Rev. Roum. Sci. Techn.– Électrotechn. et Énerg.
Vol. **68**, *4*, pp. 388–393, Bucarest, 2023

Électronique et transmission de l'information
Electronics and Information Technology

COMPARISON OF ALGORITHMS FOR FUNDAMENTAL FREQUENCY DETECTION IN THE CONTEXT OF AUDIO PLUG-INS

SILVIU-MIHAIL NECHIFOR[1], DORIN-MIHAIL DINULESCU[2]

Keywords: Fundamental frequency; Time domain; Frequency domain; Perfect reconstruction; Frame size.

This scientific article presents a comprehensive study of audio signals' fundamental frequency detection methods, focusing on both time-domain and frequency-based approaches and audio file processing since it is crucial for the post-processing part of the audio plug-ins for which this study is intended. Additionally, the article introduces self-repairing algorithms that adaptively identify and correct errors in the detected signals, ensuring robustness and accuracy in signal processing tasks and enhancing the overall detection performance. The findings from this study offer valuable insights into advancing signal processing techniques with broader implications across various domains.

1. INTRODUCTION

When it comes to sound engineering, especially in audio plugins, such as reverb, parametric equalizers, and the infamous autotune, one of the most important parameters of the audio signal we want to process is its fundamental frequency. Therefore, doing it as precisely as possible is mandatory to achieve the expected outcomes. Moreover, time complexity is an aspect in which one should take great interest. Since most of the applications require overlapping frames to prevent the appearance of distortions and artifacts, the second and third sections will cover simple ways to verify that perfect reconstruction is achieved based on some parameters that will be introduced later for both the time domain and frequency domain methods and discuss choosing the size of the frames. The fourth chapter will discuss both time domain and frequency domain fundamental frequency detection algorithms and explain in which scenarios the time domain is better than the frequency ones and vice versa. The last chapter will introduce some self-repairing algorithms that can correct inaccurately detected frequencies.

2. FRAME PROCESSING

2.1. FRAME SIZE

In signal processing and digital audio analysis, a frame size is pivotal in segmenting continuous audio signals into smaller, manageable units for further analysis. The size of the frame is given by (1). The frame size, denoted by N, represents the number of samples contained within each frame and is calculated as the product of the sampling frequency f_s and the frame duration T. Usual values for the frame duration are between 10 and 30 ms. The number of samples must be chosen so that it can be rewritten as a power of 2, and the condition of T is met for the specific sampling frequency of the audio file.

$$N = f_s \cdot T. \tag{1}$$

2.2. VOICED / UNVOICED FRAMES

For audio plug-ins such as autotune, processing only the voiced frames and not introducing artifacts by altering noisy frames, which could have a very high/low fundamental frequency, is essential. Under this section, some criteria can be used to determine if a signal is voiced or unvoiced. If a signal is voiced, all/ most of the following conditions should be met: the detected fundamental frequencies take values between 50 Hz and 500 Hz, the zero-cross rate of the signal should be smaller than an arbitrary value (usually 0.1), and the energy of the frame should be higher than –60 dB [1]. If an application requires more precise algorithms for voiced/unvoiced decision making, [2] can be referred to.

3. PERFECT RECONSTRUCTION

Perfect reconstruction ensures that the processing of each frame will not cause any information loss and overlapping and combining the frames should not cause any aliasing. If perfect reconstruction is not successfully realized, it can lead to various artifacts and distortions in the output, which can significantly degrade the quality of the processed signal. In this section, we will give an overview of the methods to ensure perfect reconstruction is realized for both frequency domain and time domain. For more advanced and efficient methods than the ones presented in this paper, [3] covers some of them.

3.1. TIME-DOMAIN CONDITION

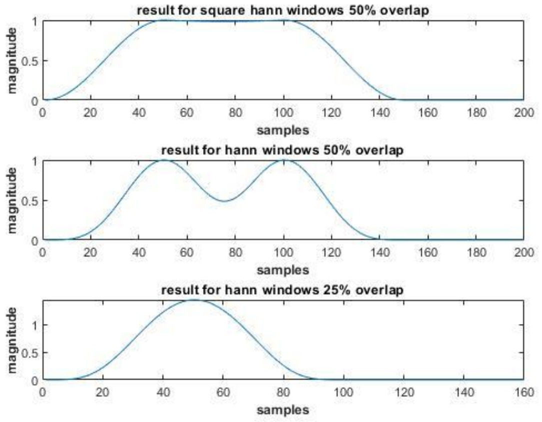
In the time domain, overlapping windows can ensure perfect reconstruction. The most common choice is the Hann window. Two conditions must be met simultaneously for it to take place eqs. (3) and (4). Fulfilling the first condition ensures that all the information from the original signal has been captured and that the tr. The second condition states that the chosen window must be symmetric, which is fulfilled by most window types, such as Hann and Kaiser. For a given window, the amount of overlap will be crucial in assuring perfect reconstruction, and the user should verify for a specific amount of overlap that best fits their application if the two conditions are met. The second condition can be verified straight forward. In Fig. 1, we tested using MATLAB if different types of windows meet the first condition with varying amounts of overlap. As it can be seen only the root Hann window with 50 % overlap ensures perfect reconstruction.

[1] Faculty of Electronics, Telecommunication and Information Technology, Department of Technologies and Telecommunication Systems, National University of Science and Technology POLITEHNICA of Bucharest, Romania, E-mail: silviu.nechifor@stud.etti.upb.ro
[2] Faculty of Mechanical Engineering and Mechatronics, Department of Mechatronics and Precision Mechanics, National University of Science and Technology POLITEHNICA of Bucharest, Romania, E-mail: dorin.dinulescu@stud.mec.upb.ro

DOI: 10.59277/RRST-EE.2023.68.4.11

$$w(k)^2 + w(k+N)^2 = 1, \tag{2}$$

$$w(k) = w(2N-1-k). \tag{3}$$


result for square hann windows 50% overlap

result for hann windows 50% overlap

result for hann windows 25% overlap

Fig. 1 – Results of verifying eq. (3) on 3 different types of windows/overlaps. Root Hann window 50 % overlap (TOP), Hann window with 50% overlap (MIDDLE), and HANN window 25 % overlap (BOTTOM).

3.2. FREQUENCY DOMAIN CONDITION

To ensure that unaltered spectra can be successfully reconstructed in the context of frequency domain methods, the analysis window must adhere to the constraint overlap-add (COLA) principle. Generally, if the analysis window conforms to the condition (4), the window is COLA-compliant. Furthermore, COLA compliance can be classified into weak or strong categories.

$$\sum_{m=-\infty}^{\infty} g^{a+1}(n-mR) = c, \forall n \in \mathbb{Z}. \tag{4}$$

Weak COLA compliance signifies that the Fourier transform of the analysis window features zeros aligned with frame-rate harmonics, represented as:

However, spectral alterations disrupt alias cancellation. Weak COLA relies on alias cancellation within the frequency domain. Hence, perfect reconstruction is feasible using weakly COLA-compliant windows, provided the signal remains unaltered spectrally. The “iscola” function can be utilized to verify weak COLA compliance. The window length and hop size determine the number of summations employed for COLA compliance assessment.

4. **TIME-DOMAIN METHODS**

The time domain methods for fundamental frequency detection are most suitable for applications that work on only sound sources (for instance, a human voice or a single instrument).

4.1. AUTOCORRELATION FUNCTION (ACF)

The autocorrelation method and its modification can be classified as the most used fundamental frequency detection method for its simplicity and efficiency. The ACF is represented by the similarity ratio of the selected input signal to itscopy shifted by m samples. The function of one-side autocorrelation is defined as follows at (5), where $R(m)$ is the autocorrelation value, $s(n)$ is the input speech signal, n is the sample order, m is the mutual shift, and N is the total number of samples embedded in processed signal segment. Figure 2 illustrates a frame from a baritone interpretation of a song, to which all the time domain methods under this chapter will be applied, and its relevant ACF is shown in Fig. 3, where local peaks can be observed. The distance between these peaksis related to the fundamental frequency of the analyzed signal. By the ACF peak position, the fundamental frequency is calculated as the mean value of the sum of partial fundamental frequencies between two consequence peaks leading to equation (6), where f_s is the sampling frequency, a is the position of partialpeaks of the total peak number A

$$R(m) = \sum_{n=0}^{N-1-m} s(n)s(n+m)\,, \tag{5}$$

$$m = 0, 1, 2, \ldots N\text{–}1\,,$$

$$F_0 = \frac{f_s \cdot A}{\sum_{i=0}^{A-1} a_i + a_{i+1}}\,. \tag{6}$$

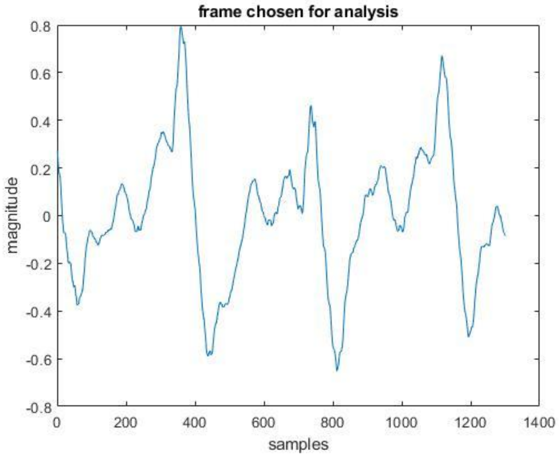

frame chosen for analysis

Fig. 2 – Example of real input signal waveform on one frame.

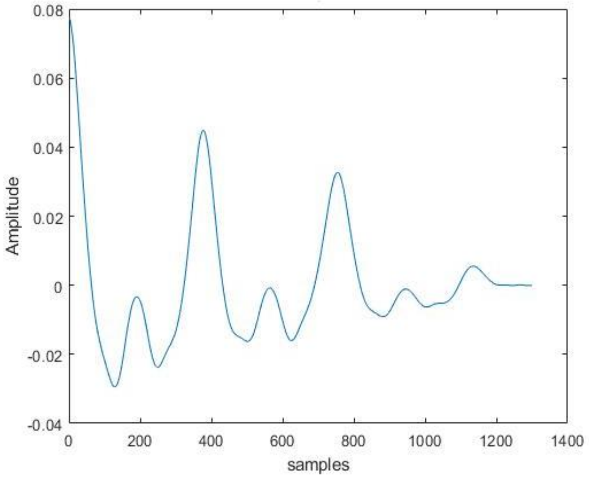


Fig. 3 – The ACF of the input signal.

4.2. MODIFIED AUTOCORRELATION FUNCTION (MACF)

MACF is formed by ACF method modified by so-called central clipping leading to analysis of observed signal peaks and removing its central part. The central clipping is defined by eq. (7), where $y(n)$ is signal value after central clipping and C_L isclipping threshold mostly set as the 50% value of maximal signal amplitude on current segment. After the central clipping, the autocorrelation function is applied on the current segment. Figure 4 presents centrally clipped signal waveform and Fig. 5 shows its autocorrelation function. The big advantage of this function canbe found in the formant attenuation of the input signal, leading to better fundamental frequency detection [4]

$$y(n) = \begin{cases} (s(n) - C_L) & , s(n) \geq C_L \\ 0, & |s(n)| < C_L \\ (s(n) + C_L) & , s(n) \leq -C_L \end{cases} . \quad (7)$$

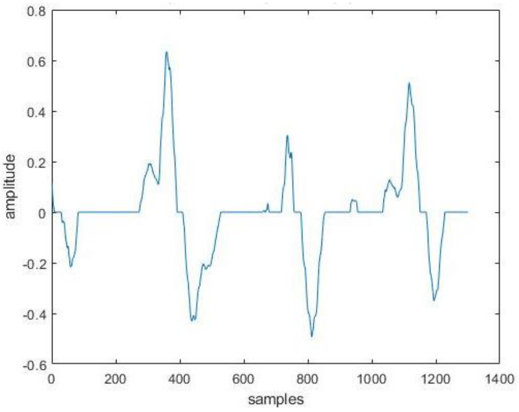


Fig. 4 – Centrally clipped input signal.

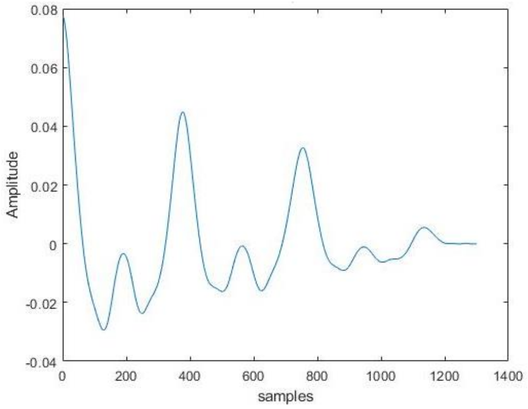


Fig. 5 – ACF of the centrally clipped input signal.

4.3 NORMALIZED CROSS-CORRELATION AUTOCORRELATION FUNCTION (NCCF)

The NCCF method is very similar to ACF and improves itslacks. The value of NCCF is calculated as shown in (8), where NCCF(m) is the final value, M_0 is the total number of autocorrelation points that must be calculated, and m takes values between 0 and M_0.

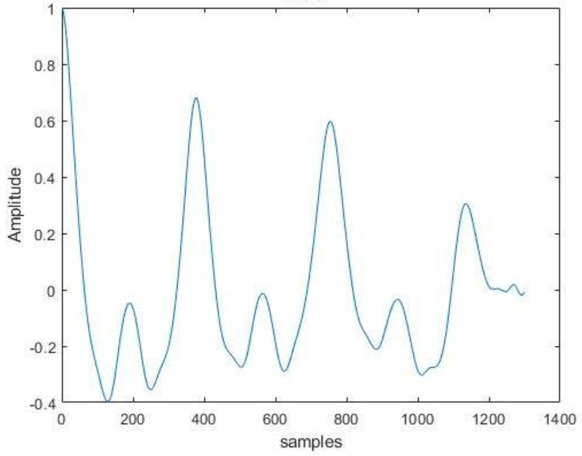


Fig. 6 – The NCCF of the input signal.

Figure 6 illustrates the final waveform [3].

$$\mathrm{NCCF}(m) = \frac{\sum_{n=0}^{N-1-m} s(n) \cdot s(n+m)}{\sqrt{\sum_{n=0}^{N-1-m} s^2(n) \cdot \sum_{n=0}^{N-1-m} s^2(n+m)}} . \quad (8)$$

4.4. AVERAGE MAGNITUDE DIFFERENCE FUNCTION (AMDF)

The AMDF can also be defined as modifying the autocorrelation function using multiplying substituted by subtraction. The Average Magnitude Difference function is defined by eq. (9), where $R(k)$ is the AMDF function final value and k is the time shiftof the input signal. The big advantage of the AMDF function is sharper and narrower local extremes, leading to better resolution offundamental frequency detection. The AMDF result of the input signal illustrated in Fig. 2 is shown in Fig. 7.

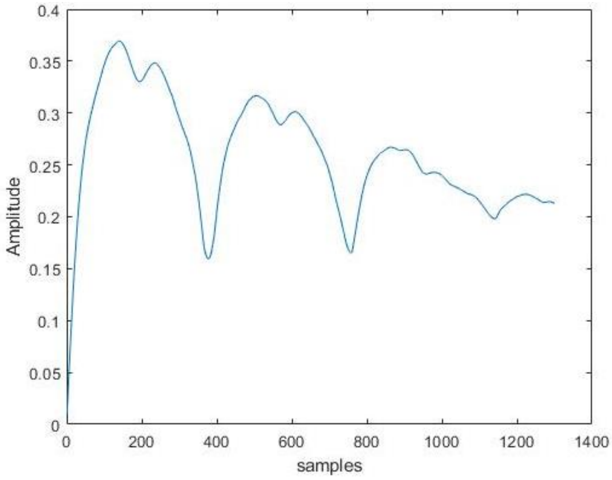


Fig. 7 – The AMDF of the input signal.

5. FREQUENCY-DOMAIN METHODS

The discrete Fourier transform (DFT) offers a fixed-in-time portrayal of signal frequencies. As music contains diverse frequency components, it becomes necessary to compute the DFT at multiple time instances to extract meaningful insights. Introducing a DFT size denoted as N, along with a window function $w[n]$ that operates within the range $-N/2 \leq n \leq N/2$, and an input signal $x[n]$ of length Ns samples, the short-time Fourier transform (STFT) extends this concept by incorporating the temporal dimension. It is mathematically defined as follows:

This concept can be envisioned as performing multiple DFT calculations on a signal, multiplying the signal by a sliding window that varies over time. In this context, I will refer to these modified signal versions as "frames." Each instance of sliding the window and performing the multiplication on the input signal creates a new frame. Subsequently, a DFT computation is conducted on each of these frames. The STFT is used in the frequency domain algorithms discussed in this section.

5.1. SPECTRAL YIN

The classic version of the YIN algorithm is a time domain method used for fundamental frequency detection [5]; however, it first requires a modified version of the AMDF function, and the algorithm has a time complexity of. In contrast, the spectral YIN has a time complexity of $O(n \cdot \log(n))$. As a result, only the spectral YIN will be analyzed. Moreover, when subjected to testing using pure sinusoidal signals, the relative error achieved by spectral YIN consistently remains under 0.01Hz.

These compelling advantages position spectral YIN as the favored and preferred pitch detection technique. It is called spectral YIN because this function is computed in the frequency domain. If $x_t[k]$ is the input frame and $X_t[k]$ is its DFT, then the tapered AMDF computation in the frequency domain is as shown

$$\mathrm{d_t}(\tau) = \frac{2}{N}\sum_{k=0}^{N-1} |X[k]|^2 \left(1 - \cos\left(\frac{2\pi k\tau}{N}\right)\right). \quad (9)$$

5.2. SPECTRAL METHOD

This method is based on analyzing the input signal in the frequency domain. In the case of spectral method usage, the conversion into the frequency domain is performed by discrete Fourier transform defined by:

$$S(k) = \sum_{n=0}^{N-1} S(n) \cdot e^{-\mathrm{j}k\frac{2\pi}{N}n}, \quad k = 0,1,2,\ldots,N-1, \quad (10)$$

where $S(k)$ is the value of k-th spectral component. The spectrum of an input signal is shown in Fig. 8.

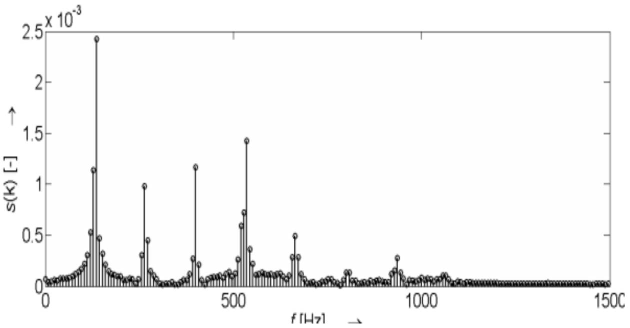


Fig. 8 – Fundamental frequency (the highest peak) detection of the input signal by spectral method.

6. SELF-REPAIRING ALGORITHMS

It is advisable to incorporate a self-repairing algorithm after the selected pitch detection method to enhance the efficiency of fundamental frequency detection. Generally, within the domain of recognition tasks, a range of self-repairing algorithms are available to maximize efficiency or likelihood. For instance, in speech recognition scenarios, the Baum-Welch algorithm is commonly employed in conjunction with Hidden Markov Models. At the same time, more advanced self-repairing algorithms have been proposed. For the specific application mentioned, a developed self-repairing algorithm could be designed to be simpler yet effective.

6.1. BAUM-WELCH ALGORITHM

As a preliminary step to introducing the self-repairing algorithm, a basic algorithm is integrated, focusing on associating the identified fundamental frequency with the nearest musical note, precisely aligning it with the note's corresponding frequency for potential subtle adjustments. Fig. 10. ACF detected fundamental frequency (blue) and the nearest note assignment (red) on the part of the recorded soprano female voice. The resultant assigned note values are stored within a workspace table for subsequent processing [3].

In the case of recorded real soprano female voice, the disparity between the determined notes and the assigned notes in the processed segments (pertaining to fundamental frequencies) is presented in Fig. 9. The blue curve represents the found notes. In contrast, the nearest detected notes are depicted in red, showcasing the visual distinction.

The self-repair mechanism is based on the previously identified notes enumerated within the workspace table. The underlying concept revolves around the notion that a single segment cannot adequately represent an entire musical note due to the temporal extent of the segment. This prompts a comparison of frequencies between neighboring segments and the current segment.

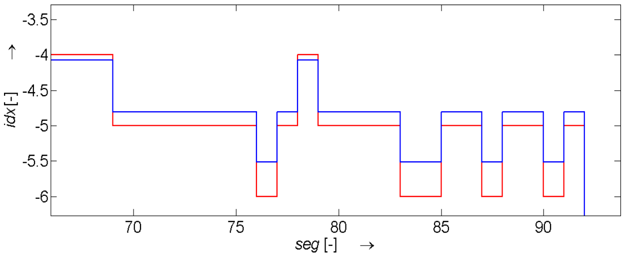


Fig. 9 – ACF detected the fundamental frequency (blue), and the nearest note assignment (red) was on the part of the recorded soprano female voice.

The algorithm for self-repair, as presented, can be described as a sliding window traversing the entire input signal and scrutinizing three consecutive segments at a time. Suppose the identified frequency (and the corresponding note) remains consistent for the boundary segments while differing for the central segment. In that case, the frequency of the central segment is adjusted to match the found frequency value of the boundary segments.

It is important to note that a distinct approach is necessary for the initial and concluding three segments of the input recorded signal. These segments undergo verification once the repair process for all other segments has been completed. For the first triplet, if the pitch value of the initial segment differs from that of the subsequent two, it is rectified to align with the pitch value of the second and third segments.

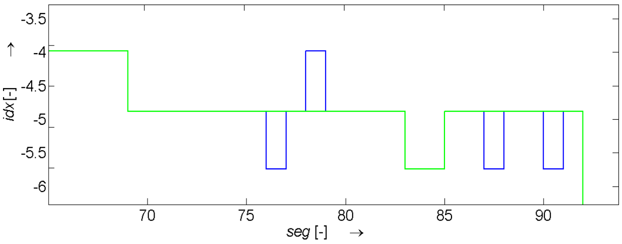


Fig. 10 – The function of the self-repairing algorithm (green) applied on detected fundamental frequencies of real female soprano voice (blue) by ACF.

The final segment of the analyzed input signal undergoes assessment based on the preceding two segments, specifically considering their respective pitch values. If the detected pitch value in the final segment deviates, it is adjusted to match the pitch values of the two preceding segments.

For this reason, establishing the shortest detectable note (segment) as half the duration of the actual shortest note present in the input signal is strongly recommended and necessary. This is essential to ensure the effective operation of the self-repairing algorithm. The fundamental concept draws inspiration from a previously published retroactive-checking algorithm employed in vowel recognition within fluent speech. This technique resulted in an average 36.7 % reduction in false detections. Figure 10 offers a visualization of a portion of the analyzed input signal (depicted in blue in Fig. 9, which has been corrected by the self-repairing algorithm (displayed in green).

Applying the self-repairing algorithm merits minimizing the error rate in fundamental frequency detection. Its judicious integration with the detection method can achieve highly satisfactory outcomes.

6.2. ADAPTIVE AUTOCORRELATION

In this subsection, we delineate the operational essence of the AAC algorithm [6]. The algorithm's primary aim is to gauge the fundamental frequency of a given sampled speech signal, denoted as x. This objective is achieved by computing autocorrelations between signal x and a designated self-segment, referred to as s.

The chosen segment encompasses the initial M_s data points of the speech signal. M_s is determined as $M_s = f_s / F_l$, where f_s represents the sampling frequency, and F_l signifies the lowest resolvable frequency. The duration of this segment, denoted as T_s, equals M_s / f_s in seconds.

Estimating the fundamental frequency of the speech signal involves identifying maxima within an autocorrelation function, as discussed in the third section.

The operational principle is elucidated through the aid of a periodic signal. The diagram illustrates sequential algorithmic steps, progressing from left to right. The top row highlights the fixed signal segment, shaded for emphasis, while the periodic signal, slightly shifted concerning the segment, is displayed in the bottom row.

Function z_k exhibits maxima at k values where the segment and the shifted signal align most effectively, including at $k = 0$ and integer multiples of the period linked to the signal's fundamental frequency. However, due to the potential presence of extra maxima resulting from higher frequency components, more than relying on the correlation function z_k maxima is required. To address this, a peak detector function y_k is introduced, defined as $y_k = z_{k0} \exp(-(k-k_0)/(f_s\tau))$.

The visual representation in the illustration captures the evolution of functions z_k and y_k. Only segments of these functions pertinent to fundamental frequency estimation at specific algorithmic snapshots are depicted as solid lines. Conversely, values that do not contribute to the estimation at those moments, encompassing future values and values that have already led to an estimation, are illustrated as dotted lines.

Column (A) portrays the algorithm's initial phase, wherein the signal segment and the signal overlap. Column (B) illustrates the algorithm's progression, ultimately culminating in determining the fundamental frequency in column (C). Here, the first peak of the correlation function z_k, after its intersection with the peak detector function y_k, yields the estimation of the period N_{period} (in samples), and, consequently, the fundamental frequency $F_0 = fs/N_{\text{period}}$ of the signal. Upon obtaining a new F_0 estimate, the algorithm resets, employing the signal starting from the estimation point as the new reference. This iterative process unfolds as represented in column (D) for successive fundamental frequency estimations. If k surpasses the maximum expected period $T_{\max} = 1/F_l$, the algorithm returns the most recent valid estimate and initiates anew with a fresh segment.

Figure 11 presents an instance wherein the algorithm is applied to an authentic speech signal, which typically lacks strict periodicity.

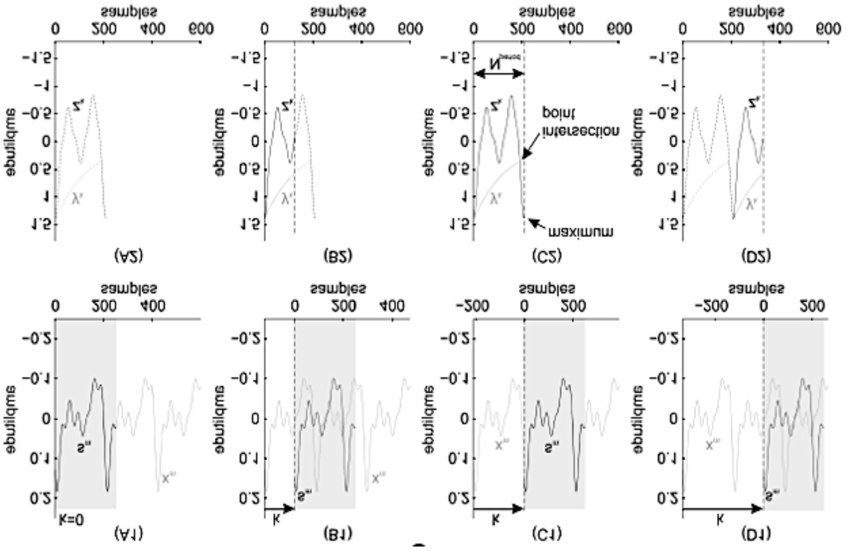


Fig. 11 – Principle of operation of the AAC algorithm in four steps.

7. **CONCLUSIONS**

The Baum-Welch algorithms perform exceptionally well on noisy inputs, which is the case for most real-world recording environments. However, the computational complexity is high. The adaptive autocorrelation has a lower time complexity but performs worse on noisy inputs. Although, considering the affordability and accessibility of quality recording equipment, for most applications in the domain of audio engineering, like designing audio plugins that depend on fundamental frequency detection, the ACF or MACF provides satisfactory results with the bonus of being able to provide other useful parameters of the input signal that then can be used for detection of voiced/unvoiced frames.

Received on 1 September 2023

REFERENCES

1. H. Yang, L. Qui, S.N. Koh, *Application of instantaneous frequency estimation for fundamental frequency detection*, Proceedings of IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, Philadelphia, PA, USA, pp. 616–619, 1994.
2. L. Wang, Z. Li, X. Zhuang N.E. Mastorakis, *Voiced/unvoiced pronunciation judgement based on sparse representation and learning dictionary*, 3rd European Conf. on Electrical Engineering and Computer Science (EECS), Athens, Greece, pp. 147–150, 2019.
3. Z. Chen, Z. Shen, D. Guo, X. Wang, *Design of near perfect reconstruction prototype filter with FFT interpolation*, IEEE International Conference on Digital Signal Processing (DSP), Beijing, China, pp. 159–163, 2016.
4. M. Stanek, T. Smatana, *Comparison of fundamental frequency detection methods and introducing simple self-repairing algorithm for musical applications*, IEEE, 25th International Conference Radioelektronika (Radioelektronika), pp. 217–221, 2015.
5. T. Royer, *Pitch-shifting algorithm design and applications in music*, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, Degree Project in Electrical Engineering, Stockholm, Sweden, 2019.
6. M. Staudacher, V. Steixner, A. Griessner, C. Zierhofer, *Fast fundamental frequency determination via adaptive autocorrelation*, EURASIP, Journal on Audio, Speech, And Music Processing, Article number: 17 (2016).