists of two or more melodic tones, whether played by single or multiple instruments.

Many notation systems have been used in the history of music from cuneiform tablets, used during the Babylonian era, to alphabetical notation employed during ancient Persia and Greece, to modern notation system which originated in European classical music. In modern notation system, instruments are represented by staff lines and the tones are symbolized by musical notes, placed on or in between the lines [46]. Examples of single-instrument monophonic and orchestral polyphonic pieces have been given in figures 3.1 and 3.2, respectively.

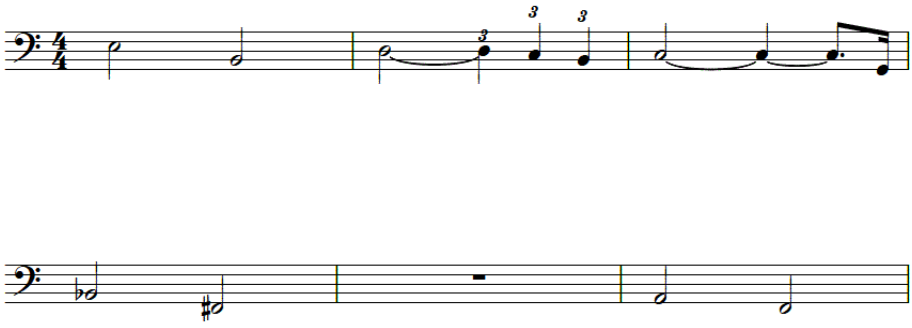


Figure 3.1: A Monophonic sample, Rimsky Korsakov's Scheherazade symphonic suite, Opus 35, Movement I, Contra-Bass

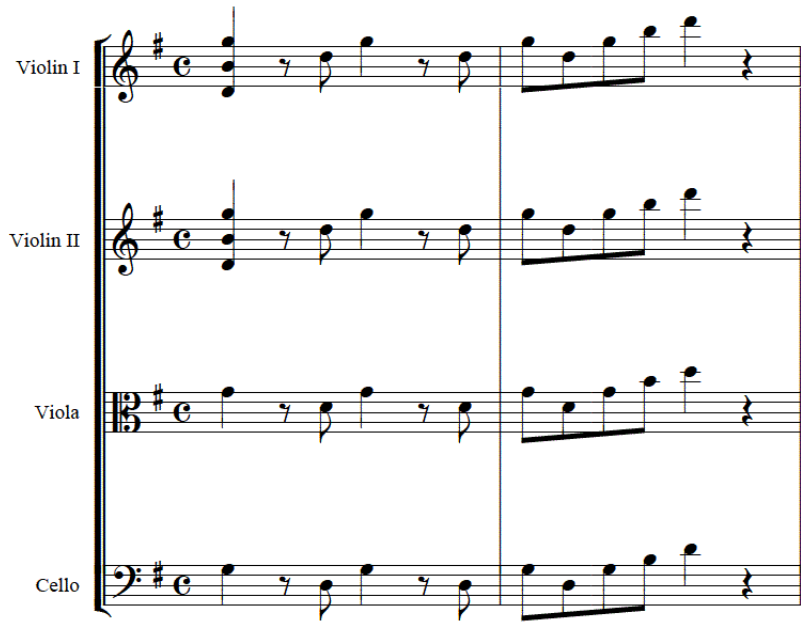


Figure 3.2: A Polyphonic sample, Mozart's Serenade in G, Köchel 525, No. 13, Movement I, String Quartet

3.2 The Music Transcription Problem

The Automatic Music Transcription (AMT) problem is explained in [52, 71]. The concept of AMT has been studied for nearly forty years, resulting in various publications. The AMT process, in general, may be broken into sub-processes such as pitch detection, chord analysis and identification, melody extraction, or more specifically timbral analysis, beat tracking, as well as post-processes such as analysis of keys and accidentals, which result in full score generation [70]. The process of music transcription itself is a difficult task, which requires several years of musical training. Automating such a process is therefore extremely challenging. The process has not been fully automated with a desired level of accuracy especially in case of complex polyphonic signals [52].

Monophonic transcription is known to be one of the earliest attempts in automating

the music transcription process, resulting in melody transcription. Melody transcription is generally achieved by using a pitch detection algorithm. Poliner et al. studied and evaluated melody transcription approaches in [92]. Earlier approaches used STFT for audio spectral analysis. Some authors have shown interest in STFT, even recently. For instance Gang et al. [44] used STFT in polyphonic music transcription by employing spectrographic features.

Several theses have been written on the topic of music analysis and transcription. Klapuri discussed signal processing methods for music transcription in [70]. His thesis concerned source separation and multi-pitch estimation. Probabilistic approaches have been discussed in [21, 57]. Martin discussed sound-source recognition in [80]. Tzanetakis [105] provided an audio texture segmentation methodology for feature extraction in genre clas-sication and Music Information Retrieval (MIR) systems. Benetos [10] provided automatic transcription based on note events. Harmonic collisions using sinusoidal analysis has been discussed in [35]. This thesis concerns note event modeling based on the sinusoidal analysis that is provided by the IHA transform.

3.3 Note Events Analysis

Note events detection and modeling plays a great role in the AMT process [98]. Abdallah and Plumbley [1] provided a note event detection using Independent Component Analysis (ICA) as a conditional density model. A complete survey may be found in [10]. Note events also form the essential part of the Musical Instrument Digital Interface (MIDI) standard [81]. MIDI is an industry specification for encoding, storing, synchronizing, and transmitting the musical performance, primarily based on note events. As such, it has been used by music authoring software not only for playback but also for score visualization. One of the important characteristics of the MIDI format is that the MIDI data is represented

in time rather than beats. Therefore, post-processes such as tempo detection may be applied on the note event model, in an isolated manner although semi-automated techniques such as tapping may also be employed for fine-tuning the result [22, 70]. Processing MIDI information is also beneficial in MIR systems [105].

Pitch detection is the crucial process in MFFE. The purpose of multiple-F0 estimation is to detect the fundamental frequencies (F0s), also known as pitches, of all the components in a mixture signal. Harmonic structures play a great role in multi-pith estimation [26]. A review on objective music structure analysis is provided by Li et al. in [74]. Numerous pitch detection algorithms have been proposed by the researchers, most of which are based on a Blackboard system [79]. Previously, Abe and Ando [2] provided time-frequency domain operators for decomposing sounds into loudness, pitch and timbre. Yeh et al. discussed the polyphony inference in the MFFE process in [109]. They used a frame-based system by combining an F0 candidate selection with a joint F0 evaluation.

While early methods were based on DFT, other short-time frequency analysis approaches have later been used for pitch detection. For instance, Fitch and Shabana proposed a wavelet-based pitch detector in [40]. Wavelets has been widely used in the literature. They are highly beneficial in template-based approaches such as classification and retrieval. For instance, recently, [37] Fan et al. used wavelets to implement a humming music retrieval. Auto-Correlation Function (ACF) has also been prominently used for multiple-F0 estimation. [82], for instance, developed a monophonic transcription system using ACF [82]. de Cheveigné and Kawahara [32] proposed an algorithm, called YIN, based a modified ACF. Using pre-processing algorithms, such as source separation [23], also improves the MFFE process.

3.4 Music Transcription Techniques

AMT is a broad topic that involves various techniques, from frequency analysis to probabilistic model, genetic algorithms, neural networks, and holistic approaches. Frequency -domain based methods, in general, have better overall performance, especially in multi-pitch detection [4]. While ZC and temporal ACF were originally used in the context of pitch detection, other combined approaches have been applied in AMT. Leman [73], for instance, proposed a tonality induction system using a self-organizing map based on neural networks, so called Kohenon map, which used a twelve bin analyzer for twelve semitones, independent of octave. Marolt used neural networks for transcription of polyphonic piano music in [77, 78]. Pichevar and Rouat [90] used neural networks for monophonic sound source separation, which is a supplementary technique used in music transcription. Bruno and Nesi [17] provided an music transcription system supporting different instruments. Kitahara et al. [69] provided an instrument identification based on F0-dependent multivariate normal distribution. Reis et al. [97] used genetic algorithms for polyphonic music transcription. Grindlay and Ellis [50] developed an Eigen-instrument model for multi-voice polyphonic music transcription. Table 3.1 lists some reported results form the literature.

Method	*Rate (%)*		*Details*
	Lowest	*Highest*	
Argent et al. [4]	53.8	72.1	polyphonic, using constant-Q bispectral analysis
Bertin et al. [12]	22.4	36.4	polyphonic, NMF and K-SVD
Yeh et al. [109]	58.9	61.9	MFFE, MIREX dataset
Bertin et al. [13]	32.0	75.8	polyphonic, NMF, various implementations
Benetos and Dixon [11]		58.2	polyphonic, multiple-instrument
Bruno and Nesi [17]		91.3	AMT supporting different instruments
Costantini et al. [29]		96.9	J. S. Bach Inventions
da C. B. Diniz et al. [31]		80.0	Filter Banks, Korsakov's Flight of the Bumblebee
Grindlay and Ellis [50]	64.0	98.0	polyphonic, multi-voice, Eigen-instruments
Ryynänen and Klapuri [98]		41.0	polyphonic, using note event modeling

Table 3.1: Some Reported Results from the Literature

Non-Negative Matrix Factorization (NMF) has been extensively used in AMT. Smaragdis and Brown [102] along with Sophea and Phon-Amnuaisuk [104] used NMF for polyphonic music transcription. Bertin et al. [12] used NMF as well as non-negative K-Means Singular Value Decomposition (SVD) as two blind signal decomposition algorithms for AMT. Bertin et al. [13] used Bayesian NMF for enforcing harmonicity and smoothness in polyphonic music transcription. Costantini et al. [29] used NMF in transcription of piano music. Ganseman et al. [45] recently used Probabilistic Latent Component Analysis (PLCA), a variant of NMF for source separation using invertible CQT. Fuentes et al. [42] also recently published a paper on the topic of melody extraction using probabilistic models based on CQT and NMF.

The music transcription problem is still an unsolved problem [71]. Although most transcription systems focus on multi-pitch analysis, others have target atonal areas. For example, Tzanetakis et al. [106] provided a subband-based drum transcription. Multi-pitch analysis, in general, delivers a basis for note events detection, which results in melodic or harmonic transcription. It can also be used in a more high level analysis, i.e., key detection, harmonic analysis, etc. For example, Gerhard [47] provided an interval analysis using relative pitch. Chuan and Chew [27] implemented a key finding method.

3.5 The Applications of CQT in Music Analysis

CQT has been applied on monophonic as well as polyphonic music analysis such as chord identification [85, 54], chord transcription [72], key detection [112], and score transcription [4]. Purwins et al. [93] used CQT for modulation tracking. CQT has also been used in source separation [96]. Different variations of Constant-Q approach have been used in the literature. For example, Graziosi et al. earlier proposed a modified version of the Constant-

Q, so called mCQFFB, by improving the response characteristics of Fast Filter Bank (FFB) [34, 49]. da C. B. Diniz et al. [31] provided a practical design of filter banks in the AMT context. Argent et al. [4] proposed using CQT for both pitch and onset estimation. A comprehensive survey may be found in [57].

3.6 Summary

An overview of music signal analysis with the focus on music transcription was provided. A survey of related work was given with an emphasis on CQT in polyphonic transcription, and the importance of MFFE and note events analysis in music transcription. In the next chapter, we develop our harmonic analysis algorithm, which in nature is a time-frequency transformation targeted for music analysis. The algorithm is used in this thesis for MFFE and the note events detection.

Chapter 4

The IHA Transform

This chapter presents our novel short-time frequency analysis algorithm. Given a set of reference pitches, the objective of the algorithm is to transform the real-valued time-domain signal into a set of complex time-domain signals in such a way that the amplitude and phase of the resulting signals represent the amplitude and phase of the signal components with respect to a set of reference pitches. The specification of the IHA algorithm as well as the phasor construct for both continuous and discrete forms is provided, in here. The chapter also contributes to the fast real-time implementation of an $M + 1$-delay IHA algorithm.

The IHA problem may be defined as the following. Given an input signal, the objective of the approach is to find a set of pairs of time-varying functions, representing instantaneous amplitude and phase of the signal components. The problem may be broken into two main sub-problems: signal decomposition, and instantaneous amplitude phase estimation. As there are numerous approaches for decomposing an arbitrary signal into its components, our proposed approach is based on constant-Q analysis which is well suited for the processing musical signals.

The IHA transform is derived by using a Constant-Q filtering and performing a linear amplitude and phase estimation scheme, as follows. We use WSFs with regards to the logarithmic spectrum that is used by CQT in order to implement the decomposition. Estimating the instantaneous amplitude and phase components is carried out by applying the constraints mentioned in section 2.4. We used a linear approach, although non-linear methods have also been suggested in the literature [18].

4.1 Constant-Q Filtering

Let $x(t)$ represent the real-valued input signal whose Fourier transform exists [48], and $h_\omega(t)$ be the transfer function of the ideal low-pass filter in the time domain with cut-off frequency of ω:

$$h_\omega(t) = \frac{\omega}{\pi}\text{sinc}\left(\frac{\omega t}{\pi}\right),$$

where $\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$.

Recall the decomposition scheme in (2.2). By applying a series of band-pass filters represented by B_k's, the input signal can be decomposed into a set of $C_k(t)$, as in the following:

$$C_k(t) = \text{P}((1+\frac{\lambda}{2})\omega_k, t) - \text{P}((1-\frac{\lambda}{2})\omega_k, t), \tag{4.1}$$

where $\text{P}(\omega, t)$ represents the output of the low-pass filter

$$\text{P}(\omega, t) = \frac{\omega}{\pi}\int_{-\infty}^{+\infty} x(t-\tau)\text{sinc}\left(\frac{\omega\tau}{\pi}\right)\text{d}\tau. \tag{4.2}$$

Remark 1. *In the above equation, the linear centric approach in* (2.6) *is used. In case of*

using (2.7)*, the output components will be:*

$$C_k(t) = \mathrm{P}(\omega_k\sqrt{\gamma}, t) - \mathrm{P}(\omega_k\frac{1}{\sqrt{\gamma}}, t).$$

Recall the decomposition scheme in section 2.2. In our model, the input signal is composed of quasi-sinusoidals whose frequencies are ω_k's. The quasi-sinusoidal, in here, may be understood as a wave form that best fits a piecewise sinusoid. Various models of quasi-sinusoidals have been used in the literature. For instance, in using K-partials [35], a blind estimation approach can be used for estimating sinusoidal parameters. Partials refer to the fundamental frequency and the overtones. Shahnaz et al. [101] used a different model of K-partials, similar to our approach, for single pitch estimation. We will show later on how their model fits in our model.

Eq. (4.1) specifies the implementation of our constant-Q filtering. Graziosi et al. used a discrete constant-Q filter bank in [49]. Our approach applies the filter bank in the continuous form. Using the continuous implementation has many advantages. For instance, it provides utilizing theoretical approaches by using a continuous model of the signal. We will show how our implementation can improve the estimation of the instantaneous amplitude and phase functions.

4.2 Amplitude and Phase Estimation

Recall (2.2). Assuming B_k's are sufficiently small and $C_k(t)$ approximately represents the k^{th} quasi-sinusoidal component, we can write:

$$C_k(t) \approx a_k(t)\cos(\omega_k t + \Phi_k(t)).$$

By taking derivative of both sides of the equation, we obtain:

$$\frac{\mathrm{d}}{\mathrm{d}t}C_k(t) \approx \left(\frac{\mathrm{d}}{\mathrm{d}t}a_k(t)\right) \cdot \cos(\omega_k t + \Phi_k(t)) - a_k(t) \cdot \left(\omega_k + \left(\frac{\mathrm{d}}{\mathrm{d}t}\Phi_k(t)\right)\right) \cdot \sin(\omega_k t + \Phi_k(t))$$

Since $a_k(t)$ and $\Phi_k(t)$ are locally constant, for small Δt, we write:

$$\begin{cases} C_k(t - \Delta t) \approx a_k(t)\cos(\omega_k(t - \Delta t) + \Phi_k(t)) \\ C_k(t + \Delta t) \approx a_k(t)\cos(\omega_k(t + \Delta t) + \Phi_k(t)) \end{cases}.$$

As $\Delta t \to 0$

$$\tfrac{\mathrm{d}}{\mathrm{d}t}a_k(t) \approx 0, \quad \tfrac{\mathrm{d}}{\mathrm{d}t}\Phi_k(t) \approx 0.$$

therefore:

$$C_k(t) - \frac{\mathbf{j}}{\omega_k}\frac{\mathrm{d}}{\mathrm{d}t}C_k(t) \approx a_k(t)\mathrm{e}^{\mathbf{j}(\omega_k t + \Phi_k(t))}.$$

Thus, we define $\mathrm{H}_k(t)$, the continuous IHA transform of $x(t)$ as

Definition 1 (The continuous IHA transform)**.** *The continuous IHA transform of $x(t)$ with respect to the map function Ω is defined as:*

$$\mathrm{H}_k(t) \stackrel{def}{=} \mathrm{e}^{-\mathbf{j}\omega_k t}\left[1 - \frac{\mathbf{j}}{\omega_k}\frac{\mathrm{d}}{\mathrm{d}t}\right]C_k(t). \tag{4.3}$$

where $\Omega(k) = \langle \omega_k, B_k \rangle$ as specified in (2.5)*.*

$x(t)$ may then be fully reconstructed using:

$$x(t) = \Re\left\{\sum_k \mathrm{H}_k(t)\mathrm{e}^{\mathbf{j}\omega_k t}\right\}$$

The magnitude of the time-varying complex function $\mathrm{H}_k(t)$ in (4.3) delivers the estimates for instantaneous amplitude of the component C_k. It also presents the existence of reference

frequency ω_k in time.

4.3 Discretization

In practical applications, real signals are generally sampled and represented in the discrete form. Therefore, the Constant-Q filtering as well as the amplitude and phase estimation algorithms must be provided in the discrete form, as well. Various approaches have been proposed in the literature for implementing band-pass filtering in the discrete form, most of which are based on DFT. CQT uses STFT for applying short time windows in order to calculate the DFT. It provides acceptable results, especially in combination with other approaches (cf. [4, 49, 31]). We propose using a different approach by remodeling the signal in its original continuous form. This significantly improves the estimation of the instantaneous amplitude and phase functions in the discrete form[1].

In our approach,

1. the discrete signal is initially interpolated in order to be remodeled in the continuous form;
2. the continuous signal is then filtered and the filter outputs are generated;
3. and the result is subsequently represented in the discrete form.

For simplicity, we use the Nyquist-Shannon algorithm among other interpolation techniques. The details are given in the following.

[1] The performance analysis is provided in chapter 7.

Let $x[n]$ represent our discrete signal. If the input signal is a perfectly pitched audio, it can be assumed that the signal consists of a set of piece-wise quasi-sinusoidal components. Therefore, given $x[n]$, a sequence of numbers in the real domain, representing a sampled signal with the sampling period T, where n is an integer and denotes the sample index, the signal decomposition will become:

$$x[k] = \sum_{k} C[k, n],$$

where

$$C[k, n] = a[k, n] \cos(n\omega_k T + \Phi[k, n]).$$

$C[k, n]$'s in the above equation represent the harmonic sinusoidals.

4.4 Derivation of the IHA Transform

Suppose x is band-limited and contains no frequencies higher than π/T where T represents the sampling period. Using sampling theorem[2], x can be reconstructed using the following [76]:

$$\tilde{x}(t) = \sum_{m=-\infty}^{+\infty} x[m] \text{sinc}\left(\frac{t - mT}{T}\right). \tag{4.4}$$

It can be shown that $x[m] = \tilde{x}(mT)$.

To derive the IHA transform, we apply the band-pass filters represented by B_k's on the signal model in the continuous form. Using (4.2), the output of the ideal band bass filter

[2]The Shannon interpolation formula is used for simplicity, as the convolution of the sinc kernel and the low-pass filter function is indeed a sinc function. However, one may use a more modern sampling approach. Comprehensive overview of modern sampling theory may be found in [55, 36, 107]. The infinite latency property of the sinc filters is addressed in section 4.5.1.

will become:

$$\tilde{\mathrm{P}}(\omega, t) = \sum_{m=-\infty}^{+\infty} x[m] \frac{\omega T}{\pi} \mathrm{sinc}\left(\frac{\omega t - m\omega T}{\pi}\right),$$

where $0 \leq \omega \leq \pi/_T$.

By rewriting the above equation in the discrete form ($t = nT$), we will have:

$$\tilde{\mathrm{P}}(\omega)[n] = \sum_{m=-\infty}^{+\infty} x[m] \frac{\omega T}{\pi} \mathrm{sinc}\left(\frac{\omega T}{\pi}(m-n)\right).$$

$\tilde{\mathrm{P}}$ may also be rewritten as:

$$\tilde{\mathrm{P}}(\upsilon)[n] = \sum_{m=-\infty}^{+\infty} \upsilon \mathrm{sinc}\left(\upsilon(m-n)\right) x[m], \tag{4.5}$$

where

$$\upsilon = \tfrac{\omega T}{\pi}, \quad 0 \leq \upsilon \leq 1, \tag{4.6}$$

is to be called the normalized frequency, as (4.5) is independent of T. The normalized frequency is expressed in number of half-cycles per sample. Some authors have used the product of the frequency and the sampling period as the normalized frequency (cf. [4]).

Definition 2 (The normalized frequency)**.** *The normalized frequency υ with respect to the sampling period T is defined as* (4.6)*.*

By using (4.2) and (4.5), we may obtain the filter response $C[k, n]$ as in the following[3]. For simplicity, m is shifted by n.

$$C[k, n] = \sum_{m=-\infty}^{+\infty} x[m+n] F_k[m],$$

[3]cf. Appendix A.1

where

$$F_k[m] = \upsilon_k \cdot (1 + \frac{\lambda}{2}) \cdot \text{sinc}\left(\upsilon_k(1 + \frac{\lambda}{2})m\right) - \upsilon_k \cdot (1 - \frac{\lambda}{2}) \cdot \text{sinc}\left(\upsilon_k(1 - \frac{\lambda}{2})m\right),$$

represents the discrete constant-Q filter, $m \in \mathbb{Z}$, and υ_k represents the k^{th} normalized frequency with respect to T. $F_k[m]$ may also be simplified as:

$$F_k[m] = \lambda \cdot \upsilon_k \cdot \text{sinc}\left(\frac{\lambda m \upsilon_k}{2}\right) \cos(\pi \upsilon_k m). \tag{4.7}$$

Remark 2. *In case of using logarithmic centric approach as in* (2.7)*:*

$$F_k[m] = \upsilon_k \sqrt{\gamma} \text{sinc}\left(m\upsilon_k \sqrt{\gamma}\right) - \upsilon_k \frac{1}{\sqrt{\gamma}} \text{sinc}\left(m\upsilon_k \frac{1}{\sqrt{\gamma}}\right).$$

The final step to formulate the discrete IHA transform is to implement the estimation algorithm in the discrete form. In order to best fit $C[k, n]$ into a piece-wise sinusoid, we assume that $a[k, n]$ and $\Phi[k, n]$ are locally constant. Hence, we can write:

$$C[k, n] \approx a[k, n] \cos(n\pi\upsilon_k + \Phi[k, n]).$$

By using the similar approach as in section 4.2, we can estimate the instantaneous amplitude and phase components by using two consecutive samples, and maximizing the likelihood of having equal amplitudes and a $\pi\upsilon_k$ phase lag:

$$\begin{cases} C[k, n-1] \approx a[k, n] \cos((n-1)\pi\upsilon_k + \Phi[k, n]) \\ C[k, n+1] \approx a[k, n] \cos((n+1)\pi\upsilon_k + \Phi[k, n]) \end{cases}.$$

By representing $a[k, n]$ and $\Phi[k, n]$ in their phasor representation,

$$\text{H}[k, n] = a[k, n] \text{e}^{\mathbf{j}\Phi[k,n]},$$

we can estimate the phasor coefficients using[4]:

$$\mathrm{H}[k,n] \approx \begin{pmatrix} 1 \\ \mathbf{j} \end{pmatrix}^{T} \cdot D(\upsilon_k, n)^{-1} \cdot \begin{pmatrix} C[k,n-1] \\ C[k,n+1] \end{pmatrix},$$

where

$$D(\upsilon, n) = \begin{pmatrix} \cos(n\pi\upsilon - \pi\upsilon) & -\sin(n\pi\upsilon - \pi\upsilon) \\ \cos(n\pi\upsilon + \pi\upsilon) & -\sin(n\pi\upsilon + \pi\upsilon) \end{pmatrix}.$$

By resolving $D(\upsilon, n)^{-1}$ we can simplify the result. Therefore, using the map function Ω, we derive the discrete IHA transform of $x[n]$, a sequence in the complex domain, as given in the following. The definition of the map function Ω is given in (2.5).

Definition 3 (The discrete IHA transform)**.** *The discrete IHA transform of $x[n]$ with respect to the map function Ω is defined as:*

$$\mathrm{H}[k,n] \stackrel{def}{=} \mathrm{e}^{-\mathbf{j}\pi\upsilon_k n} \cdot \wp_{\upsilon_k}\left(C[k,n]\right), \tag{4.8}$$

where $\wp$ is the IHA phasor construct, as defined below:

$$\wp_{\upsilon}(x[n]) \stackrel{def}{=} \frac{\mathbf{j}}{\sin 2\pi\upsilon} \begin{pmatrix} \mathrm{e}^{-\mathbf{j}\pi\upsilon} \\ -\mathrm{e}^{\mathbf{j}\pi\upsilon} \end{pmatrix}^{T} \cdot \begin{pmatrix} x[n-1] \\ x[n+1] \end{pmatrix}.$$

The above construction transforms the real-valued component into its instantaneous phasor representation.

[4] cf. Appendix A.2

4.5 Implementation

This section presents the implementation of constant-Q filtering based on WSFs and the signal model in its original continuous form, which was presented in section 4.3.

4.5.1 Using a Window Function

Consider the filter equation in (4.7). Since F_k is symmetric and attenuating on both sides, one may consider an absolute upper bound for m such that $m \in [-M_k, M_k]$. Smith [103] showed the improvement of filter response by using windowed sync filters, among which Blackman [103] was demonstrated to have the smoothest response:

$$W_{Blackman}[m] = 0.42 - 0.5\cos\left(\frac{\pi(m+M_k)}{M_k}\right) + 0.08\cos\left(\frac{2\pi(m+M_k)}{M_k}\right).$$

CQT suggests only Q number of cycles would be sufficient for the filter implementation [16]:

$$\omega(2M_k+1)T \approx 2\pi Q.$$

By applying (2.8) and (4.6), M_k may be estimated by:

$$M_k = \left\lfloor \frac{2}{\upsilon_k \lambda} \right\rfloor. \tag{4.9}$$

Thus our modified filter response will be:

$$\tilde{C}[k,n] = \sum_m x[m+n] \cdot F_k[m] \cdot W_k[m], \tag{4.10}$$

where $W_k[m]$ represents the symmetric window function[5].

[5] $W_k[m] = W_k[-m]$

4.5.2 Frequency Upper Bound

One of the key constraints in using sampling theorem is that the signal is to be band-limited. As a result, the center frequencies υ_k's are bounded, as well. For instance, in case of using (2.6), the upper bound for υ_k may be obtained by[6]:

$$\upsilon_k \leq \frac{2}{\lambda + 2}$$

Remark 3. *In case of using logarithmic centric approach, the following inequality may be used:*

$$\upsilon_k \leq \frac{1}{\sqrt{\gamma}}$$

Therefore, given υ_0, the normalized global reference frequency were $0 < \upsilon_0 < 1$, the upper bounds for k, the frequency index, may be obtain by using (2.5), as in the following:

$$k \leq -\frac{1}{\log \gamma} \left(\log \upsilon_0 + \log(\lambda + 2) - \log 2\right)$$

Remark 4. *Similarly, in case of using logarithmic centric approach, upper bound for k will be:*

$$k \leq -\frac{\log \upsilon_0}{\log \gamma} - \frac{1}{2}$$

4.5.3 Frequency Lower Bound

Eq. (4.9) also indicates that the number of samples that are required for each iteration increases as υ_k approaches lower frequencies. In practice, where there exists a time quantum, one may consider an upper bound for M. Hence, the lower bound for the reference frequency

[6] cf. Appendix A.3

can be derived using $\gamma\upsilon_k$ as the frequency resolution (cf. [28]):

$$\upsilon_k \geq \frac{2}{M_{max}\lambda}.$$

A lower bound for k may also be obtained by:

$$k \geq \frac{1}{\log\lambda}\left(\log 2 - \log M_{max} - \log \upsilon_0\right) - 1.$$

4.5.4 Unity Gain and Zero Phase Shift

Recall the decomposition scheme presented in section 4.3. In practice, both k and n are bounded. Hence, using a window function delivers undesirable but inconsiderable noise in the frequency response, cf. [18]. Using such windows results in amplitude loss and phase lag. To overcome this, in order to preserve the unity gain, we may include an amplification factor in the window function. To calculate such amplification factor, a test signal, i.e.,

$$\cos(m\pi\upsilon_k), \quad m \in [-M_k, M_k],$$

may be used. Therefore, the amplification factor may be estimated using calculating the inverse of the absolute value of the median of the output (where $m = 0$).

Furthermore, a similar approach may be used to overcome the phase lag issue. The amplification factor may then be generalized into a complex number whose amplitude and phase equate the amplification factor and the inverse phase of the median, respectively. Therefore, the modified phasor coefficients can be rewritten as:

$$\mathrm{H}[k,n] \approx z_k \cdot \mathrm{e}^{-\mathbf{j}\pi\upsilon_k m} \cdot \wp_{\upsilon_k}\left(C[k,n]\right), \tag{4.11}$$

where z_k represents the complex amplification factor. Since both k and n are bounded, the filter banks produce undesirable non-zero H$[k, n]$ for non-existing frequencies. To overcome this, a simple threshold technique may also be used. The overall error, caused by z_k's, will consequently be minimized.

4.5.5 The Resolution Pyramid

Using (4.9), one may minimize M by performing the convolution in a lower resolution, where υ_k is maximized. In order to perform the IHA transform in the original resolution, a linear interpolation technique may be applied on both the amplitude and the phase, individually. It can be shown that the latency, as calculated in the following, is constant, regardless of the resolution in use. The latency, in here, may be interpreted as the amount of time that the filter requires to produce unity gain response:

$$l = M_k \cdot T \tag{4.12}$$

Using our approach, the number of samples is adjusted according to the frequency bin, whereas in mCQFBB a fixed number is used [49].

4.5.6 Fast Realtime Implementation

Several papers have been published on the efficiency of the various implementations of CQT, cf. [100]. In our approach, the real-time sliding window may be implemented by using an $M_k + 1$-delay component. Eq. (4.9) indicates that the number of samples that are required for each iteration increases as υ_k approaches lower frequencies. We may minimize M_k by performing the convolution operation in a lower resolution, where υ_k is maximized. As a

Algorithm 1: Realtime IHA algorithm

Input: k: the frequency index,
ϵ: amplitude threshold,
$x[n]$: a stream of real numbers representing the input signal

Output: $\mathrm{H}[k, n]$: a delayed stream of complex numbers representing the IHA transform of $x[n]$

```
choose appropriate resolution pyramid based upon frequency index, as explained in 4.5.5
perform the M + 1-delay IHA algorithm, as specified in Alg. 2
perform the following filter on H[k, n]
    foreach H[k, n] do
        if H[k, n] ≤ ε then
            H[k, n] ← 0
perform an up-resolution operation on H[k, n], if necessary, using a linear interpolation approach on both instantaneous amplitude and phase, individually, as specified in 4.5.5
output H[k, n]
```

result, the phasor coefficients in the original resolution may be estimated by using a linear interpolation technique on both the amplitude and the phase, individually. Due to the limited number of frequencies in practice, a maximum of 8-level resolution-pyramid may be used.

Fig. 4.1 presents the block diagram of our real-time IHA system. As illustrated, it consists of two delay buffers, a total of $M + 1$ samples delay. The complexity of the above construct in time and space is therefore $\mathcal{O}(Mn)$ and $\mathcal{O}(M)$, respectively (cf. [61, 56]). Algorithm 1 specifies the implementation of the real-time system. The input signal x, in this implementation, is represented by an input stream and is assumed to be zero-padded.

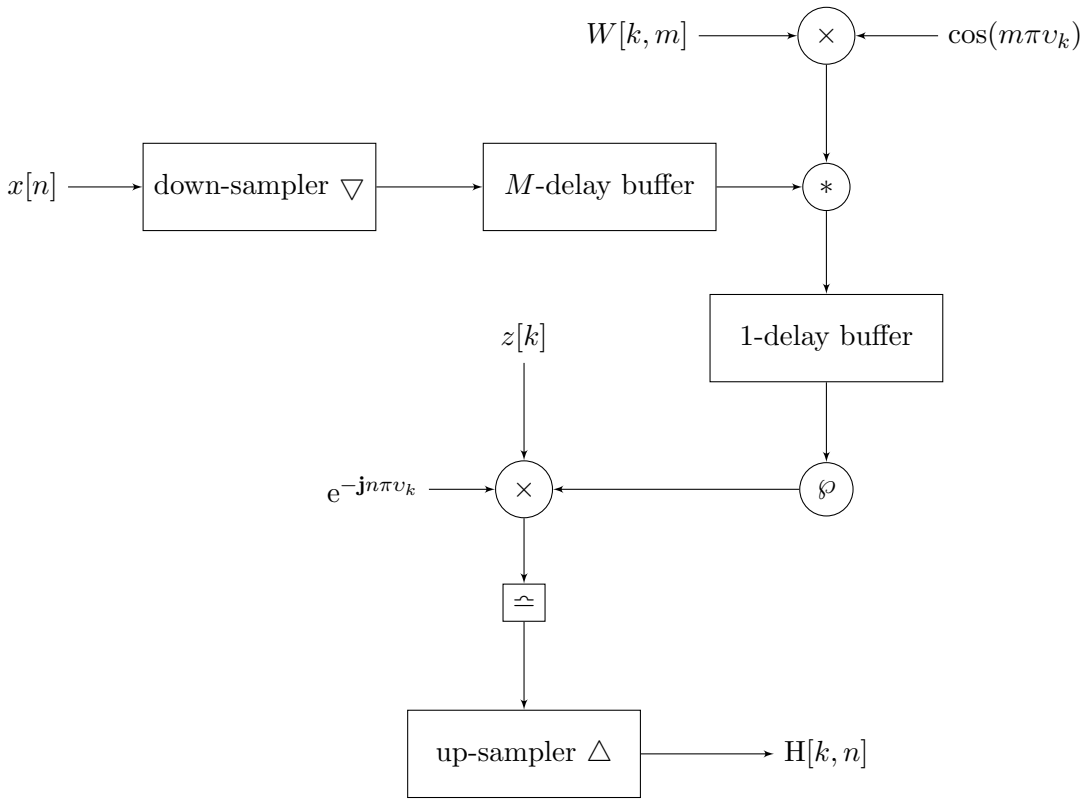


Figure 4.1: Block diagram of the real-time IHA system

Algorithm 2: The $M+1$-delay fast online discrete IHA block

Input: k: the frequency index,
$x[n]$: a stream of real numbers representing the input signal
Output: $\mathrm{H}[k,n]$: a delayed stream of complex numbers representing the IHA transform of $x[n]$

estimate M using (4.9)
construct the window function $W[k,m]$ using desired window (i.e. Blackman)
construct the template signal: $\cos(m\pi\upsilon_k)$
construct the filter: $F_k \leftarrow W[k,m] \cdot \cos(m\pi\upsilon_k)$
adjust z_k using the template signal and the approach specified in 4.5.4
input $x[n]$ using and M-delay buffer
foreach $x[n]$ **do**
    calculate the filter response using (4.10)
    estimate $\mathrm{H}[k,n]$ using a 1-delay buffer and the modified phasor construct in (4.11)
    output $\mathrm{H}[k,n]$

4.6 Summary

The IHA algorithm was formalized by using the decomposition scheme, presented in section 2.2, and employing the phasor construct, specified in (4.8). The normalized frequency, presented in the discretization section, makes the derivation independent of sampling frequency. We provided a fast implementation of the IHA algorithm for being used in realtime systems. The analysis of the algorithm will be provided in the chapter 7.

Chapter 5

Generalization into the Multi-Dimensional Space

This chapter contributes to the instantaneous harmonic analysis of multi-dimensional signals. A quaternion representation is proposed to support multiple phase elements. The space is extended into hyper-complex in order to address multiple phase elements.

Hamilton's quaternions are the extension of complex numbers into a higher dimension [53]. They are based on one real and three imaginary components represented by three imaginary units such as **i**, **j**, and **k** where $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. Using the Cayley-Dickson construct, quaternions can also be extended into hyper-complex numbers in which each number consists of one real and $2^M - 1$ imaginary components where $M > 2$. Quaternions have been widely used by many researchers and they are reported in the literature for the processing and representation of multidimensional data such as computer graphics, computer aided geometry and design, signal processing, etc. Whilst they were introduced in 1843 by Hamilton, they have not been applied in signal processing until recent years [99]. In

signal processing, many quaternion applications can be found in time-frequency analysis of multi-dimensional signals: Quaternion Fourier Transform (QFT) [99, 110], Quaternion Discrete Cosine Transform (QDCT) [39], quaternion wavelets [24], and hyper-complex wavelet transform [25].

Similar to the one-dimensional IHA algorithm, the goal of the multi-dimensional IHA transform is to transform the real-valued multi-dimensional time-domain input signal into a set of hyper-complex time-domain signals in such a way that the amplitude and phase elements of the resulting signals represent the amplitude and phase elements of the signal components. To achieve this, the space is extended into a hyper-complex in order to address multiple phase elements.

The input signal, in our approach, is represented by a multi-dimensional real function whereas in the literature other models have been used. For instance, QFT in [99] the signals in the hyper-complex space. Using such model provides a transformation algorithm that can be applied in higher dimensions. For instance, Sangwine suggests forming the hyper -complex signal by putting the RGB color components into the hyper-complex imaginary parts and leaving the real part with zero.

In the following sections, we will extend the IHA algorithm to address multi-dimensional signals in both continuous and discrete forms. The generalization is given in four parts: the two-dimensional continuous algorithm, the multi-dimensional continuous algorithm, the two-dimensional discrete algorithm, and the multi-dimensional discrete algorithm.

5.1 Quaternion Representation

In order to construct the phasor operator in the multi-dimensional space, we suggest taking the same approach as in section 2.4. To do so, we take the complex value $a(\cos\phi + \mathbf{j}\sin\phi)$ and extend it into quaternion space: $a(\cos\phi_1 + \mathbf{i}\sin\phi_1)(\cos\phi_2 + \mathbf{j}\sin\phi_2)$ where $\mathbf{i}$ and $\mathbf{j}$ are the 1st and the 2nd imaginary units in the quaternion space. This will result in the following quaternion:

$$a\cos\phi_1\cos\phi_2 + a\sin\phi_1\cos\phi_2\mathbf{i} + a\cos\phi_1\sin\phi_2\mathbf{j} + a\sin\phi_1\sin\phi_2\mathbf{k}$$

Therefore, using a above redundant construct, the tuple $\langle\alpha, \phi_1, \phi_2\rangle$ may be represented by a quaternion were α represents the amplitude and ϕ_1, ϕ_2 represent the phases in $\mathbb{R}^2$.

Definition 4 (The quaternion representation)**.** *A tuple $\langle\alpha, \phi_1, \phi_2\rangle$ may be represented by a quaternion using the following construct:*

$$\mathcal{Q} \stackrel{def}{=} a \cdot \left(\left(\begin{array}{c} \cos(\phi_1) \\ \sin(\phi_1) \end{array} \right) \otimes \left(\begin{array}{c} \cos(\phi_2) \\ \sin(\phi_2) \end{array} \right) \right)^T \cdot U_2 \tag{5.1}$$

where

$$U_2 = \left(\begin{array}{cccc} 1 & \mathbf{i} & \mathbf{j} & \mathbf{k} \end{array} \right)^T.$$

U_2 represents the vector of units in the quaternion space. T and $\otimes$ denote the matrix transpose and Kronecker product operations, respectively [53].

Likewise, given any arbitrary quaternion $\mathcal{Q}$, it can be shown that in order for $\mathcal{Q}$ to be a redundant representation, the condition $e(\mathcal{Q}) = 0$ must be satisfied. $e(\mathcal{Q})$ is referred to as the eccentricity of $\mathcal{Q}$ and defined as the following.

Definition 5 (The eccentricity of quaternion representation)**.** *The eccentricity of a quater-*

nion $H = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ *is defined as:*

$$e(\mathcal{Q}) \stackrel{def}{=} \frac{|ad - bc|}{\max(|ad|, |bc|)} \tag{5.2}$$

The quaternion representation may also be extended into the M-dimensional space, where U_M represents the vector of units in the hyper-complex space:

$$U_M = \begin{pmatrix} 1 & \mathbf{j}_1 & \cdots & \mathbf{j}_{2^M-1} \end{pmatrix}^T,$$

and $\mathbf{j}_i$ designates the i^{th} imaginary unit in the hyper-complex space [53].

Definition 6 (The hyper-complex representation)**.** *A tuple* $\langle \alpha, \phi_1, \cdots, \phi_M \rangle$ *may be represented by a hyper-complex number using the following construct:*

$$\mathcal{Q} \stackrel{def}{=} a \cdot \left(\bigotimes_{i=1}^{M} \begin{pmatrix} \cos(\phi_1) \\ \mathbf{j}_{2^{i-1}} \sin(\phi_1) \end{pmatrix} \right)^T \cdot U_M \,. \tag{5.3}$$

Using the above construct, the number of redundancies becomes $2^M - M - 1$. As a result, in order for $\mathcal{Q}$ to be a hyper-complex representation of the tuple $\langle \alpha, \phi_1, \cdots, \phi_M \rangle$, the quaternion property must be satisfied for all $M(M-1)/2$ combinations of every two dimensions i, j that are used in the calculation. The vector of units may alternatively be derived using the following construct, as well:

$$U_M = \bigotimes_{i=1}^{M} \begin{pmatrix} 1 \\ \mathbf{j}_{2^{i-1}} \end{pmatrix}.$$

5.2 The Two-Dimensional Continuous IHA Algorithm

The two-dimensional continuous IHA algorithm is derived by extending the approach that has been used in section 4.2 into the quaternion space and using the quaternion representation that was specified in section 5.1;

Let $x(T)$ be a Fourier-transformable two-dimensional time-domain continuous signal where $T = (t_i) \in \mathbb{R}^2$ represents the time vector and i represents the dimension index. Let $K = (k_i) \in \mathbb{Z}^2$ represent the reference frequency index vector, $\Omega_K = (\omega_{k_i}) \in \mathbb{R}^{+2}$ be the reference frequency vector, and $B_{k_i} = [{\omega^1}_{k_i}, {\omega^2}_{k_i}]$ be a band around ω_{k_i} where ${\omega^2}_{k_i}$ and ${\omega^1}_{k_i}$ represent the upper bound and lower bound of the band and $0 \leq {\omega^1}_{k_i} < \omega_{k_i} < {\omega^2}_{k_i} \leq 1$, using a map function similar to (2.5)[1]. Similar to the approach in section 4.1, the filter output $C_K(T)$ may be derived as in the following:

$$C_K(T) = x(T) * \zeta(T), \tag{5.4}$$

where $\zeta(T)$ represents the filter function:

$$\zeta(T) = \frac{1}{\pi^2} \prod_{i=1}^{2} \left[{\omega^2}_{k_i} \mathrm{sinc}(\frac{{\omega^2}_{k_i} t_i}{\pi}) - {\omega^1}_{k_i} \mathrm{sinc}(\frac{{\omega^1}_{k_i} t_i}{\pi}) \right],$$

and $*$ denotes the two-dimensional convolution.

If B_{k_i} is sufficiently small, we can assume $C_K(T)$ forms a quasi-sinusoidal, therefore:

$$C_K(T) \approx a_K(T) \cos(\omega_{k_1} t_1 + \Phi_{k_1} t_1) \cos(\omega_{k_2} t_2 + \Phi_{k_2} t_2).$$

Thus, we define $\mathrm{H}_K(T)$, the IHA transform of $x(T)$ as the following:

[1] We differentiate Ω, the frequency vector, from Ω in (2.5) which represents the quantization map function

Definition 7 (The continuous two-dimentional IHA transform)**.** *The continuous two-dimentional IHA transform of* $x(T)$ *is defined as:*

$$\mathrm{H}_K(T) \stackrel{def}{=} \Xi_K\{C_K(T)\} \tag{5.5}$$

where

$$\Xi_K = {U_2}^T . \bigotimes_{i=1}^{2} (\wp(\omega_{k_i}, t_i)),$$

$$\wp(\omega, t) = \begin{pmatrix} \cos(\omega t) & -\sin(\omega t) \\ -\sin(\omega t) & -\cos(\omega t) \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \frac{1}{\omega}\frac{\partial}{\partial t} \end{pmatrix}.$$

5.3 The Multi-Dimensional Continuous IHA Algorithm

The multi-dimensional IHA algorithm can similarly be derived by using the same approach presented in 5.2 and extending the number of dimensions to M. The definition of the transformation is given in the following where Ξ_K represents the phasor construct in M-dimentional space. $C_K(T)$, the filter output is derived using the approach in (5.4) where ζ represents the filter function in the M-dimensional space:

$$\zeta(T) = \frac{1}{\pi^M} \prod_{i=1}^{M} \left[{\omega^2}_{k_i} \mathrm{sinc}(\frac{{\omega^2}_{k_i} t_i}{\pi}) - {\omega^1}_{k_i} \mathrm{sinc}(\frac{{\omega^1}_{k_i} t_i}{\pi}) \right].$$

Definition 8 (The continuous multi-dimentional IHA transform)**.** *The continuous multi-dimentional IHA transform of* $x(T)$ *is defined as:*

$$\mathrm{H}_K(T) \stackrel{def}{=} \Xi_K\{C_K(T)\} \tag{5.6}$$

where

$$\Xi_K = {U_M}^T \cdot \bigotimes_{i=1}^{M} (\wp(\omega_{k_i}, t_i))$$

$$\wp(\omega, t) = \begin{pmatrix} \cos(\omega t) & -\sin(\omega t) \\ -\sin(\omega t) & -\cos(\omega t) \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \frac{1}{\omega}\frac{\partial}{\partial t} \end{pmatrix},$$

It can be shown that in case of $M = 1$, the above equation becomes (4.3)[2].

5.4 The Two-Dimensional Discrete IHA Algorithm

The two-dimensional discrete IHA algorithm may be derived by using a similar approach in section 4.3 and extending the number of dimensions to two. The derivation is given in the following.

Let $x[N] : \mathbb{Z}^2 \to \mathbb{R}$ be a two-dimensional sequence of numbers in the real domain where $N = (n_i) \in \mathbb{Z}^2$ represents the sample index vector, and $\mathbf{T} = (T_i) \in \mathbb{R}^2$ represents the sampling period vector[3], and i represents the dimension index. Using sampling theorem, we can derive the continuous form of the signal as:

$$\tilde{x}(T) = \sum_N x[N] \cdot \operatorname{sinc}\left(\frac{t_1 - n_1 T_1}{T_1}\right) \operatorname{sinc}\left(\frac{t_2 - n_2 T_2}{T_2}\right). \tag{5.7}$$

where $T = (t_i) \in \mathbb{R}^2$ represents the time vector.

Let $K = (k_i) \in \mathbb{Z}^2$ be the reference frequency index vector. Using similar approach in (4.6), we derive the normalized frequency vector $\Upsilon_K = (\upsilon_{k_i}) \in \mathbb{R}^2$ as:

$$\Upsilon_K = \frac{1}{\pi}\left({\Omega_K}^T \mathbf{T}\right), \tag{5.8}$$

[2] cf. Appendix A.4

[3] $\mathbf{T}$ represents the sampling period vector whereas T represents the time vector in the continuous form

where $\Omega_K = (\omega_{k_i}) \in \mathbb{R}^{+2}$ represents the actual reference frequency vector.

The filter output can similarly be derived as:

$$C[K,N] = \sum_{\dot{N}} \prod_{i=1}^{2} \left[{\upsilon^2}_{k_i} \mathrm{sinc}\left({\upsilon^2}_{k_i} (\dot{n}_i - n_i) \right) - {\upsilon^1}_{k_i} \mathrm{sinc}\left({\upsilon^1}_{k_i} (\dot{n}_i - n_i) \right) \right] \cdot x[\dot{N}], \tag{5.9}$$

where $0 \leq {\upsilon^1}_{k_i} < \upsilon_{k_i} < {\upsilon^2}_{k_i} \leq 1$. ${\upsilon^2}_{k_i}$ and ${\upsilon^1}_{k_i}$ represent the upper bound and lower bound of $B_{k_i} = [{\upsilon^1}_{k_i}, {\upsilon^2}_{k_i}]$, the normalized band around υ_{k_i} similar to section 5.2.

If B_{k_i} is sufficiently small, we can assume $C[K,N]$ forms a quasi-sinusoidal, therefore:

$$C[K,N] \approx a[K,N] \cos(n_1 \pi \upsilon_{k_1} + \Phi[k_1, n_1]) \cos(n_2 \pi \upsilon_{k_1} + \Phi[k_2, n_2]).$$

Using the 4 neighbors at N in all four directions: N, E, S, W (using the points of a compass), we may write:

$$\begin{cases} C[K, N + P^1] \approx a[K,N] \cos((n_1 - 1)\pi \upsilon_{k_1} + \Phi[k_1, n_1]) \cos(n_2 \pi \upsilon_{k_1} + \Phi[k_2, n_2]) \\ C[K, N + P^2] \approx a[K,N] \cos((n_1 + 1)\pi \upsilon_{k_1} + \Phi[k_1, n_1]) \cos(n_2 \pi \upsilon_{k_1} + \Phi[k_2, n_2]) \\ C[K, N + P^3] \approx a[K,N] \cos(n_1 \pi \upsilon_{k_1} + \Phi[k_1, n_1]) \cos((n_2 - 1)\pi \upsilon_{k_1} + \Phi[k_2, n_2]) \\ C[K, N + P^4] \approx a[K,N] \cos(n_1 \pi \upsilon_{k_1} + \Phi[k_1, n_1]) \cos((n_2 + 1)\pi \upsilon_{k_1} + \Phi[k_2, n_2]) \end{cases}$$

where P^i is the i^{th} column of the following matrix:

$$P = \begin{pmatrix} -1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

By using the quaternion representation, as described in section 5.1, and technique used in section 4.4, we may derive the two-dimensional discrete IHA transform, as in the following:

Definition 9 (The two-dimentional discrete IHA transform)**.** *The two-dimentional discrete IHA transform of* $x[N]$ *is defined as:*

$$\mathrm{H}[K,N] \stackrel{def}{=} \Xi_K\{C[K,N]\} \tag{5.10}$$

where Ξ *is the two-dimensional phasor construct:*

$$\Xi_K(x[N]) \stackrel{def}{=} \prod_{i=1}^{2} \mathrm{e}^{-\mathbf{j}_{2^{i-1}}\pi\upsilon_k m} \cdot \wp_{\upsilon_k,i}\left(x[N]\right),$$

$$\wp_{\upsilon,i}(x[N]) \stackrel{def}{=} \frac{\mathbf{j}_{2^{i-1}}}{\sin 2\pi\upsilon} \begin{pmatrix} \mathrm{e}^{-\pi\upsilon\mathbf{j}_{2^{i-1}}} \\ -\mathrm{e}^{\pi\upsilon\mathbf{j}_{2^{i-1}}} \end{pmatrix}^T \cdot \begin{pmatrix} x[N+P^i{}_1] \\ x[N+P^i{}_2] \end{pmatrix},$$

$$P = \begin{pmatrix} -1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and $P^i{}_1$ *,* $P^i{}_2$ *are the first and second elements of the* i^{th} *row of the matrix* P*.*

5.5 The Multi-Dimensional Discrete IHA Algorithm

The multi-dimensional discrete IHA algorithm may be derived by using a similar approach in section 5.4 by extending the number of dimensions to M, as specified in the following:

Definition 10 (The multi-dimentional discrete IHA transform)**.** *The multi-dimentional discrete IHA transform of* $x[N]$ *is defined as:*

$$\mathrm{H}[K,N] \stackrel{def}{=} \Xi_K\{C[K,N]\} \tag{5.11}$$

where Ξ *is the multi-dimensional phasor construct:*

$$\Xi_K(x[N]) \stackrel{def}{=} \prod_{i=1}^{M} \mathrm{e}^{-\mathbf{j}_{2^{i-1}}\pi\upsilon_k n} \cdot \wp_{\upsilon_k,i}\left(x[N]\right),$$

$$\wp_{\upsilon,i}(x[N]) \stackrel{def}{=} \frac{\mathbf{j}_{2^{i-1}}}{\sin 2\pi\upsilon} \begin{pmatrix} \mathrm{e}^{-\pi\upsilon\mathbf{j}_{2^{i-1}}} \\ -\mathrm{e}^{\pi\upsilon\mathbf{j}_{2^{i-1}}} \end{pmatrix}^T \cdot \begin{pmatrix} x[N+P^i_{\ 1}] \\ x[N+P^i_{\ 2}] \end{pmatrix},$$

$$P = \begin{pmatrix} -1 & 1 \end{pmatrix} \otimes I_M,$$

I_M *represents the identity matrix in* $M \times M$*, and* $P^i_{\ 1}$ *,* $P^i_{\ 2}$ *are the first and second elements of the* i^{th} *row of the matrix* P*.*

It can be shown that in case of $M = 1$, the above equation becomes (4.8)[4].

5.6 Remarks

By definition, the quaternion representation is redundant. It can be shown that in order for H in (5.10) to satisfy the quaternion representation property, the following matrix must also satisfy the property:

$$\Gamma = \begin{pmatrix} C[K, N+P^1] \\ C[K, N+P^2] \\ C[K, N+P^3] \\ C[K, N+P^4] \end{pmatrix}.$$

This cannot generally be satisfied. Thus one may apply a normalization operation on Γ before using it in (5.10). However, for simplicity, this can be bypassed due to the special interest in estimating the instantaneous amplitude. Both approaches yet provide acceptable

[4] cf. Appendix A.5

results. Similar approach may be used in the M-dimensional case. The statement may be generalized into the multi-dimensional space, in which case the quaternion property is applied on all $M(M-1)/2$ combinations of every two dimensions i, j, which are used in the calculations of $\wp_{v,i}(x[N])$ and $\wp_{v,j}(x[N])$, as specified in (5.11).

Although the IHA transform outperforms the QFT approach in terms of providing the instantaneous amplitude and phase elements, the IHA algorithm cannot be used for the multi-channel signals unless each channel is processed individually. The IHA transform is quite similar to the quaternion wavelet transform in [24] in terms of using a redundant representation. Our approach, however, outperforms the latter in terms of estimating the instantaneous amplitude of the signal vs. the oscillating wavelet response. And finally, the time complexity of the algorithm may be reduced by utilizing symmetric property of the matrices.

Chapter 6

Generalization of IHA Algorithm for Multiple Fundamental Frequency Estimation

This chapter contributes to the generalization of the IHA algorithm by utilizing a composite kernel. The generalized IHA algorithm contributes to the MFFE process. A bottom-up overtone elimination approach is proposed by utilizing the representation of the kernel function by a sequence of complex values. The generalized IHA algorithm forms the core of our audio-to-MIDI system, which will be specified in chapter 8.

In this chapter, we generalize the IHA transform that was specified in chapter 4, by utilizing a composite kernel function. Considering IHA to provide the instantaneous pure components of the input signal, the generalized IHA estimates the instantaneous amplitude and phase elements of the components based upon a generalized kernel function. Such generalization makes it possible to use IHA in an MFFE process, where a multi-pitch

analysis is required for detecting multiple fundamental frequencies.

To overcome the harmonic collisions, several approaches may be used. Recently, Chen and Liu used a modified harmonic product spectrum technique by calculating the magnitude of the STFT for all the integer harmonics [26]. We propose using a bottom-up approach in order to eliminate the overtones, as follows. In our model, the timbre is modeled by a set of complex numbers, representing the amplification and phase lag of the subsequent overtones, generated from the fundamental tone. Thus, starting from the lowest frequency, the overtones are estimated and subsequently removed from the higher frequencies. Our approach is sensitive to the fundamental frequencies while in [26] a product of the magnitude of all integer harmonics is used.

In the following sections, the derivation of the generalized IHA algorithm by specifying the properties of the kernel function and the constraints that are used in the decomposition scheme is provided.

6.1 Overview

The IHA transform provides a decomposition framework for transforming an arbitrary signal into a set of pure sinusoidals, given a set of reference frequencies. In here, we generalize the kernel function $e^{\mathbf{j}t}$ in (2.10), into a generic periodic function such that the input signal can be decomposed into a set of composite components. Hence, the objective of the algorithm is to generalize the kernel function $e^{\mathbf{j}t}$ into a set of generic periodic functions $\psi(t)$ such that it can be used in the following decomposition scheme:

$$x(t) = \Re\{\sum_{k} \mathrm{H}_k(t)\psi_A(\omega_k t)\} + r(t), \tag{6.1}$$

where $\psi_A(t)$ denotes the analytical form of $\psi(t)$:

$$\psi_A(t) = \psi(t) + \mathbf{j} \cdot \mathcal{H}\{\psi(t)\},$$

$\mathcal{H}$ denotes the Hilbert transform, and $r(t)$ represents a residual signal.

6.2 Kernel Properties

Let $\psi(t) : \mathbb{R} \to \mathbb{R}$ be an arbitrary Fourier tranformable smooth periodic function in $\mathbb{R}$ with the following properties:

$$\psi(t) = \psi(t + 2\pi), \tag{6.2}$$

$$\int_{-\pi}^{\pi} \psi(t)\mathrm{d}t = 0.$$

Remark 5. *The period value 2π is chosen for simplicity. It does not impose any restriction on the algorithm. A rescaling operation may be applied to any given kernel to achieve a 2π period.*

Using the theory of Fourier series, we can show that the kernel function may alternatively be derived by:

$$\psi(t) = \frac{1}{\sqrt{2\pi}} \sum_{k=-\infty}^{+\infty} \cdot \Psi_k \mathrm{e}^{\mathbf{j}kt}$$

where Ψ_k represents the k^{th} coefficient and is obtained by:

$$\Psi_k = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \psi(t)\mathrm{e}^{-\mathbf{j}kt}\mathrm{d}t.$$

It must be noted that $\Psi_0 = 0$ and $\Psi_k = \bar{\Psi}_{-k}$, where $\bar{\Psi}_k$ represents the complex conjugate of Ψ_k. Hence the kernel function $\psi(t)$ can be represented by a sequence of complex coefficients,

as the following:

$$(\Psi_k)_{k=1}^{\infty}\,.$$

6.3 Estimating the Kernel Function

Estimating the Kernel function from a sampled wave-form is a straightforward practice, as kernels are represented by a series of complex coefficients Ψ_k, where $k \in \mathbb{N}$. In practice, the input wave-form is band-limited, and therefore, there exist an upper bound for k such that $k \leq k_{\max}$.

Let $\psi[k]$ represent sampled wave-form of the kernel $\psi(t)$ with frequency f and sampling frequency f_s where $f_s \in \mathbb{N}$. Using sampling theorem, we can write:

$$\psi(2\pi f t) = \sum_{m=-\infty}^{+\infty} \psi[m]\mathrm{sinc}\,(f_s t - m) \tag{6.3}$$

Since ψ is periodic, using (6.2), we can obtain:

$$\psi[m] = \psi(\frac{2\pi m}{M}), \tag{6.4}$$

where $m \in [0, M]$, $M \in \mathbb{N}$, and $M = \frac{f_s}{f}$.

For simplicity, f is assumed to be is a divisor of the sampling frequency. In general, a simple interpolation technique may be applied by virtually choosing a new sampling frequency as:

$$f_{s\mathrm{new}} = \gcd(f_{s\mathrm{old}}, f),$$

where gcd represents the greatest common divisor.

Using (6.4), and by applying Fourier transform on both sides of (6.3), we obtain:

$$\mathcal{F}\{\psi(2\pi ft)\} = \frac{1}{\sqrt{2\pi}} \sum_{n=-\infty}^{+\infty} \sum_{m=0}^{M-1} \frac{1}{f_s} \psi[m] \mathrm{e}^{\frac{-\mathbf{j}(nM+m)\omega}{f_s}}, \tag{6.5}$$

where $0 \leq \omega \leq \pi f_s$.

Using (6.2), we can also derive the Fourier transform of ψ:

$$\mathcal{F}\{\psi(2\pi ft)\} = \sqrt{2\pi} \sum_{k=-\infty}^{+\infty} \Psi_k \cdot \delta(\omega - 2\pi fk), \tag{6.6}$$

where $1 \leq k \leq \frac{M}{2}$.

By equating (6.5) and (6.6), and using the limit theorem on $\omega \to 2\pi fk$, we can estimate Ψ_k by:

$$\Psi_k = \frac{2}{M} \sum_{m=1}^{M} \psi[m] \cdot \mathrm{e}^{\frac{-\mathbf{j}2\pi mk}{M}}. \tag{6.7}$$

The existence of inharmonicity in real audio signals makes the kernel estimation process difficult. Inharmonicity is the measure to which the frequencies of the overtones do not equate the integer multiples of the fundamental frequency. String instruments, for instance, are known to produce imperfect harmonic tones. To overcome the inharmonicity issue, we use a regression-based estimation algorithm, as specified in Alg. 3.

Although perfect harmonics are desirable, inharmonic sounds are not necessarily unpleasant. Among musicians, inhamonicity is sometimes referred to as the warmth property of the sound. The topic has been researched in the audio processing field, especially in sound synthesis [64].

Algorithm 3: Kernel estimation using regression

Input: $x[n]$: a stream of real numbers, representing the sample wave
M: number of coefficients to estimate

Output: (Ψ_k): kernel function represented by a sequence of complex coefficients

1 estimate υ_0, the normalized fundamental frequency of $x[n]$

2 set $q_k \leftarrow \arg_n(\upsilon[n], k\upsilon_0), \forall k \in [1, M]$ `/* overtone indices */`

3 calculate $\mathrm{H}[q_k, n]$, the IHA algorithm of $x[n]$, for $k \in [1, M]$

4 set $X \leftarrow \{\mathrm{H}[q_1, n], |\mathrm{H}[q_1, n]| \geq \alpha\}$

5 set $Y \leftarrow \{\mathrm{H}[q_k, n], k \in [2, M], |\mathrm{H}[q_1, n]| \geq \alpha\}$ `/* n corresponds the elements in X */`

6 estimate $\langle A, \Theta \rangle$ by applying a linear regression on $|X|, |Y|$ and $\angle X, \angle Y$, respectively.

7 estimate Ψ_k's using

$$\Psi_k = \begin{cases} 1 & k = 1 \\ A_k \mathrm{e}^{\mathbf{j}\theta_k} & k > 1 \end{cases}$$

8 output (Ψ_k)

6.4 On the Decomposition Scheme

The Fourier transform is an integral transform that uses an orthonormal kernel: $\mathrm{e}^{\mathbf{j}t}$ [48]. An integral transform may generally be represented by:

$$X(\omega) = \int_{t_1}^{t_2} x(t)\xi^*(\omega, t)\mathrm{d}t,$$

which delivers the following decomposition scheme:

$$x(t) = \int_{\omega_1}^{\omega_2} X(\omega)\xi(\omega, t)\mathrm{d}\omega,$$

where ξ represents the decomposition kernel function, associated with an forward kernel ξ^*[1]. In Fourier transform, $\xi(\omega, t)$ is set to $\mathrm{e}^{\mathbf{j}\omega t}$. Our decomposition scheme suggests using $\xi(\omega, t) = \psi(\omega t)$, cf. (6.1). Such a kernel, however, does not provide an orthornormal basis for the integral transformation, mainly for the reason that the solution to the integral

[1] In some texts, ξ^* is referred to as the kernel and ξ is identified as the reverse kernel.

transformation does not exist[2].

Several papers have been published in the literature to find an orthonormal basis for using composite kernels. The wavelet transform, for instance, suggests using two variables, and therefore, provides a two-dimensional transform. The main reasons behind the divergence of the integral transform, using the kernel $\psi(\omega t)$, is that the residual error from the lower frequencies is propagated and accumulated in the higher frequencies. For that reason, we proposed the residual function $r(t)$ in 6.1. The solution is derived in the following.

6.5 The Generalized Discrete IHA Algorithm

The generalized discrete IHA algorithm forms the basis of our MFFE process. Using the harmonic structures [26], it delivers a timbral analysis of input signal where a multi-pith estimation is performed. The algorithm uses the harmonic structure, provided by the IHA core, and transforms them into multi-pitch estimation. The algorithm is based upon the idea behind the integral transformation, presented in section 6.4, with regards to the following suppositions:

1. The discrete IHA delivers the instantaneous complex coefficients;

2. We interpret the presence of the wave by examining the lower frequency and its

[2]The problem may be defined as finding $\mathcal{J}\{x(t), \psi\}$ such that it satisfies:

$$x_A(t) = \int_{0^+}^{\infty} \mathcal{J}\{x(t), \psi\}(\omega) \cdot \psi_A(\omega t) \mathrm{d}\omega.$$

By applying a Fourier transform on both sides of the equation, it can be shown that the following recursive definition, which represents the solution to the integral, does not converge [108]:

$$\mathcal{J}\{x(t), \psi\}(\omega) = \frac{1}{\Psi_1} \left(\frac{1}{2} \mathcal{F}\{x_A(t)\}(\omega) - \sum_{k=2}^{\infty} \Psi_k \cdot \mathcal{J}\{x(t), \psi\}(\frac{\omega}{k}) \right).$$

Algorithm 4: Generalized discrete IHA algorithm

Input: $x[n]$: a stream of real numbers, representing the input signal
k_0: starting frequency index,
$k_{\max}$: ending frequency index,
(Ψ_i): kernel function represented by a sequence of complex coefficients,
M: number of kernel coefficients

Output: $\mathrm{J}[k,n]$: a stream of complex numbers, representing the generalized discrete IHA,
$r[n]$: a stream of real numbers, representing the residual signal

quantize the frequency axis and store the normalized frequency values in $\upsilon[k]$
set $r[n] \leftarrow x[n]$ `/* residual signal */`
for $k \in [k_0, k_{\max}]$ **do**
  `// number of overtones`
  set $M' \leftarrow \min(\underset{i}{argmax}(i|\forall i \in [1,M], i\upsilon[k] < \upsilon[k_{\max}]), M)$
  `// overtone indices`
  set $q_i \leftarrow \underset{n}{\arg}(\upsilon[n], i\upsilon[k]), \forall i \in [1, M']$ `/*` $\underset{n}{\arg}(x[n],\alpha) := \{n|\forall n : x[n] = \alpha\}$ `*/`
  foreach q_i **do**
    calculate $\mathrm{H}[q_i, n]$, using the IHA transform of $r[n]$
  estimate $a[k,n] = \min\left(\left|\frac{\mathrm{H}[q_i,n]}{\Psi_i}\right|\right), \forall q_i$
  estimate $\phi[k,n] = \angle \mathrm{H}[k,n] - \angle\Psi_1$
  set $\mathrm{J}[k,n] \leftarrow a[n] \cdot \mathrm{e}^{\mathbf{j}\phi[k,n]}$
  output $\mathrm{J}[k,n]$
  set $r[n] \leftarrow r[n] - \sum_{i=1}^{M'} a[k,n] \cdot \Phi_i \cdot \mathrm{e}^{\mathbf{j}i\phi[k,n]}$
output $r[n]$

overtones, by matching them against the kernel coefficients; and

3. Since, in practice, the input signal is band-limited, a limited number of coefficients is required for overtone estimation.

By taking the above into consideration, we proposed using a bottom-up overtone elimination approach for estimating the instantaneous complex coefficients. Algorithm 4 presents our generalized IHA algorithm, which implements our MFFE process.

Our algorithm delivers a decomposition algorithm using a composite kernel. Shahnaz et al. provided a pitch estimation based on harmonic sinusoidal autocorrelation model sim-

ilar to our decomposition scheme in 2.2 [101]. Our approach delivers multi-pitch estimation using composite kernels, which resembles the mother wave in wavelet transform. Also, Ding et al. provided a pitch estimation using Harmonic Product Spectrum (HPS) [33]. The authors used HPS as the product of spectral frames for constant number of overtones. Chen and Liu utilized a modified HPS for multi-pith estimation [26]. Our method uses the kernel coefficients Ψ_k's in calculating the overtones.

6.6 Summary

We presented a signal decomposition algorithm using composite kernels. We used the harmonic structure delivered by the IHA core and transformed them into multi-pitch information. A bottom-up overtone reconstruction and elimination process was carried out for the MFFE process. This forms the core of our audio-to-MIDI system, which is specified in the chapter 8.

Chapter 7

Performance Analysis

This chapter presents the performance analysis of the presented IHA algorithm. A new relation between CQT and WSFs is provided, in here. It is shown that CQT can alternatively be implemented by applying a series of logarithmically scaled WSFs while its window function is adjusted, accordingly. Both approaches yet provide a short-time cross-correlation measure between the input signal and the corresponding pure sinusoidal kernels whose frequencies are equal to the center of the filter band. It is shown that the IHA phasor construct significantly improves the instantaneous amplitudes estimation.

WSFs have been extensively used in the literature [103]. Both CQT and WSFs provide a decomposition scheme based on a set of reference frequencies (cf. invertible CQT [56] and the WSF decomposition, presented in section 4.5). In this chapter, we present a new relation between CQT and WSFs. The derivation of the relation as well as the performance analysis of our IHA algorithm are given in the following sections.

7.1 The Relation between CQT and WSF

We derive the relation between CQT and WSF by deriving a sliding version of CQT, by using a delay operation [49]. The details are given in the following.

7.1.1 Deriving the Relation

Recall that the formal definition of CQT is given in a form of STFT, as presented in (2.1). The sliding version of CQT may be formalized by using the following two assumptions:

1. x is zero padded;
2. and the STFT window is centered at n.

Thus:

$$X[k,n] = \frac{1}{N[k]} \sum_{m=0}^{N[k]-1} W[k,m] \cdot \mathrm{e}^{\frac{-\mathbf{j}2\pi Q m}{N[k]}} \cdot x[m+n-\frac{N[k]-1}{2}].$$

It must be noted that, by definition, $N[k]$ is an odd number. Therefore, by substituting $N[k]$ with $2M_K+1$, shifting m by $-M_k$, and also substituting

$$\frac{2Q}{2M_k+1}$$

with υ_k using (4.6), we will have:

$$X[k,n] = \frac{(-1)^Q}{2M_k+1} \sum_{m=-M_k}^{M_k} \mathrm{e}^{-\mathbf{j}\pi\upsilon_k m} \cdot x[m+n] \cdot W[k,m+M_k].$$

Without losing generality, we can rewrite the window $W[k, m + M_k]$ as $W[k, m]$, where m is shifted by M_k. Therefore,

$$X[k, n] = \xi \left(x, \mathrm{e}^{\mathrm{j}\pi\upsilon_k m}, \frac{(-1)^Q}{2M_k + 1} \cdot W_k[m] \right), \tag{7.1}$$

where ξ denotes the windowed cross-correlation, as defined in the following:

$$\xi(x, \tau, W) = \sum_{m=-M}^{M} x[n - m] \cdot \tau[m] \cdot W[m], \tag{7.2}$$

and W is a symmetric window with $2M + 1$ samples in length.

Corollary 1. *Eq.* (7.1) *suggests that CQT is equivalent to the cross correlation of the input signal with the sinusoidal kernel* $\cos(\pi\upsilon_k m)$ *in its analytical form within the following window:*

$$\frac{(-1)^Q}{2M_k + 1} \cdot W_k[m].$$

Similarly, (4.10) can be derived by means of the above cross-correlation. By choosing:

$$M_k = \frac{2}{\lambda\upsilon_k},$$

we may derive $C[k, n]$ as:

$$C[k, n] = \xi \left(x, \cos(\pi\upsilon_k m), \frac{2}{M_k} \mathrm{sinc} \left(\frac{m}{M_k} \right) \cdot W_k[m] \right). \tag{7.3}$$

Thereby:

Corollary 2. *We interpret* (7.3) *as the cross-correlation of the input signal with the sinusoidal kernel* $\cos(\pi\upsilon_k m)$ *within the window:*

$$\frac{2}{M_k} \mathrm{sinc} \left(\frac{m}{M_k} \right) \cdot W_k[m].$$

Hence, by comparing (7.1) and (7.3), it can be deduced that:

Corollary 3. *CQT is the equivalent of performing a series of WSF whose bandwidths correspond to* (2.6)*, where*

$$\lambda = \frac{2}{Q},$$

and the window function is adjusted by the following:

$$W_{\text{sinc}}[m] = (-1)^Q \cdot \left(\frac{4M_k + 2}{M_k}\right) \cdot \text{sinc}\left(\frac{m}{M_k}\right) \cdot W_{\text{CQT}}[m].$$

By assuming that Q is generally an even number, and M_k is also considerably large, the above adjustment may be simplified as:

$$W_{\text{sinc}}[m] = 4\text{sinc}\left(\frac{m}{M_k}\right) \cdot W_{\text{CQT}}[m]. \tag{7.4}$$

Corollary 4. *CQT and WSF can interchangeably be used as both provide a cross-correlation measure of the function with a sinusoidal kernel.*

7.1.2 Interpretation

The decomposition scheme presented in section 2.2 makes it possible to perform an IHA of the input signal. This was achieved by estimating the complex values $\text{H}[k, n]$'s which designate the phasor representation of the instantaneous amplitude and phase lag of the signal's constituent components. Both CQT and our presented WSF provide instantaneous complex values, which can be used in such estimation.

One interesting property of CQT's complex kernel $\text{e}^{-\mathbf{j}\pi\upsilon_k m}$ is that the magnitude of the resulting transformation can be interpreted as a representation for instantaneous

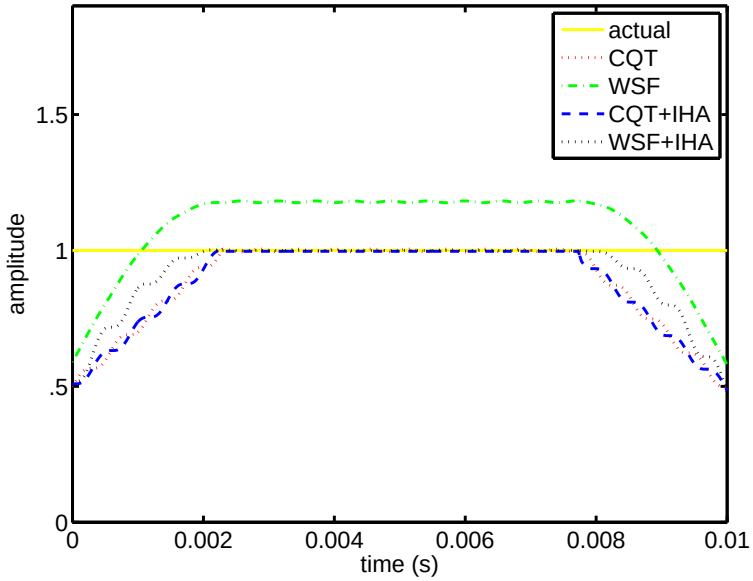
amplitudes of the corresponding components. At first glance, it may seem that the WSF approach lacks such a property. However, to resolve this, one may simply substitute the $\cos(\pi \upsilon_k m)$ kernel with its analytical form:

$$\tau[m] = \mathrm{e}^{-\mathbf{j}\pi\upsilon_k m},$$

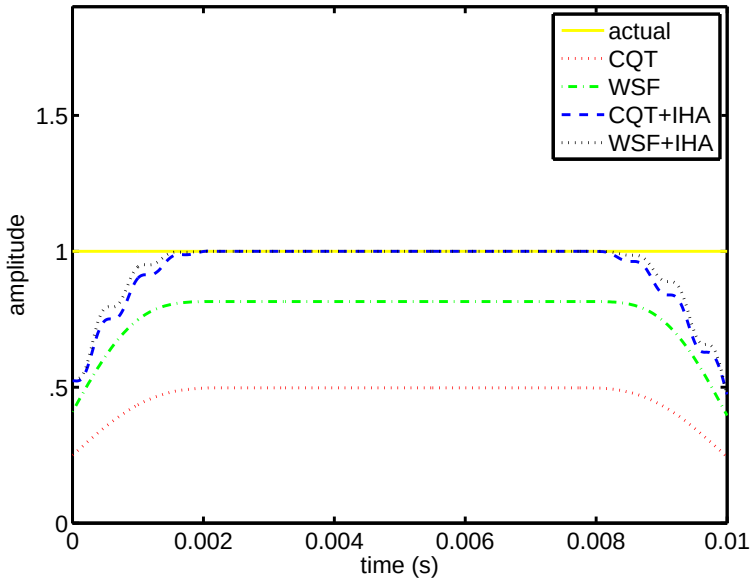
without changing the relation. Such WSF will be equivalent to performing the filters on the analytical form of the input signal using Hilbert transform. Since we demonstrated the sliding CQT could be derived by means of WSF, an invertible CQT is achievable, and thereby, a reconstruction algorithm may be used, cf. (2.10).

Fig. 7.1 presents a sample output of the IHA algorithms using flat and Blackman windows in 7.1(a) and 7.1(b), respectively. A single-component piece-wise sinusoidal with unit amplitude and frequency of 880 Hz has been used. The sampling frequency were also 44 kHz. In order to condense the temporal window as much as possible, $\gamma = 2$ was used. Table 7.1 summarizes the overall instantaneous amplitude estimation rate. The overall rate has been estimated by averaging the absolute values of distances between estimated and actual amplitudes. The ratio of the signal length over latency, as well as the latency itself were 2.20 and 2.30×10^{-3}, respectively. In this particular example, IHA delivered a stabilized estimate even though the underlying filtering algorithm provided unstable oscillating response, cf. WSF-IHA flat vs. WSF Blackman.

In Fig. 7.2, a sample output for a two-component input signal using a semi-tone bandwidth and Blackman Window has been demonstrated. The details are given in Table 7.2, correspondingly. In both examples, it is shown that using the combination of WSF and IHA not only maximizes the estimation rate, but also minimizes the overall reconstruction error.

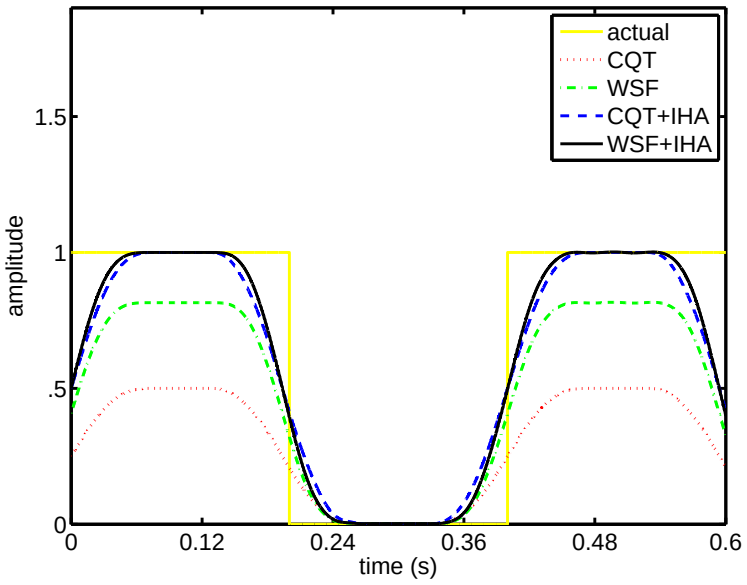


(a) flat window

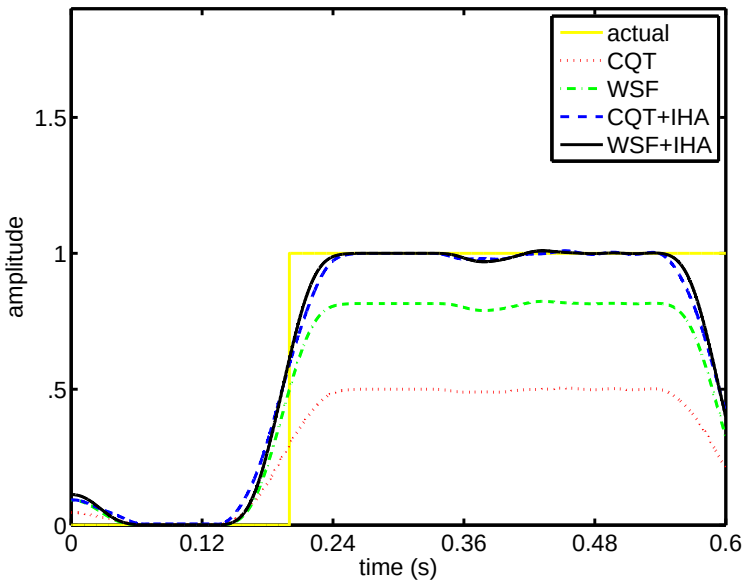


(b) Blackman window

Figure 7.1: Sample output of CQT vs. WSF and IHA



(a) first component



(b) second component

Figure 7.2: Sample output of two-component signal using a semitone bandwidth

Method	Overall Estimation Rate			
	flat-window		*Blackman-window*	
	Amp.	*Rec.*	*Amp.*	*Rec.*
CQT	88.20	87.24	46.28	65.54
WSF	83.12	86.00	76.73	83.52
CQT+IHA*	88.26	87.24	46.30	65.54
WSF+IHA*	83.28	86.00	76.78	83.52
CQT+IHA	88.11	89.68	93.06	95.68
WSF+IHA	**92.03**	**95.01**	**94.18**	**96.39**

Amp. amplitude response

Rec. signal reconstruction

* not using eq. (4.11)

Table 7.1: Overall estimation rate for instantaneous amplitude and full signal reconstruction of a sample signal using CQT, WSF and IHA.

Method	Overall Estimation Rate			
	Amp.			*Rec.*
	C_1	C_2	$C_1 + C_2$	
CQT	60.75	63.36	55.20	53.78
WSF	79.55	83.07	81.31	82.46
CQT+IHA	88.18	93.31	87.41	92.03
WSF+IHA	**90.02**	**94.25**	**89.26**	**93.45**

Table 7.2: Overall estimation rate for instantaneous amplitude and full signal reconstruction of a two-component signal

Although CQT was originally defined by means of STFT [16], a number of versions of it have been used in the literature. For instance, Graziosi et al. utilized a sliding filter bank approach to calculate the CQT [49] while Holighaus et al. [56] proposed using a sliced version. In their approach the input signal is uniformly sliced and consequently converted into a set of atom coefficients. Since those coefficients can form a matrix, they proposed obtaining the reconstructed signal using a pseudo-inverse approach.

Such matrix representation seems to be highly efficient for signal registration and compression. Nagathil and Martin,for instance, provided an optimal signal reconstruction

from constant-Q spectrum [84]. However, depending on the application, other representation schemes may be also used. For example, in melody extraction, one major concern is the detection of on-set/off-set events, which can be achieved by processing an instantaneous amplitude response. In such a model, whether a linear centric or a logarithmic centric approach is used, the sliding transformation is always invertible (cf. (2.6) and (2.7)). The window function, however, produces an small noise, cf. [103].

7.2 Relation to Wavelets

An interesting property of logarithmic quantization is that the resulting filter outputs inherit wavelet properties. For instance, in case of octave bandwidth ($\lambda = 2$), the filter output simulates a quasi-Shannon wavelet:

$$C_k(t) = \frac{1}{\pi}\sqrt{\frac{\omega_k}{\pi\sqrt{2}}} \cdot WT_\psi(Sha)\{x\}\left(\frac{\pi\sqrt{2}}{\omega_k}, t\right). \tag{7.5}$$

In the above equation, $WT_\psi(Sha)\{x\}$ represents the continuous wavelet transform of x using Shannon wavelet [75] with

$$\frac{\pi\sqrt{2}}{\omega_k}, \quad t,$$

as the scale and translational values, respectively. Choosing a lower value for λ results in a higher Q factor and therefore provides rational dilation wavelets [9]. Although the wavelet output is rescaled by a factor of $\sqrt{\omega_k}$, both approaches provide full reconstruction algorithms.

Although ω_k's are logarithmically scaled, (2.6) shows that they are yet linearly centered within the bands. This poses certain restrictions in practice, especially in music analysis, where the border frequencies are also desired to be logarithmically scaled, i.e.

(2.7). The border frequencies are referred to the upper-bound and the lower-bound of B_k.

CQT suggests that Q is integer. However, in order to have side-by-side bandwidths, a real-valued Q must be used in general. Moreover, a lower bound for Q may also be derived as $Q > 1$, due to the fact that in case $Q = 1$, the quantization of the frequency axis by non-overlapping frequency bands is not possible, as the minimal frequencies of all B_k's approach zero.

Fig. 7.3 demonstrates the distance between the center frequency and the upper and lower border frequencies in form of musical intervals. As illustrated, for lower values of Q, the difference between the minimal and maximal frequencies is considerably large (i.e. octave vs. perfect fifth, at $Q = 2$). However, as Q approaches 34, both values merge into a semitone, which makes the filter banks suitable for a 12-equal temperament system implementation [7]. Therefore, for $Q > 34$, the two linear-and logarithmic-centric approaches merge (cf. (2.7), (2.6)).

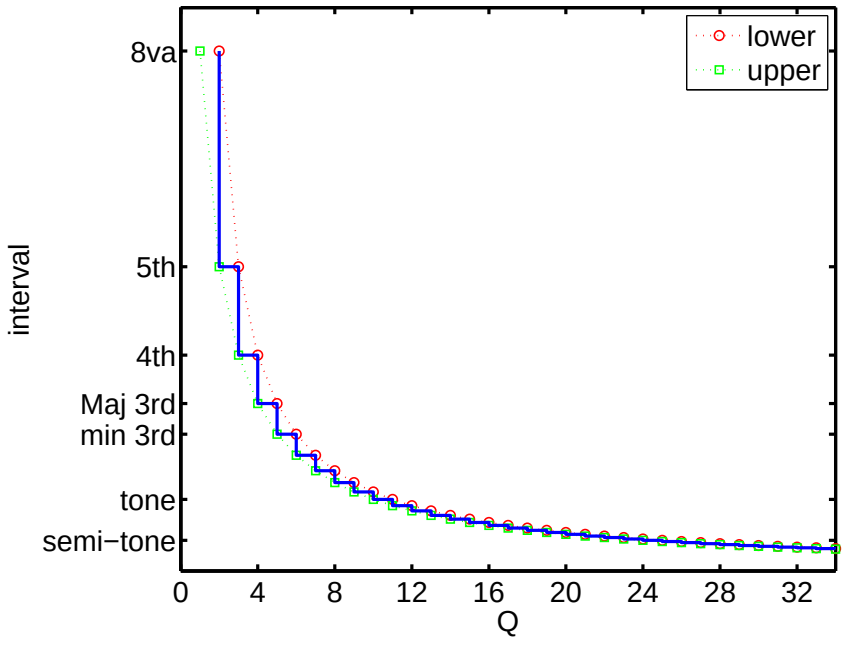


Figure 7.3: Lower-limit vs. upper-limit intervals of various Q's

7.3 Simulations

The performance analysis of the CQT in (7.1) against WSF with and without using window functions is provided here. A Blackman window is used in this comparison. We apply the IHA phasor construct on the real-part of the CQT output as well as the WSF approach. The details are given in the following.

In order to evaluate the relation in (7.4), we examined the performance of the CQT in (7.1) against WSF with and without using window functions. A Blackman window was used in the experiment. We also applied our phasor construct on the real-part of the CQT output as well as the WSF approach for the instantaneous harmonic analysis. The details are given in the following.

In the simulations, we used two sets of synthetic data as well as one set of mid-range real-audio. We applied the algorithms on the frequency range between 110 and $1,760$Hz with semitone steps, resulting in full four octaves. Both CQT and WST using Blackman window along with IHA were applied to the data sets.

We calculated the overall amplitude estimation success rate for the two synthetic data sets with respect to the actual amplitudes. The threshold value, as explained in section 4.5.4, was set to one-tenth of the maximum amplitude and was applied to all four methods. The following formula was used for calculating the success rate:

$$rate = 100 \times \left(1 - \tfrac{1}{N} \sum_{k,n} \tfrac{|a[k,n]-\tilde{a}[k,n]|}{a[k,n]}\right), \quad a[k,n] \neq 0,$$

where a and $\tilde{a}$ denote the actual and estimated amplitudes, respectively. The summary of the results is given in table 7.3.

Method	Overall Estimation Rate	
	DS1	DS2
CQT	43.51	31.87
WSF	70.35	51.26
CQT+IHA	79.87	55.69
WSF+IHA	**82.28**	**57.75**

Table 7.3: Overall estimation rate on synthetic data

In the case of real audio data, since the actual amplitudes were unknown, we calculated the reconstruction rate, using a weighted averaging method as given in the following:

$$rate = 100 \times \left(1 - \frac{\sum_n |x[n] - \tilde{x}[n]| \cdot |x_A[n]|}{\max(|x_A[n]|) \cdot \sum_n |x_A[n]|} \right),$$

where x, $\tilde{x}$, and x_A denote the original, estimated, and analytical signals, respectively. In our experiment we modified the signals by applying a low pass filter in order to eliminate the frequencies above $1,760$Hz. The results are shown in table 7.4.

Method	Overall Estimation Rate
	DS3*
CQT	86.47
WSF	89.72
CQT+IHA	90.03
WSF+IHA	**90.27**

* modified

Table 7.4: Overall reconstruction rate on real audio signals

Tables 7.3 and 7.4 demonstrate that WSF, in general, outperformed CQT in both amplitude estimation as well as signal reconstruction. The combination of IHA and WSF significantly improves the amplitude estimation and reduces the overall reconstruction error.

7.4 Summary

A new relation between CQT and WSF in the presented continuous model was provided. The invertibility of our IHA was discussed. We demonstrated that the performance of CQT can be enhanced by adjusting its window function, as specified in (7.4). Our simulation results supported the theory. In the following chapters we examine the applicability of the IHA algorithm for improving the MFFE and note on-set/off-set detection processes.

Chapter 8

The Transcription System

This chapter contributes to the post-processing algorithms, used in the proposed transcription system. The output of the MFFE process using the generalized IHA algorithm is translated into note on-set/off-set events. The specification of our proposed audio-to -MIDI system as well as the note event representation, used in the simulation, are also discussed here.

The music transcription in this thesis is defined as a process of analyzing audio signals in order to detect on-set/off-set events of the notes, in form of an audio-to-MIDI practice. Percussion analysis or processing of musical accidentals is not included (cf. [71, 10]). Our proposed transcription system is built on top of the generalized IHA transform, which was presented in the previous chapters. It is based upon the note events modeling, which is specified in the following.

In the following sections, an overview of the note events modeling is given; an equivalent matrix representation is provided; the proposed transcription system is specified; the

MFFE process is reviewed; and the post-processing algorithms for extracting the note events are provided. The simulation results are provided.

8.1 An Overview of Note Events Modeling

The music notations deliver three primary pieces of information: beat information, note events, and instrumentation:

1. the beat information specifies how the note events are translated in time;
2. the note events provide the information about note on-sets and off-sets;
3. and the instrumentation represents the information about the tone color, also known as timbre.

Given the beat information, a piece of music may be encoded using a set of tuples, as specified in the following. Coding and synthesis of articulations are out of the scope of our discussion.

One important characteristic of the MIDI format is that the MIDI data is represented in time rather than beats. Therefore, we exclude the beat information from the note event representation. Hence, we use a set of 5-tuples $\langle i, n, v, t_{\text{on}}, t_{\text{off}} \rangle$ where i represents the instrument index, n represents the note index, v represents the loudness, t_{on} represents the on-set, and t_{off} represents the off-set. The note index is associated with a fundamental frequency. The loudness factor in the MIDI context is also referred to as the velocity [81]. In some context, t_{off} is substituted with note duration. Translating such representation into MIDI format is a straightforward process. Each tuple is translated into two individual

MIDI events: one for on-set and one for off-set [81]. Our audio-to-MIDI system is based upon this representation.

8.2 Matrix Representation

For the MFFE process, we used a quantized time model, as follows. In our model, the time axis is quantized into a set of time intervals, each of which covers a beat resolution. As a result, each individual instrument channel is encoded using a melody matrix where each row represents a tone whereas each column represents a discrete time. A non-zero element indicates an on-set whereas the consequent zero element within the same row indicates the corresponding off-set.

Figure 8.1 illustrates a sample melody matrix in crotchet [46] beat resolution using the **A440** just-intonation system [7]. As illustrated, the melody is represented by matrix M, with respect to the reference frequency matrix F, corresponding to the notes: **A**, **C#**, **E**, **G**, and **A+8va**, and **A-8va**. Each row in matrix M corresponds to a particular note and each column represents a crotchet beat. For instance, row 1 corresponds to the note **A** (i.e. the leftmost note, fourth note, etc. on the staff lines), row 2 corresponds to **C#**, etc. For simplicity, only five reference frequencies have been shown whereas in the simulation the full range (an 88-row) matrix is used.

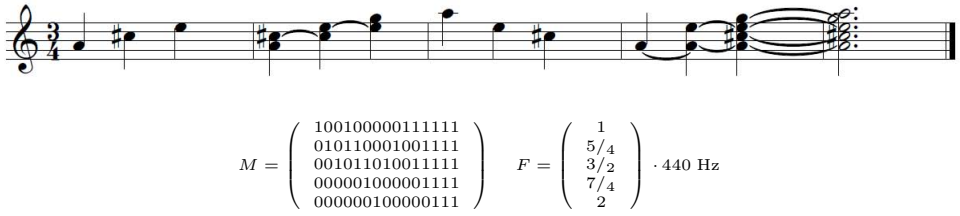


Figure 8.1: A sample melody matrix in crotchet beat resolution

In general, the beat resolution for coding is set to a $1/_2$ of the beat resolution used in the transcription. Beat resolution is normally chosen depending on the desired transcription assurance.

8.3 The Proposed Transcription System

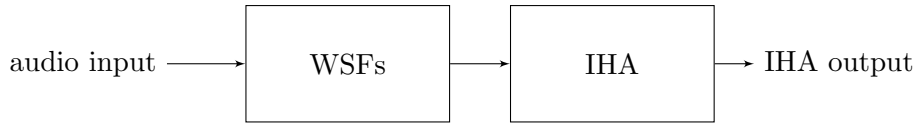
Figure 8.2 illustrates the block diagram of our proposed transcription system. To extract the note events from the audio input, we apply the generalized IHA algorithm on the input signal. The instantaneous amplitude for each frequency index is interpreted as the presence of the tone in time. Therefore, a post-processing algorithm is performed to translate the instantaneous amplitudes into note events.

As illustrated, the transcription system is implemented in two phases. Phase I produces the raw output by applying the IHA transform on the input signal. The output of the IHA block consists of the harmonic sinusoidals and their overtones. The raw output is then fed into Phase II for the MFFE process. By using the timbral data captured by an offline process, the generalized IHA transforms the raw output into a set of sequences of complex coefficients, representing the notes.

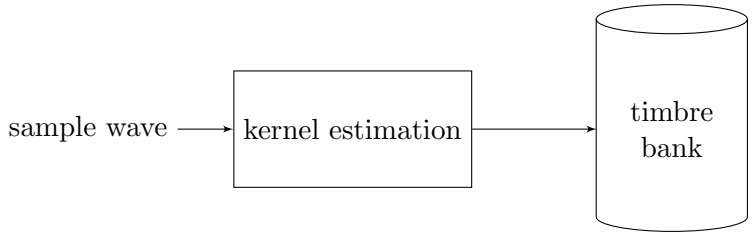
Phase I is implemented by applying Algorithms 1 and 2, as presented in chapter 4. The Offline Phase is implemented by using Algorithm 3, as given in the previous chapter. The generalized IHA block in phase II delivers the MFFE using Algorithm 4. In our model, each instrument is represented by a sequence of kernel coefficients.

In order to translate the complex coefficients into note events a down-sampler is used. We use a gradient operation to detect the on-set/off-set events. To increase the efficiency

Phase I: IHA Transform



Offline Phase: Instrument Processing



Phase II: Timbral Analysis

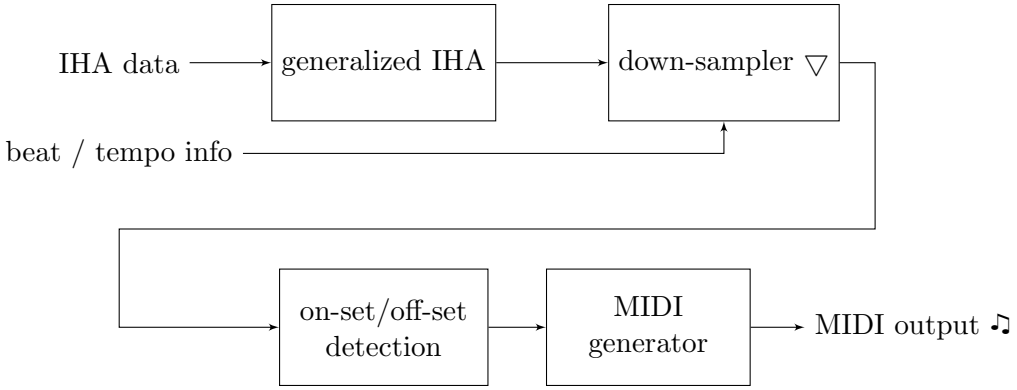


Figure 8.2: Block diagram of the proposed transcription system

of the note events detection, we apply the gradient operation in a lower resolution. Using this, the time complexity of the process is also reduced. The time resolution is reduced into beat quantum. The note events are then translated and consequently represented in MIDI format. The details are given in the following sections.

8.4 The MFFE Process

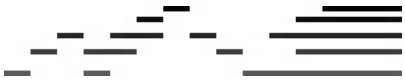
We perform the MFFE analysis by applying Algorithm 4 on the input signal using $(\Psi_k)_i$, representing the i^{th} instrument from the timbre bank, where k denotes the coefficient index. Klapuri [70] proposed two different iterative spectral subtraction systems for MFFE. He took advantage of prime number harmonics to detect the fundamental frequencies. In IHA, we use the harmonic relations by sweeping the frequency bins from lower frequencies to the top. For a fundamental frequency to exist, the IHA amplitude for the fundamental frequency and its overtones must co-exist.

Figures 8.3(a), 8.3(b) illustrate the output of the algorithm for the sample melody in Figure 8.1 using zero and two overtones, respectively. Figure 8.3(b) demonstrates the harmonic collusion caused by non-prime overtones. The output of the Algorithm 4 is provided in Figure 8.3(c) where the effect of the harmonic collisions on the estimated fundamental frequencies is highlighted.

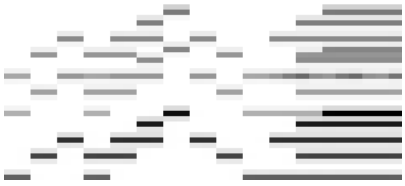
Figure 8.3(a) was generated from the output of the IHA for the twelve semitones in the octave between **A** and **A+8va**. Figure 8.3(b) demonstrates the output of the IHA for three octaves, covering two overtones, and figure 8.3(c) demonstrates the output of the Algorithm 4 for the first octave. The output of the algorithm outside the range was zero and not included in the image.

8.5 Extracting the Note Events

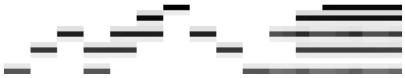
Extracting the note events is performed by scanning the output of the generalized IHA for note on-set/off-set events. Before processing the IHA output, one may perform an inverse-



(a) no overtone



(b) two overtones



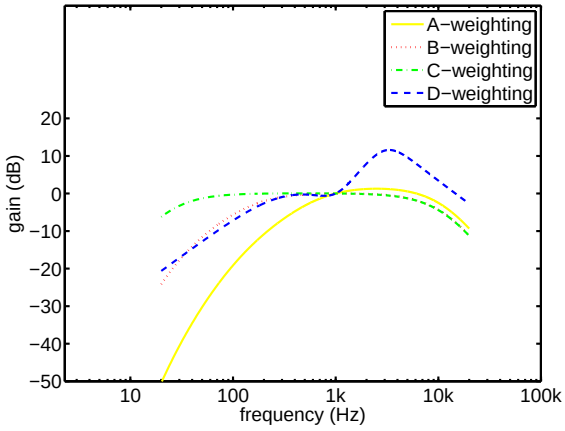
(c) output the generalized IHA

Figure 8.3: The output the IHA algorithm on the sample melody in Fig. 8.1

weighting operation to equalize the frequency outputs. Frequency weighting is an operation for measurement of the sound loudness [3]. Figure 8.4 illustrates various weighting curves relative to 1 kHz, as defined by ANSI [3].

To identify the note events, we perform a down-sampling operation on the output of the generalized IHA algorithm, to transform the output into beat quantum. The down sampling is applied on the magnitude of the generalized IHA. A threshold / mapping technique is then applied to transform the real data into a 128-level gray-scale, depending on the type of desired MIDI output[1]. A gradient operation is consequently applied on the gray -scale data, for detecting the on-sets and off-sets. The details are given in Algorithm 5.

[1]In most transcription systems a binary-level MIDI velocity is used.



$$A(f) = \frac{12200^2 f^4}{(f^2+20.6^2)\sqrt{(f^2+107.7^2)(f^2+737.9^2)}(f^2+12200^2)}$$

$$B(f) = \frac{12200^2 f^3}{(f^2+20.6^2)\sqrt{f^2+158.5^2}(f^2+12200^2)}$$

$$C(f) = \frac{12200^2 f^2}{(f^2+20.6^2)(f^2+12200^2)}$$

$$D(f) = \frac{f}{6.896688849647610^{-5}}\sqrt{\frac{h(f)}{(f^2+79919.29)(f^2+1345600)}}$$

where

$$h(f) = \frac{(1037918.48-f^2)^2+1080768.16f^2}{(9837328-f^2)^2+11723776f^2}$$

Figure 8.4: Different weighting curves relative to 1 kHz

8.6 Simulations

The performance of the generalized IHA algorithm has also been examined by applying the algorithm on three sets of synthesized signals, as explained in the following. Using the tempo information and given a desired beat resolution, a melody matrix was generated for each input MIDI where each row designated the presence of a particular note in time and each column represented a unit time interval in beat resolution, as specified in 8.2.

The synthesized wave files were then generated by using the input matrices and a set of pure sinusoidals corresponding to each tone and its overtones. In our experiment three versions were generated by applying pure-harmonic, one-overtone, and two-overtone timbre banks. The accuracy of the note detection process has been determined by applying the same coding process on the output signals. The accuracy rate was calculated using the F-

measure:

$$a = \frac{2TP}{2TP + HD} \times 100,$$

where TP represents the total number of matching 1s between the input and output matrices and HD represents the Hamming distance between the two.

Five different MIDI data sets have been used in our experiment, as in the following. DS1: treble-range, DS2: easy full-range, DS3: four-part choral, DS4: full-range piano, and DS5: full-range multi instrument. The details of the datasets are provided in Appendix B. The MIDI parts were extracted and synthesized using three timbre banks. The data items were created by splitting the original files into five-measure segments. The accidentals were not used during the synthesizing process. The algorithm was applied to individual segments and the overall accuracy was calculated for each data set. In our experiment a maximum of two overtones was used. The result is listed in table 8.1. Notes below **C3** were not used in the simulation.

Accuracy	*AP*	0-*overtone*	1-*overtone*	2-*overtone*
DS1	1.9	99.5	98.3	97.2
DS2	2.7	99.8	98.8	98.3
DS3	4.0	99.4	97.8	97.8
DS4	2.6	91.1	88.0	78.6
DS5	3.2	81.3	76.8	72.5

AP Average Polyphony

Table 8.1: Overall AMT Simulation Results

In the above table, the melody matrix was used as the ground truth. Our accuracy measurement supports the Music Information Retrieval Evaluation eXchange (MIREX) specification [83]. MIREX specifications define the accuracy of note event detection as the following:

1. The onset is detected within a ±50 ms range of a ground-truth onset

2. and the detected F_0 is within a quarter tone of the ground-truth pitch

8.7 Summary

The applicability of the generalized IHA algorithm in MFFE and note event modeling was demonstrated. An audio-to-MIDI transcription system based on generalized IHA was presented.

While the transcription output seems satisfactory for mid-range and slow pieces (DS1 -3), our simulation shows that the accuracy has been decreased by 25% in the case of using full-range data. Such errors occur mainly for the following reasons: the response latency associated with the low frequencies, the difference between the responses from the fundamental tone and the overtones, and the harmonic collision. The existence of inharmonicity in real audio signals, where the frequencies of the overtones do not follow the integer multiples of the fundamental frequency, makes the transcription process rather difficult.

Algorithm 5: Algorithm for generating note events

Input: i: instrument index,
τ: tempo *(in crotchets per minute)*,
u: beat resolution *(in whole note unit)*,
T: sampling frequency,
$\mathrm{J}[k, n]$: a stream of complex numbers, representing the generalized discrete IHA
k_0: starting frequency index,
$k_{\max}$: ending frequency index,
ϵ: amplitude threshold

Output: e: collection of $\langle b, k, l\rangle$ representing the note events where $b \in \mathbb{N}$, $k \in \mathbb{Z}$, and $l \in \mathbb{N}$ represent the beat index, the note index, and the velocity level. *($l = 0$ denotes off-set)*

construct the melody matrix M by performing a down-resolution on every row of J using the ratio r, where $\mathrm{J}[k, n] \geq \epsilon$ where

$$r = \left(\frac{1}{T}\right) : \left(\frac{\tau}{240u}\right)$$

set $e \leftarrow \{\}$
$M \leftarrow M : m_{i,j} \geq \epsilon$ `/* eliminate the noisy output */`
$O \leftarrow \mathrm{sgn}(m_{i,j} - m_{i,j-1})$ where sgn represent the signum function `/* on-set */`
$F \leftarrow \mathrm{sgn}(m_{i,j+1} - m_{i,j})$ `/* off-set, in binary level */`
for $k \in [k_0, k_{\max}]$ **do**
    for $b \in [0, b_{\max}]$ **do**
        if $O_{k,n} \geq 0$ **then**
            add $\langle b, k, O_{k,n}\rangle$ to e
        if $F_{k,n} \leq 0$ **then**
            add $\langle b, k, 0\rangle$ to e
output $e[n]$

Chapter 9

Conclusions and Future Directions

Techniques and challenges regarding both time-frequency analysis and automatic music transcription were reviewed. A novel approach for estimating the instantaneous amplitude and phase elements of real-valued signals based on enhanced Constant-Q filtering and a unique phasor construct was presented. It was demonstrated that the proposed algorithm delivers a stable estimation by maintaining the instantaneous amplitudes and the phase lags locally constant. The magnitude of such a construct provides a finer estimate for instantaneous amplitude, compared to the analytical kernel that is used in CQT.

A fast real-time $M+1$-delay IHA algorithm was implemented. A generalization of the IHA algorithm for multi-dimensional signal analysis was also formalized in the hyper-complex space. A quaternion representation was proposed to support multiple phase elements.

A theoretical analysis of the IHA algorithm was provided. A new relation between the CQT and WSFs based on the original signal model in the continuous form was presented. It was shown that CQT can alternatively be implemented by applying a series of logarithmically scaled WSFs while its window function is adjusted, accordingly. Both approaches yet provide a short-time cross-correlation measure between the input signal and the corresponding pure sinusoidal kernels whose frequencies are equal to the center of the filter band. It was shown that the IHA phasor construct significantly improves the instantaneous amplitude estimation.

A generalization of the IHA algorithm was provided by utilizing composite kernels for timbral analysis. The IHA algorithm contributed to an MFFE process where a post-processing overtone elimination algorithm was used. Due to the limited number of overtones in practice, a representation of the kernel function by a sequence of complex values was used.

An audio-to-MIDI system was designed based upon the generalized IHA algorithm. The proposed system was implemented using the generalized IHA algorithm as the MFFE block and a set of post-processing algorithms for detecting the note on-set/off-set events. By using the timbral data captured by an offline process, the generalized IHA provides a set of sequences of complex coefficients, designating the presence of notes in time.

The performance of the proposed audio-to-MIDI system was analyzed by applying the algorithms on five data sets: treble-range, full-range, four-part choral, full-range piano, and full-range multi instrument. The MIDI parts were extracted and synthesized using timbre banks. The algorithm was applied to each

individual segment and the overall accuracy was measured for each data set. The data sets were explicitly selected to examine the performance of our MFFE algorithm with regards to latency, amplitude weighting, and harmonic collision. Our simulation demonstrated compelling results, with regards to the MIREX specification and existing state-of-the-art AMT systems, as reported in the literature [83].

The latency factor, especially in the lower frequency range imposes a risk where a low beat resolution is desired. While it can be assumed that, in practice, no consecutive low-range tones coexist simultaneously, a lower quality factor resulting in overlapping bands (i.e. 200- or 400-cent width) may be used for very low range tones. A post-processing selection algorithm, such as maximum likelihood, may be used to classify best candidates among the neighbored bins [72].

One of the interesting extensions of the proposed algorithm is to utilize the algorithm in interactive music retrieval systems such as humming. The IHA algorithm may also be used in music registration and classification systems, with regards to the provided normalized temporal spectral analysis. Another interesting extension would be to evaluate the performance of the multi-dimentional IHA and its applicability to multi-dimensional signals.

The proposed algorithms can be used as a basis for improving existing transcription systems. The ability of the IHA algorithm to simultaneously decompose an audio input into its fundamental components and deliver their instantaneous amplitudes, makes it exceptionally beneficial in music processing applications. The IHA algorithm may be applied in various AMT sub-systems such as note

tracking, chord analysis, we well as instrument identification.

During recent years, many contributed to individual areas of AMT. Although there have been significant improvements in the state-of-the-art AMT techniques, the overall performance of the existing systems is yet not comparable to human experts [11]. AMT is essentially a complex problem and requires various techniques. It is still an open problem, especially in the field of polyphonic transcription, and requires experts from different disciplines. Although the number of components that are possibly required to implement a functional AMT system is beyond the scope of this thesis, we hope our fundamental contributions to the harmonic analysis motivate future research directions.

Bibliography

[1] Abdallah, S. A. and Plumbley, M. D. [2003], Probability as metadata: Event detection in music using ICA as a conditional density model, *in* '4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)', Nara, Japan, pp. 223–238.

[2] Abe, M. and Ando, S. [1995], Nonlinear time-frequency domain operators for decomposing sounds into loudness, pitch and timbre, *in* 'International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95', Vol. 2, pp. 1368–1371.

[3] American Institute of Physics, Acoustical Society of American Standards & Secretariat and American National Standards Institute [2001, Revised in 2011], *ANSI/ASA S1.42–2001 (R2011): American National Standard, Design Response of Weighting Networks for Acoustical Measurements*, Acoustical Society of America standards, American Institute of Physics, Melville, N.Y., USA.

[4] Argent, F., Nesi, P. and Pantaleo, G. [2011], 'Automatic transcription of polyphonic music based on the constant-Q bispectral analysis', *IEEE Transactions on Audio, Speech, and Language Processing* **19**(6), 1610–1630.

[5] Arroabarren, I., Rodet, X. and Carlosena, A. [2006], 'On the measurement of the instantaneous frequency and amplitude of partials in vocal vibrato', *IEEE Transactions on Audio, Speech, and Language Processing* **14**(4), 1413–1421.

[6] Balazs, P., Dörfler, M., Jaillet, F., Holighaus, N. and Velasco, G. A. [2011], 'Theory, implementation and applications of nonstationary gabor frames', *Computational and Applied Mathematics* **236**(6), 1481–1496.

[7] Barbour, J. M. [1951], *Tuning and Temperament: A Historical Survey*, Michigan State College Press, Michigan, USA.

[8] Bari, N. K. [1964], *A Treatise on Trigonometric Series [Vols. I & II]*, Pergamon Press.

[9] Bayram, I. and Selesnick, I. W. [2009], 'Frequency-domain design of overcomplete rational-dilation wavelet transforms', *IEEE Transactions on Signal Processing* **57**(8), 2957–2972.

[10] Benetos, E. [2012], Automatic Transcription of Polyphonic Music Exploiting Temporal Evolution, PhD thesis, School of Electronic Engineering and Computer Science, Queen Mary University of London.

[11] Benetos, E. and Dixon, S. [2013], 'Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model', *The Journal of the Acoustical Society of America* **133**(3), 1727–1741.

[12] Bertin, N., Badeau, R. and Richard, G. [2007], Blind signal decompositions for automatic transcription of polyphonic music: Nmf and k-svd on the benchmark, *in* 'IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)', Vol. 1, pp. I–65–I–68.

[13] Bertin, N., Badeau, R. and Vincent, E. [2010], 'Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription', *IEEE Transactions on Audio, Speech, and Language Processing* **18**(3), 538–549.

[14] Boashash, B. [1992*a*], 'Estimating and interpreting the instantaneous frequency of a signal, part i: Fundamentals', *Proceedings of the IEEE* **80**(4), 520–538.

[15] Boashash, B. [1992*b*], 'Estimating and interpreting the instantaneous frequency of a signal, part ii: Algorithms and applications', *Proceedings of the IEEE* **80**(4), 540–568.

[16] Brown, J. [1991], 'Calculation of a constant-q spectral transform', *The Journal of the Acoustical Society of America* **89**(1), 425–434.

[17] Bruno, I. and Nesi, P. [2005], 'Automatic music transcription supporting different instruments', *Journal of New Music Research* **34**, 139–149.

[18] Cancela, P., Rocamora, M. and López, E. [2009], An efficient multi-resolution spectral transform for music analysis, *in* 'Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)', Kobe, Japan, pp. 309–314.

[19] Carleson, L. [1966], 'On convergence and growth of partial sums of Fourier series', *Acta Mathematica* **116**(1), 135–157.

[20] Casazza, P. G. [2000], 'The art of frame theory', *Taiwanese Journal Mathematics* **4**, 129–202.

[21] Cemgil, A. T. [2004], Bayesian Music Transcription, PhD thesis, Radboud University of Nijmegen.

[22] Cemgil, A. T. and Kappen, B. [2003], 'Monte Carlo methods for tempo tracking and rhythm quantization', *Journal of Artificial Intelligence Research* **18**, 45–81.

[23] Chafe, C. and Jaffe, D. [1986], Source Separation and Note Identification in Polyphonic Music, *in* 'Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)', Tokyo, pp. 1289–1292.

[24] Chan, W. L., Choi, H. and Baraniuk, R. [2004*a*], Quaternion wavelets for image analysis and processing, *in* 'International Conference on Image Processing (IICIP'04)', Vol. 5, pp. 3057–3060.

[25] Chan, W. L., Choi, H. and Baraniuk, R. G. [2004*b*], Directional hypercomplex wavelets for multidimensional signal analysis and processing, *in* 'IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)', Vol. 3, pp. iii–996–999.

[26] Chen, X. and Liu, R. [2013], Multiple pitch estimation based on modified harmonic product spectrum, *in* 'Proceedings of the 2012 International Conference on Information Technology and Software Engineering', Vol. 211 of *Lecture Notes in Electrical Engineering*, Springer Berlin Heidelberg, pp. 271–279.

[27] Chuan, C.-H. and Chew, E. [2005], Polyphonic audio key finding using the spiral array CEG algorithm, *in* 'IEEE International Conference on Multimedia and Expo (ICME'05)', pp. 21–24.

[28] Cohen, L. [1994], The uncertainty principle in signal analysis, *in* 'Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis', pp. 182–185.

[29] Costantini, G., Todisco, M. and Saggio, G. [2011], A sensor interface based on sparse nmf for piano musical transcription, *in* '4th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI'11)', pp. 157–161.

[30] Cranitch, M., Cychowski, M. T. and FitzGerald, D. [2006], Towards an inverse constant-Q transform, *in* '120th Audio Engineering Society Convention', pp. 1–5.

[31] da C. B. Diniz, F., Biscainho, L. and Netto, S. [2007], Practical design of filter banks for automatic music transcription, *in* '5th International Symposium on Image and Signal Processing and Analysis (ISPA'07)', pp. 81–85.

[32] de Cheveigné, A. and Kawahara, H. [2002], 'YIN, a fundamental frequency estimator for speech and music', *The Journal of the Acoustical Society of America* **111**(4), 1917–1930.

[33] Ding, H., Qian, B., Li, Y. and Tang, Z. [2006], A method combining lpc-based cepstrum and harmonic product spectrum for pitch detection, *in* 'Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia', IIH-MSP '06, IEEE Computer Society, Washington, DC, USA, pp. 537–540.

[34] dos Santos, C. N., Netto, S. L., Biscainho, L. W. P. and Graziosi, D. B. [2004], A modified constant-Q transform for audio signals, *in* 'IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)', Vol. 2, pp. ii–469–472.

[35] Ehmann, A. F. [2010], High-Resolution Sinusoidal Analysis for Resolving Harmonic Collisions in Music Audio Signal Processing, PhD thesis, University of Illinois at Urbana-Champaign.

[36] Eldar, Y. and Michaeli, T. [2009], 'Beyond bandlimited sampling', *Signal Processing Magazine, IEEE* **26**(3), 48–68.

[37] Fan, Z., Sufen, D., Guifa, T. and Jie, Y. [2010], Improved humming music retrieval method based on wavelet transformation and dynamic time warping, *in* 'International Conference on Internet Technology and Applications', pp. 1–4.

[38] Farhat, H. [1990], *The Dastgah Concept in Persian Music*, Cambridge University Press.

[39] Feng, W. and Hu, B. [2008], Quaternion discrete cosine transform and its application in color template matching, *in* 'Congress on Image and Signal Processing (CISP '08)', Vol. 2, pp. 252–256.

[40] Fitch, J. and Shabana, W. [1999], A wavelet-based pitch detector for musical signals, *in* 'Proceedings of 2nd COST-G6 Workshop on Digital Audio Effects (DAFx99)', Norwegian University of Science and Technology, Trondheim, pp. 101–104.

[41] Folland, G. B. and Sitaram, A. [1997], 'The uncertainty principle: a mathematical survey', *Journal of Fourier Analysis and Applications* pp. 207–238.

[42] Fuentes, B., Liutkus, A., Badeau, R. and Richard, G. [2012], Probabilistic model for main melody extraction using constant-Q transform, *in* 'IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'12)', pp. 5357–5360.

[43] Gabor, D. [1946], 'Theory of communication. part 1: The analysis of information', *Journal of the Institution of Electrical Engineers - Radio and Communication Engineering* **93**(26), 429–441.

[44] Gang, R., Bocko, M. F., Headlam, D. and Lundberg, J. [2009], Polyphonic music transcription employing max-margin classification of spectrographic features, *in* 'IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'09)', pp. 57–60.

[45] Ganseman, J., Scheunders, P. and Dixon, S. [2012], Improving plca-based score-informed source separation with invertible constant-Q transforms, *in* 'Proceedings of the 20th European Signal Processing Conference (EUSIPCO'12)', pp. 2634–2638.

[46] Gehrkens, K. W. [1930 [2006]], *Music Notation and Terminology*, Project Gutenberg Literary Archive Foundation, Salt Lake City, USA.

[47] Gerhard, D. [1998], Automatic interval naming using relative pitch, *in* 'In Bridges: Mathematical Connections in Art, Music and Science', pp. 37–48.

[48] Grafakos, L. [2000], *Modern Fourier Analysis*, Graduate Texts in Mathematics, Springer, New York, USA.

[49] Graziosi, D. B., dos Santos, C. N., Netto, S. L. and Biscainho, L. W. P. [2004], A constant-Q spectral transformation with improved frequency response, *in* 'Proceedings of the 2004 International Symposium on Circuits and Systems (ISCAS '04)', Vol. 5, pp. V–544–V–547.

[50] Grindlay, G. and Ellis, D. P. W. [2009], Multi-voice polyphonic music transcription using eigeninstruments, *in* 'IEEE Workshop on Applications of Signal Processing to

Audio and Acoustics (WASPAA'09)', Mohonk Mountain House, New Paltz, NY, USA, pp. 53–56.

[51] Gröchenig, K. [2001], *Foundations of Time-Frequency Analysis: Applied and Numerical Harmonic Analysis*, Birkhauser, Boston.

[52] Hainsworth, S. and Macleod, M. [2003], 'The automated music transcription problem'. **URL:** *http://citeseer.ist.psu.edu/636235.html*

[53] Hamilton, Sir. W. R. [1866], *Elements of Quaternions*, Longman, London UK.

[54] Harte, C. A. and Sandler, M. [2005], Automatic Chord Identification using a Quantised Chromagram, *in* 'Proceeding of the 118th Convention of the Audio Engineering Society (AES)'.

[55] Higgins, J. R. [1996], *Sampling Theory in Fourier and Signal Analysis*, Vol. I. Foundations, Oxford University Press.

[56] Holighaus, N., Dörfler, M., Velasco, G. A. M. and Grill, T. [2013], 'A framework for invertible, real-time constant-Q transforms', *IEEE Transactions on Audio, Speech and Language Processing* **21**(4), 775–785.

[57] Hu, D. J. [2012], Probabilistic Topic Models for Automatic Harmonic Analysis of Music, PhD thesis, University of California, San Diego.

[58] Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C. and Liu, H. H. [1998], The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, *in* 'Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences', Vol. 454, pp. 903–995.

[59] Huang, N. E., Wu, Z., Long, S. R., Arnold, K. C., Chen, X. and Blank, K. [2009], 'On instantaneous frequency', *Advances in Adaptive Data Analysis* **1**(2), 177–229.

[60] Hunt, R. A. [1968], On the convergence of Fourier series, *in* 'Orthogonal Expansions and their Continuous Analogues (Proc. Conf., Edwardsville, Ill., 1967)', Southern Illinois Univ. Press, Carbondale, Ill., pp. 235–255.

[61] Ingle, A. N. and Sethares, W. A. [2012], 'The least-squares invertible constant-Q spectrogram and its application to phase vocoding.', *The Journal of the Acoustical Society of America* **132**(2), 894–903.

[62] Jannatpour, A., Krzyżak, A. and O'Shaughnessy, D. [2013*a*], A new approach to short-time harmonic analysis of tonal audio signals using harmonic sinusoidals, *in* '26th Annual IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'13)', pp. 1–6.

[63] Jannatpour, A., Krzyżak, A. and O'Shaughnessy, D. [2013*b*], 'On the interpretation of the constant-Q transform by windowed-sinc filters and cross-correlation of harmonic sinusoidals', *submitted to Journal of Acoustical Society of America* .

[64] Jarveläinen, H., Välimäki, V. and Karjalainen, M. [1999], 'Audibility of inharmonicity in string instrument sounds, and implications to digital sound systems', *Acoustics Research Letters Online (ARLO)* **2**, 79–84.

[65] Kadambe, S. and Boudreaux-Bartels, G. F. [1992], 'A comparison of the existence of 'cross terms' in the Wigner distribution and the squared magnitude of the wavelet transform and the short-time Fourier transform', *IEEE Transactions on Signal Processing* **40**(10), 2498–2517.

[66] Kates, J. N. [1979], Constant-Q analysis using the chirp Z-transform, *in* 'IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '79)', Vol. 4, pp. 314–317.

[67] Katznelson, Y. [2004], *An Introduction to Harmonic Analysis*, third edn, Cambridge University Press.

[68] Khan, N. A., Taj, I. A. and Jaffri, M. N. [2010], Instantaneous frequency estimation using fractional Fourier transform and Wigner distribution, *in* 'International Conference on Signal Acquisition and Processing, (ICSAP'10)', pp. 319–321.

[69] Kitahara, T., Goto, M. and Okuno, H. G. [2003], Musical instrument identification based on f0-dependent multivariate normal distribution, *in* 'Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)', Vol. 5, pp. V–421–424.

[70] Klapuri, A. [2004*a*], Signal Processing Methods for the Automatic Transcription of Music, PhD thesis, Tampere University of Technology.

[71] Klapuri, A. P. [2004*b*], 'Automatic music transcription as we know it today', *Journal of New Music Research* **33**(4), 269–282.

[72] Lee, K. and Slaney, M. [2008], 'Acoustic chord transcription and key extraction from audio using key-dependent hmms trained on synthesized audio', *IEEE Transactions on Audio, Speech, and Language Processing* **16**(2), 291–301.

[73] Leman, M. [1995], *Music and Schema Theory: Cognitive Foundations of Systematic Musicology*, Springer-Verlag, Berlin-Heidelberg.

[74] Li, X., Liu, R. and Li, M. [2009], A review on objective music structure analysis, *in* 'International Conference on Information and Multimedia Technology (ICIMT'09)', pp. 226–229.

[75] Mallat, S. G. [1999], *A Wavelet Tour of Signal Processing*, Academic Press, Burlington, MA, USA, chapter 7, pp. 210–211.

[76] Marks II, R. J. [1991], *Introduction to Shannon Sampling and Interpolation Theory*, Spinger-Verlag, New York.

[77] Marolt, M. [2001], Sonic: Transcription of polyphonic piano music with neural networks, *in* 'Audiovisual Institute, Pompeu Fabra University', Vol. 11, pp. 217–224.

[78] Marolt, M. [2004], 'A connectionist approach to automatic transcription of polyphonic piano music', *IEEE Transactions on Multimedia* **6**(3), 439–449.

[79] Martin, K. D. [1996], A blackboard system for automatic transcription of simple polyphonic music, Technical Report 385, M.I.T Media Laboratory Perceptual Computing Section.

[80] Martin, K. D. [1999], Sound-Source Recognition: A Theory and Computational Model, PhD thesis, Massachusetts Institute of Technology.

[81] MIDI Manufacturers Association Incorporated [2010], 'The Complete MIDI 1.0 Detailed Specification'. Accessed: 2013-08-05.
URL: *http://www.midi.org/techspecs/midispec.php*

[82] Monti, G. and Sandler, M. [2000], Monophonic transcription with autocorrelation, *in* 'Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFx-00)', Verona, Italy, pp. 111–116.

[83] *Music Information Retrieval Evaluation eXchange (MIREX)* [2013]. Accessed: 2013-08-05.
URL: *http://music-ir.org/mirexwiki/*

[84] Nagathil, A. and Martin, R. [2012], Optimal signal reconstruction from a constant-Q spectrum, *in* 'IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'12)', pp. 349–352.

[85] Nawab, S. H., Abu Ayyash, S. and Wotiz, R. [2001], Identification of musical chords using constant-Q spectra, *in* 'IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASP'01)', Vol. 5, pp. 3373–3376.

[86] Nho, W. and Loughlin, P. J. [1999], 'When is instantaneous frequency the average frequency at each time?', *IEEE Signal Processing Letters* **6**(4), 78–80.

[87] Oliveira, P. M. and Barroso, V. [1999], 'Instantaneous frequency of multicomponent signals', *IEEE Signal Processing Letters* **6**(4), 81–83.

[88] Oliveira, P. M. and Barroso, V. [2000], Uncertainty in the time-frequency plane, *in* 'Proceedings of the Tenth IEEE Workshop on Statistical Signal and Array Processing', pp. 607–611.

[89] Oung, H. and Forsberg, F. [1998], 'Theory and applications of adaptive constant-Q distributions', *IEEE Trans. on Signal Processing* **46**(10), 2616–2625.

[90] Pichevar, R. and Rouat, J. [2007], 'Monophonic sound source separation with an unsupervised network of spiking neurones', *Neurocomputing* **71**(1-3), 109–120.

[91] Picinbono, B. [1997], 'On instantaneous amplitude and phase of signals', *IEEE Transactions on Signal Processing* **45**(3), 552–560.

[92] Poliner, G., Ellis, D., Ehmann, A., Gómez, E., Streich, S. and Ong, B. [2007], 'Melody transcription from music audio: Approaches and evaluation', *IEEE Transactions on Audio, Speech and Language Processing* **15**(4), 1247–1256.

[93] Purwins, H., Blankertz, B. and Obermayer, K. [2000], A new method for tracking modulations in tonal music in audio data format, *in* 'International Joint Conference on Neural Networks (IJCNN)', Vol. 6, IEEE Computer Society, pp. 270–275.

[94] Qian, S. and Chen, D. [1996], *Joint Time–Frequency Analysis: Methods and Applications*, Upper Saddle River, NJ: PTR Prentice Hall.

[95] Rabiner, L. R. and Schafer, R. W. [1978], *Digital Processing of Speech Signals*, Prentice-Hall Series, Englewood Cliffs, NJ, USA, chapter 6, pp. 250–344.

[96] Rafii, Z. and Pardo, B. [2011], Degenerate unmixing estimation technique using the constant-Q transform, *in* 'IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '11)', pp. 217–220.

[97] Reis, G., Fonseca, N. and Ferndandez, F. [2007], Genetic algorithm approach to polyphonic music transcription, *in* 'IEEE International Symposium on Intelligent Signal Processing (WISP'07)', pp. 1–6.

[98] Ryynänen, M. P. and Klapuri, A. [2005], Polyphonic music transcription using note event modeling, *in* 'IEEE Workshop on Applications of Signal Processing to Audio and Acoustics', pp. 319–322.

[99] Sangwine, S. J. [1997], The discrete quaternion Fourier transform, *in* 'Sixth International Conference on Image Processing and Its Applications IPA'97)', Vol. 2, pp. 790–793.

[100] Schörkhuber, C. and Klapuri, A. [2010], Constant-Q transform toolbox for music processing, *in* '7th Sound and Music Computing Conference', Barcelona, Spain.

[101] Shahnaz, C., Zhu, W.-P. and Ahmad, M. O. [2012], 'Pitch estimation based on a harmonic sinusoidal autocorrelation model and a time-domain matching scheme', *IEEE Transactions on Audio, Speech and Language Processing* **20**(1), 322–335.

[102] Smaragdis, P. and Brown, J. C. [2003], Non-negative matrix factorization for polyphonic music transcription, *in* 'IEEE Workshop on Applications of Signal Processing to Audio and Acoustics', pp. 177–180.

[103] Smith, S. W. [1997], *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, San Diego, CA, USA, chapter 16, pp. 285–296.

[104] Sophea, S. and Phon-Amnuaisuk, S. [2007], Determining a suitable desired factors for nonnegative matrix factorization in polyphonic music transcription, *in* 'International Symposium on Information Technology Convergence (ISITC07)', pp. 166–170.

[105] Tzanetakis, G. [2002], Manipulation, Analysis and Retrieval Systems for Audio Signals, PhD thesis, Princeton University.

[106] Tzanetakis, G., Kapur, A. and Mcwalter, R. I. [2005], Subband-based drum transcription for audio signals, *in* 'IEEE International Workshop on Multimedia Signal Processing'.

[107] Unser, M. [2000], 'Sampling–50 years after Shannon', *Proceedings of the IEEE* **88**(4), 569–587.

[108] Wheelon, A. D. and Robacker, J. T. [1968], *Table of Summable Series and Integrals Involving Bessel Functions*, Holden-Day Advanced Physics Monographs, Holden-Day; First Thus edition, San Francisco.

[109] Yeh, C., Roebel, A. and Rodet, X. [2010], 'Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals', *IEEE Transactions on Audio, Speech, and Language Processing* **18**(6), 1116–1126.

[110] Yeh, M.-H. [2008], 'Relationships among various 2-D quaternion Fourier transforms', *IEEE Signal Processing Letters* **15**, 669–672.

[111] Zhang, Y., Li, H. and Bi, L. [2008], Adaptive instantaneous frequency estimation based on EMD and TKEO, *in* 'Congress on Image and Signal Processing (CISP'08)', Vol. 1, pp. 60–64.

[112] Zhu, Y., Kankanhalli, M. S. and Gao, S. [2005], Music key detection for musical audio, *in* 'Proceedings of the 11th International Multimedia Modelling Conference (MMM'05)', pp. 30–37.

[113] Zygmund, A. [1988], *Trigonometric Series: [volumes I & II Combined]*, Cambridge Mathematical Library, Cambridge University Press.

Appendix A

Mathematical Derivations and Proofs

A.1 Derivation of (4.7) in section 4.4

$C[k,n]$ is derived by:

$$C[k,n] = \mathrm{P}((1+\frac{\lambda}{2})\upsilon_k)[n] - \mathrm{P}((1-\frac{\lambda}{2})\upsilon_k)[n]$$

Using

$$\tilde{\mathrm{P}}(\upsilon)[n] = \sum_{m=-\infty}^{+\infty} \upsilon \mathrm{sinc}\left(\upsilon(m-n)\right) x[m],$$

we derive the output of the band pass filter for B_k as:

$$C[k,n] = \sum_{m=-\infty}^{+\infty} \upsilon_k \cdot (1+\frac{\lambda}{2}) \cdot \mathrm{sinc}\left(\upsilon_k(1+\frac{\lambda}{2})(m-n)\right) x[m] - \upsilon_k \cdot (1-\frac{\lambda}{2}) \cdot \mathrm{sinc}\left(\upsilon_k(1-\frac{\lambda}{2})(m-n)\right) x[m].$$

By shifting m by n ($m \leftarrow m+n$),

$$C[k,n] = \sum_{m=-\infty}^{+\infty} \upsilon_k \cdot (1+\frac{\lambda}{2}) \cdot \text{sinc}\left(\upsilon_k(1+\frac{\lambda}{2})m\right) x[m+n] - \upsilon_k \cdot (1-\frac{\lambda}{2}) \cdot \text{sinc}\left(\upsilon_k(1-\frac{\lambda}{2})m\right) x[m+n].$$

Therefore:

$$F_k[m] = \upsilon_k \cdot (1+\frac{\lambda}{2}) \cdot \text{sinc}\left(\upsilon_k(1+\frac{\lambda}{2})m\right) - \upsilon_k \cdot (1-\frac{\lambda}{2}) \cdot \text{sinc}\left(\upsilon_k(1-\frac{\lambda}{2})m\right).$$

To simplify $F_k[m]$ we expand the sinc functions. Without loosing the generality, we exclude the special case where $m = 0$. The result is similar. Therefore:

$$F_k[m] = \frac{1}{\pi m}\left(\sin\left(\pi\upsilon_k(1+\frac{\lambda}{2})m\right) - \sin\left(\pi\upsilon_k(1-\frac{\lambda}{2})m\right)\right).$$

By expanding above, we obtain:

$$F_k[m] = \frac{1}{\pi m}\left(\sin\left(\pi\upsilon_k m + \pi\upsilon_k m\frac{\lambda}{2})\right) - \sin\left(\pi\upsilon_k m - \pi\upsilon_k m\frac{\lambda}{2})\right)\right).$$

By combining the two sines, we get:

$$F_k[m] = \frac{2}{\pi m}\cos(\pi\upsilon_k m)\sin(\pi\upsilon_k m\frac{\lambda}{2}).$$

By rewriting the sine function by sinc, we obtain:

$$F_k[m] = \cos(\pi\upsilon_k m)\lambda\upsilon_k \text{sinc}(\upsilon_k m\frac{\lambda}{2}).$$

■

A.2 Derivation of the Phasor Coefficients in section 4.4

Using two consecutive samples: $s[n]$ and $s[n \pm 1]$, we write:

$$\begin{cases} C[k, n-1] \approx a[k, n] \cos((n-1)\pi\upsilon_k + \Phi[k, n]) \\ C[k, n+1] \approx a[k, n] \cos((n+1)\pi\upsilon_k + \Phi[k, n]) \end{cases}.$$

By expanding the cosine functions, we obtain:

$$\begin{cases} a[k, n] \cos(\Phi[k, n]) \cos((n-1)\pi\upsilon_k) - a[k, n] \sin(\Phi[k, n]) \sin((n-1)\pi\upsilon_k) \approx C[k, n-1] \\ a[k, n] \cos(\Phi[k, n]) \cos((n+1)\pi\upsilon_k) - a[k, n] \sin(\Phi[k, n]) \sin((n+1)\pi\upsilon_k) \approx C[k, n+1] \end{cases}.$$

By rewriting the above equations in matrix form, we get:

$$\overbrace{\begin{pmatrix} \cos((n-1)\pi\upsilon_k) & -\sin((n-1)\pi\upsilon_k) \\ \cos((n+1)\pi\upsilon_k) & -\sin((n+1)\pi\upsilon_k) \end{pmatrix}}^{D(\upsilon_k, n)} \cdot \begin{pmatrix} a[k, n] \cos(\Phi[k, n]) \\ a[k, n] \sin(\Phi[k, n]) \end{pmatrix} \approx \begin{pmatrix} C[k, n-1] \\ C[k, n+1] \end{pmatrix}.$$

Since

$$\mathrm{H}[k, n] = \begin{pmatrix} 1 \\ \mathbf{j} \end{pmatrix}^T \cdot \begin{pmatrix} a[k, n] \cos(\Phi[k, n]) \\ a[k, n] \sin(\Phi[k, n]) \end{pmatrix},$$

we obtain $\mathrm{H}[k, n]$ using:

$$\mathrm{H}[k, n] \approx \begin{pmatrix} 1 \\ \mathbf{j} \end{pmatrix}^T \cdot D(\upsilon_k, n)^{-1} \cdot \begin{pmatrix} C[k, n-1] \\ C[k, n+1] \end{pmatrix}.$$

$D(\upsilon, n)^{-1}$ may also be derived by:

$$\frac{1}{\det D(\upsilon, n)} \cdot \begin{pmatrix} -\sin((n+1)\pi\upsilon_k) & \sin((n-1)\pi\upsilon_k) \\ -\cos((n+1)\pi\upsilon_k) & \cos((n-1)\pi\upsilon_k) \end{pmatrix}$$

where

$$\begin{aligned}\det D(\upsilon, n) &= \cos((n+1)\pi\upsilon_k)\sin((n-1)\pi\upsilon_k) - \sin((n+1)\pi\upsilon_k)\cos((n-1)\pi\upsilon_k)\\ &= \sin\left((n-1)\pi\upsilon_k - (n+1)\pi\upsilon_k\right)\\ &= -\sin(2\pi\upsilon_k).\end{aligned}$$

The derivation of the phasor construct is given in the following:

$$\begin{aligned}\begin{pmatrix} 1 \\ \mathbf{j} \end{pmatrix}^T \cdot D(\upsilon_k, n)^{-1} &= \tfrac{-1}{\sin(2\pi\upsilon_k)} \cdot \begin{pmatrix} 1 \\ \mathbf{j} \end{pmatrix}^T \cdot \begin{pmatrix} -\sin((n+1)\pi\upsilon_k) & \sin((n-1)\pi\upsilon_k) \\ -\cos((n+1)\pi\upsilon_k) & \cos((n-1)\pi\upsilon_k) \end{pmatrix}\\ &= \tfrac{1}{\sin(2\pi\upsilon_k)} \cdot \begin{pmatrix} \sin((n+1)\pi\upsilon_k) + \mathbf{j}\cos((n+1)\pi\upsilon_k) \\ -\sin((n-1)\pi\upsilon_k) - \mathbf{j}\cos((n-1)\pi\upsilon_k) \end{pmatrix}^T\\ &= \tfrac{\mathbf{j}}{\sin(2\pi\upsilon_k)} \cdot \begin{pmatrix} \mathrm{e}^{-\mathbf{j}(n+1)\pi\upsilon_k} \\ -\mathrm{e}^{\mathbf{j}(1-n)\pi\upsilon_k} \end{pmatrix}^T\\ &= \mathrm{e}^{-\mathbf{j}n\pi\upsilon_k} \cdot \underbrace{\frac{\mathbf{j}}{\sin(2\pi\upsilon_k)} \cdot \begin{pmatrix} \mathrm{e}^{-\mathbf{j}\pi\upsilon_k} \\ -\mathrm{e}^{\mathbf{j}\pi\upsilon_k} \end{pmatrix}^T}_{\wp_{\upsilon_k}}\end{aligned}$$

■

A.3 Derivation of the equations in section 4.4

Using (2.6), the upper bound for B_k's is obtained by $(1 + \frac{\lambda}{2})\upsilon_k$. Since $\forall\omega, \omega \leq 1$, we can write:

$$(1 + \frac{\lambda}{2})\upsilon_k \leq 1$$

Therefore:

$$\upsilon_k \leq \frac{2}{2 + \lambda}$$

■

Using $\upsilon_k = \gamma^k \upsilon_0$, we can write:

$$(1 + \frac{\lambda}{2})\gamma^k \upsilon_0 \leq 1.$$

By taking logarithm from both sides, we obtain:

$$k \log \gamma + \log \upsilon_0 + \log(1 + \frac{\lambda}{2}) \leq 0.$$

Using (2.4), we know that $\log \gamma > 0$, therefore:

$$k \leq -\frac{1}{\log \gamma} \left(\log \upsilon_0 + \log(\lambda + 2) - \log 2\right).$$

■

In case of using (2.7), the upper bound for B_k's is obtained by $\sqrt{\gamma}\upsilon_k$. Since $\sqrt{\gamma}\upsilon_k \leq 1$, we obtain:

$$\upsilon_k \leq \frac{1}{\sqrt{\gamma}}.$$

■

The upper bound for k can similarly be derived using $\upsilon_k = \gamma^k \upsilon_0$:

$$\gamma^k \sqrt{\gamma} \log \upsilon_0 \leq 1$$

By taking logarithm from both sides, we get:

$$k \log \gamma + \frac{1}{2} \log \gamma + \log v_0 \leq 0.$$

Since we know that $\gamma > 1$, the upper bound for k will be:

$$k \leq -\frac{\log v_0}{\log \gamma} - \frac{1}{2}.$$

■

A.4 Multi-Dimensional Continuous IHA – Special Case

In case of $M = 1$,

$$
\begin{aligned}
\mathrm{H}_k(t) &= \Xi_k\{C_k(t)\} \\
&= {U_1}^T \cdot \wp(\omega_k, t)\{C_k(t)\} \\
&= \begin{pmatrix} 1 \\ \mathbf{j} \end{pmatrix}^T \cdot \begin{pmatrix} \cos(\omega_k t) & -\sin(\omega_k t) \\ -\sin(\omega_k t) & -\cos(\omega_k t) \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \frac{1}{\omega_k}\frac{\partial}{\partial t} \end{pmatrix} \{C_k(t)\} \\
&= \begin{pmatrix} \cos(\omega_k t) - \mathbf{j}\sin(\omega_k t) \\ -\sin(\omega_k t) - \mathbf{j}\cos(\omega_k t) \end{pmatrix}^T \cdot \begin{pmatrix} 1 \\ \frac{1}{\omega_k}\frac{\partial}{\partial t} \end{pmatrix} \{C_k(t)\} \\
&= \begin{pmatrix} \mathrm{e}^{-\mathbf{j}\omega_k t} \\ -\mathbf{j}\mathrm{e}^{-\mathbf{j}\omega_k t} \end{pmatrix}^T \cdot \begin{pmatrix} 1 \\ \frac{1}{\omega_k}\frac{\partial}{\partial t} \end{pmatrix} \{C_k(t)\} \\
&= \mathrm{e}^{-\mathbf{j}\omega_k t} \cdot \begin{pmatrix} 1 \\ -\mathbf{j} \end{pmatrix}^T \cdot \begin{pmatrix} 1 \\ \frac{1}{\omega_k}\frac{\partial}{\partial t} \end{pmatrix} \{C_k(t)\} \\
&= \mathrm{e}^{-\mathbf{j}\omega_k t} \left[1 - \frac{\mathbf{j}}{\omega_k}\frac{\partial}{\partial t}\right] C_k(t).
\end{aligned}
$$

■

A.5 Multi-Dimensional Discrete IHA – Special Case

In case of $M = 1$,

$$\begin{aligned}
\mathrm{H}[k,n] &= \Xi_k\{C[k,n]\} \\
&= \mathrm{e}^{-\mathbf{j}\pi\upsilon_k n} \cdot \wp_{\upsilon_k,1}\left(C[k,n]\right) \\
&= \mathrm{e}^{-\mathbf{j}\pi\upsilon_k n} \cdot \tfrac{\mathbf{j}}{\sin 2\pi\upsilon} \begin{pmatrix} \mathrm{e}^{-\pi\upsilon_k\mathbf{j}} \\ -\mathrm{e}^{\pi\upsilon_k\mathbf{j}} \end{pmatrix}^T \cdot \begin{pmatrix} C[k,n+(-1)] \\ C[k,n+(+1)] \end{pmatrix} \\
&= \mathrm{e}^{-\mathbf{j}\pi\upsilon_k n} \cdot \tfrac{\mathbf{j}}{\sin 2\pi\upsilon} \begin{pmatrix} \mathrm{e}^{-\mathbf{j}\pi\upsilon_k} \\ -\mathrm{e}^{\mathbf{j}\pi\upsilon_k} \end{pmatrix}^T \cdot \begin{pmatrix} C[k,n-1] \\ C[k,n+1] \end{pmatrix}
\end{aligned}$$

■

Appendix B

The Specifications of the Simulation Data Sets

Data Set	*Specification*	*Details*
DS1	treble	pieces from Beyer Op.101 piano right hand, medium level
DS2	easy full-range	pieces from Beyer Op.101 John Thompson's Easiest Piano Course
DS3	four-part choral	pieces from Nikolai Rimsky-Korsakov's harmony exercises
DS4	full-range piano	Invention No. 1 by Johann Sebastian Bach BWV.772 Ludwig van Beethoven's Für Elise, WoO 59 Flight of the Bumblebee, Nikolai Rimsky-Korsakov
DS5	orcherstral & piano-duet	Flight of the Bumblebee, version for string orchestra selected pieces for piano and violin selected pieces for piano and flute