Music Signal Processing Techniques

Odysseas Zafeiriou

8/1/2019

Department of Electrical and Computer Engineering

Technical University of Crete

Committee members

Professor Georgios Karystinos

Professor Michail Lagoudakis

Professor Aikaterini Mania

Abstract

Note onset and pitch detection is a standard tool in music signal analysis and processing. In this thesis, after a short introduction to the nature of musical notes and piano notes in particular, we present an overview of various techniques for detecting piano note onsets as well as pitch. Several onset detection algorithms are analyzed and tested against single piano notes and more complex monophonic passages. Subsequent thresholding techniques are used to extract the results, which are finally compared and evaluated. As far as pitch or note detection is concerned, some basic concepts are presented as well as an attempt to tackle the constraints presented by the innate inharmonicity present in piano tuning.

Contents

1 Introduction

1.1 Motivation

The research field this thesis wanted to explore and contribute to was automatic music transcription. The initial goal was to implement an algorithm for extracting the harmonic content of piano chords. However, the first phase concerning the detection of onset events, was already proven to be an interesting and challenging task, also due to the unique nature of the piano. Thus, an overview of the available onset detection techniques and their implementation on real piano signals was considered a valuable study. Along with the comparison of the aforementioned methods two implementations for detecting the pitch of the under examination note were presented.

1.2 Outline

The second section of this thesis acts as an introduction to the piano as an instrument, its mechanics and some of its unique properties but also the characteristics of piano notes both in the time and frequency domain. The section concludes with a special reference to the effect of inharmonicity.

The third section introduces in more detail the signals used in the course of this thesis, from the way these sample signals were obtained to fundamental concepts and processes(e.g. the *Fourier* transform) used throughout the implementation of the algorithms.

The fourth section covers the overview of the implemented onset detection methods. After the introduction to basic concepts(e.g. *onset, attack* and *transient*) and general approach to onset detection algorithms, the methodology followed in obtaining the detection functions is presented. More specifically, the input signal was subject to time-block analysis, i.e. split in windows of certain samples which were further analyzed separately. These windows were in some implementations overlapping over half the window size. In order to avoid *spectral leakage* caused by the windowing process, a Hamming window was used to minimize its effect. In total six onset detection algorithms were implemented, two in the time domain and four in the frequency domain. More specifically in the time domain, the energy of the window and the ratio between two successive window maxima were the measures whose change in time were followed. In the frequency domain, first the *High-frequency content* was used as a measure, hoping to highlight and take advantage of the *percusiveness* of the onset and its broadband nature. Then an implementation based on the spectral difference, through the magnitude part of the Fourier transform of the windows, was tested. The last two methods, extracted onset information through the phase part of the frequency domain signal. Both take advantage of the stationarity of a periodic signal, as measured in the progression of its phase and how it is disrupted when an onset event occurs. The last method expresses this stationarity in terms of the phasor of the signal, namely measuring the phase and magnitude of the signal and calculating its euclidean distance from respective estimated values. The detection function obtained were then thresholded using an adaptive threshold which follows the local median value of a window follower. The algorithms were first tested and compared against a three note piano excerpt and in turn against a noisy recording of a nocturne by F.Chopin. The latter succeeded in highlighting pros and cons of the methods used.

In the last section, two methods for note detection are presented. The first one, uses the *Cauchy-Schwarz* inequality, to find the most likely candidate for the note played among predefined libraries of spectra of all the 88 notes of a piano. A case for libraries of averaged spectra from multiple pianos and that of ideal self-defined spectra were examined. The second method is based on the energy of the harmonics of a piano note. It aims to pick at least a pair of consecutive order harmonics in its candidates in order to decide the fundamental frequency. For this reason the method is run thrice; for 3, 5 and 10 maxima, ultimately offering a set of three note candidates.

2 Introduction to piano notes

2.1 Piano's physical characteristics

The piano is an acoustic, stringed musical instrument invented around the year 1700 (the exact year is uncertain), in which the strings are struck by hammers. It is played using a keyboard, which is a row of keys (small levers) that the performer presses down or strikes with the fingers and thumbs of both hands to cause the hammers to strike the strings [1].

The supporting structure of the piano is the rim, a rigid, curve-shaped wooden structure that supports all the other elements and which, theoretically speaking, does not vibrate. The soundboard is a piece of 'Picea' wood that fits into the rim, with its borders stuck to the inner perimeter of the rim. The soundboard vibrates with its edges clamped, it is the actual radiation element of the piano. It is about 10 mm thick, and it presents an anisotropic stiffness. The vibration of the soundboard is due to the transmission of the strings' vibration through a piece called the bridge, which is stuck to the soundboard. Two bridges exist, one for the bass strings and the other for the remaining strings. The point of change is called the bass break. Strings are the main vibrating elements. They are excited by the hammers (action mechanism) when the key is pressed by the pianist. The string is tensioned with one end (the one far from the keyboard) fixed on the rigid cast iron frame and the other (that nearest the keyboard) rolled onto the tuning pin of the wrest plank. The "speaking" length is the portion of the string between the two points of support the bridge and the agrafe. In certain strings the agrafe is replaced by the capo d'astro as the support. They are both constructed on the iron frame. The speaking length determines the main vibration, because the remainder of the string is highly damped to prevent vibration. All the string parameters will be related to the speaking portion. The strings are made of steel. The bass strings, which lie on the bass bridge, are wound with one or two layers of copper wire to increase their linear density without increasing the diameter of the steel string. The length and diameter (and thus the linear density) of the string are different from note to note. Most of the piano notes have two or even three strings. This combination is called unison. Only one string belonging to the unison has the right tuning. The others are tuned out slightly by about one or two cents [2]. This adjustment is responsible for the vibrato and double decay characteristics of the piano sound. The lowest notes use only a single string per note [3].

2.2 Piano note in time and frequency domain

2.2.1 Time domain

The representation of a piano note is seen in Fig. 1a. The y-axis refers to the air pressure levels, or time magnitude of the time signal. When a note event occurs, the envelope of the signal raises drastically. This event is referred to as the *attack*. The *attack* is followed by a decay, sustain period, during which the amplitude of the signal drops. These events will be studied more explicitly at section 3 where onset detection methods are presented.

2.2.2 Frequency domain

Single piano notes are generally assigned and referred to as a single frequency, known as the pitch frequency. The pitch values (see Table 1) increase logarithmically as we ascend in the piano *clavier* (keyboard), and always derive from the tuning pitch of A4.

$$P(i) = 2^{\frac{i-49}{12}} P(49)) \tag{1}$$

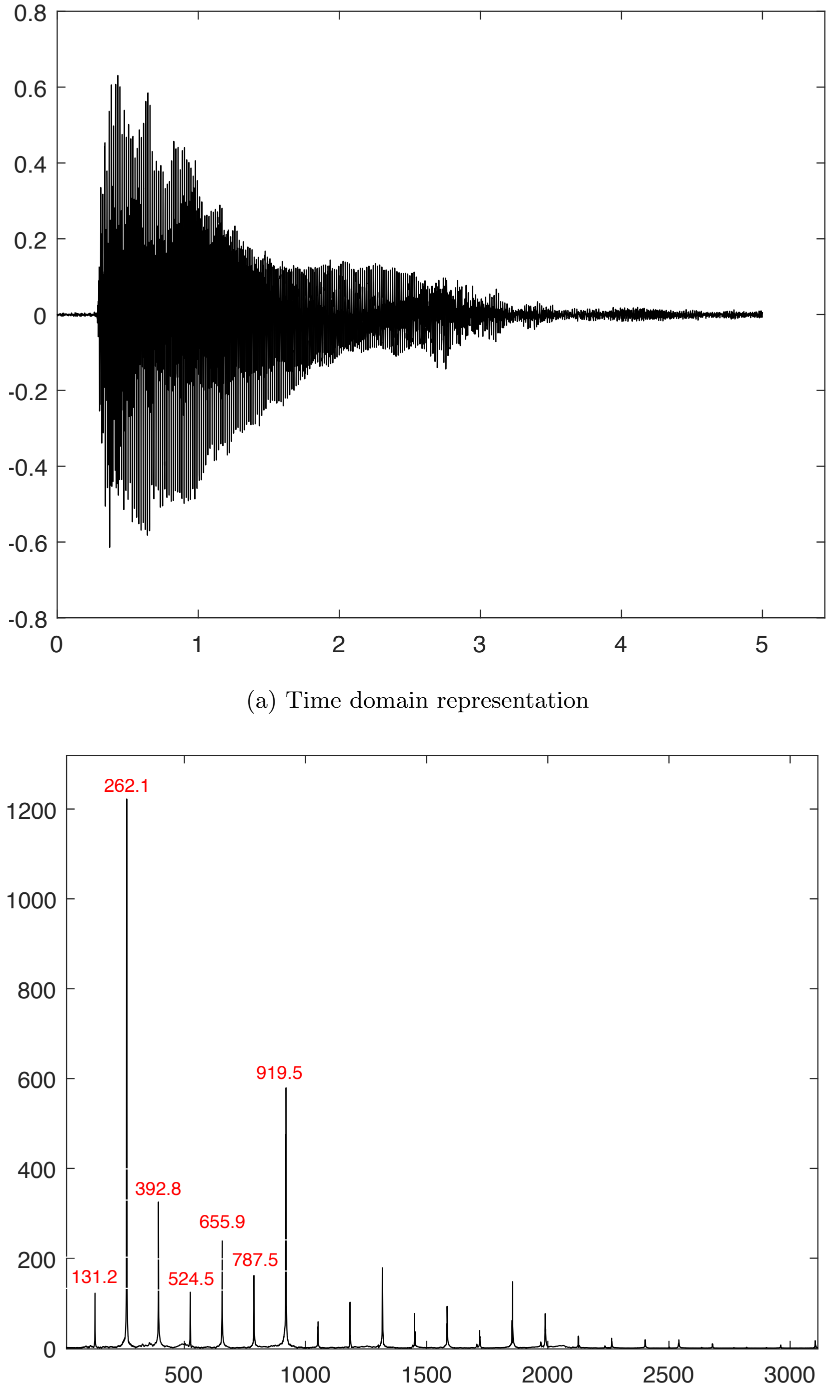
where $i = 1, 2, \ldots, 88$ denoting note increasing number and $P(49)$ refers to the tuning pitch of note A4.

The piano note spectrum however, consists of theoretically infinite sinusoids all with a frequency close to positive integer multiples of the pitch. The fundamental frequency F_0 is referring to the pitch of the note. Subsequent harmonics are labeled as $F_1 \simeq 2F_0$, $F_2 \simeq 3F_0$ etc.

Timbre The *timbre* of any musical instrument, is basically its identity. It is what makes a flute sound different than a violin, a fagotto or a piano. It is described by the ratios of the harmonics in relation to the fundamental. These ratios may differ even from piano to piano, and are also affected by the interpretation, as the latter affects some features concerning the note onset.

Harmonically related notes By *harmonically related notes* we refer to notes who share a number of high-energy harmonics, i.e. whose spectra have an increased degree of matching. It is this matching that makes certain notes sound *in harmony* when played together and is the basis for Harmony rules and limitations found in music Theory.[1].

[1] E.g. The interval of an augmented 4th , for example F4-B4 was widely avoided during the classical era, since these notes share no harmonics of significant energy, thus introducing disharmony.Basic examples of harmonic notes are octaves(8th), minor or major 3rd and perfect 5th.[http://musictheoryblog.blogspot.gr/2007/01/intervals.html]



(a) Time domain representation

(b) Frequency magnitude representation(zoomed to [0-3kHz])

Figure 1: Time and frequency magnitude frequency representation of piano note C3

Piano Note	Pitch(Hz)	**Piano Note**	Pitch(Hz)	**Piano Note**	Pitch(Hz)	**Piano Note**	Pitch(Hz)
A0	27.5	***G2***	98	***F4***	349.23	***D#6***	1244.51
A#0	29.14	***G#2***	103.83	***F#4***	369.99	***E6***	1318.51
B0	30.87	***A2***	110	***G4***	392	***F6***	1396.61
C1	32.7	***A#2***	116.54	***G#4***	415.3	***F#6***	1479.98
C#1	34.65	***B2***	123.47	***A4***	440	***G6***	1567.98
D1	36.71	***C3***	130.81	***A#4***	466.16	***G#6***	1661.22
D#1	38.89	***C#3***	138.59	***B4***	493.88	***A6***	1760
E1	41.2	***D3***	146.83	***C5***	523.25	***A#6***	1864.66
F1	43.65	***D#3***	155.56	***C#5***	554.37	***B6***	1975.53
F#1	46.25	***E3***	164.81	***D5***	587.33	***C7***	2093
G1	49	***F3***	174.61	***D#5***	622.25	***C#7***	2217.46
G#1	51.91	***F#3***	185	***E5***	659.25	***D7***	2349.32
A1	55	***G3***	196	***F5***	698.46	***D#7***	2489.02
A#1	58.27	***G#3***	207.65	***F#5***	739.99	***E7***	2637.02
B1	61.74	***A3***	220	***G5***	783.99	***F7***	2793.83
C2	65.41	***A#3***	233.08	***G#5***	830.61	***F#7***	2959.96
C#2	69.3	***B3***	246.94	***A5***	880	***G7***	3135.96
D2	73.42	***C4***	261.63	***A#5***	932.33	***G#7***	3322.44
D#2	77.78	***C#4***	277.18	***B5***	987.77	***A7***	3520
E2	82.41	***D4***	293.66	***C6***	1046.5	***A#7***	3729.31
F2	87.31	***D#4***	311.13	***C#6***	1108.73	***B7***	3951.07
F#2	92.5	***E4***	329.63	***D6***	1174.66	***C8***	4186.01

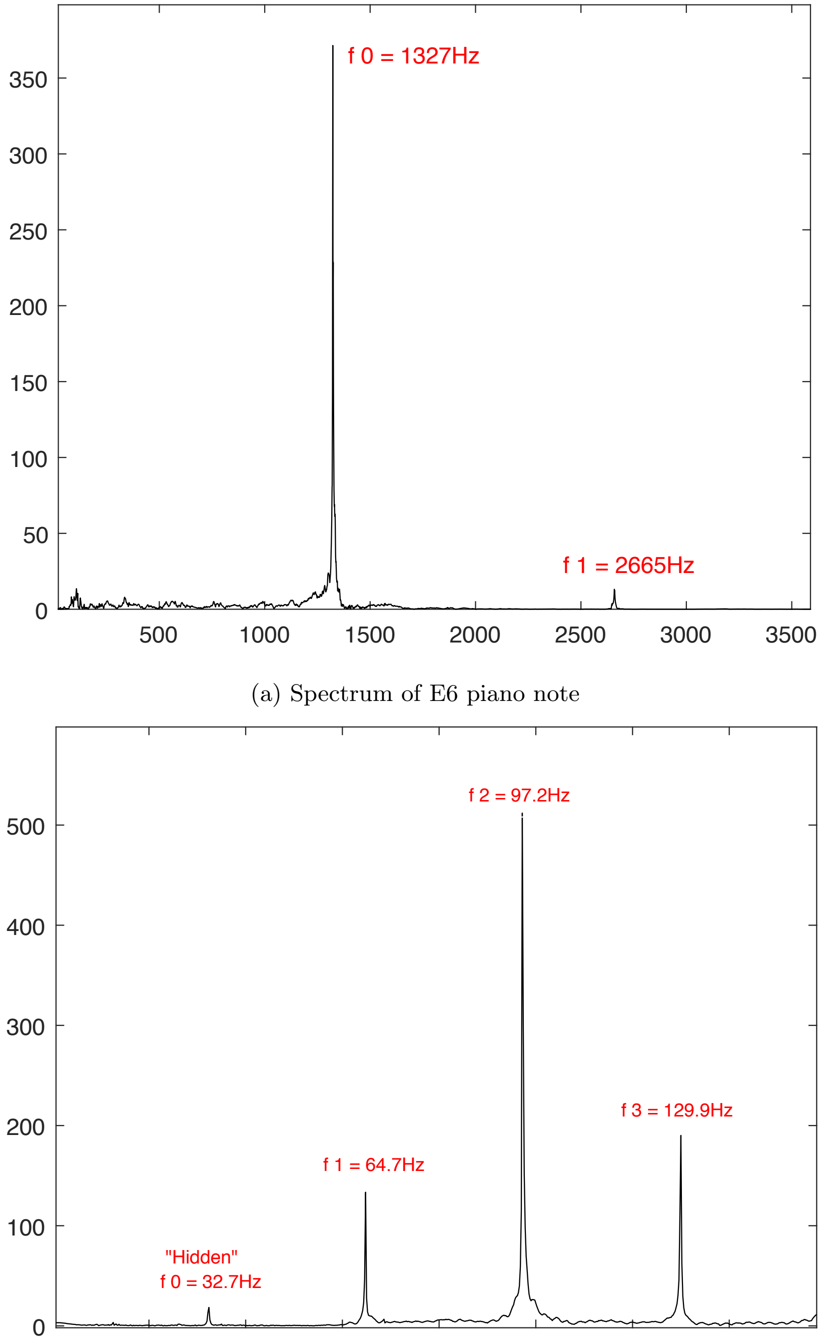
Table 1: Piano note tempered pitches for A4=440 Hz tuning

Piano note special spectral features

2.2.3 Inharmonicity

The characteristics of the piano elements lead to a set of specific aspects of the vibrational parameters. The position where the string is struck by the hammer and the width of the striking zone are not the same for all notes. Both aspects affect the spectral envelope, acting like a comb filter, so the spectral envelope of the notes changes as the octaves increase. The stiffness of the steel string causes an increase in frequency of the harmonics above the harmonic values, so the string vibration is clearly inharmonic. Because inharmonicity [4] depends on the physical parameters of the string, every note has a different degree of inharmonicity. Inharmonicity also affects piano tuning. When tuning a note that is one octave higher than the other, if the higher note is tuned at twice the frequency of the lower one, a beat will appear. That beat is produced by the fundamental of the higher note and the second harmonic of the lower note (which is higher than the former due to inharmonicity). To avoid the occurrence of this beat, the higher note is tuned to be coincident with the second harmonic of the lower note, and thus slightly above its "tempered" value. As a result the piano notes are not tuned to the tempered values, with the exception of the A4 note and basically the octave from C3-C4. The piano tuning is generally described by Railsback curve(see Fig. 3). Furthermore the end of the string held by the bridge is not fixed; it is moving. This changes the vibration frequency in a way that is different for every harmonic of the note. This modification depends on the mechanical impedance of the soundboard. The resulting frequency can be higher or lower than that for fixed ends. Finally, it is understood that in most cases any piano(unless very recently hard tuned), will present some slightly or more evidently out of tune notes [3].

Consequently, as we can see in Fig. 1b the actual frequency of the fundamental harmonic F_0 of a piano C3 note is slightly out-of-tune than the theoretical 130.81 Hz. Similarly, subsequent harmonics have different values than the exact multiples of the F_0. In Fig. 2, for the sake of showing the diversity between different piano note spectra, we see the spectrum of notes E6 and C1. An other interesting point seen in the case of the C1 note (Fig. 2b) is the fact that the fundamental frequency F_0 is "hidden", its energy is very low compared to the respective values of higher harmonics. This phenomenon is present when dealing with very low note and cedays as we surpass the first octave of the piano range. Psychoacoustic researches however, suggest that human ear distinguishes between notes, not only by the pitch, but also from the relative magnitudes and frequency distances between higher harmonics. The implications that derive from the inharmonicity and "hidden fundamental" features of piano note spectra, will be further discussed in section 4, where note detection algorithms will be presented.



(a) Spectrum of E6 piano note

(b) Part of C1 spectrum showing the hidden fundamental at 32.7 Hz

Figure 2: Spectra of notes E6 and C1

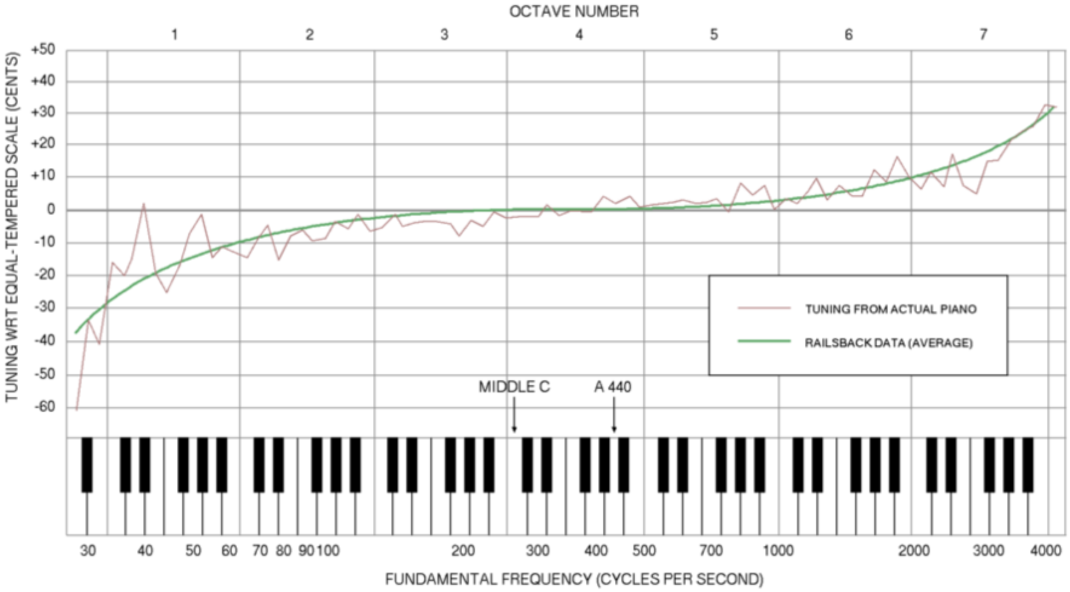


Figure 3: The Railsback curve(green) indicates the average difference in cents (1 semitone consists of 100 cents) between tempered and actual tuning value.

3 Signal model

For all the tests performed throughout the course of this study, three types of short piano music pieces $I(t)$ were used. Their duration was customized using *Cubase* software.

1. Stereo[1] non-anechoic[2] 5 second recordings of all piano notes played in a *Steinway & Sons model B* piano interpreted at fortissimo intensity found in the site of University of Iowa dept. of Electronic Music library [5].

2. Stereo non-anechoic recording of piano passage (C4 - E4 - D4) using the built-in microphones of a *Samsung Galaxy S6 Edge.*

3. Sample of a stereo non-anechoic recording of ***Chopin's*** *Nocturne op.9 No.1.*

[1] **Stereo recording** is a technique involving the use of two microphones to simultaneously record one instrument. The mono signals from each microphone are assigned to the left and right channels of a stereo track to create a sense of width in the recording [6].

[2] **Non - anechoic** means the recording took place in a room whose boundaries do not fully absorb sound waves , resulting in some echoing signals.

Time domain signal All the above pieces were recorded at a sampling frequency $F_\mathrm{s} = 44.1\mathrm{kHz}$. With Matlab's *audioread* function we imported the respective .wav files to our project. Since the recordings were stereo, the imported signals also consisted of two separate channels of equal length N found as

$$N = TF_\mathrm{s} \tag{2}$$

where T is the duration of the piece's continuous time signal $I(t)$ in seconds. Each separate channel of the .wav format is described as

$$x_c(n) = I_c(nT_\mathrm{s}) \tag{3}$$

where $x(n)$ is the sampled version(.wav file) of $I(t)$, $T_\mathrm{s} = 1/F_\mathrm{s}$ is the sampling period and c refers channels 1 or 2 of the stereo recordings.

The two channels were averaged, to ultimately constitute the signal used in the implemented algorithms. This signal basically represents the discrete resentation of the pieces.

$$x(n) = \frac{x_1(n) + x_2(n)}{2} \tag{4}$$

Frequency domain signal

Fourier transform The Fourier transform transforms a signal from its original domain (in this case, time) to the frequency domain and vise versa. The forward transform (time to frequency) of the continuous time input signal $I(t)$ is given as

$$X(F) = \int_{-\infty}^{\infty} I(t)e^{-j2\pi tF} \tag{5}$$

F referring to an infinite possible frequency values. However, for easier manipulation the values of F are discetized as suggested by the Discrete Fourier transform of the discrete time singal $x(n)$.

Discrete Fourier transform The *Discrete Fourier Transform* or DFT represents the signal in the frequency domain, in terms of discrete bins rather than continuous frerquencies. The DFT frequency representation signal is

$$X(k) = \sum_{n=1}^{N} x(n)e^{-j2\pi\frac{nk}{N}} \tag{6}$$

where k now refers to frequency bins. The *Fast Fourier Transform* in *Matlab*, is basically the same as the DFT with some improvements in the algorithm's complexity. So by running the FFT for our discrete real signal *x(n)* we end up with the complex signal *X(k)*.

The amplitude part of the k_{th} bin is calculated as

$$|X(k)| = \sqrt{\Re(X(k))^2 + \Im(X(k))^2} \tag{7}$$

and the phase component is found as

$$\phi_k = \tan^{-1}\left(\frac{\Re(X(k))}{\Im(X(k))}\right). \tag{8}$$

To further improve the FFT's frequency representation and computing speed it is optional to choose an input signal of size

$$N' = 2^a \tag{9}$$

where

$$a = \lceil \log_2 N \rceil \tag{10}$$

We now call the FFT for the specified input size N'.

$$X(k) = \mathrm{FFT}(x(n), N') \tag{11}$$

This procedure in *Matlab* is performed by assigning a 0 value to the extra $N' - N$ samples added at the right of $x(n)$, also known as *right zero padding*.

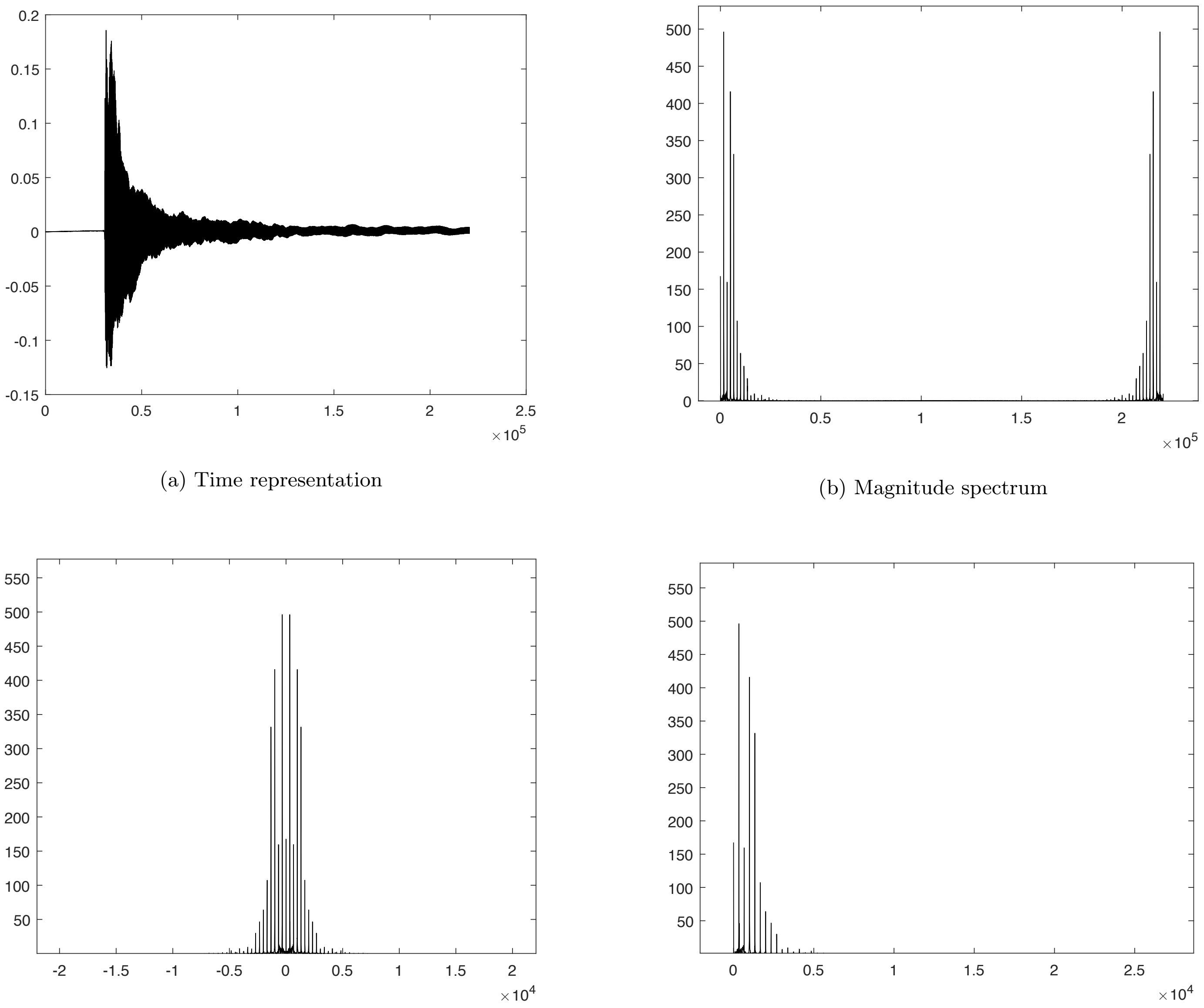
The FFT output $X(k)$ is symmetric around the central bin. With the use of the *fftshift* function in *Matlab* we center this symmetric representation around bin#1 referring to the frequency of 0 Hz(DC component) and finally keep the values referring to positive frequencies, both for amplitude and phase. The whole procedure is shown in Fig. 4.

The actual frequency value associated with the k^{th} bin is found as

$$F(k) = (k-1)\frac{F_{\mathrm{s}}}{N'} \tag{12}$$

and the FFT's frequency resolution, i.e. the difference between successive bins' associated frequency in Hz is

$$\mathrm{FR} = \frac{F_{\mathrm{s}}}{N'} \tag{13}$$



(a) Time representation

(b) Magnitude spectrum

(c) Spectrum around 0

(d) Final form

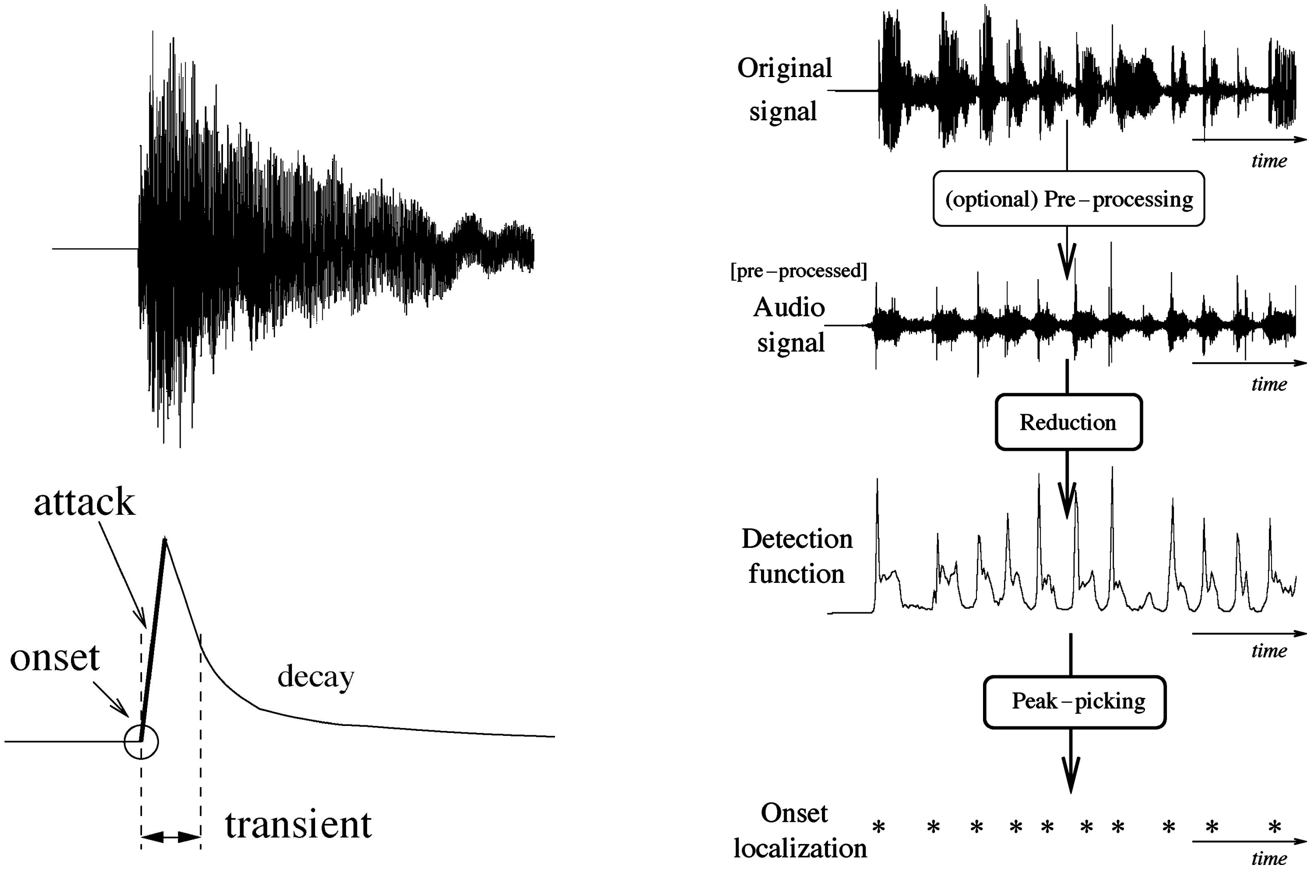
Figure 4: Obtaining the final spectrum of note E4

4 Onset Detection

Basic concepts Automatically detecting music events, gives huge possibilities in a series of music analysis applications such as compression, retrieval and automatic transcription. In the scope of this paper onset detection methods we reviewed, in order to later proceed to recognition of the single piano note samples. Before actually reviewing some onset detection algorithms, a clear distinction between the related concepts of *attack*, *transients* and *onsets*, also shown in Fig. 5a, is mandatory.

- ***Attack*** : The attack refers to the time interval during which the amplitude envelope increases.
- ***Transient*** : The concept of transient is more difficult to describe precisely. As a preliminary informal definition, transients are short intervals during which the signal evolves quickly in some non trivial or relatively unpredictable way. In the case of acoustic instruments, the transient often corresponds to the period during which the excitation (in piano, a hammer strike)is applied and then damped, leaving only the slow decay at the resonance frequencies of the body. Central to this time duration problem is the issue of the useful time resolution: we will assume that the human ear cannot distinguish between two transients less than 10 ms apart. Note that the release or offset of a sustained sound can also be considered a transient period.
- ***Onset*** : The onset of the note is a single instant chosen to mark the temporally extended transient. In most cases, it will coincide with the start of the transient, or the earliest time at which the transient can be reliably detected.

The above terms however, are not so discrete and easily detectable when it comes to real music signals, mainly due to noise presence and multiple sound sources. Piano onsets are considered percussive, due to the mechanics of the hammer, in contrast to e.g. a soft violin onset produced by the "smooth" bowing. However the "percusiveness" of piano onsets might still vary from note to note, depending on the chromatics of the interpretation. *Staccato* and *forte* interpretations, generally tend to increase the attack's gradient and might even enhance some harmonics (due to soundboard resonance) which would be less present in *legato* and *piano* playing. In the latter interpretation, onsets might appear a lot "softer" too. In general, it is impossible to successfully detect onsets without first observing the time-varying "transientness" of the signal [7].



(a)

(b)

Figure 5: (a) Onset parts visualized, (b) Overview of onset detection methods

4.1 Onset detection overview

The general approach of onset detection algorithms consists of three main parts.(Fig. 5).

- ***Pre-Processing*** : In this stage the input signal is processed in such a way, as to enhance certain aspects of it. It is these aspects that will be then used during the construction of the detection function. However, this is an optional step with its importance according to the task in hand and mainly takes place when a simple reduction method is to implemented. A very common such pre-process, is to independently analyze information through multiple frequency bands. This leads to the division of the spectrum to a number of overlapping or non-overlapping bands, either to ultimately combine changes in individual bands to obtain a global estimate or to focus on a particular band to increase the robustness of a certain reduction method [8].

- ***Reduction*** : This step is the core in the procedure of onset detection. The input signal, either pre-processed or not, is analyzed in subsequent time blocks which in almost all cases overlap at a fraction of the analysis block length. For every such block, a distinct property is examined and measured. This property's time evolving output obtained in this step is a signal which aims to make the later step of ultimately peak picking and determining the onsets, simpler and more robust. In general all these methods lie under a more general approach, very common in Automatic Music Transcription, known as time-frequency analysis. *Short-time Fourier*, *Gabor* and *Wigner* are the most broadly used transforms. in this paper the concepts and performance of numerous onset detection approaches is analyzed and evaluated.

- ***Peak Picking*** : This last step before obtaining the possible onset position is peak picking , or in a more broad sense meant as post processing the detection function found on the previous step. First task, if needed, is low-pass filtering the detection function to follow more general schemes in its temporal evolution and then "following" this smooth version to mark local maxima. Later in this paper we will propose a method to obtain these maxima and modify their respective values. The latter is done , in order to make the final task of thresholding more effective.

4.2 Obtaining a Detection Function

Time - block analysis In the introduction to the reduction process of onset detection algorithms, a need for a time-frequency analysis of the input signal was highlighted, in order to interpret the evolution of certain of its features through the analysis blocks.

The idea is to analyse the input $x(n)$ in successive analysis blocks of L samples, with a hop distance of h samples describing the jump between successive analysis blocks(Fig. 6).

The total number of analysis blocks is given as

$$\mathrm{NoB} = \left\lfloor \frac{N-L}{h} \right\rfloor \tag{14}$$

The floor function assures we do not exceed the limit of N samples of our $x(n)$ signal.

The m^{th} analysis block is given as

$$b_m(l) = x((m-1)h + l) \quad , \quad l = 1, 2, \ldots, L \tag{15}$$

Block analysis time resolution The hop distance h is set as a fraction of the block size L (L,$L/2$,$L/4$ etc) and determines the time-block analysis' time resolution, or in other words, how often we check for onsets in the time domain signal.

$$\mathrm{TR} = hT_{\mathrm{s}} \tag{16}$$

Setting the time resolution In the implementations of the algorithms towards obtaining a detection function, several values of hop distance h were chosen, according to the respective chosen block size L. Common values of h were set at 512 samples. A sampling frequency of 44.1kHz or sampling period of $2.27 \cdot 10^{-5}$ seconds led to a time resolution of $512 \cdot (2.27 \cdot 10^{-5}) \simeq 11.6$ ms. Even though when comparing to real piano music [2] and human ear time resolution(minimum of 10ms), this value might seem excessive, the fact that rapid changes in the transient state(which as seen in Fig. 5a only constitutes a small time interval proportionally to the whole note) indicate the onset, justifies this decision. On top of that, by choosing a seemingly small time resolution at this stage allows for more "freedom" in the process of low pass filtering the detection function while still being able to have practically reliant onset positioning.

[2] a *32nd* note at a tempo of 112 *4th* notes per minute (Very fast passage in *La Campanella* by ***Liszt***) constitutes a duration of 67 ms

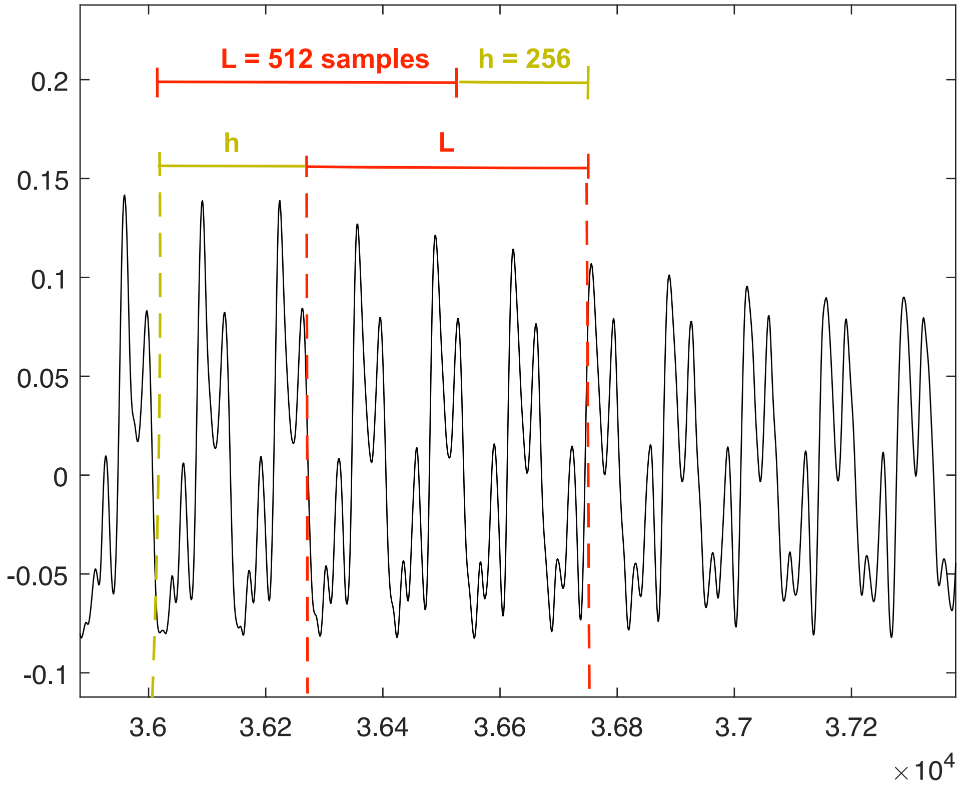
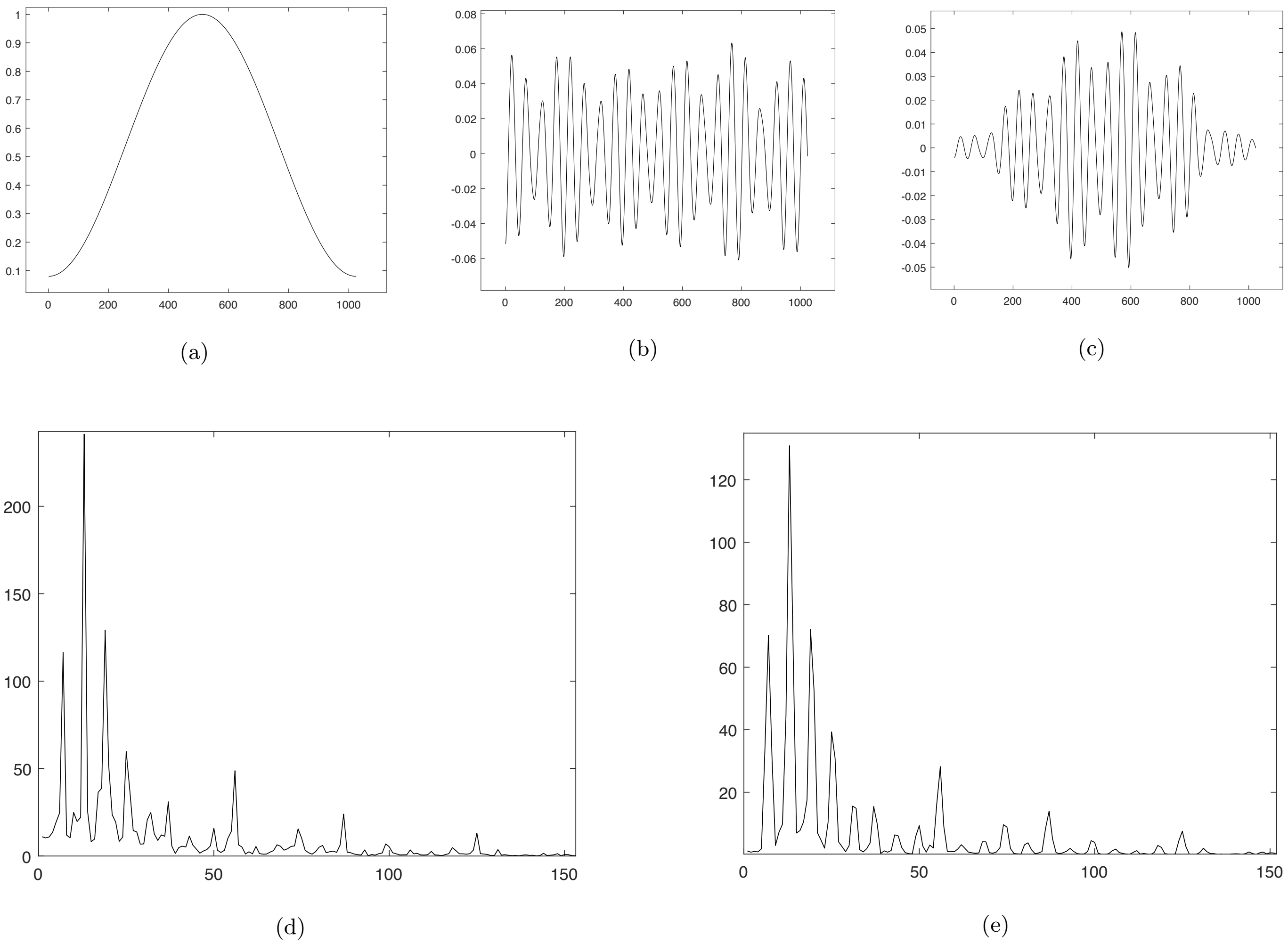


Figure 6: Time block analysis, with L = 512 and h = 256 samples

Frequency resolution Frequency resolution is inversely proportionate to the block size L, as seen in Eq.13. This suggests that when dealing with methods based on the dime domain, where frequency resolution is of no use, the block size can take lower values. On the other hand, in some frequency domain approaches, block sizes could be set to a bigger value if a clearer frequency representation is needed to increase the method's robustness. In these examples, in frequency domain based onset detection methods, a block size of $L = 1024$ was used, ultimately offering a frequency resolution of $F_s/1024 \simeq 43.06$Hz. It is enough of a resolution to follow frequency related events.

Spectral leakage In practical signal-sampling applications, one can obtain only a finite record of the signal. This finite sampling record results in a truncated waveform that has different spectral characteristics from the original continuous-time signal. These discontinuities produce leakage of spectral information, resulting in a discrete-time spectrum that is a smeared version of the original continuous-time spectrum. This phenomenon is known as spectral leakage. To reduce this on our sampled blocks, we multiply them with a hamming window of L size. This procedure minimizes the transition edges between successive blocks ultimately reducing spectral leakage [9].



(a)

(b)

(c)

(d)

(e)

Figure 7: The effect of the Hamming window(a), in the block's time signal(b,c) and spectrum(d,e)

Hamming window The Hamming window is obtained as

$$H(l) = \alpha - \beta \cos\left(\frac{2\pi l}{L}\right) \tag{17}$$

with $\alpha = 0.54$, $\beta = 1$ - $\alpha = 0.46$ and $l = 1, 2, \ldots, L$. The window is then multiplied with the time block.

$$b'_m(l) = b_m(l)H(l) \quad , \quad l = 1, 2, \ldots, L \tag{18}$$

The effect of the Hamming window in both time and frequency domain are shown in Fig. 7.

Results As part of this introduction to the several reduction methods examined, single piano notes are tested, more specifically the a3 note [3]. In the detection function comparison afterwards we will compare their performance on the more complex input signals of the C4-E4-D4 passage and that of the ***Chopin's*** *Nocturne*.

Time domain

4.2.1 Energy follower

First attempts in onset detection took advantage of the basic temporal effect of an onset event in an audio signal [10]. It is understood that during the transient the signal's amplitude increases, while on the other hand, during the note sustain(steady state) the respective amplitude steadily decreases. A basic idea to capitalize on this fundamental concept is to apply an "energy follower", which follows the changes of the signal's energy by adding the squared samples foe every block.

Algorithm Steps

1. In this case we opted for a block and hop size of 512 samples, resulting to a time resolution of 11.6 ms.
2. Continuing, for each analysis block we extract a measurement which refers to the local energy of the signal.

$$E(m) = \frac{1}{L}\sum_{l=1}^{L} b_m(l)^2 \tag{19}$$

   where L is the block size(512 samples), b_m is the m^{th} block.

3. We finally get our detection function by taking the rectified first derivative of $E(m)$, since we are interested in rise of energy.

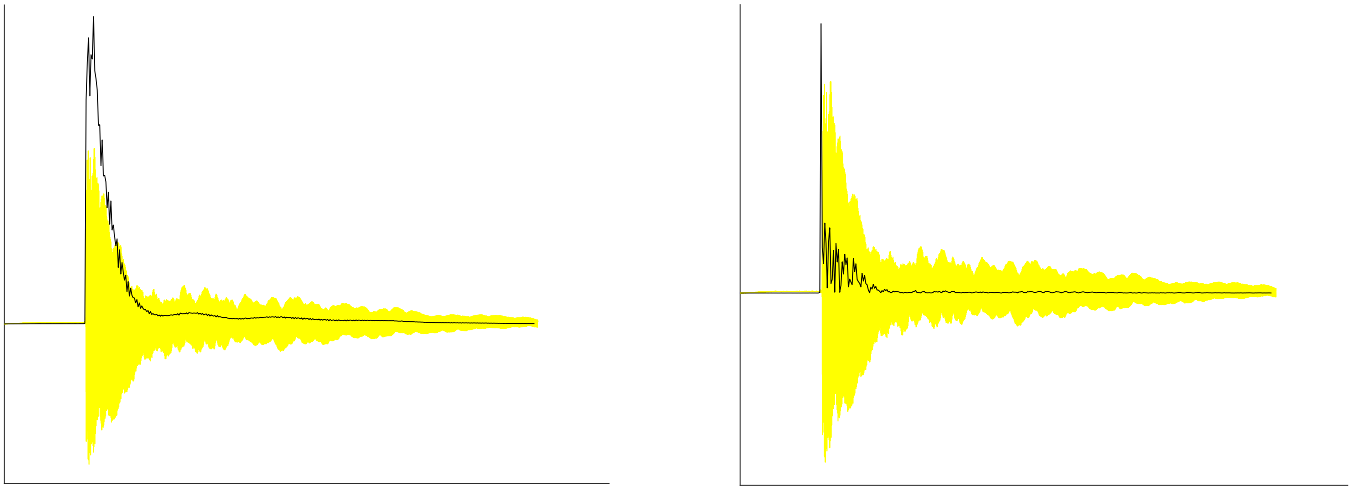
$$\mathrm{DF}(m') = \mathrm{H}(E(m'+1) - E(m')), \quad for \quad m' = 1, 2, \ldots, \mathrm{NoB} - 1 \tag{20}$$

   where

$$\mathrm{H}(y) = \frac{y + |y|}{2} \tag{21}$$

   is the rectifying process.

[3] All comparison figures to be shown, have the y axis scaled to offer a visible comparison. In the peak picking stages we will deal with normalized detection function outputs



(a)

(b)

Figure 8: (a) Block-by-block energy, (b) Rectified energy derivative - Detection function

4.2.2 Peak follower

An also rather straightforward and immediate approach proposed in [11], is based on the same idea as energy envelope follower, this time following maxima of successive blocks. In this case the envelope is calculated using a peak follower algorithm that mimics the behavior of an ideal capacitive voltage peak follower circuit.

Algorithm steps

1. The block and hop size are once again set at 512 samples.

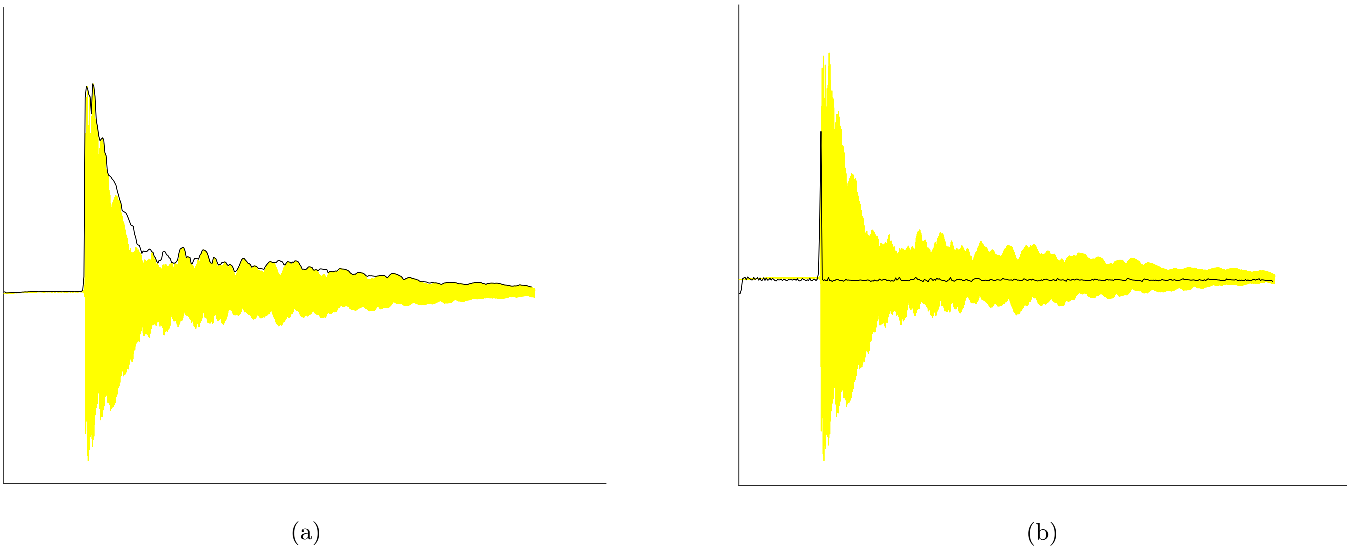
2. For every block we measure the peak (max) of its absolute value.

$$\text{maxima}(m) = \max\left(|b_m|\right) \tag{22}$$

3. At each block, the peak follower is updated to the new block value if that is higher, or a fixed proportion (close to 1) of the previous block peak follower value, so as to simulate exponential decay.

$$\text{PF}(m) = \max\{\text{maxima}(m), \text{PF}(m-1)\kappa_{\text{decay}}\} \tag{23}$$

, where $\kappa_{\text{decay}} = 0.99$ is the decay factor, which simulates exponential decay when the Peak Follower drops.



(a)

(b)

Figure 9: (a) Peak follower adjusted to simulate exponential decay, (b) Peak follower method's output

4. The detection function is finally obtained by calculating the ratio of successive peaks, as these where given by the peak follower.

$$\mathrm{DF}(n) = \frac{\mathrm{PF}(n)}{\mathrm{PF}(n-1)} \tag{24}$$

Comments

Following a detection or at the start of the scan, it is possible for the detector to falsely trigger on the first few samples, simply because the peak follower has been reset and may initially lie in an insignificant envelope trough. However in our 512 sample block implementation covers that dropback which could occur if a small block at the start of the scan had a very small value.

This procedure follows a similar feature to the energy follower. However in this case we follow a measure of the ratio between successive values to get the final form of our detection function. This method on one hand will produce sharper peaks but is more prone to oversharpening them when an onset happens against a quiet background while risking the opposite effect when an onset happens on top of a note sustain.

Frequency domain Since the frequency content of a signal provides a huge amount of information and processing capabilities, many onset detection techniques are based on spectral features of the input audio signal. These techniques described below, not only skip the need of preprocessing as described earlier(such as band predetection and separation) but also offer very good results when dealing with more complex inputs, such as polyphonic signals played by one or even more instruments or percussion onsets.

4.2.3 High Frequency Content

As far as frequency domain is concerned, onset events are typically described as a broadband event. That said, we can suppose that since most of the energy of the signal is gathered in lower frequencies, we can search for onset behavior in higher frequencies. This can be implemented by weighing the spectrum accordingly. Here, the High Frequency Content(HFC) algorithm as proposed by Masri in [11] is presented.

Algorithm steps

1. We again analyze the signal accordingly. We obtain blocks of 1024 samples with a hop of 512 samples. The blocks are multiplied by the hamming window as shown in Eq.(18) and graphically in Fig. 7c.

2. For every time analysis block, constructed in step 1, we obtain the FFT of our block B_m, as shown in Eq.(6).

3. We now construct the "ramp" signal as proposed by Masri in the HFC (High-Frequency content) function.

$$R(k) = |k| \tag{25}$$

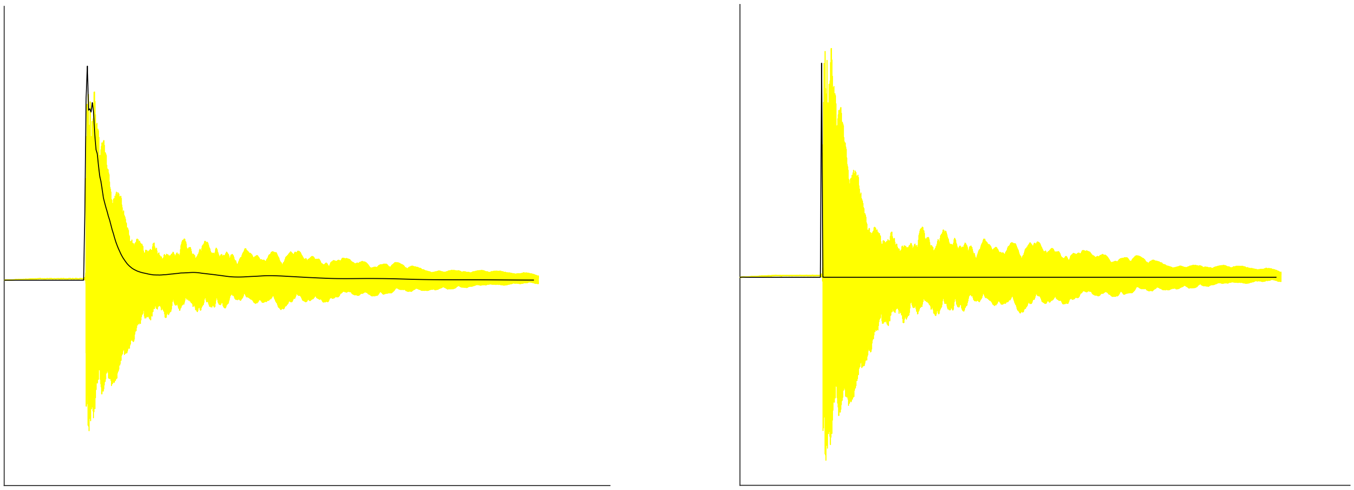
   k referring to the bins of the respective FFT spectrum. This ramp linearly weights the spectrum towards high frequencies.

4. We calculate the High - Frequency energy content by summing the weighted energy of every bin. To avoid dc component interference and alteration of our metric , we neglect the first bin.

$$\mathrm{HFE}(m) = \sum_{k=2}^{L/2} |B_m(k)|^2 R(k) \tag{26}$$

5. In order to once again come up with detection function easier to undergo a successful peak picking procedure , we normalize this intermediate detection function.

$$\mathrm{DF}(m) = \frac{\mathrm{HFE}(m)}{\mathrm{HFE}(m-1)} \quad \cdot \quad \frac{\mathrm{HFE}(m)}{E(m)} \tag{27}$$



(a)

(b)

Figure 10: (a) High frequency energy, (b) High frequency content normalized output

where HFE is the High Frequency energy as calculated in Eq.(26) and $E(m)$ refers to the energy of the block

$$E(m) = \sum_{k=2}^{L/2} |B_m(k)|^2 \tag{28}$$

The first part of the detection function product, suggests that we will enhance instances where the absolute HFE of the current block is bigger than the respective value of the previous block. However this ratio could present significant values for small absolute value differences. Hence, comes the second part of the product , which balances the detection function values to instances where the ratio of HFE to overall energy is larger. For avoidance of *dividing by zero* we set lowest possible values for $E(n)$ and HFE(n) to 1.

4.2.4 Spectral difference

As we have seen so far in the previously presented onset detection algorithms, after the time-block analysis of the input, we measure a quantity related to every such block and consequently end up with our novelty function. In this next case however, we will look at a method which takes some information regarding subsequent analysis' blocks and compares them in order to come up with a proper onset detection scheme. An approach based on the changes of spectra in respect of time is used, in order to formulate the detection function as the "distance" between two subsequent spectra, as the latter are treated as points in the N-dimensional space. Several metrics are proposed to calculate this distance(otherwise referred to as spectral flux), like Masri's proposal of calculating the L1 - Norm of the difference between two subsequent analysis' blocks' spectra [11]. Duxbury proposes the L2 - norm of the difference [12], combined with the High Frequency Content as described earlier after preprocessing the input signal and studying changes in five different subbands. In this paper, we tested a method that measures the L2 - norm of the difference. In this case too, the rectified difference is taken into account, meaning we once again keep only the effect of frequencies that show an increase in their energy between two analysis blocks. As we discussed earlier, this procedure emphasizes onsets rather than offsets in the music signal.

Algorithm Steps

1. We start once again by setting the size of the time analysis block this time to $L = 1024$ and the hop distance between adjacent blocks to $h = 512$ samples. The blocks are once again multiplied with a *Hamming* window.

2. We now move on to calculating our metrics. First up, for each of the first 512 bins(positive frequencies) of the two - under examination - blocks we extract their spectral bin difference.

$$\mathrm{BD}_m(k) = \mathrm{H}(|B_m(k)| - |B_{m-1}(k)|) \tag{29}$$

   we once again keep the rectified bin difference(Eq.(21)) since we are interested in rises of the respective frequency bin amplitudes.

3. After obtaining these bin differences for every bin, we extract our detection function for every block of reference by summing up the squared bin differences.

$$\mathrm{DF}(m) = \sum_{k=2}^{L/2} \mathrm{BD}_m^2(k) \tag{30}$$

   where once again we ignore the first bin.

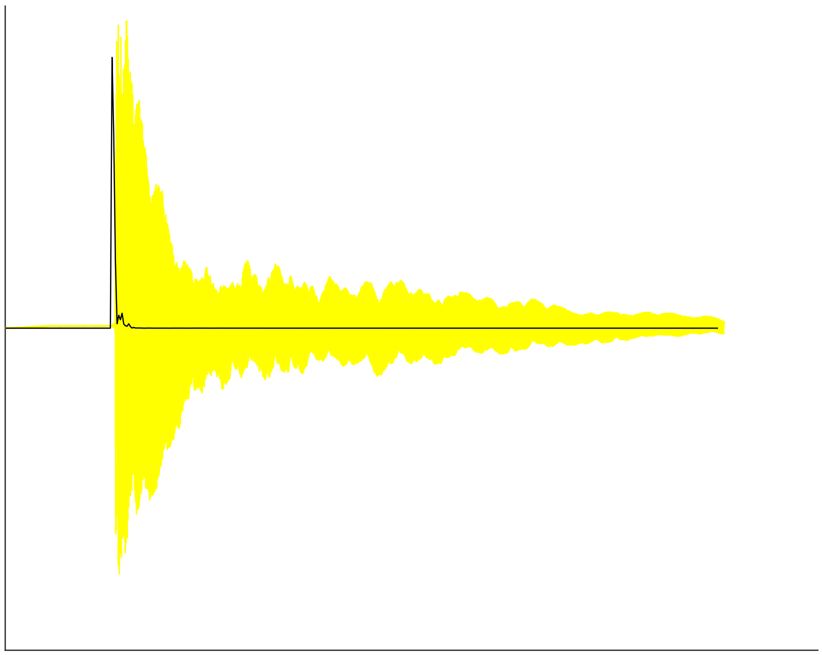


Figure 11: Spectral difference output for note A3

4.2.5 Weighted Phase Deviation

As far as frequency content information is concerned, we have so far examined algorithms that only use the amplitude information of the Fast-Fourier transform output, ignoring phase information. As expected, this phase information has a strong correlation with the time evolution of the signal. The angle part of the FFT output associates every frequency bin with the phase shift calculated at the start of the time analysis block and its value is obtained as shown in Eq.(8).

The idea The idea of onset detection capitalizing on phase information, lies in a basic difference between the transient and steady state of the signal. As we discussed in the introduction of this paper, a steady state is described as the state of the signal that can be analyzed as a finite summation of sine waves, in contrast to the transient state which can not be sufficiently modeled in such a manner. However, in music signal processing these terms are not so strictly defined. Instead we define the steady state as the part of the note or notes that can be reconstructed by the tonal part and only that, even though in theory every note has an infinite amount of harmonics. During the steady state, the phase deviation between subsequent analysis blocks for any frequency bin, should remain constant. This is true as we expect that instantaneous frequency of a bin should remain approximately constant over adjacent windows.

Wrapped-Unwrapped phases One thing to keep in mind before we examine the phase evolution of a single bin in respect of time, is that Matlab by default "wraps" the values between -π and π. The *unwrap* function basically "smoothens" the wrapped version. It corrects the radian phase angles in a vector by adding multiples of $\pm 2\pi$ when absolute jumps between consecutive elements are greater than or equal to the default jump tolerance of π radians. This procedure can be seen graphically in Fig.s 13a and 13b.

Instantaneous frequency and phase deviation We define the instantaneous frequency of every bin as

$$F_k(m) = \left(\frac{\widetilde{\phi}_k(m) - \widetilde{\phi}_k(m-1)}{2\pi h} \right) F_{\mathrm{s}} \tag{31}$$

where $\widetilde{\phi}_k(m)$ refers to the unwrapped phases of the blocks, h is the hop size and F_{s} is the sampling frequency. Since this instantaneous frequency should remain constant over adjacent windows as suggested in the idea, it is the numerator that should remain approximately constant.

$$\widetilde{\phi}_k(m) - \widetilde{\phi}_k(m-1) \simeq \widetilde{\phi}_k(m-1) - \widetilde{\phi}_k(m-2) \tag{32}$$

Equivalently, the phase deviation can be defined as the second difference of the phase

$$\Delta\phi_k(m) = \widetilde{\phi}_k(m) - 2\widetilde{\phi}_k(m-1) + \widetilde{\phi}_k(m-2) \simeq 0 \tag{33}$$

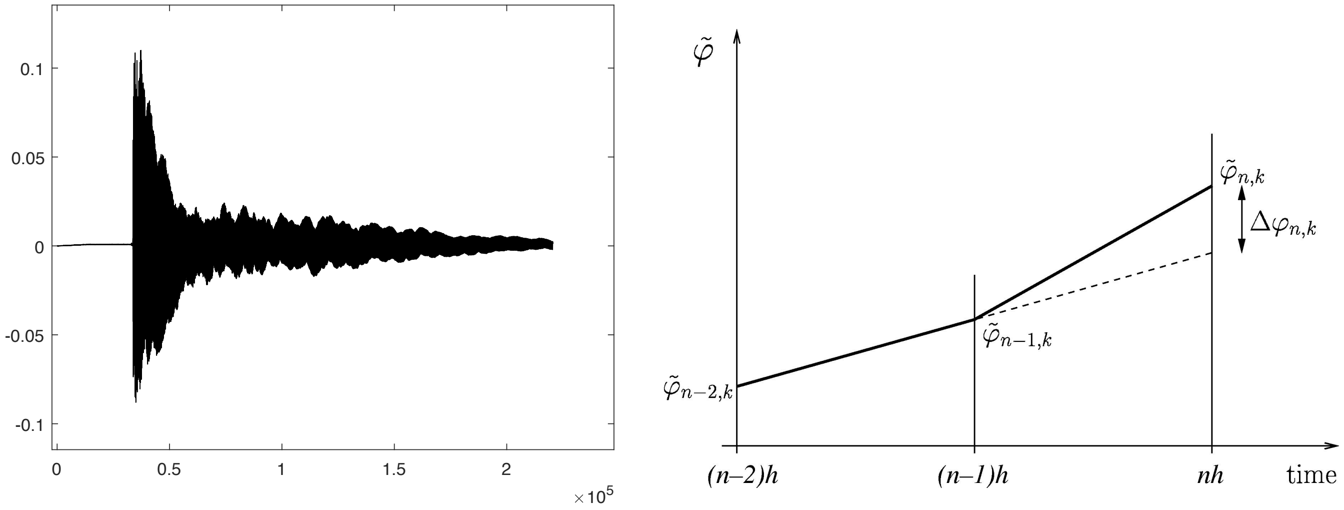
On the other hand when an onset event occurs(transient state), an "unexpected" event will alter this deviation. It is this property, on which phase onset detection algorithms are based. In [13], Bello proposes a method that analyzes the instantaneous distribution (in the sense of a probability distribution or histogram) of phase deviations across the frequency domain. In [14] the mean absolute phase deviation is used. Here we use a measure of the weighted phase deviation.

Algorithm Steps

1. Similarly to previous algorithms, the first step consists of the analysis of the input signal in time blocks of $L = 1024$ and hop size of $h = 512$ samples. This is once again to obtain a somewhat better frequency resolution. The analysis blocks are once again multiplied by a hamming window.

2. For every bin we calculate its phase deviation $\Delta\phi_k$ as shown in Eq.(33)

3. Angle values are computed for every bin as the inverse tangent of the imaginary part divided by the real part of the complex FFT output(Eq.(8)). This means that even bins with very low contribution to the signal (very small



(a)

(b)

Figure 12: (a)Original signal(A3), (b)Phase deviation scheme at an onset point

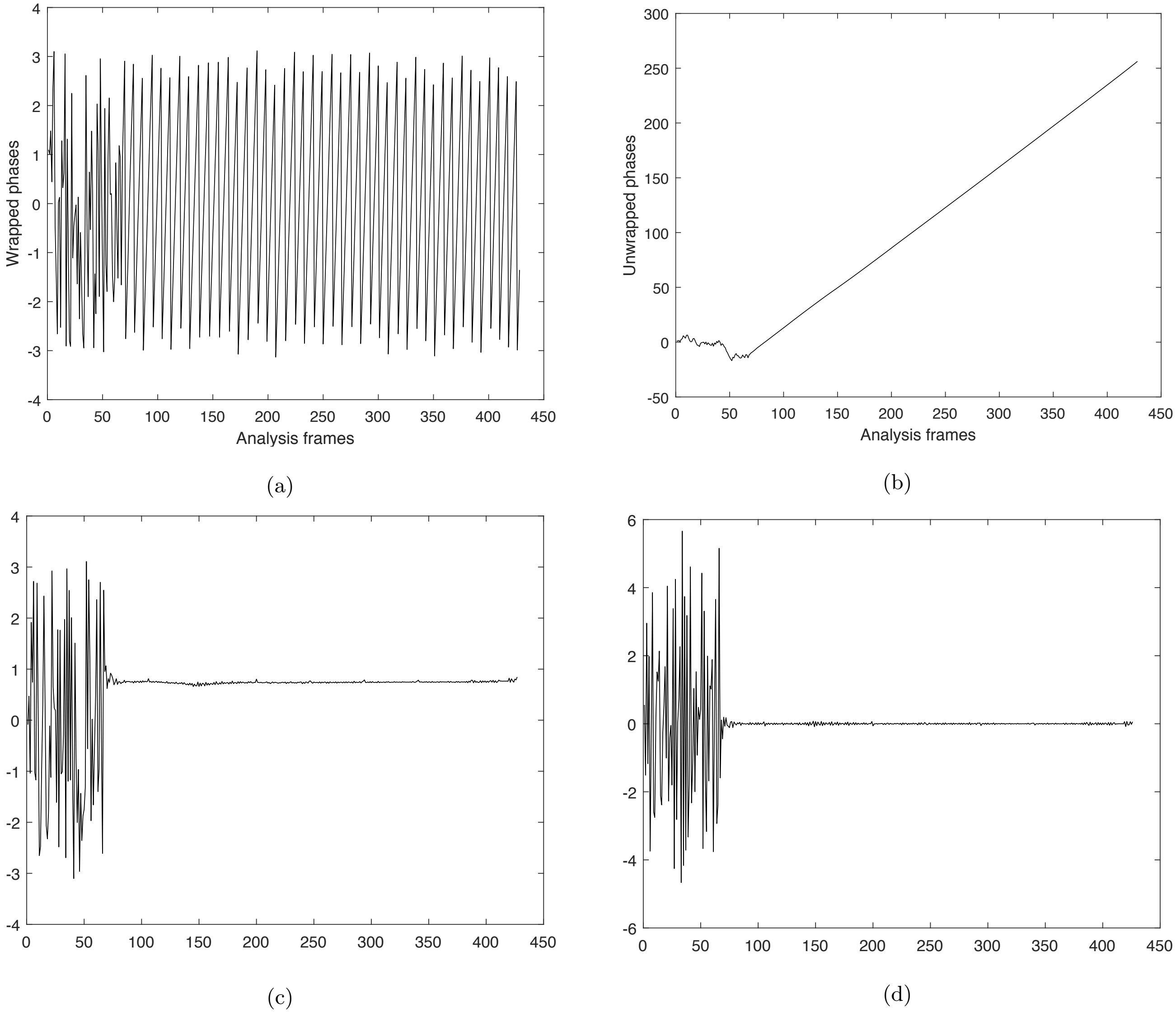
magnitude) can still project a significant angle value, and ultimately provide "noisy" information regarding the bin's phase deviation. Thats why we propose a weighted phase deviation extraction, meaning that we weigh each bin's phase deviation by multiplying with the bin's magnitude at the block of reference. This way we minimize the noise components introduced in our detection function by these "unimportant" bins.

$$\mathrm{WPD}_k(m) = \Delta\phi_k(m)|B_k(m)| \tag{34}$$

, where $\mathrm{WPD}_k(m)$ is the weighted phase deviation of bin k, computed as the product of $\Delta\phi_k(m)$ and the magnitude of that bin, all referring to analysis block b_m.

4. We finally obtain our detection function by measuring the sum of phase deviations for every analysis block f_m.

$$\mathrm{DF}(m) = \sum_{k=2}^{L/2} \mathrm{WPD}_k(m) \tag{35}$$



(a)

(b)

(c)

(d)

Figure 13: Wrapped (a) and unwrapped (b) phase for bin 83. First(c) and second(d) unwrapped phase derivatives for bin 83

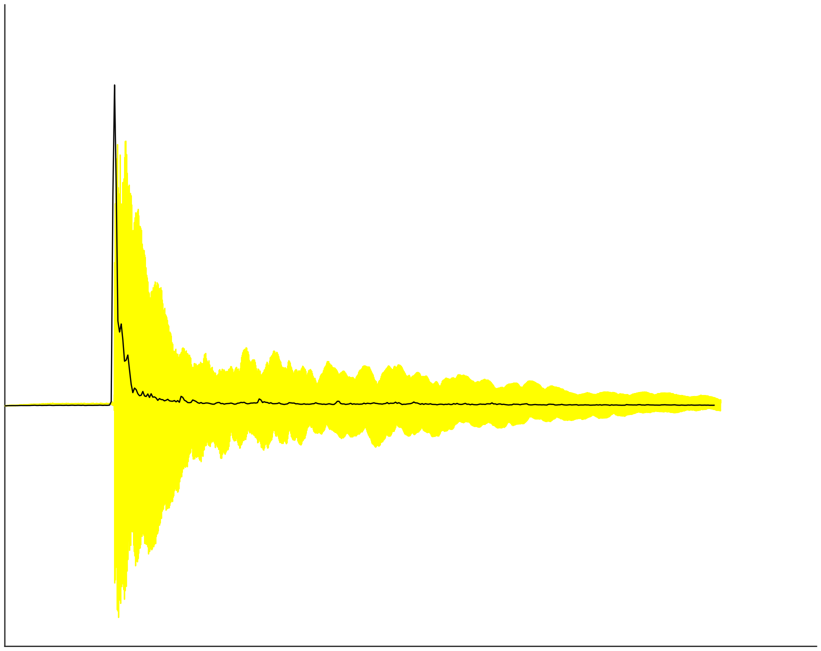


Figure 14: Weighted phase deviation output for note A3

4.2.6 Phasor stationarity

The last case of onset detection we are going to examine in the scope of this paper, is a method proposed in [15] and takes place in the complex domain. The idea is to combine energy and phase information as extracted in the methods we discussed earlier. A straight forward approach would suggest a plain multiplication of spread measures of the two -statistically similar- functions which would indeed produce sharper for the detected onsets. However, the approach we will study, energy and phase information is simultaneously analyzed.

In Sections 4.2.4 and 4.2.5, onsets were located by inspecting changes in either magnitude or phase. In this method however, we will simultaneously take these changes into account, by predicting the to be examined values of analysis' blocks. During the steady state of a music signal (no transient present), we assume that frequency and amplitude values remain constant. So by estimating the FFT phasor for a bin at block b_m, through information extracted by its phasors at previous blocks(b_{m-1} ,b_{m-2}) and calculating its *Euclidean* distance from the actual phasor (Fig. 15a), we expect this distance to approach 0 during steady state and show peaks during onset events, as both magnitude and phase will deviate as seen in previous sections.

Algorithm steps

1. We once again choose a block size of $L = 1024$ with a hop size of $h = 512$ samples.

2. We denote the phasor for the k^{th} bin as

$$B_k(m) = R_k(m)e^{j\phi_k(m)} \tag{36}$$

and the estimation of $B_k(m)$ based on previous blocks as

$$\hat{B}_k(m) = \hat{R}_k(m)e^{j\hat{\phi}_k(m)} \tag{37}$$

Magnitude estimation The estimated magnitude is considered to remain constant between two adjacent blocks. This is a good estimate even though in practice a piano note's sustain drops in respect of time. So we assign it the magnitude of the previous block's FFT.

$$\hat{R}_k(m) = |B_k(m-1)| \tag{38}$$

Phase estimation The estimated unwrapped phase $\tilde{\hat{\phi}}_k(m)$ can be measured as

$$\tilde{\hat{\phi}}_k(m) = 2\tilde{\hat{\phi}}_k(m-1) + \tilde{\hat{\phi}}_k(m-2) \tag{39}$$

However, since $\phi_k(m)$ in Eq.(39) refers to wrapped phase values, so does $\hat{\phi}_k(m)$. So we wrap the unwrapped phase$\tilde{\hat{\phi}}_k(m)$ between -π and π using the *princarg* function.

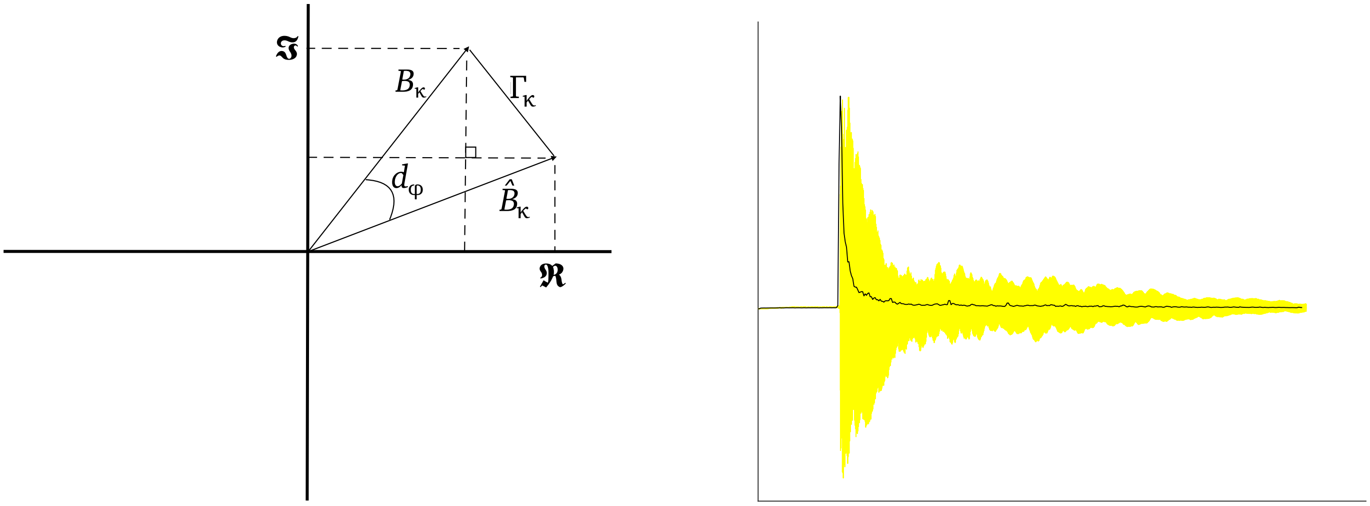
$$\hat{\phi}_k(m) = \text{princarg}(\tilde{\hat{\phi}}_k(m)) \tag{40}$$

3. We now measure the *Euclidean* distance between $B_k(m)$ and $\hat{B}_k(m)$. This gives us a measure of bin stationarity through blocks.

$$\Gamma_k(m) = \left\{ \left[\Re(\hat{F}_k(m)) - \Re(F_k(m)) \right]^2 + \left[\Im(\hat{F}_k(m)) - \Im(F_k(m)) \right]^2 \right\}^{1/2} \tag{41}$$

4. We get the final form of our detection function in the form of block stationarity, by summing the bin stationarities calculated in step 3 across all the bins.

$$\text{DF}(m) = \sum_{k=1}^{L/2} \Gamma_k(m) \tag{42}$$



(a)

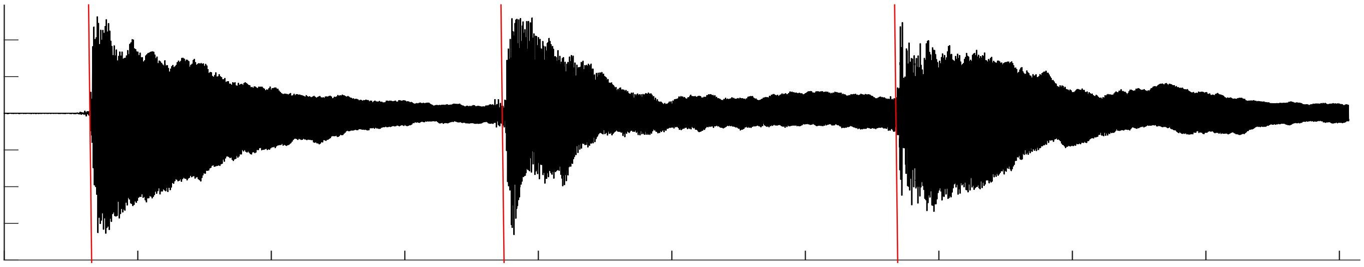
(b)

Figure 15: (a)Phasor stationarity scheme , (b)Detection function output

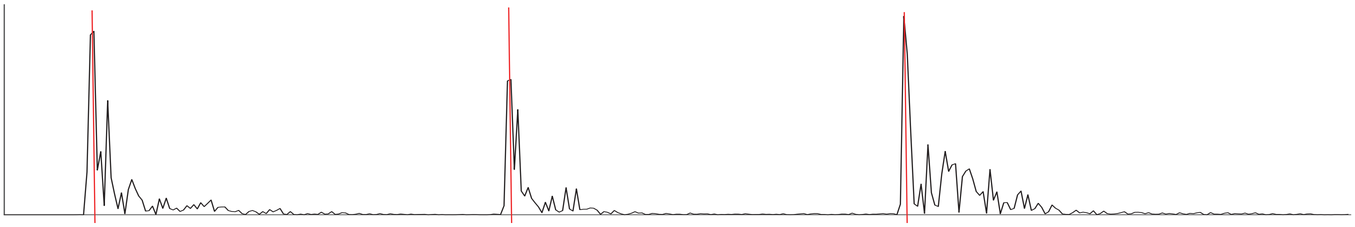
4.3 Comparison between detection functions

C4-E4-D4 This piano passage consists of the three notes mentioned. The recording starts with just the background noise (environment, my breath etc.). The notes are played at similar intensity level, more specifically *fortissimo*. The consistently loud playing ensures a similar attack period of the notes, and increased "percusiveness" as explained in the introduction to Section 4.

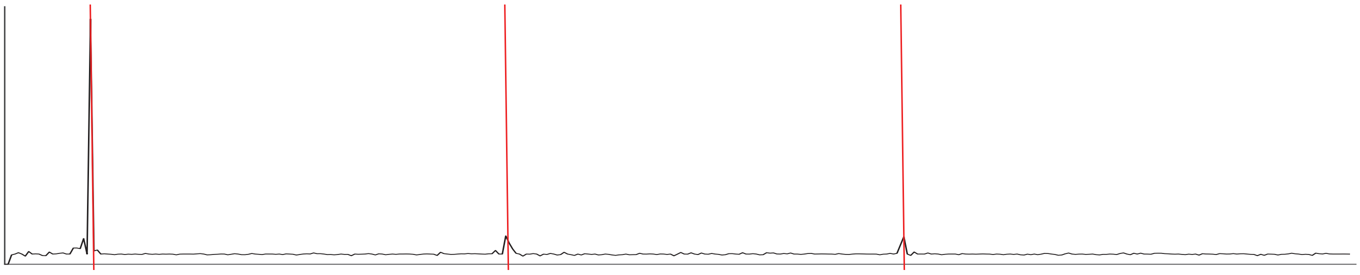
C4 - E4 - D4 excerpt



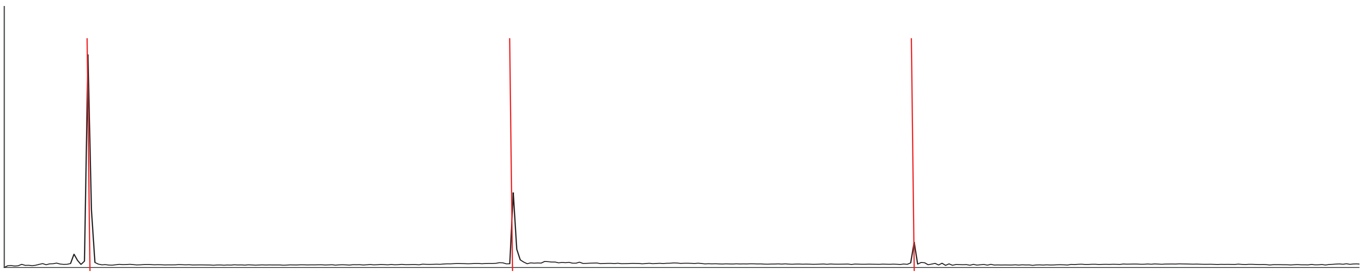
Energy Follower

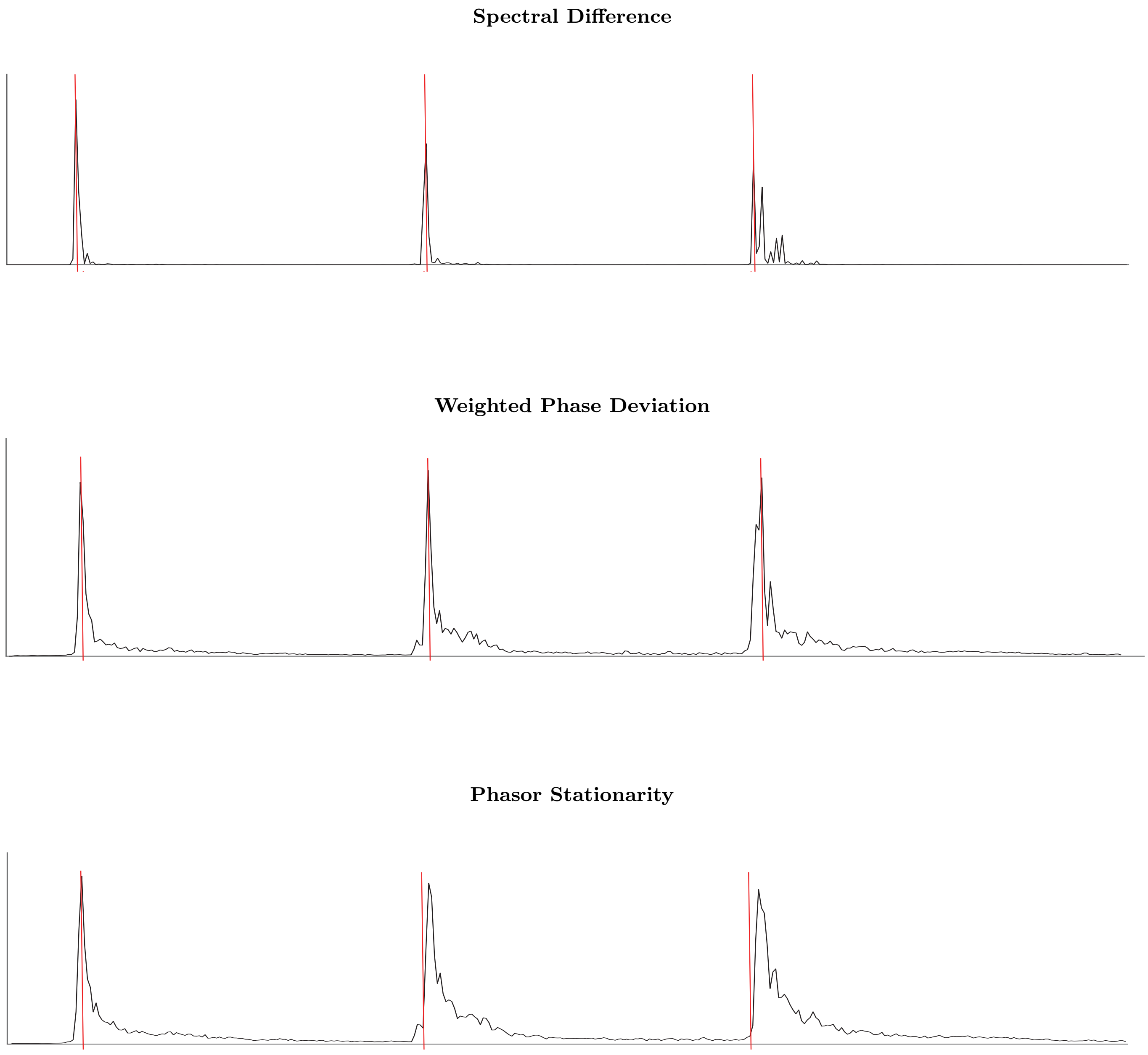


Peak Follower



High Frequency Content





Spectral Difference

Weighted Phase Deviation

Phasor Stationarity

Figure 16: Detection Functions outputs for our input piece C4 - E4 - D4

Nocturne This example probably constitutes the most complex single note passage we could come up with for several reasons.

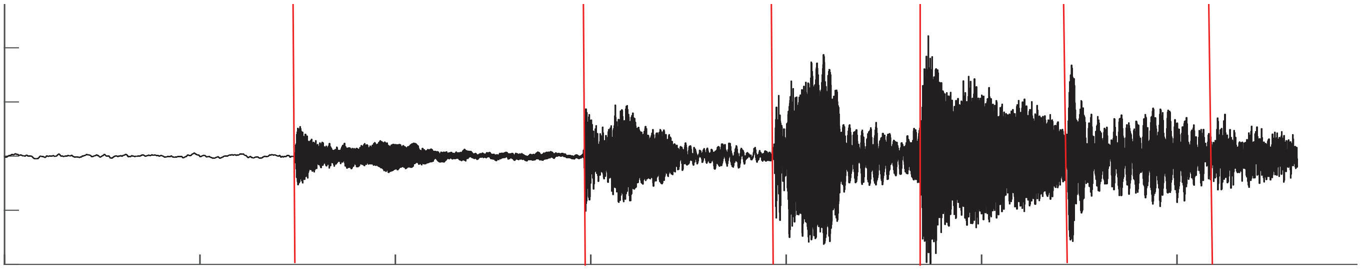
1. Even though it is just 6 notes, the diversity in dynamics is evident. This extreme interpretation is a distinct characteristic in first introduced in the Romantic period, where this *Nocturne* belongs.

2. As also seen in the resentation of this piece, the specific piano used for the recording has a characteristic wavy note sustain in some notes.

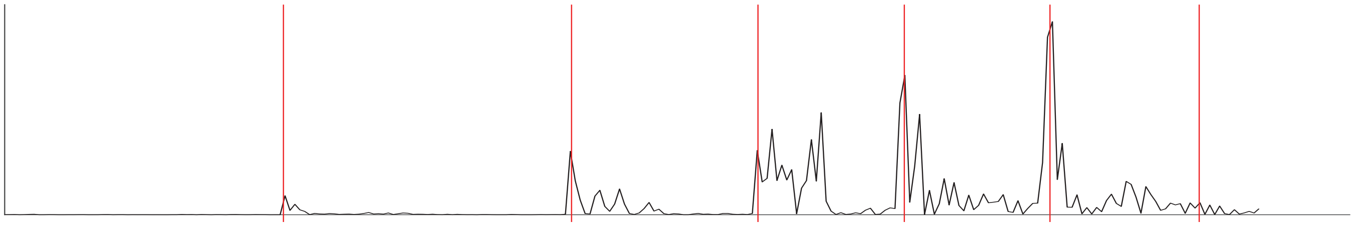
3. All these features are further enhanced leading to a complex signal by the over use of sustain pedal[4], through the whole duration of this excerpt.

4. Lastly, the note sequence B-flat5 - C6 - D-flat6 - B-flat5 - C6 - A-flat5 offers repeating notes, and harmonic notes(i.e. notes with common harmonics), thus potentially confusing some spectral, even temporal, measurements.

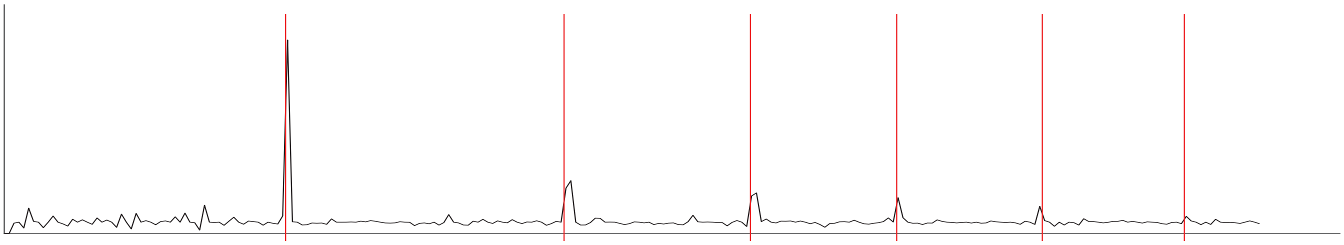
Nocturne excerpt



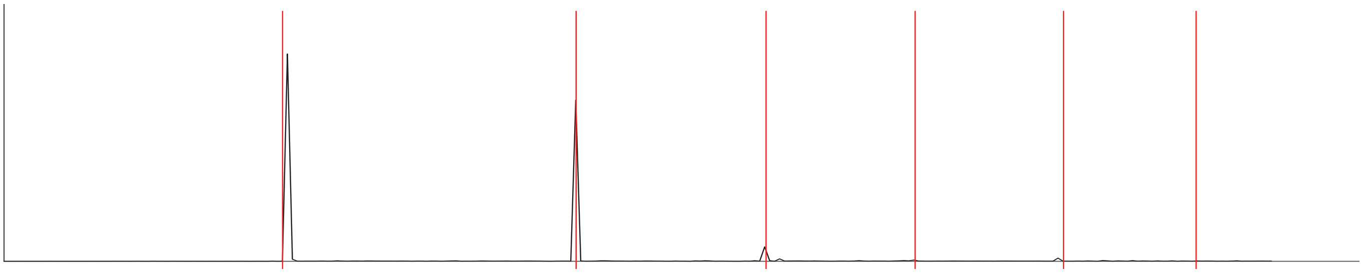
Energy Follower

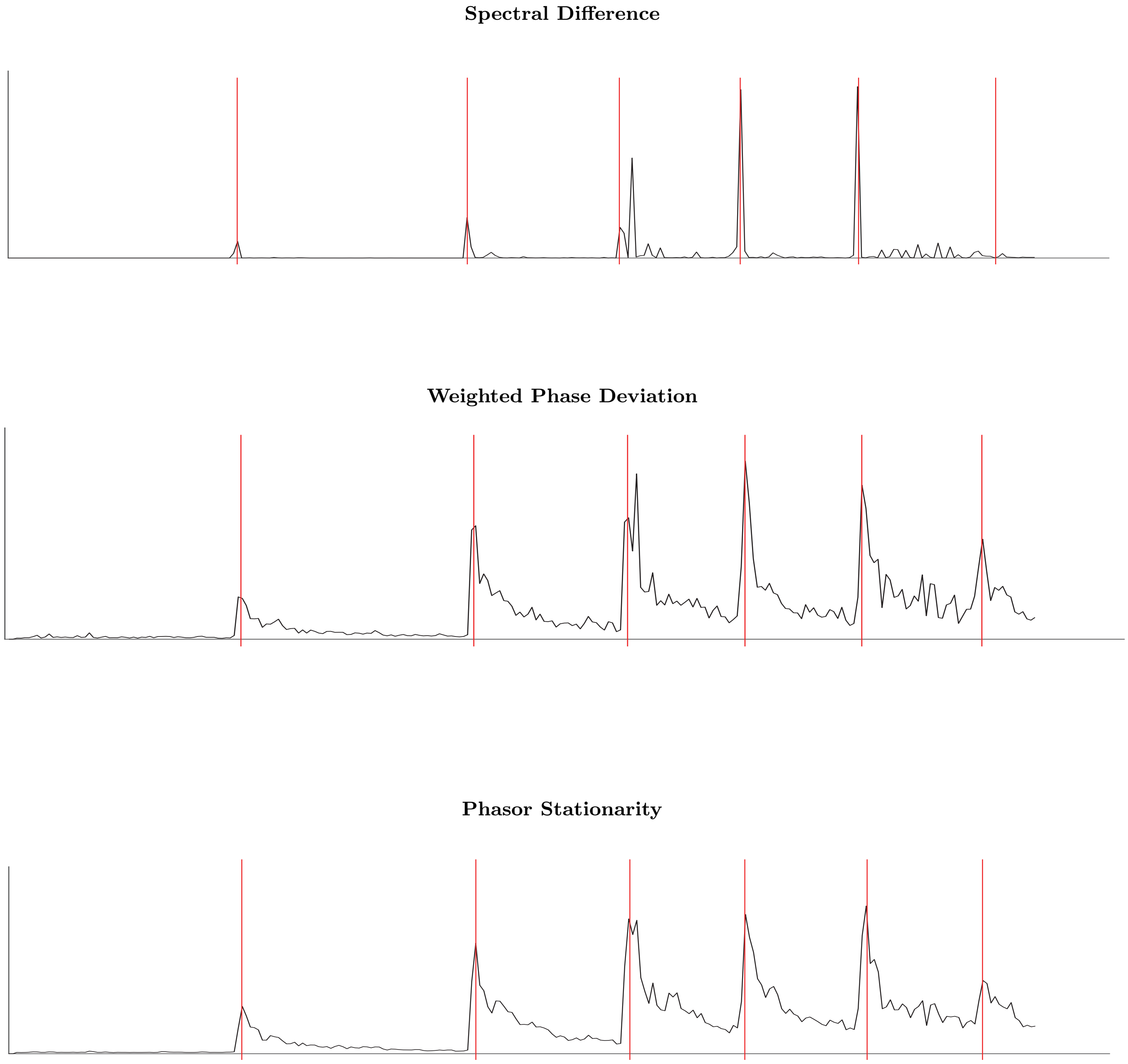


Peak Follower



High Frequency Content





Spectral Difference

Weighted Phase Deviation

Phasor Stationarity

Figure 17: Detection Functions outputs for our input piece Nocturne op.9 no.1

Peak evaluation

C4-E4-D4 : As we can see in Fig. 16 all methods have successfully shown peaks close to the onset events, shown with a red line in the original signal. In the case of the energy follower, the time derivative of the energy used to follow onsets, works fine in this clear note case but still shows peaks relatively high caused by the vibrato of the note. The normalization using ratios for the peak follower and high frequency content, improves the false positive detection, but tends to overemphasize onsets occurred against a quiet background(in this case the first note). Spectral flux, while still being somewhat dependent on the actual energy of the signal at time analysis blocks, offers distinct peaks, with however small peaks following vibrato(soundboard resonance) phenomena. These minor peaks are not present in the phase deviation as expected. The phasor stationarity method finally, also works as expected, by somewhat combining the spectral flux representation with that of the phase deviation.

Nocturne For the methods performed to the more complex case of the *Nocturne* (Fig. 17), first up we notice the the last onset, is not easily visible by looking at the signal. This alerts us, as to how the basic approach of energy methods will perform. Indeed, besides the energy follower with its already mentioned false positive peaks, the peak follower and high frequency content, show relatively low peaks for notes 3,4 and 5 and fail to detect the 6th. The reason why High frequency content method does not work as expected, is the very low "percusiveness" of onsets, which is a typical finding in *Romantic* interpretations. The spectral difference method also seems to suffer in the detection of the 6th onset. The fast that this note is played fairly *piano*, with heavy sustain pedal effect accumulated throughout the so-far played excerpt, combined with the fact the the 5th(B-flat5) with the 6th(G-flat5) are heavily harmonically related, highlights the method's drawback at such cases. The last two methods however once again manage to show a more distinct peak at the 6th onset, yielding them more efficient for our two excerpts.

[4] The **Sustain pedal** keeps the dumper lifted even when the piano key is released. This allows for all the harmonics and resonances to keep on sounding over other note events, leading to a more "round" or "full" sound.

4.4 Thresholding the detection functions

Smoothing The purpose of smoothing our detection function $\mathrm{DF}(m)$ is to is to facilitate the tasks of thresholding and peak-picking by increasing the uniformity and consistency of event-related features in the detection function, ideally transforming them into isolated, easily detectable local maxima.

We obtain our smooth $\mathrm{DF}(m)$ by sliding a rectangular window of length W samples over it.

$$\hat{\mathrm{DF}}(m) = \frac{1}{R}\sum_{w=1}^{W} \mathrm{DF}(m+r) \tag{43}$$

4.4.1 Fixed thresholding

This pretty trivial approach, suggests that onset periods $O(m))$ "exist" when the smooth detection function $\hat{\mathrm{DF}}(m)$ is higher than a certain positive constant T, referring to the fixed threshold.

$$O(m) = \hat{\mathrm{DF}}(m) - T \quad , \quad \forall m \in \{m : \hat{\mathrm{DF}}(m) > T(m)\} \tag{44}$$

and

$$O(m) = 0 \quad , \quad elsewhere$$

where $\hat{\mathrm{DF}}$ is our smooth detection function, m referring to the analysis block and T is the fixed threshold.

Even though the implementation of such a threshold is fairly simple and easy to obtain, the results it returns for cases of real music signals like the ones we are studying, are not so satisfactory. Mainly, diversity of peak values caused by complexity in music interpretation or other characteristics, contributes to erroneous results of a fixed threshold.

Fixed thresholding could be used for simple signals like our *C4-E4-D4*, for some methods like the energy follower. Moreover, the normalization performed in the detection function of the peak follower and the high frequency content methods, since offering steeper local peaks, might allow the use of a delicately valued fixed threshold. Still, however, the implication following the use of a fixed threshold, led us to seeking better alternatives.

4.4.2 Adaptive thresholding

In general an adaptive threshold, instead of having a fixed value, follows $\hat{\mathrm{DF}}(m)$ and takes values based on local events [7].

Adaptive threshold using local median More specifically, in the implementation of our onset detection algorithms we opted for an adaptive threshold constructed using a *moving-median filter* derived out of the smooth $\hat{\mathrm{DF}}(m)$. Such an adaptive threshold is obtained as shown below.

$$T'(m) = \lambda \cdot \mathrm{median}\{|\hat{\mathrm{DF}}(m-K)|, \ldots, |\hat{\mathrm{DF}}(m+K)|\} \tag{45}$$

where λ is a positive constant,scaling the median, in our case $\lambda = 1$ and median returns the median value of a window of length $2K+1$ sliding through our smooth detection function $\hat{\mathrm{DF}}(m)$.

We now fine tune the adaptive threshold by raising it for a positive constant T .

$$T''(m) = T + T'(m) \tag{46}$$

where

$$T = a \cdot \delta, \tag{47}$$

and

$$\delta = \frac{1}{M}\sum_{m=1}^{M} |\hat{\mathrm{DF}}(m) - T'(m)| \tag{48}$$

i.e. the mean absolute deviation of $\hat{\mathrm{DF}}(m) - T'(m)$.

Similarly to the fixed threshold, the onset periods $O(m)$ are found as

$$O(m) = \hat{\mathrm{DF}}(m) - T''(m) \quad , \quad \forall m \in \{m : \hat{\mathrm{DF}}(m) > T''(m)\} \tag{49}$$

and

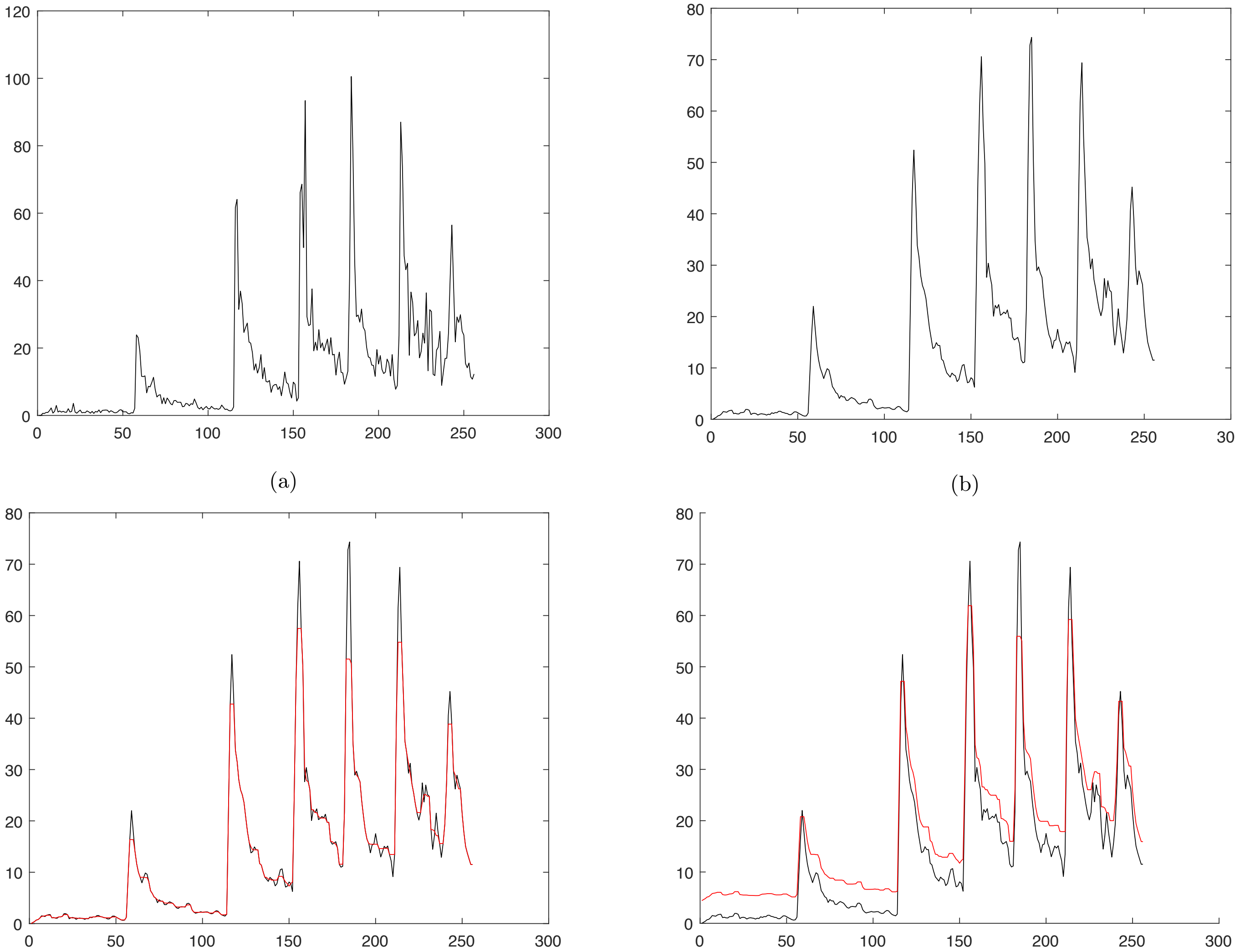
$$O(m) = 0 \quad , \quad elsewhere \tag{50}$$

Thresholding efficiency It is clear in Fig. 18d that the results of the thresholding depend a lot on the fine tuning of the adaptive threshold $T''(m)$ in respect of λ and δ, for every method. However certain methods with fixed appropriate values of λ and δ behaved well for both our inputs *C4-E4-D4* and *Nocturne*.

4.5 Peak picking

After applying our adaptive threshold and obtaining the onset periods $O(m)$ as shown in Eq.(49) and Eq.(50), we decide upon the final onset events as the local maxima of $O(m)$

$$\hat{O}(m) = 1 \quad , \quad \forall m \in \{m : O(m-1) < (O(m) < O(m+1))\} \tag{51}$$



(a)

(b)

(c)

(d)

Figure 18: Original detection function $DF(m)$ for the phase deviation method(a), smooth detection function $\hat{DF}(m)$ (b), adaptive threshold $T'(m)$ (c) and raised adaptive threshold $T''(m)$ (d) both denoted as the red line.

.

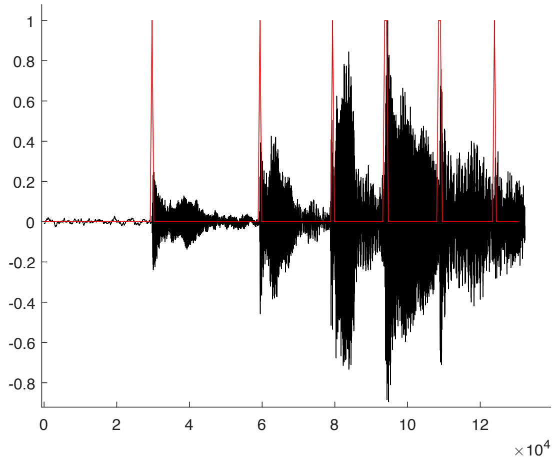


Figure 19: $\hat{O}'(m)$(red line) shows successful detection of onsets in the *Nocturne* excerpt, using the method of phase deviation. Smoothing rectangular window of $R = 3$ and adaptive threshold of $K = 2$ and $a = 6$, were used for detection.

and

$$\hat{O}(m) = 0 \quad , \quad elsewhere$$

with $\hat{O}(m)$ showing the decided onset events.

4.6 Time block accuracy

Before moving on to finding the note whose onset was detected, we must define the time block accuracy of the peaks in $\hat{O}(m)$. Time block accuracy refers to which block shows the onset event for every method explained.

Energy follower, Peak follower For these two methods since their detection functions relies on energy measures of the time signal will return a detection block corresponding to the end of the attack,i.e. the block with highest energy or highest maximum respectively (Fig.s 8b and 9b). However, the onset event, most probably started a few blocks before the detected block. This suggests that these two methods slightly suffer in accurately spotting the onset.

High frequency content,Spectral difference,Weighted phase deviation, Phasor stationarity For these methods however, time block accuracy is improved, as the parameters measured refer to blocks closer the actual onset block.

5 Note Detection

After successfully identifying note onsets, we move on to methods of finding which piano note was played. Similarly to onset detection methods, many note detection methods are based either in time or frequency domain representations of the note.

A part of $x(n)$ after the onset event is selected such as to be able to trace changes even in the lowest frequencies, i.e. bigger wavelengths.

Finding the wanted excerpt length As we will see the length of the excerpt changes depending in the method we use to find the note.

Time domain methods Since the lowest piano note's fundamental frequency is $P(1) = 27.5Hz$ as shown in Table 1, the biggest period for any piano note is

$$T_{\max} = \frac{1}{P(1)} = 3.6 \cdot 10^{-2} \quad \text{seconds} \tag{52}$$

So for a sampling period of $T_{\text{s}} = 1/F_{\text{s}} \simeq 2.27 \cdot 10^{-5}$ seconds we get the minimum length of our excerpt as

$$N_{\min} = \left\lceil \frac{T_{\max}}{T_{\text{s}}} \right\rceil = 1604 \quad \text{samples} \tag{53}$$

Frequency domain methods However, the $N_{\min}$ also determines the frequency resolution for our FFT as we saw in Eq.(13). The smallest difference in fundamental frequencies as shown in Table 1, is the difference between notes A0 and A#0.

$$\Delta F_{\min} = P(2) - P(1) = 1.64\text{Hz} \tag{54}$$

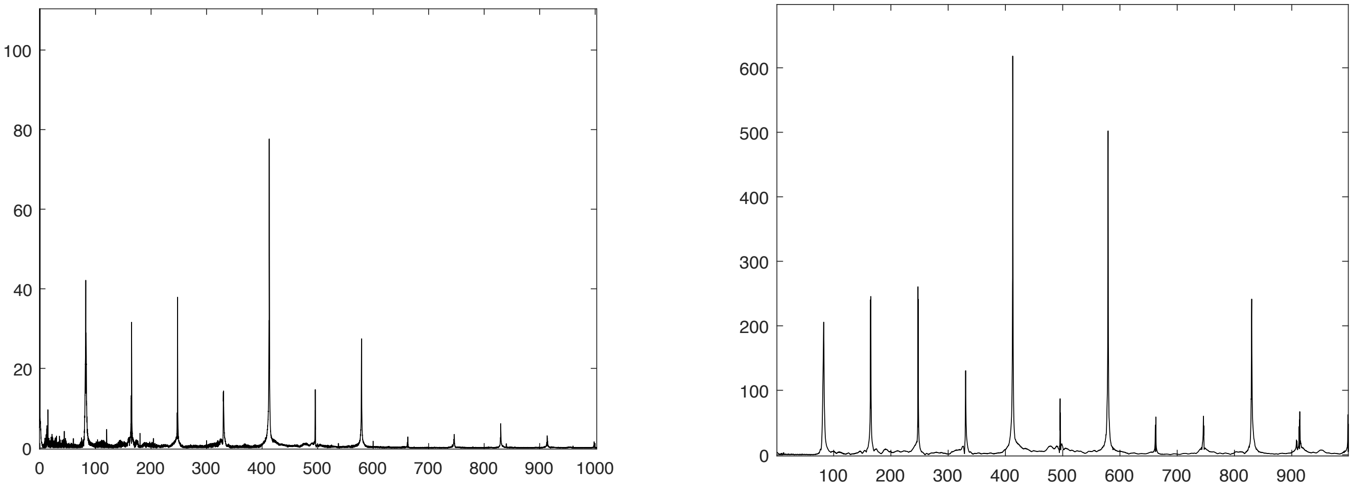
where $P(1)$ and $P(2)$ refer to the pitches of notes A0 and A#0 respectively. So, we defy a good enough frequency resolution as

$$\text{FR}_{\min} = 0.5\text{Hz} \tag{55}$$

That means

$$N_{\min} = \frac{F_{\text{s}}}{\text{FR}_{\min}} = 88200 \quad \text{samples} \tag{56}$$

Our excerpt length Since time domain note detection(e.g. Zero crossing [16], autocorrelation [17] to name a couple) are not in the analysis scope of this paper, for our methods we used the size of excerpt $N_{\min}$ as shown in Eq.(56) in order to theoretically successfully detect notes as low as A0.



(a)

(b)

Figure 20: Different harmonic magnitude relation for the first 12 harmonics of E2 between *pianissimo*(a) and *fortissimo*(b) intensity.

My initial goal My initial goal during this thesis was to ultimately find a way to recognize piano pieces containing chords(i.e. more than one note being played at the same time). Time domain approaches do not seem suitable for this task, so i opted to start for frequency domain methods.

5.1 Piano note frequency representation

How a piano note "looks" like if we obtain the magnitude part of its *Fourier* transform depends on several factors.

5.1.1 Harmonics' relative magnitude

First thing to notice, is that there is no definitive rule as to how the partials' or harmonics' magnitude relate. This means that the assumption that the fundamental frequency for the i^{th} note $F_0(i)$ will have the largest magnitude out of all harmonics, falls short. This suggests that obvious peak picking methods searching for the maximum of the note's magnitude spectrum are erroneous for some notes. A typical example highlighting this, is the concept of "hidden" fundamentals, seen at Fig. 2b. The harmonics' magnitude relation is also heavily affected by the intensity or interpretation of the piano player. These two aspects our shown in Fig. 20.

5.1.2 Inharmonicity

We had a basic idea of what inharmonicity is in Section 2.2.3. Let us sum up the aspects of inharmonicity regarding piano notes.

Partial inharmonicity This form of inharmonicity refers to how partials(i.e. harmonics) tend to have bigger frequency than the theoretical perfect multiple of the fundamental frequency F_0.

$$F_j(i) = (j+1)F_0(i) + I_j(i) \tag{57}$$

gives the actual frequency at harmonic j of note i, taking the inharmonicity $I_j(i)$ of note i at harmonic j into account. Since $I(i)$ depends on physical properties of the piano strings, such as diameter and stiffness, so it is different from note to note. Also, characteristics of the piano bridge and soundboard resonance factors, differ the inharmonicity of note i from harmonic j to harmonic j'.In general,

$$I_j(i) > I_{j'}(i) \quad , for \quad j > j' \tag{58}$$

Moreover this increase of inharmonicity with the increase of harmonic number is not linear, so it becomes more difficult to model.

Fundamental inharmonicity It is better to discuss fundamental inharmonicity with an example.

Let us assume an A4 note(note #49) tuned to its tempered pitch value as shown in Table 1, i.e. $P(49) = 440Hz$. The first harmonic of the A4 piano note would theoretically appear at $F_1(49) = 880\text{Hz}$. However, because of the partial inharmonicity (Eq,(57)), this harmonic will appear at a higher frequency.

$$F_1(49) > 2F_0(49)\text{Hz} \tag{59}$$

So, if we were to now tune note A5(note #61) at its tempered value of $P(61) = 880\text{Hz}$, the difference $\Delta_F = F_0(61) - F_1(49)$, would lead to a discomforting beat with a frequency of Δ_F Hz, if A4 and A5 were to be played together. To overcome this apparent beat, piano tuners tune the piano so as to eliminate the beat. This leads to actual piano note pitches as seen in the Railsback curve in Fig. 3.

It now becomes apparent that inharmonicity makes piano pitch detection a lot more challenging than imagined. Modeling this inharmonicity is an option [3]. Some notes can be chosen to model their inharmonicity and according to these measurements estimate the inharmonicity levels for every note and its respective harmonics. However in my thesis, i did not go deeper into trying to do these but instead tried to minimize its degree of involvement to measures.

5.2 Signal model

As we showed in Eq.(55) we are aiming for a frequency resolution of $\mathrm{FR} = 0.5\mathrm{Hz}$. However in our excerpts (Section 3) and in fact in most realistic cases the time period between successive detected onsets will be less than 2 seconds(i.e. 88200 samples for $F_\mathrm{s} = 44100\mathrm{Hz}$). To optimize our calculations and keep a consistent method we need to have input signals of same FFT length. To do that without errors we first isolate every note before obtaining its final spectral representation.

5.2.1 Excerpt in the time domain

Deciding on onset and offset times To come up with the correct signal representing a single note, we first refer to the output of our onset times $\hat{O}(m)$ (Eq.(51)).The peaks of $\hat{O}(m)$ refer to the onset blocks b_d. However, to have a clearer spectral representation(since onsets are a broadband event as described in Section 4.2.3, it is preferable to set the actual note detection start block b_s as the v^{th} block after the detected block b_d assigned to the peak of $\hat{O}(m)$.

$$b_s = b_{d+v} \tag{60}$$

where b_s refers to the decided start block for note detection methods, d is the onset detection block number as found in $\hat{O}(m)$ and v is the block shift introduced to guarantee a clearer spectral representation. An appropriate value of v must be chosen, depending on the time accuracy of the used onset detection method (Section 4.6).

The sample designating the start note analysis excerpt start is given as the 1^{st} sample of b_s.

$$n_s = b_s(1) \tag{61}$$

In the same manner we obtain a block and ultimately a sample designating the end of our note analysis excerpt.

$$b_e = b_{d'-v'} \tag{62}$$

where d' is the block number of the next detected onset found in $\hat{O}(m)$ and v' a positive constant indicating an appropriate number of shift to avoid next onset interference.

Ending sample for our note detection excerpt is the last sample of b_e.

$$n_e = b_e(L-1) \tag{63}$$

L being the time block size.

Our excerpt now is

$$q(n) = x(n) \quad , \quad n \in [n_s, n_e] \tag{64}$$

with a size of

$$N = n_e - n_s \tag{65}$$

x is the discrete time input.

5.2.2 Excerpt magnitude spectrum

The discrete time series $q(n)$ referring to our excerpt to be analyzed, is transformed under the *Fast Fourier transform* to obtain its magnitude spectrum(Eq.(6) and Eq.(7)). In order to have consistent analysis we once again opted for an $N_\mathrm{t} = 88200$ bin FFT representation.

$$Q(k) = |\mathrm{FFT}(q(n), N_\mathrm{t})| \tag{66}$$

where $Q(k)$ is the magnitude spectrum of $q(n)$ and $N_\mathrm{t} = 88200$.

- If $N_\mathrm{t} > N$, i.e. the note's actual duration is less than 2 seconds(or $N < 88200$ samples in our case), the FFT is performed by right zero padding $q(n)$, as explained in Eq.(11).
- If $N_\mathrm{t} > N$, or a longer than 2 second note, $q(n)$ is truncated and set to a length of N_t samples.

5.3 Frequency domain note detection methods

In the scope of this paper we will study two methods to detect the input note based on its frequency representation and more specifically the magnitude part of the note's *Fourier* transform. In the first method, the idea is *matched filtering* while in the second method a more straightforward measuring idea involving peak picking is implemented.

5.3.1 Note detection using matched filtering

Matched filtering [18] is a method for determining the presence and location of a target signal within some other signal, or set of signals. It is based on the *Cauchy-Schwarz* inequality.

The *Cauchy-Schwarz* inequality In linear algebra, the *Cauchy-Schwarz* [19] inequality states that for all vectors u and v of an inner product space[5] it is true that

$$|\langle u,v\rangle|^2 \leq \langle u,u\rangle \cdot \langle v,v\rangle \tag{67}$$

where $|\langle\cdot,\cdot\rangle|$ is the inner product.

Lets consider the maximization $|\langle \frac{x}{||x||}, \frac{y}{||y||}\rangle|$. From the *Cauchy-Schwarz* inequality (Eq.(67)) we get

$$\left|\left\langle \frac{x}{||x||}, \frac{y}{||y||}\right\rangle\right|^2 \leq 1 \tag{68}$$

since

$$\langle x,x\rangle = ||x||^2 \tag{69}$$

and

$$\left|\left|\frac{x}{||x||}\right|\right| = 1 \tag{70}$$

Hence through Eq.(68) we can see that $|\langle \frac{x}{||x||}, \frac{y}{||y||}\rangle|$ attains a maximum when $y = ax$ for some $a \in \mathbb{C}, \mathbb{R}$. The lower bound, which is 0, is attained when x and y are orthogonal. In informal intuition, this means that the expression is maximized when the vectors x and y have the same shape or pattern and minimized when x and y are very different. A pair of vectors with similar but unequal shapes or patterns will produce relatively large value of the expression less than 1, and a pair of vectors with very different but not orthogonal shapes or patterns will produce relatively small values of the expression greater than 0. Thus, the above expression carries with it a notion of the degree to which two signals are "alike", the magnitude of the normalized correlation between the signals in the case of the standard inner products.

In the following two note detection methods we make use of the *matched filter detector* by taking the normalized version of the input spectrum(i.e. divided by its norm)

$$\hat{Q}(k) = \frac{Q(k)}{\|Q(k)\|} \tag{71}$$

[5] In linear algebra, an inner product space is a vector space with an additional structure called an inner product. This additional structure associates each pair of vectors in the space with a scalar quantity known as the inner product of the vectors. Inner products allow the rigorous introduction of intuitive geometrical notions such as the length of a vector or the angle between two vectors.[wiki]

and *matching* it with our library's similarly normalized spectra,

$$\hat{S}_i(k) = \frac{S_i(k)}{\|S_i(k)\|} \tag{72}$$

with $i \in [1, 88]$ indicating the note number.

We measure the input's spectrum $\hat{Q}(k)$ "similarity" with $\hat{S}_i(k)$ for all i.

$$C(i) = |\langle \hat{Q}(k), \hat{S}_i(k) \rangle| \tag{73}$$

and finally decide on the detected note i_d,

$$i_d = argmax_{i \in [1,88]} C(i) \tag{74}$$

where the detected note i_{d}, gives $C(i)$ its maximum value(closest to 1).

We will see two implementations of *Matched Filtering*, each one comparing the input normalized spectrum with different pre-existing sets of normalized spectra. The first case uses a library of spectra from several pianos, including the input's piano while the second one uses a library of spectra "mimicking" the representation of ideal note spectra.

Case 1. Library of several pianos' spectra The first attempt to minimize the effect of every piano's unique spectral properties and "anomalies", uses a library of spectra obtained after averaging spectra of piano notes played in PN separate pianos.

Implementation steps

1. By using the *Cubase* software we loaded all the notes from two virtual pianos, in both piano cases with a sampling frequency $F_\mathrm{s} = 44100$ Hz. The excerpts were 2 seconds in duration, comprising only from the note(i.e. no zeros before or after the note). These excerpts, along with the notes obtained by *Iowa Univeristy* (see section 2) "filled" our note library.

2. For every piano note we obtained its normalized spectrum as shown in Eq.(71).

   The pre-existing set's spectrum for every note is once again found as

   $$\hat{S'}_i(k) = \frac{S'_i(k)}{\|S'_i(k)\|} \tag{75}$$

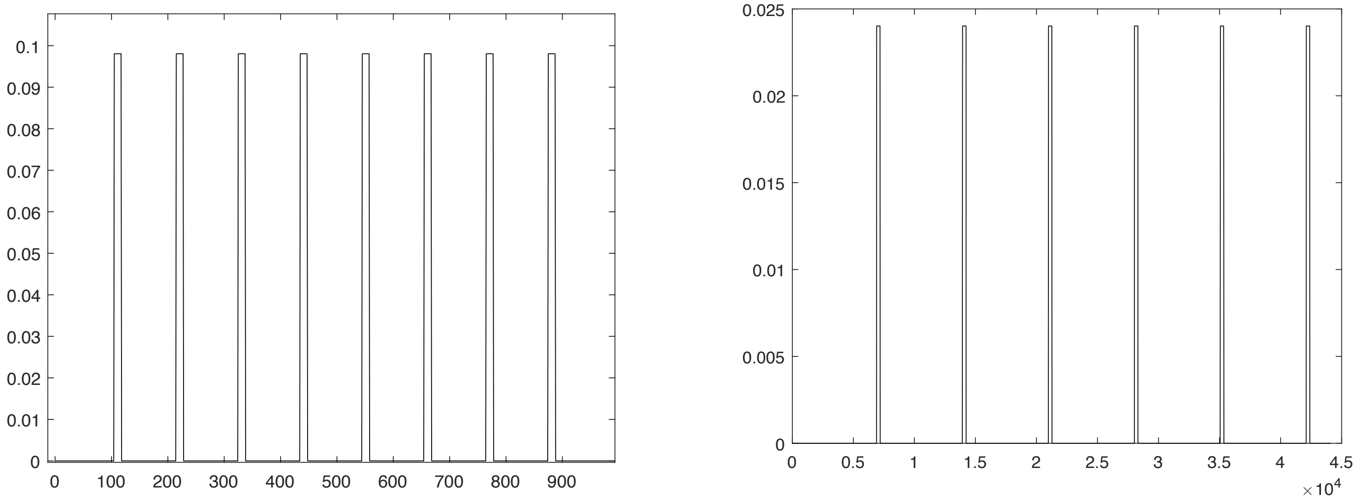
   where $S'_i(i)$ is found as the average spectrum of every note

   $$S'_i(k) = \frac{\sum^{\mathrm{PN}} \hat{S}_{\mathrm{pn},i}(k)}{\mathrm{PN}} \tag{76}$$

   over all PN pianos.

3. We complete the process as suggested by Eq.(73) and Eq.(74).

Case 2. Library of ideal spectra This time the pre-existing library consists of spectra "mimicking" ideal piano note spectra. There have been attempts to create such spectra by evaluating the total inharmonicity presented in piano notes(section) [3]. However, such an implementation requires detailed and excessive research on physical characteristics of the specific piano, which is outside the scope of this paper. In our method we suggest a rough modeling of the inharmonicity.



(a)

(b)

Figure 21: Ideal spectrum of note A2(13) (a) and note A7(85) (b). For (b) masks are less,lower and wider

Implementation steps

Constructing the ideal spectra of a single note

1. The goal is to create a signal, with rectangular pulses or *regions* of certain width and height, around frequencies where harmonics would ideally appear.

Determining harmonics' region's central frequency For the region referring to the fundamental frequency F_0 we determined its central frequency as suggested by the Railsback curve(Fig. 3).

$$H_0(i) = P(i) + R(i) \cdot (P(i+1) - P(i)) \tag{77}$$

where P refers to the tempered pitch as shown in Table 1 and R refers to the Railsback curve value for note i. For the rest of the regions we assumed their central frequencies without taking the partial inharmonicity effect into account, even though heuristic approximations can improve the accuracy of ideal "locations".

$$H_j(i) = (j+1) \cdot H_0(i) \tag{78}$$

$H_j(i)$ referring to the j^{th} harmonic's frequency for note i.

Determining harmonics' region's width The width of every region is set, taking into account a once again rough consideration of inharmonicity. In general, since magnitude peaks of higher notes are wider(i.e. 2 or 3 slightly off-tune strings) we opted for a wider, by a widening factor, region when looking at higher frequencies in general.

$$w(i) = 2 + cP(i) \tag{79}$$

where w is the width of the region aiming to detect the peak of the j^{th} harmonic of note i,c is the widening factor and P is the pitch. Notice that our approximation assumes a single width for every note irrespective of harmonic number j, thus inadequately modeling the partial inharmonicity.

2. In our implementation we constructed $J = 8$ such *regions* for every spectrum, "mimicking" the first 8 harmonics of every note. However, for higher fundamental frequencies(i.e. higher notes), some harmonics' frequency might exceed not only our analysis bandwidth, but also trace frequencies where harmonics have very little to zero energy. For these notes fewer regions were constructed.

3. To keep the comparison "fair" between ideal spectra containing different number of regions, it is of bigger importance to normalize these spectra as shown in Eq.(71).

4. After obtaining the ideal spectra and comparing them with the input, we observed that most false detections were cases where ideal spectra of harmonically related notes(Section 2.2.2), lower to the actual note, all gave a high match(Fig. 22b). To override this false matching, we used a method to verify existence of the fundamental frequency detected at this stage.

Checking for fundamental frequency "existence" This intermediate method can improve the performance of the ideal spectra method in expense of computational cost. It falls short at tracing fundamentals of low piano notes, due to the phenomenon of "hidden" fundamental, also explained in Section 2.2.2. The way to verify the existence of the fundamental is as follows.

(a) For the intermediate, still to be verified, detected note i'_{d} as found from the *matched filter detector* we introduce a spectral area, similarly to *bandpass* filtering our spectrum. This area's left(low) and right(high) frequency limits are found as

$$L(i'_{\text{d}}) = \lfloor (1-b)P(i'_{\text{d}}) \rfloor \tag{80}$$

and

$$R(i'_{\mathrm{d}}) = \lfloor (1+b)P(i'_{\mathrm{d}}) \rfloor \tag{81}$$

where b is the area width term, adjusted to ensure isolation of at most one harmonic and P once again refers to the pitch, or fundamental frequency value of the intermediately detected note i'_{d}.

The area is finally defined as

$$A_a(i'_{\mathrm{d}}) = \hat{Q}(a) \tag{82}$$

where $a = L(i'_{\mathrm{d}}), L(i'_{\mathrm{d}})+1, \ldots, R(i'_{\mathrm{d}})-1, R(i'_{\mathrm{d}})$ and $\hat{Q}$ is the input's normalized spectrum.

(b) For the area $A_a(i'_{\mathrm{d}})$ we measure a statistical property referred to as *maximality.* The maximality of the area is found as

$$M(A) = \frac{\max(A)}{\overline{A}} \tag{83}$$

measuring the ratio of the *max* of A to the area's mean value $\overline{A}$. $M(A)$ basically measures how distinct the maximum value(peak) is, compared to the surrounding frequencies' magnitude values. In general, peaks actually referring to harmonics tend to be a lot higher than the mean value of the surrounding area, resulting in high *maximity* values. The final decision about the validity of the intermediate detection is done by fixed thresholding the value of M.

$$M_{i'_{\mathrm{d}}}(A) > T \implies i_{\mathrm{d}} = i'_{\mathrm{d}} \tag{84}$$

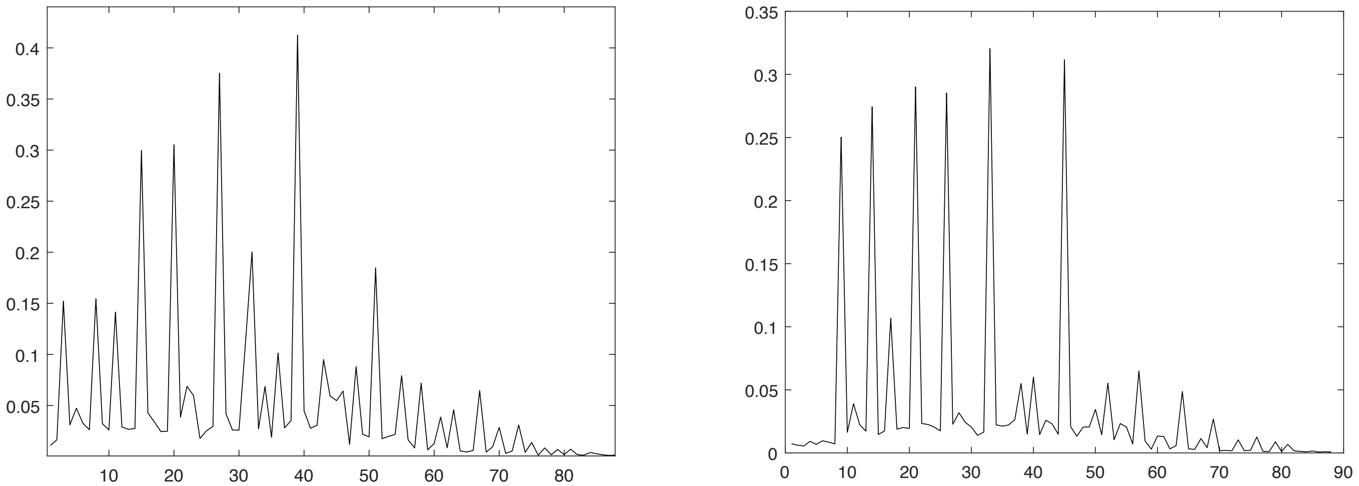
for a fixed threshold T, heavily correlated with the widening term b eq(), and i_{d} referring to the final valid detection.

(c) In the case where the intermediate detection i'_{d} is not valid, i.e. $M_{i'_{\mathrm{d}}}(A) < T$ then $C(i)$ is updated

$$C(i'_{\mathrm{d}}) = 0 \tag{85}$$

and the fundamental validity check is performed again for the maximum of the updated $C(i)$. The whole procedure is repeated, until a valid fundamental has been detected.

5. The valid fundamental decides the final detected note i_{d}.



(a)

(b)

Figure 22: Matched filter output $C(i)$ for case 1 for note B3(39) (a) and case 2 for note F4(45)(b). In case 2 (b) the high output for harmonically related to the F4 lower pitches highlights the need for the fundamental check process.

Comparison between the two *matched filtering* cases

Case 1, has the benefit of "masking" some spectral anomalies present in piano notes spectra(Section 5.1), by averaging them. However it only yields good results when looking for notes coming from a piano already accounted for in the average spectra. In other cases the library must be updated with spectra of the piano to be detected.Still there are several false detections, where the detection is the high octave of the input note. Additionally, even though in our case when adding more pianos to the library, the performance for every piano separately slightly improved, we can assume that this is not necessarily true for an ever increasing number of pianos in the library, as there is a high chance the average spectra will become extremely noisy, especially at lower frequencies where fundamental frequencies are not so distinctly separate.

Case 2, tackles the main weakness of case 1, by attempting to offer a library suitable for any piano input. However since inharmonicity(a highly unique characteristic of any piano) is not modeled adequately enough,the intention for *globality* of the library is somehow tackled. Furthermore the effectiveness of the *matched filter detector* is reduced, since in many occasions a correct detection is obtained after running the fundamental validity check a few times. This means that methods using solely methods similar to the fundamental frequency check, might be proven more effective in a certain set of note inputs. Lastly, since the regions comprising the ideal spectra are of a fixed width and height, *timbre* of notes is merely taken into account, resulting to heavily varying performance in respect of different interpretations of piano notes.

Fig. 22 shows the *matched filtering* method output for both cases.

5.3.2 Note detection using harmonics' energy

This method is in many ways, different than the method of matched filtering. The basic concept is that for every possible note i we obtain a measure of the energy of the first N peaks and come to a detected note decision based on the energy measurement made.Lets follow the algorithm, step-by-step.

The following procedure is repeated N times,once for every peak, for every note i, starting from $i = 1$ up to 88.

1. For the i^{th} note, the n^{th} peak refers to the maximum of the updated $\hat{Q}_{n-1}$ spectrum, i.e. the spectrum before the n^{th} peak picking(e.g. in Fig. 23a the first peak($n = 1$),is the maximum of input original spectrum Q_0 as obtained in Eq.(66). Such a notation for input spectrum $Q(i)$ is needed, as for after every repetition and peak detection, the under examination spectrum $Q(i)$ is updated accordingly.

2. **Updating the spectrum** After the detection of the n^{th} peak we save its respective frequency as $m_n(i)$ and define an area with and around this peak. The limits of this area are found as

$$L_n(i) = m_n(i) - \lfloor P(i) \rfloor \tag{86}$$

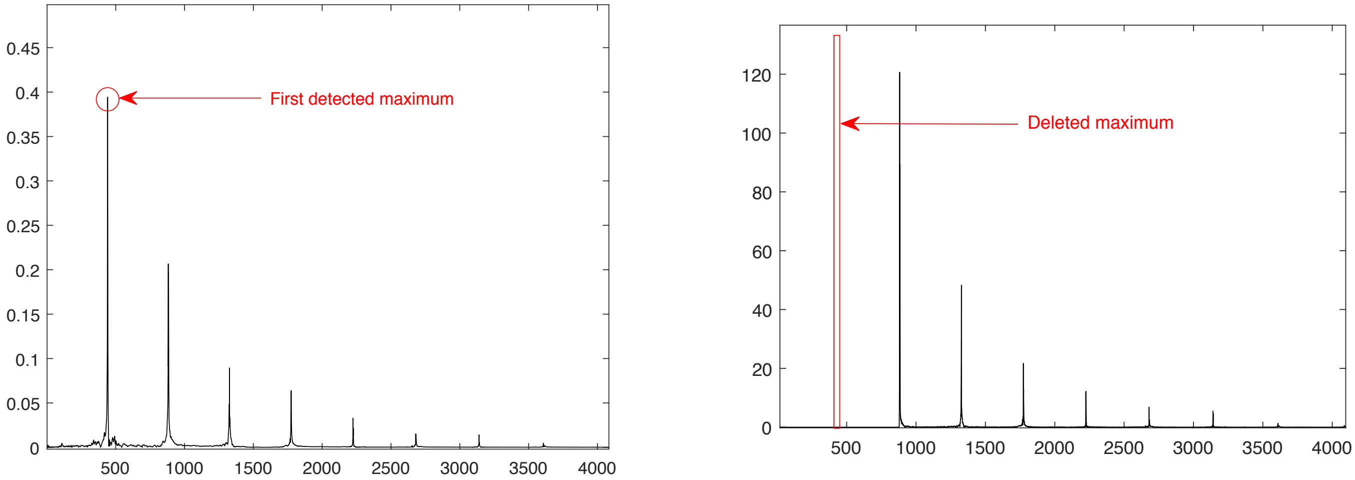
   and

$$R_n(i) = m_n(i) + \lfloor P(i) \rfloor \tag{87}$$

   where $P(i)$ is the pitch of note i. To avoid looking for indexes outside the size of Q_0, $L_n(i)$ is bounded to a minimum of 1, indicating the first element of Q_0, while $R_n(i)$ is bounded at a maximum of 44100, namely the last bin of Q_0.

3. The area between $L_n(i)$ and $R_n(i)$ referring to spectrum Q_{n-1} is now zeroed and the spectrum is updated for the next peak pick.

These 3 steps are repeated N times for every i possible note, after which we have saved the peak positions $m_n(i)$, $n = 1, 2, \ldots, N$.



(a) First peak obtained normally, as it coincides with the first iteration of the method

(b) Effect of area zeroing, as peak at 880Hz is zeroed out for $(i > 49)$ as shown in fig 18.

Figure 23: Peak picking process for A4 note($i = 49$)

Continuing, we take the positions of the peaks $m_n(i)$ and for every note we take a measurement of the overall energy of these peaks as shown.

$$E(i) = \sum_{n=1}^{N} Q_0(m_n(i))^2 \tag{88}$$

where Q_0 is the input's original magnitude spectrum.

Interpretation of the energy measured The reason why we measure the energy of the chosen peaks of all possible notes, can be examined by steps 2 and 3 mentioned above. Lets explain this graphically in Fig.s 23 and 24.

After locating the first maximum (Fig. 23a), in Fig. 24 we see how the areas to be zeroed are decided. For A4, coinciding with the input note we see that the peak on the right(first harmonic) is not included in the zeroing area(yellow line)and thus reserved for a possible later pick. However when the area referring to A$\#$4(red line) is zeroed this peak is also zeroed. So a "valuable" peak is lost and not counted for $E(50)$. as shown in Fig. 23b.This through Eq.(88) will result to an $E(50) < E(49)$, thus indicating the first occurred energy drop. This energy drop, shown in Fig. 25a, will further increase as we calculate the energy for the N first peaks found for every i.

Decision We finally take the absolute values of the first derivative of the energy $|\Delta E|$ and decide detected note i_{d} as the index of its first non-zero element, as shown in Fig. 25b.

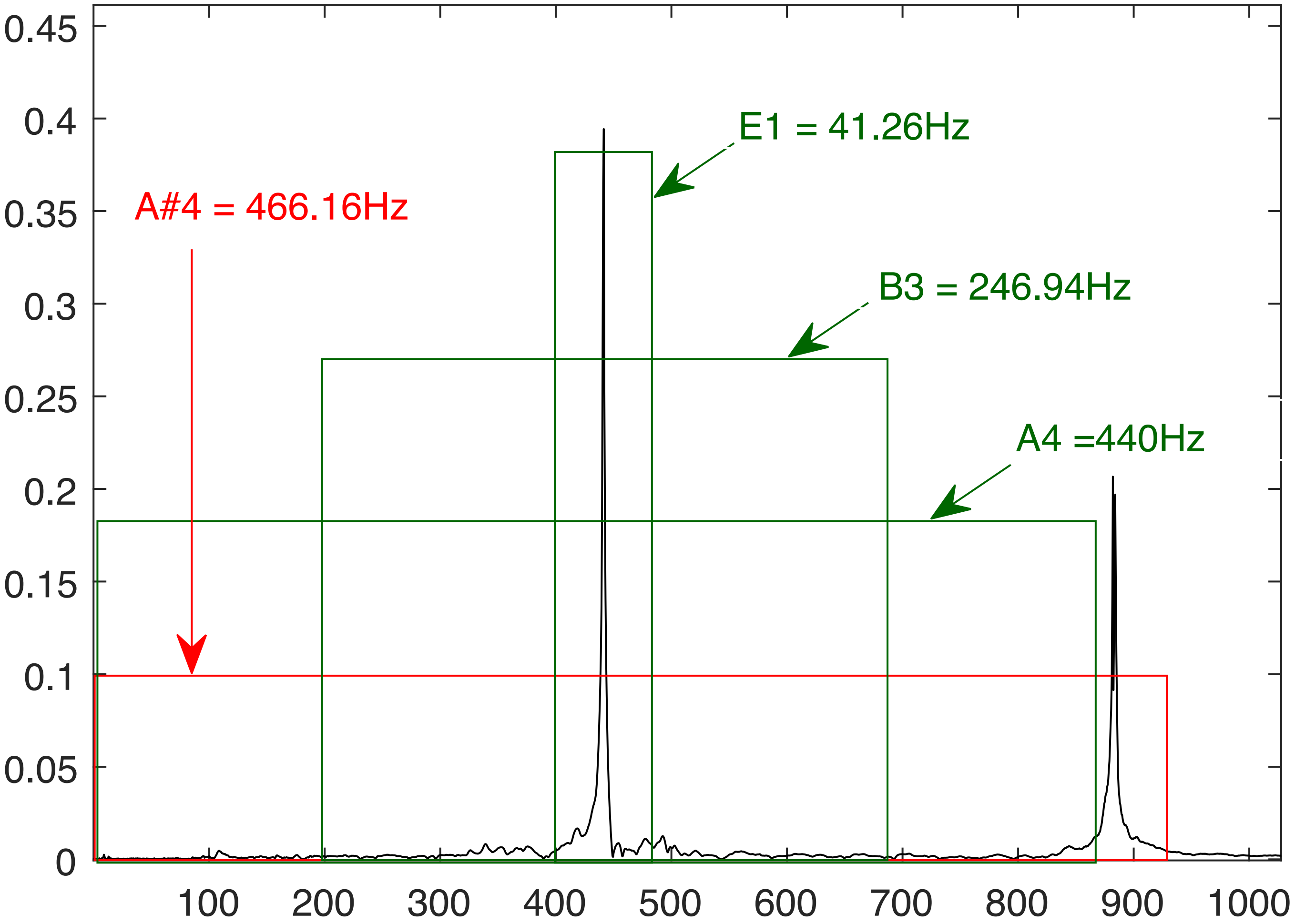
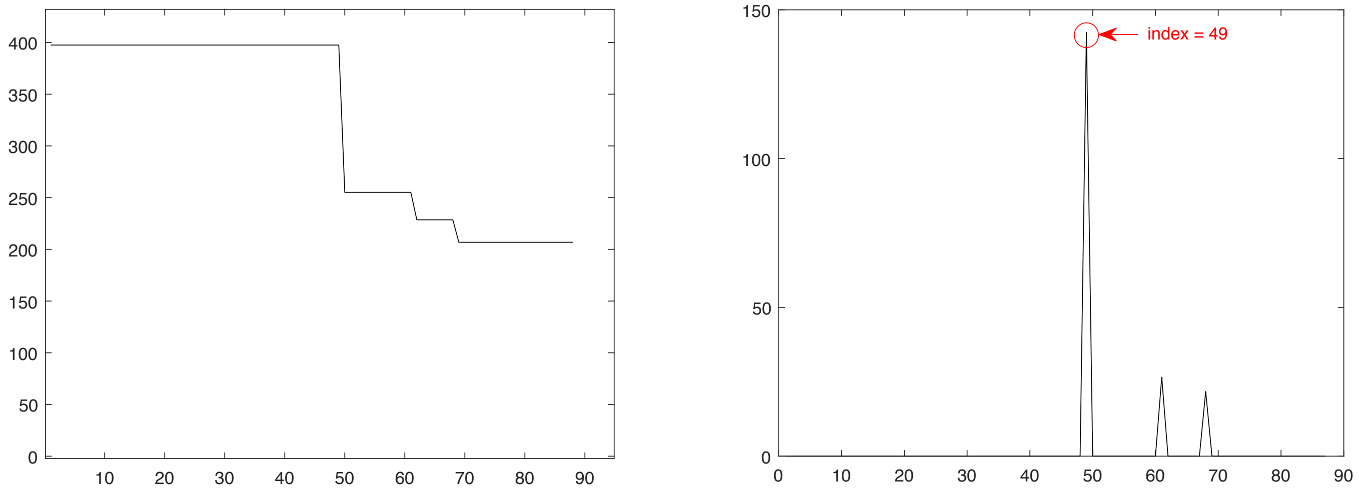


Figure 24: Zero area determination for several notes i for input spectra of A4. Red line for note $i = 50$ indicates the first area(as i increases) to zero out a "valuable" peak.



(a) Energy drop at $i = 50$, caused by zero areas

(b) Absolute first derivative of the energy, and detected index $i = 49$

Figure 25: Energy measured and decision making

Evaluation of the method The described procedure is global, in the sense that it does not take any particular piano characteristics into account. However, the issue of inharmonicity is only conditionally tackled. The method proposed, will only produce satisfactory results if the measured maxima N, all refer to actual harmonics of the spectrum and contain at least one pair of successive harmonics in order to be able to detect the first i for which an energy drop is detected. To achieve that we opted to run the algorithm for three values of N chosen under these criteria.

$N_1 = 10$ This relatively big number of peaks chosen, helps identify low notes. As we saw in previous sections, for low notes the lower harmonics tend to have less energy. At the same time lower notes have more harmonics, so we can safely assume that only useful peaks are calculated. Lastly, partial inharmonicity is tackled by using this relatively big N_1. Not only do we succeed at least a pair of successive harmonics, but moreover a pair found at a fairly low harmonic number. That ensures that the partial inharmonicity has not grown excessively at these harmonics, offering a more trustworthy zeroing area step.

$N2 = 4$ In order to achieve successful note detection for the mid range of the piano , we run the algorithm for $N_2 = 4$. N_1 will tend to take into account peaks that do not refer to actual harmonics, since notes in the middle of the piano range might show less than 10 harmonics.

$N_3 = 1$ For similar reasons we introduce the last case of $N_3 = 1$, for detecting notes in the mid-high range of the piano. This is also called the *idle* state of the algorithm, since the essential detection procedure of the algorithm is bypassed. In this case, the detected note i_d is found as

$$i_d = argmin_{i \in [1,88]} |m_{N_3} - P(i)| \tag{89}$$

where m_{N_3} is the frequency of the only peak picked, and i_d is the detected note found as the note whose tempered pitch is closer to m_{N_3}.

Candidates So for every input note spectrum I_0 the algorithms returns a set of three detected note candidates, each referring to a different number N of measured peaks. Even though this may not seem very appealing at start, the fact that the actual note is guaranteed to be in this set, some extra methods can be used to choose effectively between notes in the candidate set.

6 Conclusions

As far as onset detection is concerned, time domain methods based on the energy of the signal, although more straight-forward in their implementation, were proven less effective in detecting less percussive onsets. Same thing applies to the frequency domain, when searching for onsets based on the high frequency content of a signal. Out of all the tested methods, the ones based on phase information, namely the phase deviation and the phasor stationarity, yielded the best results in detecting soft onsets. Also, their ability to ignore pitch characteristics, helps in avoiding onsets potentially 'masked' by harmonic relevance of the notes played. The subsequent task of thresholding and peak-picking the output of the algorithms was in turn proven to be a challenging task. When dealing with music notes where interpretation is affecting the level and distinctiveness of peaks, fixed thresholding was proven to be below par. However, techniques utilizing adaptive thresholding, even though more effective, are still reliant on sometimes arbitrary trial and error fine tuning, before their execution. As far as the note detection methods are concerned, matched filtering methods assisted with predefined spectra libraries although robust in theory, are heavily affected by the inharmonicity of the piano, namely the innate deviation of higher order harmonics from the theoretical multiples of the fundamental frequency. It is worth mentioning however, that this method has more potential in chord detection as the 'masks' of detected notes are easy to be progressively subtracted from the original spectrum. The second method based on harmonics' energy, attempted to bypass the effect of inharmonicity. The algorithm proposed, aims to pick at least a pair of consecutive order harmonics in its candidates in order to decide the fundamental frequency. For this reason, and taking other piano note characteristics into consideration, the method is run thrice; for 3, 5 and 10 maxima, ultimately offering a set of three note candidates with the real note guaranteed to be a member of it. This work around was made in order to find a method to cover the whole range of the piano, more so the 'problematic' low and high regions where inharmonicity is more prominent.

In the course of this work, a lot of time and effort was spent on trying to model the effect and nature of inharmonicity. It is obvious, that machine learning techniques, might be proven very helpful in modelling the variety in harmonics' deviation and also differences between pianos. Same techniques could be applied, in order to generate more confident thresholding or peak picking steps, in determining false positives detected during the onset detection process.

References

[1] Wikipedia. Piano — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Piano&oldid=790446013`, 2017. [Online; accessed 21-July-2017].

[2] John Backus. *The Acoustical Foundations of Music.*

[3] L. I. ORTIZ-BERENGUER, F. J. CASAJUS-QUIROS, and S. TORRES-GUIJARRO. Multiple piano note identification using a spectral matching method with derived patterns. 2005.

[4] Simon Hendry. Inharmonicity of piano strings. 2008.

[5] Lawrence Fritts. University of iowa,electronic music studios. `http://theremin.music.uiowa.edu/MISpiano.html`, 2001.

[6] The 5 methods of stereo recording. `https://ehomerecordingstudio.com/stereo-microphone-techniques/`.

[7] Samer Abdallah Chris Duxbury Mike Davies Juan Pablo Bello, Laurent Daudet and Mark B. Sandler. A tutorial on onset detection in music signals. 2005.

[8] M. Goto and Y. Muraoka. Beat tracking based on multiple-agent architecture - a real-time beat tracking system for audio signals. 1996.

[9] National Instrumnets. Smoothing windows for spectral leakage. `http://www.ni.com/white-paper/4110/en/`, 2011.

[10] A. W. Schloss. On the automatic transcription of percussive music - from acoustic signal to high-level analysis. 1985.

[11] Paul Masri. Computer modelling of sound for transformation and synthesis of musical signals. 1996.

[12] M. Sandler C. Duxbury and M. Davies. A hybrid approach to musical note onset detection. 2002.

[13] J. P. Bello and M. Sandler. Phase-based note onset detection for music signals. 2003.

[14] M. Davies C. Duxbury, J. P. Bello and M. Sandler. A combined phase and amplitude based approach to onset detection for audio segmentation. 2003.

[15] M.Davies and M.Sandler J.P.Bello, C.Duxbury. On the use of phase and energy for musical onset detection in the complex domain. 2004.

[16] T. V. Ananthapadmanabha and B. Yegnanarayana. Epoch extraction of voiced speech. 1975.

[17] Chamberlin. On musical applications of microprocessors. 1980.

[18] openstax. Matched filter detector. https://cnx.org/contents/yy8aFITd@8/Cauchy-Schwarz-Inequality, 2017.

[19] Art of Problem Solving. Cauchy-schwarz inequality. https://artofproblemsolving.com/wiki/index.php?title=Cauchy-Schwarz_Inequality, 2017.