Pitch Estimation

Pitch Estimation

When looking at audio signals, one possible signal model is to distinguish between harmonic components and noise like components. The harmonic components exhibit a periodic structure in time and it is of course of interest to express this periodicity via the fundamental frequency F_0, i.e. the frequency of the first sinusoidal component of the harmonic source. This fundamental frequency is closely related to the so called pitch of the source. The pitch is defined as how ”low” or ”high” a harmonic or tone-like source is perceived. Although strictly speaking this is a perceptual property, and is not necessarily equal to the fundamental frequency, it is often used as a synonym for the fundamental frequency. We will use the term pitch in this way in the remaining text. It is also of interest how the relationships in terms of energy between the harmonic and noise like components of an audio signal are. One feature expressing this relationship is the Harmonic to Noise Ratio (HNR). The estimation of the pitch and the HNR then can be used e.g. for efficiently coding the signal, or to generate a synthetic signal based on this and other information gained from analysing the signal. In this laboratory we will concentrate on a single audio source, and we will restrict ourselves to speech, which is the primary mode of human interaction. We will use this signals to develop simple estimators for both features and compare the results to state-of-the-art solutions for estimating the pitch and the HNR.

1 Pitch Estimation

As stated above, we model an audio, or to be more specific, an speech signal as a mixture of a harmonic signal and a noise signal:

$$s(t) = h(t) + n(t) \tag{1}$$

where $s(t)$ is the speech signal, $h(t)$ is the harmonic component, and $n(t)$ ist the noise component. For time-discrete signal (and in digital signal processing of course we deal with such time-discrete signals) the equation becomes:

$$s[k] = h[k] + n[k] \tag{2}$$

k being the samples index.

In this section we will have a closer look at the harmonic component $h(t)$, which can be expressed as the sum of its partial tones, which are sinusoidals where the frequencies of the individual partial tones are integer multiples of the fundamental frequency:

$$h(t) = \sum_{n=1}^{N} a_n sin\left(\frac{2\pi n}{F_0} + \phi_n\right)$$

where a_n are the individual amplitudes and φ_n are additional phases for the individual partial tones. Unfortunately in real world signals like speech typically neither the amplitudes nor the fundamental frequency stay constant over the whole duration of the signal. But when looking closer at for e.g. speech, we see that this parameters normally only change slowly over time. This behaviour gives us the possibility to assume that the parameters stay constant if we compart the signals into small enough sections in time. Such signals are called *quasi-stationary*. So the first step towards a pitch estimation is to divide the signal into small enough blocks. The length of the block is determined by the lowest pitch we like to detect, for most algorithms at least two periods of the signal should be contained within one block to give a reliable estimate. Table 1 gives a rough overview of the pitch ranges in human speech.

	lower limit	upper limit
male	75 Hz	150 Hz
female	125 Hz	250 Hz
child		600 Hz

Table 1: typical fundamental frequencies in human speech

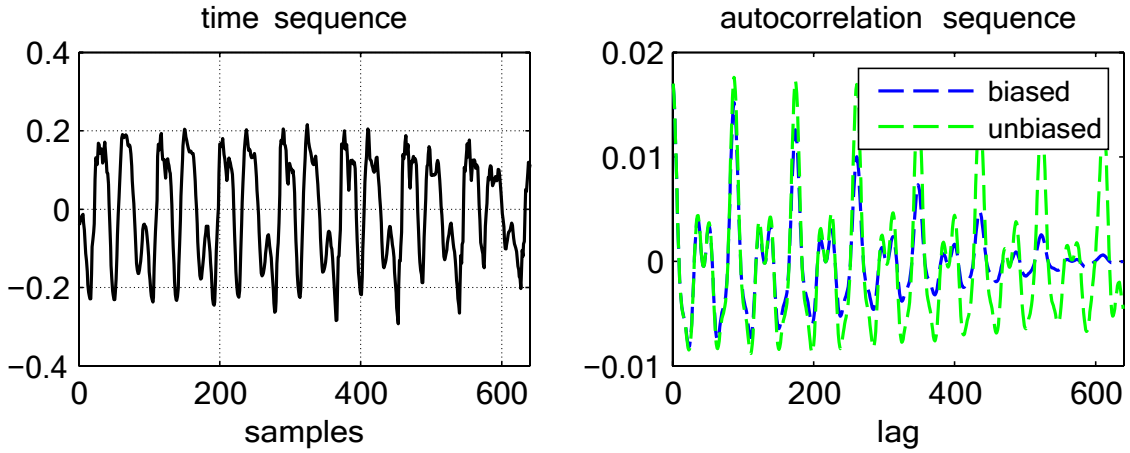

Figure 1: Comparison of the biased and unbiased autocorrelation sequence for a periodic signal (part of a vowel of a male speaker).

The simplest way would now be to just use the zero crossings of the signal. But although this method is very efficient it is not well suited if higher partials have amplitudes or if the noise component is very strong.

So most pitch algorithms are based on other methods, for a simple overview go to [1].

In this laboratory we will develop a estimation algorithm based on the autocorrelation [2]. For discrete time signals the autocorrelation is defined as:

$$R_{xx}[l] = \lim_{N\to\infty} \frac{1}{N} \sum_{k=-N}^{N} x[k]x[k-l] \qquad (4)$$

where *l* is the so called lag. Of course this is the definition for signals of infinite length, but we already divided our signal into blocks of length N each, so the autocorrelation becomes (in its *biased* form):

$$R_{xx}[l] = \frac{1}{N} \sum_{k=l}^{N-1} x[k]x[k-l] \qquad (5)$$

We only consider positive lags since the resulting autocorrelation sequence is symmetric around $l = 0$. Another form of the autocorrelation is the so called *unbiased* autocorrelation sequence

$$R_{xx}[l] = \frac{1}{N-l} \sum_{k=l}^{N-1} x[k]x[k-l] \qquad (6)$$

The difference between unbiased and biased autocorrelation is that the unbiased takes the decreasing number of samples involved in the summation into account. When looking at figure 1 we observe the difference between the biased and the unbiased autocorrelation, the biased tapers off towards high lags. When we compare the autocorrelation equations with our assumption that the signal is

periodic with a periodicity $T_0 = 1/F_0$:

$$x[k] \approx [k + mT_0],\ m \in \mathbb{Z} \tag{7}$$

we see that for such a signal we can expect local maxima of the autocorrelation sequence for lags that are a multiple of T_0. By finding the maximum of the autocorrelation we get an estimate of the fundamental frequency. Note that the autocorrelation function always has a maximum at $l = 0$, so to not erroneously detecting the zero lag as maximum, it is wise to restrict the search within lags that correspond to the upper and lower limits of the fundamental frequency range under consideration. Also the found global maximum might not be at the lag corresponding to the true fundamental frequency but can possibly be an integer multiple of that. Furthermore note that due to this, the estimate can jump between lags in consecutive frames leading also to jumps in the F_0-estimate. For a more robust estimation this must be taken into account.

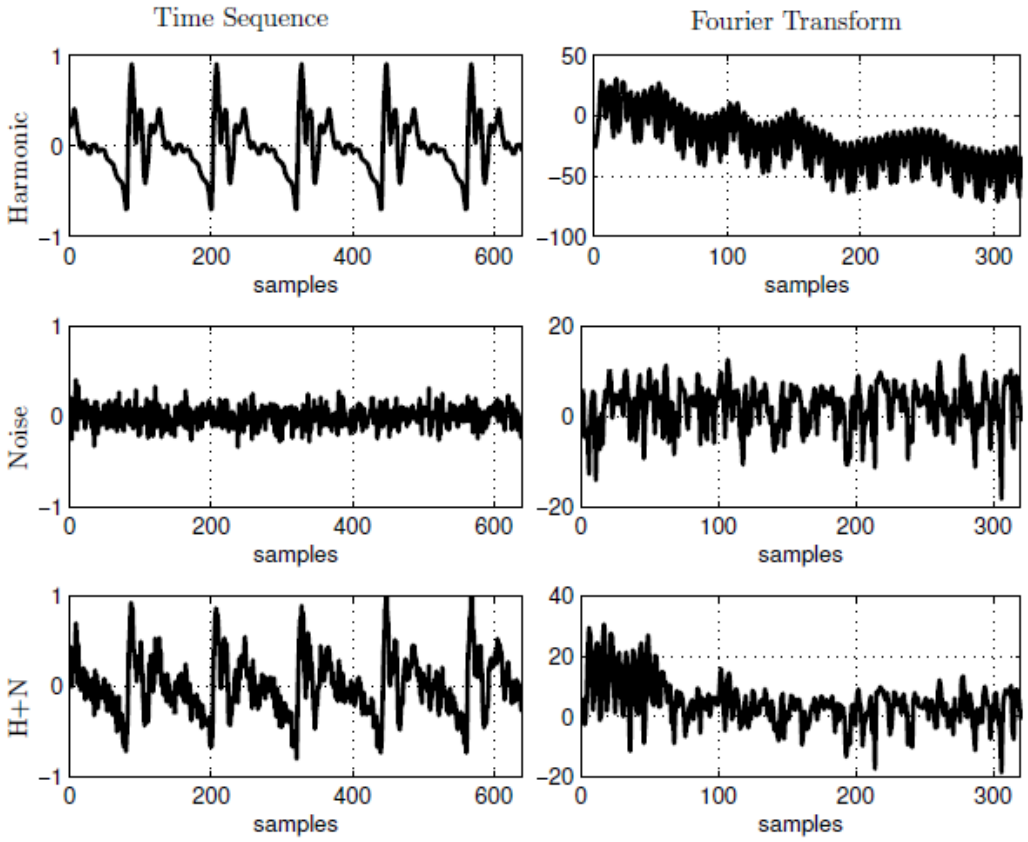

Figure 2: Example of a signal consisting of a harmonic part and a noise part.

Pitch Estimation based on Comb Filter:

The fundamental frequency is the lowest frequency component of a complex waveform that determines its perceived pitch. When you hear a musical note or any sound with a discernible pitch, what you're essentially hearing is the fundamental frequency of the sound wave.

For example, when you pluck a guitar string, the vibration produced by the string generates a complex waveform. This waveform consists of a fundamental frequency (the pitch you hear as the note) and various harmonics, which are integer multiples of the fundamental frequency. The fundamental frequency corresponds to the pitch of the note produced by the guitar string.

The fundamental frequency is called "fundamental" because it's the primary or foundational frequency upon which the other harmonics are built. It's often the most prominent frequency component in a sound wave and plays a crucial role in determining the perceived pitch of the sound.

Pitch estimation based on a comb filter is a technique commonly used in digital signal processing to determine the fundamental frequency (pitch) of a signal, particularly in the context of audio processing. The comb filter method relies on the principle of spectral analysis to identify the periodicity of a signal, which corresponds to its pitch.

Here's a detailed explanation of how pitch estimation using a comb filter works:

1. **Basic Concept of Comb Filtering:**
    - A comb filter is a type of filter characterized by regularly spaced notches or peaks in its frequency response. These notches or peaks resemble the teeth of a comb, hence the name.
    - In the context of pitch estimation, the comb filter is designed to emphasize or accentuate harmonics that are integer multiples of the fundamental frequency of the signal.
2. **Algorithm Overview:**
    - The basic idea is to create a comb filter with a fundamental frequency that is adjustable. This filter is then applied to the input signal.
    - By varying the spacing between the notches or peaks of the comb filter, the algorithm tries to find the spacing that best aligns with the harmonics present in the input signal. This spacing corresponds to the estimated fundamental frequency or pitch.
3. **Implementation Steps:**

- **Designing the Comb Filter:**
  - Choose a suitable comb filter structure. One common choice is the Finite Impulse Response (FIR) comb filter.
  - Determine the spacing between the notches or peaks of the comb filter. This spacing corresponds to the expected fundamental frequency range of the input signal.
  - Design the filter coefficients to create the desired comb filter response.
- **Applying the Comb Filter:**
  - Convolve the input signal with the comb filter. This operation accentuates the harmonics aligned with the spacing of the comb filter while attenuating others.
  - The result of this convolution will have peaks or energy spikes at frequencies corresponding to the harmonics of the fundamental frequency.
- **Pitch Estimation:**
  - Analyze the output of the comb filtering operation to identify the peaks.
  - Determine the spacing between the peaks. This spacing corresponds to the estimated fundamental frequency.
  - Convert the spacing to a frequency value, which represents the estimated pitch of the input signal.

4. **Refinement Techniques:**
   - **Peak Picking:** Apply techniques such as peak picking to accurately identify the peaks in the output spectrum of the comb filter.
   - **Interpolation:** Use interpolation methods to refine the estimated fundamental frequency by interpolating between the peaks in the output spectrum.
5. **Challenges and Considerations:**
   - Noise Sensitivity: Comb filter-based pitch estimation can be sensitive to noise, which may introduce spurious peaks in the output spectrum.
   - Resolution: The accuracy of the pitch estimation depends on the resolution of the comb filter and the spacing between its notches or peaks.
   - Computational Complexity: Depending on the implementation, comb filter-based pitch estimation can be computationally intensive, especially for real-time applications.
6. **Applications:**
   - Pitch correction in audio processing applications.
   - Music transcription and analysis.
   - Speech processing for tasks such as speech recognition and synthesis.

Overall, pitch estimation using a comb filter is a powerful technique for extracting the fundamental frequency of a signal, particularly in scenarios where the signal contains harmonics with a clear periodic structure. However, like any pitch estimation method, it has its limitations and requires careful consideration of factors such as noise robustness and computational efficiency.

Pitch estimation using a comb filter.

1. **Comb Filter Equation**:

A comb filter can be represented mathematically as a system with impulse responses spaced at regular intervals. The impulse response of a comb filter can be described as:

$$h(n)=\delta(n)+\delta(n-D)+\delta(n-2D)+\ldots$$

where:

- : $h(n)$ is the impulse response of the comb filter,
- $\delta(n)$ is the Dirac delta function,
- D is the spacing between the notches or peaks in the comb filter.

2. **Convolution**

   To apply the comb filter to an input signal $x(n)$, we perform convolution:

   $$y(n)=\sum_{k=-\infty}^{\infty} x(k)\cdot h(n-k)$$

   where:

- $y(n)$ is the output signal,
- $x(n)$ is the input signal,
- $h(n)$ is the impulse response of the comb filter.

3. **Peak Detection**:

   After applying the comb filter, we look for peaks in the filtered signal $y(n)$. Peaks in the filtered signal correspond to constructive interference between the comb filter and the harmonics present in the input signal.

4. **Pitch Estimation**:

   The spacing between adjacent peaks in the filtered signal corresponds to the fundamental period of the input signal. We can estimate the pitch f_0 using this period:

   $$f_0=1/T_0$$

   where:

- f_0 is the fundamental frequency (pitch),
- T_0 is the period between adjacent peaks.

5. **Refinement**:

   Depending on the application, further refinement steps may be applied. For example, you might use interpolation techniques to estimate the exact position of the peaks, especially if they fall between discrete samples.

6. **Comb Filter Design**:

The design of the comb filter involves selecting an appropriate spacing D between the notches or peaks. This spacing depends on the expected range of pitches in the input signal and the resolution required for accurate pitch estimation.

By adjusting the parameters of the comb filter and analyzing the peaks in the filtered signal, we can estimate the pitch of the input signal. However, it's worth noting that pitch estimation based solely on comb filtering may have limitations, especially in the presence of noise or complex audio signals. In practice, it's often used as part of a larger pitch estimation system that incorporates multiple techniques for improved accuracy and robustness.

Pitch estimation based on a harmonic sine wave model

Pitch estimation based on a harmonic sine wave model is a fundamental technique used in various audio processing applications, particularly in music and speech processing. The goal is to estimate the fundamental frequency (pitch) of a given audio signal, which represents the perceived pitch of the sound. Here's how it can be done using a harmonic sine wave model:

1. **Preprocessing**:
   - Convert the audio signal into a time-domain representation, typically a sequence of samples.
   - Apply any necessary preprocessing steps such as noise reduction, filtering, or normalization to enhance the quality of the signal.
2. **Frame Segmentation**:
   - Divide the signal into smaller frames of fixed duration (e.g., 10-30 milliseconds).
   - Overlapping frames are often used for smoother analysis.
3. **Windowing**:
   - Apply a window function (e.g., Hamming, Hanning) to each frame to reduce spectral leakage.
4. **Frequency Domain Analysis**:
   - Apply the Fourier Transform (usually Fast Fourier Transform, FFT) to each frame to convert it from the time domain to the frequency domain.
   - Calculate the magnitude spectrum of each frame.
5. **Harmonic Model**:
   - Identify the peaks in the magnitude spectrum corresponding to harmonic frequencies.
   - Harmonic frequencies are integer multiples of the fundamental frequency.
   - Use methods like peak picking or spectral peak tracking to detect harmonic peaks.
6. **Pitch Estimation**:
   - Once harmonic peaks are detected, estimate the fundamental frequency (pitch).
   - The fundamental frequency can be estimated by analyzing the spacing between harmonic peaks.
   - Common methods include autocorrelation, cepstral analysis, or using the average spacing between harmonic peaks.
7. **Refinement**:
   - Refine the estimated pitch by considering factors such as the strength of harmonic peaks, presence of noise, and temporal continuity.
   - Techniques like pitch tracking algorithms (e.g., Kalman filtering, dynamic programming) can be employed for better accuracy and smoothness in pitch tracking.
8. **Post-processing**:
   - Perform any necessary post-processing steps, such as smoothing or interpolation, to improve the accuracy and stability of the estimated pitch trajectory.
9. **Output**:
   - Output the estimated pitch trajectory over time, which represents the perceived pitch contour of the audio signal.
10. **Evaluation**:
    - Evaluate the performance of the pitch estimation algorithm using metrics such as pitch tracking error, accuracy, and robustness to various audio conditions (e.g., noise, pitch variation).

By following these steps, a pitch estimation algorithm based on a harmonic sine wave model can accurately estimate the fundamental frequency of audio signals, which is crucial for applications like music transcription, speech analysis, and pitch correction.

1. **Signal Representation**:
   - The input audio signal $x(t)$ is represented as a discrete-time signal, typically sampled at a rate of f_s Hz. Let $x[n]$ denote the discrete signal.
2. **Frame Segmentation**:
   - Divide the signal $x[n]$ into frames of length N samples with a hop size H samples between consecutive frames.
3. **Windowing**:
   - Apply a window function $w[n]$ to each frame $x_w[n]=x[n]\cdot w[n]$, where $n=0,1,2,...,N-1$.
   - Common window functions include Hamming $w[n]=0.54-0.46\cos(\frac{2\pi n}{N-1})$ or
   - Hanning $w[n]=0.5-0.5\cos(\frac{2\pi n}{N-1})$.
4. **Frequency Domain Analysis**:
   - Compute the Discrete Fourier Transform (DFT) of each windowed frame $x_w[n]$ using FFT: $X_w[k]=\sum_{n=0}^{N-1} x_w[n]\cdot e^{-j2\pi nk/N}$
   - Calculate the magnitude spectrum $|X_w[k]|$ for each frame.
5. **Harmonic Model**:
   - Identify peaks in the magnitude spectrum corresponding to harmonic frequencies.
   - Harmonic frequencies are integer multiples of the fundamental frequency f_0.
   - Let f_0 denote the estimated fundamental frequency.
6. **Pitch Estimation**:
   - Estimate f_0 based on the spacing between harmonic peaks.
   - Common methods include autocorrelation and cepstral analysis.
7. **Refinement**:
   - Refine the estimated pitch considering factors such as peak magnitude, noise, and temporal continuity.
   - Apply techniques like pitch tracking algorithms (e.g., Kalman filtering, dynamic programming) for refinement.
8. **Output**:
   - Output the estimated fundamental frequency f_0 over time, representing the pitch contour of the audio signal.

Mathematically, the pitch estimation involves analyzing the harmonic structure of the signal's spectrum to determine the fundamental frequency. This often requires careful peak detection, frequency analysis, and refinement techniques to accurately estimate the pitch, especially in the presence of noise or varying signal characteristics.

The actual implementation may vary depending on the specific requirements and the chosen algorithm for pitch estimation.