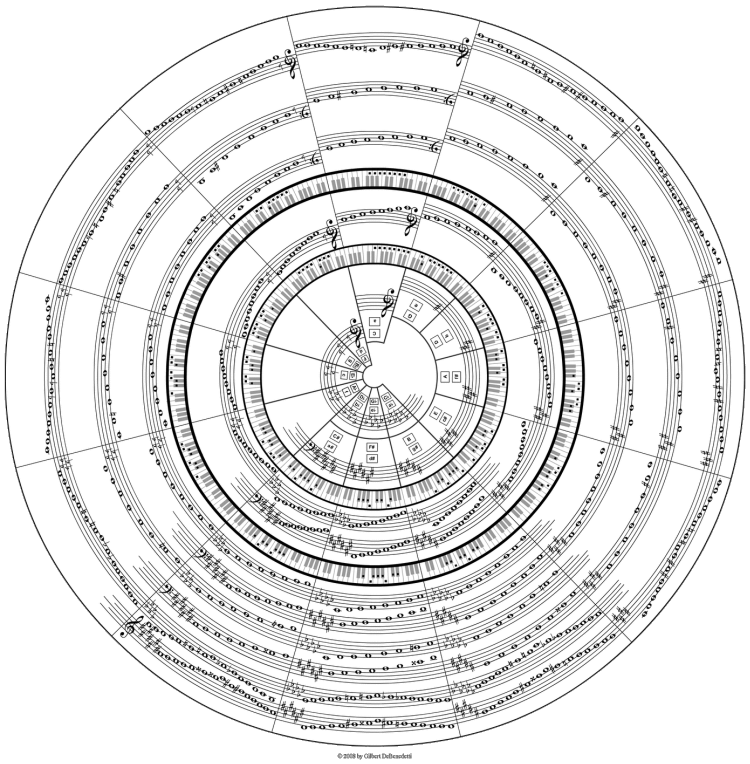
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN NORMANDIE

UNIVERSITÉ DE ROUEN NORMANDIE

Masters Thesis Project

Automatic Music Transcription

Audio Signal Processing



Rand ASSWAD
Department of Applied Mathematics
Theoretical and Applied Computer Science

Under the supervision of:
Prof. Natalie FORTIER
Prof. Jean-Philippe DUBERNARD

12 March 2020

Contents

Introduction

Music is ubiquitous ever since humans exist. Prehistoric instruments have been found and thought to be at least 40,000 years old. Music is a pilar of human civilisation; it relates to people's identities, feelings and thoughts. Hence, means of saving and sharing music are of invaluable importance. The oldest surviving notated music work *Hurrian Hymn to Nikkal* found on clay tablets dates back to 1400 BC.

Various systems were developped around the globe for *visually* representing perceived music through the use of written symbols. The modern western notation is the predominent musical notation worldwide for most music genres.

With the rise of technology, audio recordings where introduced as analog signals and eventually as digital signals, providing means for sharing and sauveguarding music *aurally*.

Music theory and musical notation have been studied for centuries, allowing humans and machines to retrieve music information from common formats. Nevertheless, *music processing* is a relatively young discipline compared to other subdomains of signal processing such as speech processing; while great results are achieved today in speech recognition, the task of retreiving music information from audio recordings is still far along.

Automatic Music Transcription (AMT) is the task of analyzing musical audio signals and producing the corresponding musical scores. This task has captured researchers interest in the late 20^{th} century and has become a wide research discipline as many of the problems in this domain remain unsolved. Furthurmore, strides in the domain of AMT would apply to numerous applications that can facilitate creating, sharing, and learning music.

The scope of this thesis is the domain of Automatic Music Transcription and the underlying tasks. We explore the state of the art and propose an implementation for a subset of the presented methods.

I have held interest for this project for quite some time, partly because I am a violinist myself but also because of my fondness of the employed mathematical principles. Most importantly, this project requires application of various mathematical notions as well as computer science skills hence serving as a demonstration of acquired knowledge throughout the Masters program.

1 Background

The focus of this project is music information retrieval from music audio signals. In this section we study defining characteristics of musical elements, human perception of music, and basic notions of modern music theory. We also review the main characteristics of a sound wave as well as analytic tools for processing digital audio signals. Furthermore, we establish the bridge between music theory and physical properties of audio signals.

1.1 Physical definition of acoustic waves

Sound is generated by vibrating objects, these vibrations cause oscillations of molecules in the medium. The varying pressure propagates through the medium as a wave, the pressure is therefore the solution of the wave equation in time and space, also known as the acoustic wave equation.

$$\Delta p = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2}$$

where p is the accoustic pressure function of time and space and c is the speed of sound propagation. The wave equation can be solved analytically with the separation of variables method, resulting in a *sinusoidal harmonic* solutions.

In audio signal processing, we are interested in the pressure at the receptor's position (listener or microphone), hence the pressure as a function of time. An audio signal is therefore defined as the deviation of pressure from the average pressure of the medium at the receptor's position.

The pressure function being harmonic, the sound signal is of the form

$$\tilde{x}(t) = \sum_{h=0}^{\infty} A_h \cos(2\pi h f_0 t + \varphi_h)$$

where

- f_0 is called the **fundamental frequency** of the signal,
- h is the harmonic number,
- A_h is the amplitude of the h^{th} harmonic,
- φ_h is the phase of the h^{th} harmonic.

In many works this formula appears in terms of the angular frequency $\omega = 2\pi f$, we denote as well $f_h = h f_0$ for $h \geq 1$.

As harmonics represent proper multiples of the fundamental frequency, $h = 0$ is excluded from the sum

$$\tilde{x}(t) = a_0 + \sum_{h=1}^{\infty} A_h \cos(2\pi h f_0 t + \varphi_h)$$

with $a_0 = A_0 \cos(\varphi_0)$.

1.2 Perception of sound and music

The human auditory system is capable of distinguishing intensities and frequencies of sound waves as well as temporal features. The inner ear is extremely sensitive to sound wave features, the brain allows furthur analysis of these features.

Music theory defines and studies *perceived features* of music signals. These features are based on the signal's intensity, frequency, and time patterns.

In music theory, a **note** is a musical symbol that represents the smallest musical object. The note's attributes define the *pitch* of the sound, its *relative duration* and its *relative intensity*.

1.2.1 Fundamental frequency and pitch

Sound signals are periodic, therefore by definition there exists a $T > 0$ such as

$$\forall t, \tilde{x}(t) = \tilde{x}(t + T)$$

which follows that there exists an infinite set of values of $T > 0$ that verify this property, indeed $\forall n \in \mathbb{N}, T' = nT, \tilde{x}(t) = \tilde{x}(t + T')$. We define the period of a signal as the smallest positive value of T for which the property holds. The **fundamental frequency** f_0 is defined formally as the reciprocal of the period. This definition holds for *any* periodic signal, regardless of its form.

In the case of sound wave, the *perception of the fundamental frequency* is referred to as the **pitch**. Pitch is the defined as the *tonal height* of a sound, it is closely related to the fundamental frequency however remaining a *relative musical concept* unlike the f_0 of a signal that is an absolute mathematical value. In fact, the relation between pitch and f_0 is neither bijective nor invariant.

In music theory, pitch is defined on a discrete space unlike the continuous frequency space. Moreover, human perception of frequency is logarithmic hence obtaining the *next pitch* corresponds to the multiplication of the frequency by a certain value r.

Finally, the frequency of the reference pitch A_4 is widely accepted today as $440Hz$ while in the baroque era it was around $415Hz$ and $440Hz$ was the frequency corresponding to A♯ pitch. Even in modern day, variations of the pitch frequency exist in different regions and even different orchestras!

1.2.2 Perception of intensity

Sound intensity is defined physically as the power carried by sound waves per unit area, whereas sound pressure is the local pressure deviation from the ambient atmospheric pressure caused by a sound wave. Human perception of intensity is directly sensitive to sound pressure, it is measured in terms of *sound pressure level* (SPL) which is a logarithmic measure of sound pressure P relative to the atmospheric pressure P_0 measured in decibels dB.

$$\text{SPL} = 20 \log_{10} \left(\frac{P}{P_0} \right) \text{dB}$$

Nevertheless, sensitivity to sound intensity is variable across different frequencies. The subjective perception of sound pressure is defined by a sound's **loudness** which is a function of both SPL and frequency ranging from quiet to loud.

In music theory, loudness is defined by a piece's **dynamics**. Dynamics are indicators of a part's loudness *relative* to other parts and/or instruments. Dynamics markings are expressed with the italian keywords *forte* ***f*** (loud) and *piano* ***p*** (soft). Subtle degrees of loudness can be expressed by the prefixes *mezzo-* or *più*, for example ***mp*** stands for *mezzo-piano* (moderately soft) and *più* ***p*** (softer), or by consecutive letters such as *fortissimo* ***ff*** (very loud) or more letters if needed.

Music dynamics also allow expressing gradual changes in loudness, indicated as symbols or italian keywords (*crescendo* and *diminuendo*).

1.2.3 Perception of duration

1.3 Audio signal processing

1.3.1 Discrete-time signals

The domain of audio signal processing deals with recorded digital/analog signals, which are discrete-time signals. The **Nyquist-Shannon sampling theorem** is the fundamental bridge between continuous-time and discrete-time signals. It establishes a sufficient condition for a sample rate that permits a discrete sequence of samples to capture all the information from a continuous-time signal. ("Nyquist–Shannon Sampling Theorem" 2020)

1.3.2 Discrete Fourier Transform (DFT)

1.3.3 Short-Time Fourier Transform (STFT)

1.4 Automatic Music Transcription

- what is it
- what are its sub-tasks

2 Pitch analysis

2.1 Introduction

Pitch analysis is the task of estimating the fundamental frequency of a periodic signal that is the inverse of the period which is defined as "the smallest positive member of the infinite set of time shifts leaving the signal invariant" (Cheveigné and Kawahara 2002). As music signal frequencies vary through time, the pitch analysis is usually performed on a short time frame (window) allowing to express the obtained pitch as a function of time, we will consider henceforth the analysis on a single frame.

Furthermore, the physical model we have considered for the signal formula is based on physical hypotheses. In fact, we considered a signal formed by a perfectly harmonic instrument travelling in a perfectly undisturbed homogenuous medium with no other iterfering waves. Since such conditions are almost never met, we base our analysis on *imperfect conditions*. Indeed, the recorded signal represents the pressure function at the receptors position. Consequently, the recorder captures the pressure at its position from *all* surrounding stimuli, recording surrounding noise, resonance effects, and the reflected wave with a certain lag. As a result, we express the observed signal as the sum of the harmonic signal $\tilde{x}$ and the residual z. (Yeh, n.d.)

$$x(t) = \tilde{x}(t) + z(t)$$

Before we move on, let's consider the *harmonicity* of a sound. In the case of perfectly harmonic instrument the frequency of harmonic partials is expressed as a proper multiple of the fundamental frequency $f_h = hf_0$. However, most musical instruments are not perfectly harmonic, for example the h^{th} harmonic frequency of a vibrating string is given as

$$f_h = hf_0\sqrt{1 + Bh^2} \quad \text{where} \quad B = \frac{\pi^3 E d^4}{64 l^2 T}$$

where B is the inharmonicity factor of the string, E is Young's modulus, d is the diameter of the string, l is its length and T is its tension. We refer to such signals as **quasi-periodic**. Pitch analysis therefore has to take into account the inharmonicity of a signal in the process of estimating its fundamental frequencies in order to prevent cases of false negatives (missed pitches). [source needed]

Pitch analysis deals with both monophonic and polyphonic signals, a monophonic signal is a signal produced by a single harmonic source whereas polyphonic signals have multiple sources, in the case of the latter the task is significantly harder. Nevertheless, pitch estimation methods for both single and multiple sourced harmonics can be classified into two categories: methods that estimate the *period* in the signal time domain and methods that estimate the f_0 from the harmonic patterns in the signal spectrum.

2.2 Single pitch

Single pitch estimation is based on finding the fundamental frequency of a monophonic sound. The quasi-periodic monophonic signal $\tilde{x}$ is expressed as

$$\tilde{x}(t) = \sum_{h=1}^{\infty} A_h \cos(2\pi f_0 t + \varphi_h)$$

For practical reasons, a finite number of harmonic partials H is used to approximate the signal.

$$\tilde{x}(t) \approx \sum_{h=1}^{H} A_h \cos(2\pi f_0 t + \varphi_h)$$

The estimation of f_0 can be approached in two different ways: by analysing the time function $x(t)$ or by analysing the signal spectrum $X(f)$.

2.2.1 Time domain

Time domain methods analyse the repetitiveness of the wave by comparing the signal with a delayed version of itself. This comparison is achieved using special functions that represent the pattern similarity or dissimilarity as a function of the **time lag** τ.

We will study and compare a the functions that appear the most in litterature.

Autocorrelation function The autocorrelation function (ACF) comes immediately to mind. By definition, autocorrelation is the similarity function between observations. Given a discrete signal of N samples, the autocorrelation function is defined as

$$r[\tau] = \sum_{t=1}^{N-\tau} x[t]x[t+\tau]$$

The value is of the ACF is at a local maximum when the lag is equal to the signal's period or its multiples. Autocorrelation is sensitive to structures in signals, making it useful to applications of speech detection. However, in the case of music signals, resonance structures appear hence the need for a better adapted function.

Difference function The Average Magnitude Difference Function (AMDF) (Ross et al. 1974) is the average unsigned difference between $x(t)$ and $x(t+\tau)$.

$$d_{\mathrm{AM}}[\tau] = \frac{1}{N} \sum_{t=1}^{N-\tau} |x[t] - x[t+\tau]|$$

The difference function is at its local minima for lags equal to proper multiples of the signals period. AMDF is more adapted than autocorrelation for applications in music processing.

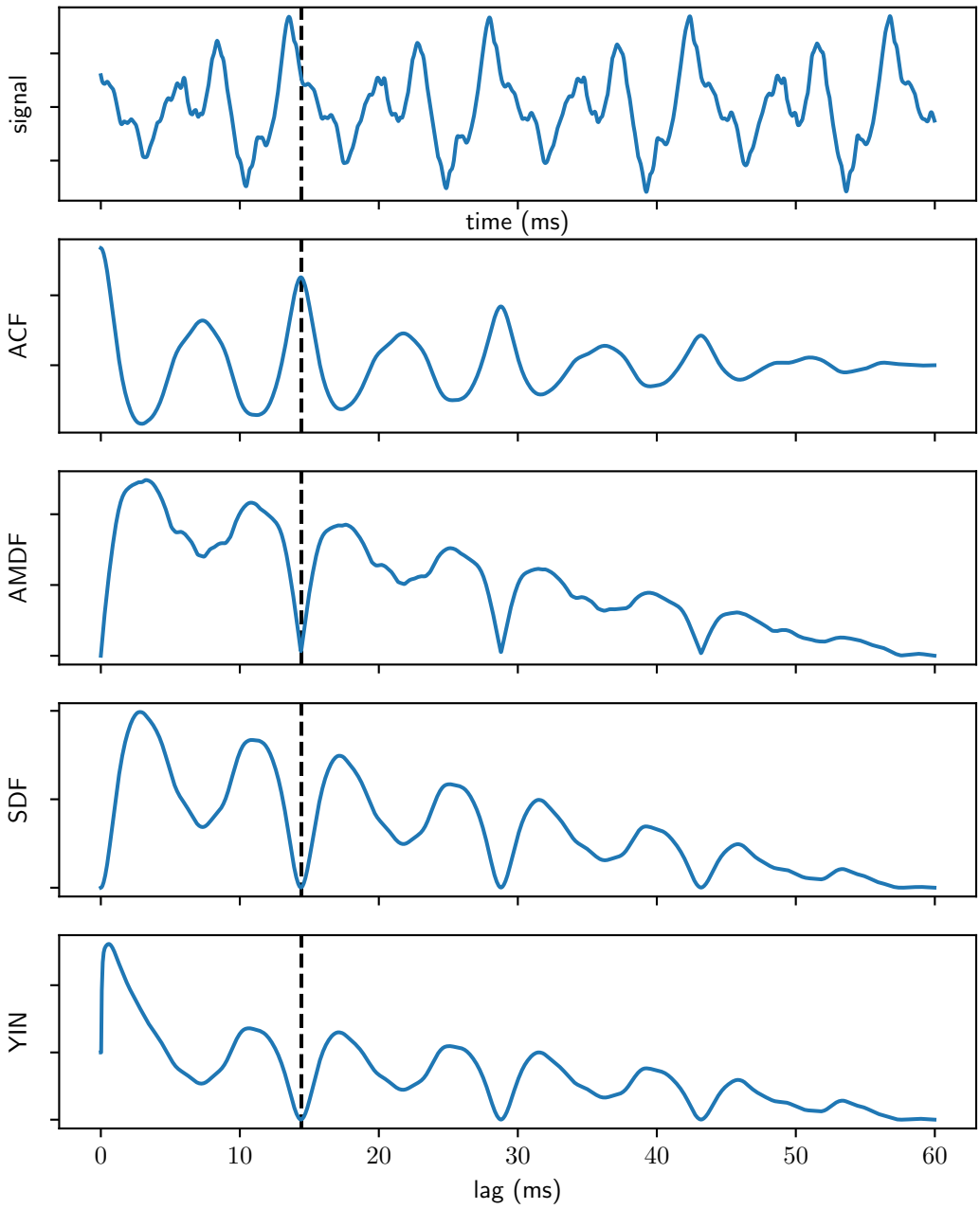
Squared difference function The Squared Difference Function (SDF) is very similar to AMDF, it accentuates however the dips at the signals period therefore indicate local extrema more clearly.

$$d[\tau] = \sum_{t=1}^{N-\tau} (x[t] - x[t+\tau])^2$$

YIN algorithm (Cheveigné and Kawahara 2002) employs the SDF as an auxiliary function for calculating the **cumulative mean normalized difference function** that divides SDF by its average over shorter lags and starts at 1 rather than 0 (in the case of SDF and AMDF); it tends to stay large at short lags and drops when SQD falls under its average.

$$d_{\mathrm{YIN}}[\tau] = \begin{cases} 1 & \text{if } \tau = 0 \\ d[\tau] / \frac{1}{\tau} \sum_{t=0}^{\tau} d[t] & \text{otherwise} \end{cases}$$

```
from muallef.io import AudioLoader
from muallef.plot import diff_functions as df

cello = AudioLoader('samples/instrument_single/cello_csharp2.wav')
cello.cut(start=2, stop=2.06)
df.time_domain_plots(cello.signal, cello.sampleRate, pitch=69.3)
```



2.2.2 Spectral domain

Fourier transform is the most adapted mathematical tool for analysing periodicity in functions. The transform produced a complex function of frequency, where the magnitude of the transform attains its local maxima at the signal's frequency and its *harmonics*.

Spectral domain methods analyse the fourier transform of the signal, which usually gives better results. Nevertheless, similar comparison functions are employed in order to get the fundamental frequency.

Spectral autocorrelation Autocorrelation measures repititive patterns, since harmonics appear at almost fixed frequency intervals, ACF allows to identify harmonic partials. (Lahat, Niederjohn, and Krubsack 1987) The autocorrelation is applied to the spectrum of the signal, that is the magnitude of the fourier transform. The function attains its local maxima at frequency shifts that are multiples of f_0, otherwise the function is attenuated since the partial peaks are not well aligned.

For a spectrum $S[f] = |X[f]|$ with K spectral bins

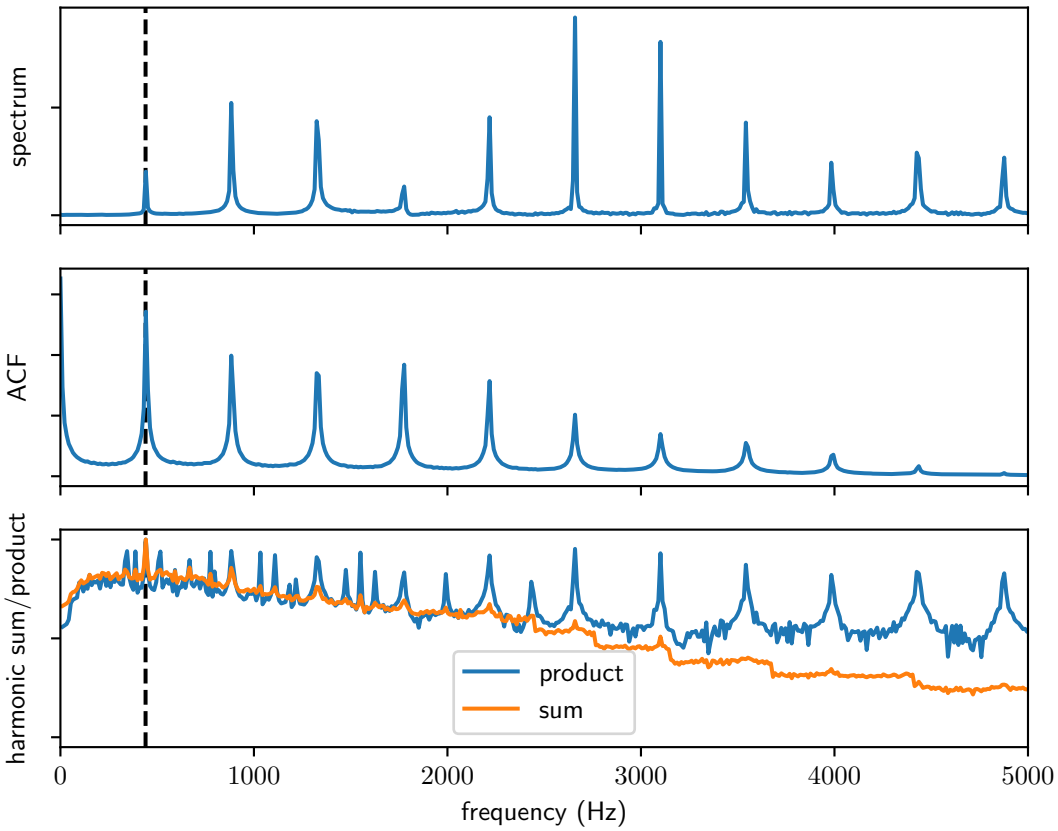
$$R[f] = \sum_{k=1}^{K-f} S[k]S[k+f]$$

Harmonic sum A *frequency histogram* represents the number of occurrences of each frequency, it does not however reflect the *amplitudes* of the harmonics of frequencies. Schroeder proposes to *weight* the contribution of each harmonic to the histogram with a monotonically increasing function of its amplitude, this is done using *log compression* where spectral harmonic bins are compressed with a logarithm. Finally, Schroeder proposed two functions of frequency that sum the compressed weighted histogram. (Schroeder 1968)

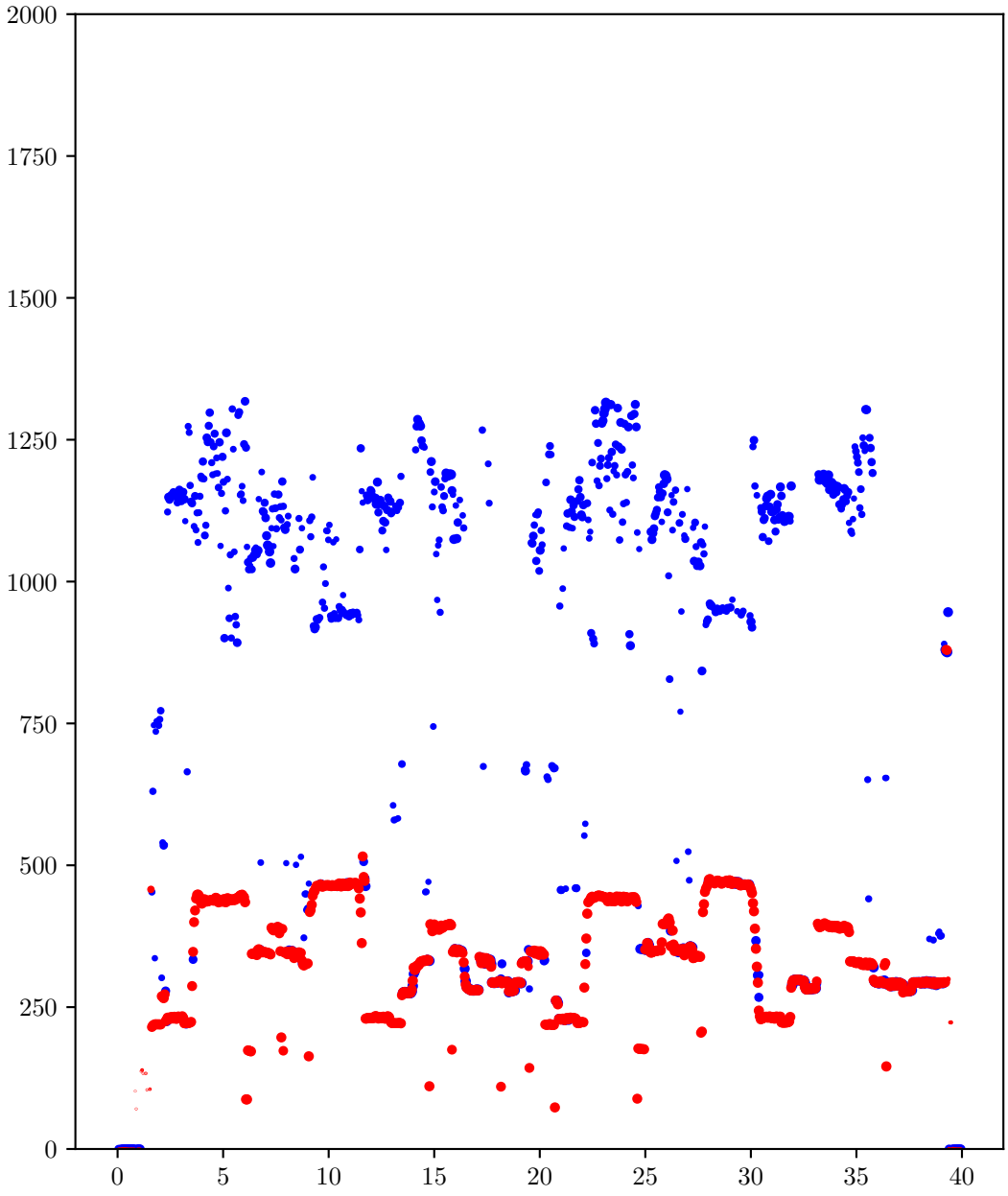
- **Harmonic sum:**
$$\Sigma(f) = \sum_{m=1}^{M} 20\log_1 0 S(nf)$$
- **Harmonic product:**
$$\Sigma'(f) = 20\log_1 0 \sum_{m=1}^{M} S(nf)$$

The sum inside the logarithm in the harmonic product can be viewed as a product because of the properties of the logarithm function.

```
oboe = AudioLoader('samples/instrument_single/oboe_a4.wav')
oboe.cut(start=0.5)
df.spectral_plots(oboe.signal[:4096], oboe.sampleRate, pitch=440)
```




```
from muallef.pitch import MonoPitch

czardas = AudioLoader('samples/monophonic/czardas_cut.wav')

yin = MonoPitch(czardas.signal, czardas.sampleRate, method='yin')
yin_f0 = yin()
yin_conf = yin.confidence
yinfft = MonoPitch(czardas.signal, czardas.sampleRate, method='yinfft')
yinfft_f0 = yinfft()
yinfft_conf = yinfft.confidence
time = czardas.time(len(yin_f0))

fig, ax = plt.subplots()
fig.set_figheight(8)
ax.scatter(time, yin_f0, c='blue', s=10*yin_conf)
ax.scatter(time, yinfft_f0, c='red', s=10*yinfft_conf**2)
_ = ax.set_ylim(0, 2000)
plt.show()
```



Spectral YIN

2.3 Multiple pitch

In polyphonic music analysis, we are interested in detecting the fundamental frequences for concurrent signals, the signals can be produced by several instruments simultanuously.

There are generally two approaches to this problem: iterative estimation and joint estimation. In iterative estimation, the most prominent f_0 is extracted at each iteration until no additional f_0 can be estimated. Generally, iterative estimation models tend to accumulate errors at each iteration step, they are however computationally cheap. Whereas joint estimation methods evaluation f_0 combinations which leads generally to more accurate estimates, however the computational cost is significantly increased.(Benetos et al. 2013)

Let's start by establishing a formalism of the task. The harmonic signal $\tilde{x}(t)$ can be expressed as the sum of M harmonic signals.

$$\tilde{x}(t) = \sum_{m=1}^{M} \tilde{x}_m(t)$$

where $\tilde{x}_m(t)$ is a harmonic monophonic signal similar to signals we've seen so far. It follows, simiilarly to before that:

$$x(t) \approx \sum_{m=1}^{M} \sum_{h=1}^{H_m} A_{m,h} \cos(2\pi h f_{0,m} t + \varphi_{m,h}) + z(t)$$

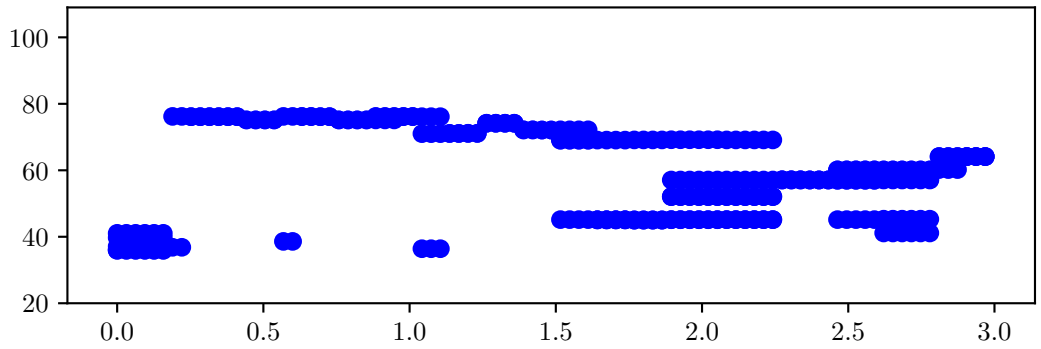
2.3.1 Iterative estimation

```
from muallef.pitch import MultiPitch
from muallef.util.units import Hz_to_MIDI

fur_elise = AudioLoader('samples/polyphonic/furElise.wav')
fur_elise.cut(stop=3)

klapuri = MultiPitch(fur_elise.signal, fur_elise.sampleRate, method='klapuri')
multif0 = Hz_to_MIDI(klapuri())
time = fur_elise.time(multif0.shape[1])

fig, ax = plt.subplots()
for m in range(multif0.shape[0]):
    pitch = multif0[m]
    ax.scatter(time, pitch, c='blue')
```

Klapuri 2006

```
<matplotlib.collections.PathCollection object at 0x7f9e9470e5e0>
<matplotlib.collections.PathCollection object at 0x7f9e94676f40>
<matplotlib.collections.PathCollection object at 0x7f9e9467d280>
<matplotlib.collections.PathCollection object at 0x7f9e9467d610>
<matplotlib.collections.PathCollection object at 0x7f9e9467d9a0>
<matplotlib.collections.PathCollection object at 0x7f9e94676e20>
```

```
_ = ax.set_ylim(20, 109)
plt.show()
```



2.3.2 Joint estimation

2.3.3 Results

3 Temporal segmentation

3.1 Introduction

- definitions of onset/offset
- explain the method

3.2 Onset Detection Function (ODF)

- explain them
- compare them

3.3 Thresholding

3.4 Results

4 Conclusion

References

Benetos, Emmanouil, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. 2013. "Automatic Music Transcription: Challenges and Future Directions." *Journal of Intelligent Information Systems* 41 (December). https://doi.org/10.1007/s10844-013-0258-3.

Cheveigné, Alain de, and Hideki Kawahara. 2002. "YIN, a Fundamental Frequency Estimator for Speech and Music." *The Journal of the Acoustical Society of America* 111 (4): 1917–30. https://doi.org/10.1121/1.1458024.

Lahat, M., Russell J. Niederjohn, and David A. Krubsack. 1987. "A Spectral Autocorrelation Method for Measurement of the Fundamental Frequency of Noise-Corrupted Speech." *IEEE Trans. Acoustics, Speech, and Signal Processing*. https://doi.org/10.1109/TASSP.1987.1165224.

"Nyquist–Shannon Sampling Theorem." 2020. *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Nyquist%E2%80%93Shannon_sampling_theorem&oldid=941933031.

Ross, M., H. Shaffer, A. Cohen, R. Freudberg, and H. Manley. 1974. "Average Magnitude Difference Function Pitch Extractor." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 22 (5): 353–62. https://doi.org/10.1109/TASSP.1974.1162598.

Schroeder, Manfred R. 1968. "Period Histogram and Product Spectrum: New Methods for Fundamental-Frequency Measurement." *The Journal of the Acoustical Society of America*. https://doi.org/10.1121/1.1910902.

Yeh, Chunghsin. n.d. "Multiple Fundamental Frequency Estimation of Polyphonic Recordings," 153.