Pitch Tracking for an automatic annotation of voice signals

1st Tomás Agustín González Orlando
Procesamiento de Voz
Instituto Tecnológico de Buenos Aires
Buenos Aires, Argentina
togonzalez@itba.edu.ar

2nd Juan Manuel Romarís
Procesamiento de Voz
Instituto Tecnológico de Buenos Aires
Buenos Aires, Argentina
jromaris@itba.edu.ar

***Abstract*—Different implementations of pitch tracking algorithms for automatic annotation of voice signals are proposed.**

Both the design and testing of the algorithms were based on the PTDB-TUG Database, achieving acceptable results.

***Index Terms*—Speech analysis, Voice signals, Pitch Tracking, High Product Spectrum, Autocorrelation algorithm, YIN, RAPT**

I. Introduction

A. Pitch definition

The pitch of a given speech/voice signal in a specific point in time is defined as the frequency *perceived* by the ear when listening to the particular sound conformed by said point in time and its surroundings. Although this is a qualitative measure and the perceived frequency of each sound is subjective to the listener, the quantitative definition of pitch that is now commonly accepted by the engineering and scientific community is the frequency separation of the different harmonics of that sound, disregarding the person or source from which that sound was produced. As a consequence, assuming these harmonics exist, the fundamental frequency or pitch of a voice signal may not have a considerable amplitude when analyzing its spectrum by means of a Short Time Fourier Transform (STFT). On the other hand, the frequency separation between the different harmonics will indeed define this pitch. In practical situations, this characteristics of the pitch may bring difficulties when trying to detect and track pitch variations of a speech signal composed by several phonemes, as this frequency may not be present in the spectrum, and even if it is present, it will not necessarily be represented by a global or local maxima in that spectrum. Several algorithms have arise to accomplish a successful tracking of this pitch. In particular, High Product Spectrum, Autocorrelation and YIN algorithms were implemented and explained below.

B. Pitch tracking - usage

The variability of the fundamental frequency (F_0) is large by nature. This frequency will vary several times within the same speech signal, even in intervals of seconds or fraction of a second. This variability is known to differ between people (for lower average male voices the frequency is 70–200 Hz, and for women it can reach 400 Hz), but it also depends considerably on the current state of the speaker. The state of the speaker would then be defined by her/his emotional state, the message she/he wants to portray, her/his real intentions and physical condition.

As a consequence, F_0 may be used in a wide range of different solutions. Some of them are mentioned below:

- Emotion Recognition [1]
- Sex determination (male/female voices)
- Speaker deterioration, or splitting the speech into phrases
- Detection of the pathological characteristics of the voice such as signs of Parkinson's disease [2]

C. Pitch Tracking Database from Graz University of Technology (PTDB-TUG)

The Pitch Tracking Database from Graz University of Technology (PTDB-TUG) is a speech database for pitch tracking that provides microphone and laryngograph signals of 20 English native speakers as well as reference pitch trajectories. Each subject read 236 out of 2342 phonetically rich sentences from the existing TIMIT corpus. The text material was selected such that each sentence was spoken by at least one female and one male speaker. In total this database consists of 4720 recorded utterances. All recordings were carried out on-site at the recording studio of the Institute of Broadband Communications at Graz University of Technology.[3]

Both laryngograph and microphone signals were digitized at 48kHz and with 16 bit resolution. Pitch annotations were provided utilizing the RAPT algorithm that will be explained later in sections below. This reference pitch was extracted using 32ms analysis windows with 10ms hopsize.

The database was validated by an external validator who was not involved in the specification and recording process. The validation included inspection of the signal files with

[1] Farrús, M., Hernando, J., Ejarque, P. Jitter and Shimmer Measurements for Speaker Recognition. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, vol. 2 (2007), pp. 1153–1156.

[2] Rusz, J., Cmejla, R., Ruzickova, H., Ruzicka, E. Quantitative acoustic measurements for characterization of speech and voice disorders in early untreated Parkinson's disease. The Journal of the Acoustical Society of America, vol. 129, issue 1 (2011), pp. 350–367.

[3] Pirker, Gregor, et al. "The Pitch-Tracking Database from Graz University of Technology." 22 Aug. 2012, www2.spsc.tugraz.at/databases/PTDB-TUG/DOCUMENTATION/PTDB-TUG$_{R}EPORT.pdf$.

respect to format, sound length, clipping, and DC offset. Furthermore, a small set of randomly chosen transcriptions (1%) was manually compared with their corresponding signal files for accuracy and completeness.[4]

II. Voiced/Unvoiced detection algorithm

Speech pitch tracking computation has to be realized only when speech is present in the audio to be analyzed. For this very important reason the use of a Voice Activity Detection algorithm (VAD) was considered for this project. Instead of developing a VAD, for this project the VAD created by the Institut national de l'audiovisuel (abbrev. INA) was utilized. This implementation can be found in [5]. This implementation provides a toolkit for voice segmentation and speech, music and noise detection. Utilizing this VAD, non-voice segments were separated from voiced segments, applying pitch detection algorithms only on voiced frames.

III. Pitch Detection Algorithms

In this section some of the known Pitch detection Algorithms will be introduced and briefly explained.

A. *Modified short-time Auto-correlation Algorithm*

This algorithm utilizes the modified short-term auto-correlation function defined as:

$$R_x(m) = \sum_{m=0}^{L-1} x(n+m)x(n+m+k), 0 \leq k < K$$

Where L is the number of signal samples used in the computation of the correlation, and K is the number of auto-correlation points to be computed. This modified short-time auto-correlation functions solves the bias problem that the normal short-time auto-correlation function presents.

It can be proven that if a function is originally periodic with period τ, its modified short-time auto-correlation function will be periodic with the same period. It can also be seen that given a lag $m = 0$, the modified short-time auto-correlation function will yield its maximum value, being the energy of the given signal in the desired window. Taking both things previously mentioned into account, the modified short-time auto-correlation function of a periodic function will have maximums in:

$$R_x(0) = R_x(\tau)$$

Taking this into account, by searching maximums in the modified short-time auto-correlation function, a signal's period can be estimated. However, although the modified short-time auto-correlation function of a section of voiced speech generally displays a prominent peak at the pitch period, modified short-time auto-correlation peaks due to the detailed formant structure of the signal are also present.

To reduce the effects of the formant structure on the tailed shape of the short-time modified short-time auto-correlation function two preprocessing non-linear functions where used prior to the modified short-time auto-correlation computation as shown in the following figure:

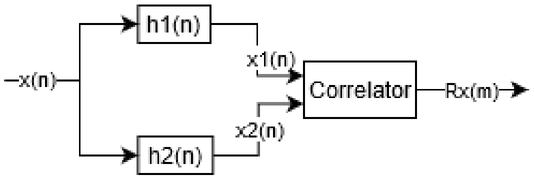

Fig. 1. Block Diagram of the nonlinear correlation processing

The non-linear preprocessing function applied is the combination of a center and peak clipper with the following input-output relation:

$$y(n) = \begin{cases} 1 & x(n) \geq C_L \\ 0 & |x(n)| < C_L \\ -1 & x(n) \leq -C_L \end{cases}$$

Where C_L is the clipping threshold value. It can be argued that a center clipper effectively attenuates the effects of first formant structure on the waveform, without seriously affecting the pitch pulse indications. The value given to the threshold previously mentioned is set as a fixed percentage (68 %) of the smaller of the maximum absolute signal level over the first and last one thirds of the analysis frame.[6]

B. *Harmonic Product Spectrum Algorithm (HPS)*

This algorithm is based on a frequency domain analysis of the speech signal. This analysis can be either done utilizing the Fast Fourier Transform or the Discrete Cosine Transform. Conventionally it is donde utilizing the FFT but as shown in [7], utilizing the DCT is also possible.

The Harmonic Product Spectrum Algorithm is derived from windowing the speech signal to be analysed and then applying the following steps to each window:

1) Obtain the FFT/DCT of the windowed signal
2) Down-sample the signal by a factor of a, where $a\epsilon[2;n]$, n being the number of harmonics to take into account.
3) Multiply all FFTs/DCTs for the windowed signal.
4) Obtain the frequency of the maximum peak of the resulting FFT/DCT. This frequency will be, in principle, the pitch of that time frame of the speech signal.
5) If the amplitude of the second maximum of the resulting FFT/DCT is located at a frequency that is half the frequency of the selected pitch, and the ratio between these amplitudes is above a certain threshold, then the pitch should be corrected to half its value.

[4] Pirker, Gregor, et al. A Pitch Tracking Corpus with Evaluation on Multipitch ... www.spsc.tugraz.at/system/files/InterSpeech2011Master$_0.pdf$.

[5] https://github.com/ina-foss/inaSpeechSegmenter

[6] L. Rabiner, "On the use of autocorrelation analysis for pitch detection," in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 25, no. 1, pp. 24-33, February 1977, doi: 10.1109/TASSP.1977.1162905.

[7] N. Sripriya and T. Nagarajan, "Pitch estimation using harmonic product spectrum derived from DCT," 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013), Xi'an, 2013, pp. 1-4, doi: 10.1109/TENCON.2013.6718976.

A detailed figure of the algorithm procedure and its result is shown below:

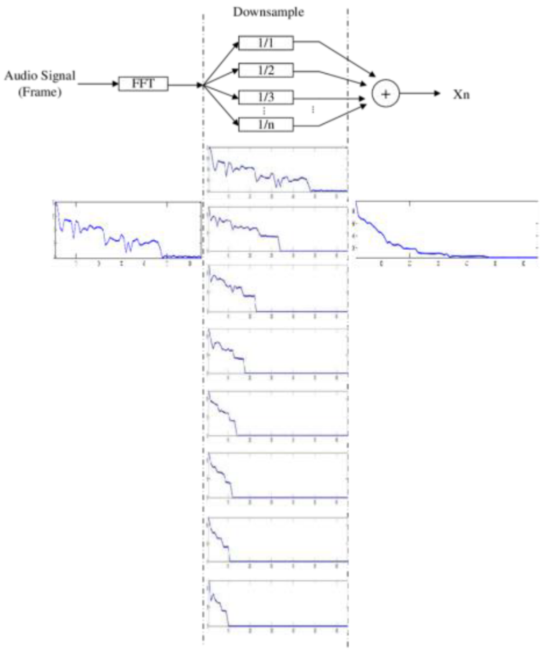

Fig. 2. HPS algorithm

When downsampling the windowed signal, if the fundamental frequency has harmonics, these would theoretically be aligned with the fundamental. After multiplying all the FFTs/DCTs, all harmonics would finally be multiplying among themselves, creating a higher peak, while all other frequencies will be reduced in amplitude. Taking this principle into account it can be seen that the peak with the highest amplitude would be associated with the correct pitch.

However, like many other pitch detection algorithms, the Harmonic Product Spectrum Algorithm can have an error of an octave on the detected fundamental frequency. To solve this problem, the criteria introduced by [8] is mentioned in the last step of the algorithm. According to this paper, for an implementation with five harmonics, the best threshold value is 0.2. Regarding the use of the DCT instead of the FFT , the better decorrelating nature of the DCT spectrum enables the pitch harmonics to appear sharper in its spectrum.Potentially, this facilitates accurate pitch estimation at lower order of the harmonic product spectrum when compared with DFT-based HPS. This algorithm has the disadvantage of needing a greater window length to achieve the correct results than the other mentioned algorithms.

C. YIN Algorithm

The YIN algorithm is proposed by [9]. This algorithm is based on the fundamental equation:

$$x(n) = x(n+\tau)$$

Where we try to find the τ which minimizes the difference between the signal $x(n)$ and $x(n+\tau)$. To achieve this, we calculate the accumulated difference:

$$d_t(\tau) = \sum_{j=1}^{w} (x(n) - x(n+\tau))^2 = 0$$

Thus, this function should be computed and its first zero should be located. The location of this zero corresponds to the period of the fundamental frequency or pitch. The direct code implementation of this function is slow, so a faster approach is found by expanding the square function and applying convolution and FFT properties:

$d_t(\tau) = r_t(0) + r_{t+\tau}(0) - r_t(\tau)$

This search can be difficult and not precise. This is why a new function, based on the last expression, is defined:

$$d'_t(\tau) = \left\{ \begin{array}{ll} 1, & \text{for } \tau = 0 \\ \frac{d_t(\tau)}{\frac{1}{\tau}\cdot\sum_{j=1}^{\tau} d_t(j)}, & \text{for } \tau \neq 0 \end{array} \right\}$$

The peak corresponding to the fundamental frequency is better defined with this new function. To find it, we choose the first minimum that is near zero and below a certain threshold.

D. RAPT Algorithm

Robust Algorithm for Pitch Tracking (RAPT) is a time domain pitch detection algorithm which was originally proposed by Talkin. It is designed to work at any sampling frequency and frame rate over a wide range of possible F0, speaker and noise conditions. It uses normalized cross-correlation (NCCF) for candidate generation. The computational complexity is slightly increased in NCCF to overcome the shortcomings of the other candidate generators like autocorrelation. To reduce the computational load, the NCCF is used in two stages. In the first stage, the signal with low-sample rate is used to generate the set of candidates and the high sample-rate signal is used in the later stage. The NCCF of the low-sample rate signal is computed for all the lags in the F0 range and the locations of the local maxima were recorded in the first pass. The NCCF of the high-sample rate signal is computed only in the vicinity of the promising peaks found in the first pass. It then searches again for the local maxima in the refined NCCF to obtain improved peak location and amplitude estimates. Finally, dynamic programming is used to select the set of NCCF peaks or unvoiced hypotheses across all frames.[10]

IV. Algorithm's implementation

A. Modified short-time Auto-correlation Algorithm

B. Harmonic Product Spectrum Algorithm

C. YIN

1) Real Time Implementation: As YIN is expected to be the algorithm with the highest accuracy of all previously presented algorithms, a real time implementation of YIN is devised.

[8] P. De la Cuadra, “Efficient Pitch Detection Techniques for Interactive Music”

[9] H. Kawahara, A. de Cheveigne, “YIN, a fundamental frequency estimator for speech and music” October 2001.

[10] L. Rabiner, ”On the use of autocorrelation analysis for pitch detection,” in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 25, no. 1, pp. 24-33, February 1977, doi: 10.1109/TASSP.1977.1162905.

A complexity analysis of YIN is performed to evaluate the better performance of the devised implementation when comparing with the direct implementation.

The complexity of the direct implementation of obtaining $d_t(\tau)$,may be expressed using big O-notation as $O(w^2)$.

Following these calculations, we may then proceed to calculate $d'_t(\tau)$ directly too, with complexity $O(w^2)$. The resulting complexity for applying YIN to a single window would then be $O(w^2)$. When applying this method over an amount of m windows of the signal, the complexity for the pitch tracking algorithm of a signal would result in $O(m \cdot w^2)$ complexity. Unfortunately, for the specified window length and overlap, this does not suffice for a real time implementation of the algorithm.

Expanding the square difference for $d_t(\tau)$ in search for a more efficient implementation, we obtain the following equation:

$$d_t(\tau) = P + r(\tau) + z(\tau)$$

Where:

$$P = \sum_{j=1}^{w} x^2(j)$$

is the power of the windowed signal. This is a constant that can be calculated in linear time $O(w)$.

$$r(\tau) = \sum_{j=1}^{w} x^2(j) \cdot x^2(j+\tau)$$

is the correlation function of the windowed signal. This may be calculated for every τ at the same time using the FFT with $O(w \cdot log(w))$ complexity.

$$z(\tau) = \sum_{j=1}^{w} x^2(j+\tau)$$

is the power of the windowed signal moved by a factor of τ. Taking into account the fact that the signal is 0 outside the interval [0, w] we rewrite $z(\tau)$ as:

$$z(\tau) = P - \sum_{j=1}^{\tau} x^2(j)$$

We may then define $z(\tau)$ as a recursive function where:

$$z(\tau+1) = z(\tau) - x^2(\tau+1)$$

This may be computed for every τ in linear time. As τ_{max} was chosen to be w, then this results in $O(w)$ complexity.

As for the calculation of $d'_t(\tau)$, we observe that the sum $g(\tau) = \sum_{j=1}^{\tau} d_t(j)$ can also be re utilized knowing that $g(\tau+1) = g(\tau) + d(\tau+1)$, so that

$$d'(\tau+1) = \frac{d_t(\tau+1)}{\frac{1}{\tau+1} \cdot (g(\tau) + d(\tau+1))}$$

By use of this method, we obtain $d'(\tau)$ for every τ in $O(w)$.

We therefore conclude that the total complexity of applying YIN to a single window is $O(w \cdot log(w))$, and for the total speech signal would be $O(m \cdot w \cdot log(w))$.

This final implementation method required 700 times less operations than the direct implementation and was able to perform the tracking of a 7 second signal in 1 second, achieving real time pitch tracking.

2) Implementation Details: The implemented algorithm has several differences with the YIN proposed by the bibliography. The implemented algorithm is different from the literature in that:

- A CMDF that has less than a specific amount of peaks (40 peaks for females, 25 for males) is considered to sonourus, otherwise the signal is considered not sonorous.
- Only sonorous signals are analyzed for determination of its pitch, for only the range of taps that determine a pith frequency between 80 Hz and 600 Hz is analyzed.
- In that interval, the minimum peak of the CMDF is determined, and that tap will be translated to the frequency that it represents by the equation $\frac{f_s}{\tau}$
- If the signal is considered to be sonorous and no peaks are found on the specific interval, then the pitch is marked as invalid and the signal is considered to be not sonorous too.

A figure showing the negative version of the CMDF of a sonorous signal (for a better visualization of peaks) is shown below:

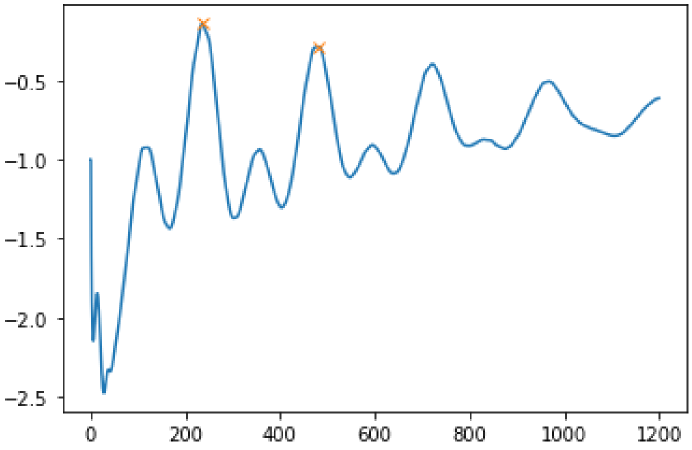

Fig. 3. Typical CMDF of a sonorous signal - negative version

From this figure, we observe few peaks, all of them equally spaced (the fundamental and its octaves). The fundamental frequency is easily identifiable by said algorithm.

A figure showing the CMDF for a non-sonorous signal (for a better visualization of peaks) is shown below:

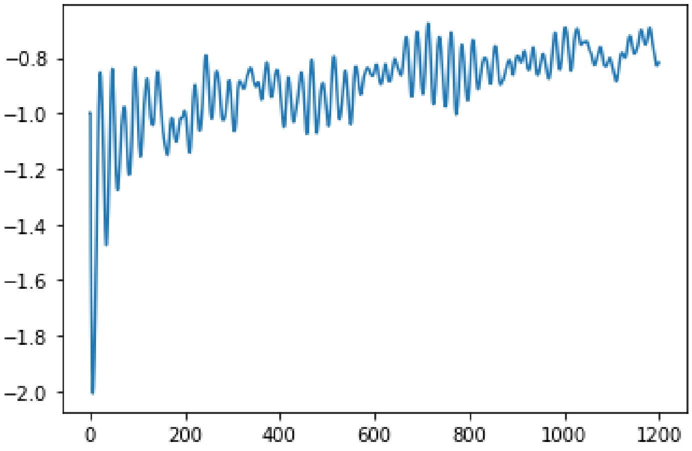

Fig. 4. Typical CMDF of a non-sonorous signal - negative version

As the reader may notice from the above, figure, several peaks prevails at different taps, and no fundamental frequency is detected.

V. Results and Analysis

A. Gross error

A gross error is defined as a frame for which the calculated pitch of the signal has at least a 20 percent difference in value in comparison with the reference value obtained from the database for that particular frame. To evaluate the performance of the algorithms, for a given speech signal, we calculate $\frac{G_e}{T}$, where:

- G_e is the amount of gross errors for the signal
- T is the amount of sample of that same signal

The tests were run on a set of 120 samples (60 male samples, 60 female samples), from 30 different subject. For each randomly selected subject, two voice samples were taken from the database (this selection was also random) and tested on one of the algorithms. $\frac{G_e}{T}$ for each algorithm was obtained and analysed.

B. YIN

For the test set of this algorithm the following results were obtained:

- Females' Mean Error (percent): 6.63
- Females' STD (percent): 2.49
- Samples: 60

The distribution of the gross error for each female sample is shown below:

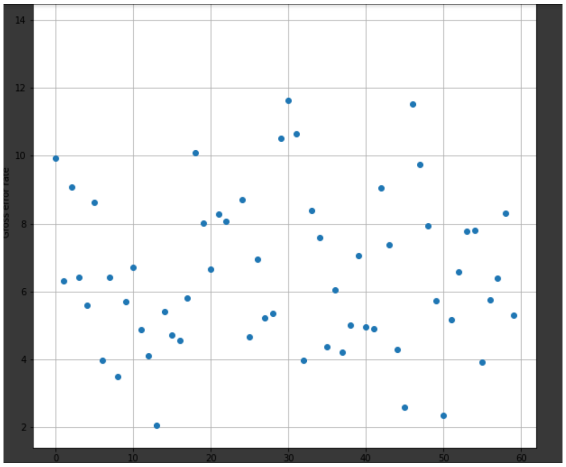

Fig. 5. Gross error distribution for female subjects

- Males' Mean Error (percent): 6.63
- Males' STD (percent): 2.49
- Samples: 60

The distribution of the gross error for each male sample is shown below:

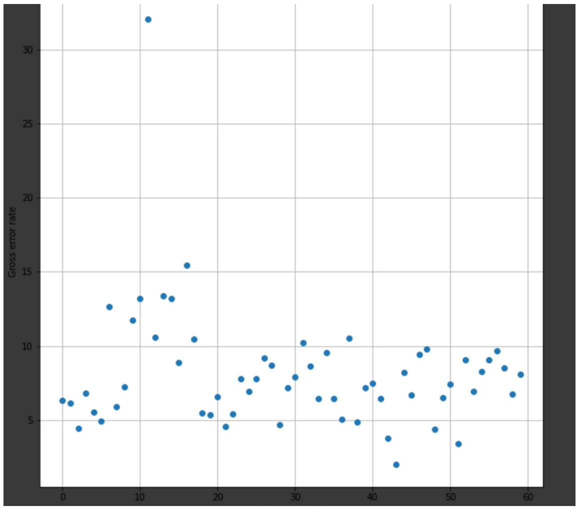

Fig. 6. Gross error distribution for male subjects

Better results were obtained for female subjects. For YIN, the parameters for the algorithm were more finely tuned for females than for males, so these results were expected.

After running a detailed analysis on the cause of high gross errors, several reasons were found to affect the results:

1) The used VAD was not always accurate and added some error to the measurements, increasing the occurence of gross errors.
2) The detection of sonourous or non-sonorous signals was the main responsible for the gross errors of the algorithm: Signals that were correctly classified as sonorous would not get gross error, but several times the algorithm classified the signal as sonorous when it was actually non-sonorous.

The classification of sonorous/non-sonorous signals may be further improved on the future, which is guaranteed to provide better results as shown in the next figure, where the speech signal is marked in blue, the reference pitch is marked in red and the estimated pitch is marked in green:

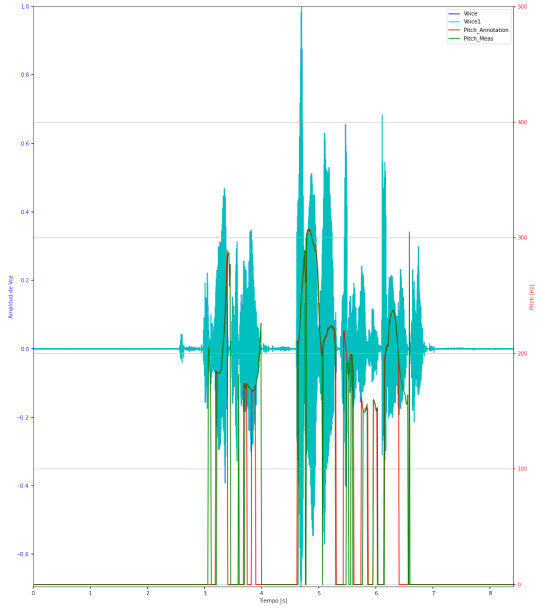

Fig. 7. Speech signal, pitch estimation and reference pitch

An external classifier may also be used to classify a window in sonorous/non-sonorous and thus may contribute to the improvement of the algorithm's precision.

A spectogram analysis was also performed to verify the accuraccy of the database and of the estimation, as shown below:

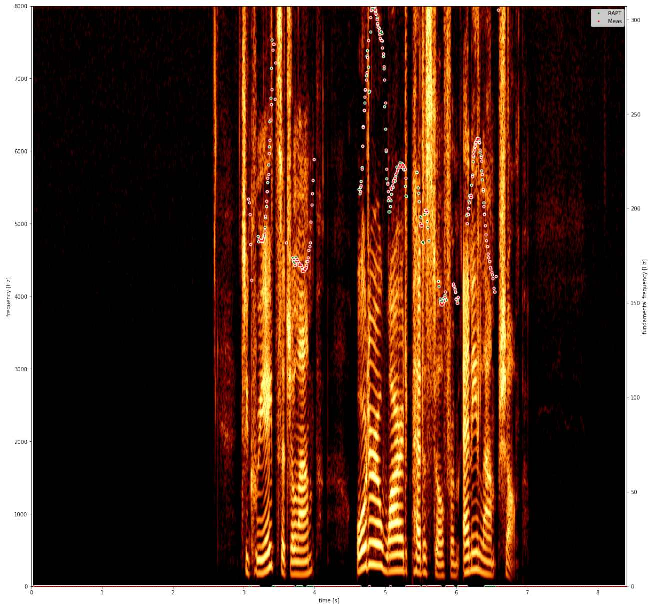

Fig. 8. Spectogram, estimation and reference values

VI. Conclusion

A. YIN

Direct implementation of the proposed YIN algorithm by the literature proved to be both slow for real time implementation and inaccurate for the particular case pitch tracking in voice signals. A method for real time implementation of the algorithm was proposed and proved to be successful for the given database. The modified version of the algorithms obtained acceptable results and there are known ways to radically improve the algorithm: An external classifier may also be used to classify a window in sonorous/non-sonorous and thus may contribute to the improvement of the algorithm's precision.

References

[1] Farrús, M., Hernando, J., Ejarque, P. Jitter and Shimmer Measurements for Speaker Recognition. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, vol. 2 (2007), pp. 1153–1156.

[2] Rusz, J., Cmejla, R., Ruzickova, H., Ruzicka, E. Quantitative acoustic measurements for characterization of speech and voice disorders in early untreated Parkinson's disease. The Journal of the Acoustical Society of America, vol. 129, issue 1 (2011), pp. 350–367.

[3] Pirker, Gregor, et al. "The Pitch-Tracking Database from Graz University of Technology." 22 Aug. 2012, www2.spsc.tugraz.at/databases/PTDB-TUG/DOCUMENTATION/PTDB-TUG_REPORT.pdf.

[4] Pirker, Gregor, et al. A Pitch Tracking Corpus with Evaluation on Multi-pitch ... www.spsc.tugraz.at/system/files/InterSpeech2011Master_0.pdf.

[5] L. Rabiner, ”On the use of autocorrelation analysis for pitch detection,” in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 25, no. 1, pp. 24-33, February 1977, doi: 10.1109/TASSP.1977.1162905.

[6] N. Sripriya and T. Nagarajan, ”Pitch estimation using harmonic product spectrum derived from DCT,” 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013), Xi'an, 2013, pp. 1-4, doi: 10.1109/TENCON.2013.6718976.

[7] hP. De la Cuadra, "Efficient Pitch Detection Techniques for Interactive Music"

[8] H. Kawahara, A. de Cheveigne, "YIN, a fundamental frequency estimator for speech and music" October 2001.

[9] Talkin, David. ”A robust algorithm for pitch tracking (RAPT).” Speech coding and synthesis 495 (1995): 518.

[10] L. Rabiner, ”On the use of autocorrelation analysis for pitch detection,” in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 25, no. 1, pp. 24-33, February 1977, doi: 10.1109/TASSP.1977.1162905.