

RESEARCH

Open Access



# Fast fundamental frequency determination via adaptive autocorrelation

Michael Staudacher\*, Viktor Steixner, Andreas Griessner and Clemens Zierhofer

## Abstract

We present an algorithm for the estimation of fundamental frequencies in voiced audio signals. The method is based on an autocorrelation of a signal with a segment of the same signal. During operation, frequency estimates are calculated and the segment is updated whenever a period of the signal is detected. The fast estimation of fundamental frequencies with low error rate and simple implementation is interesting for real-time speech signal processing.

**Keywords:** Pitch detection, Fundamental frequency, Autocorrelation

## 1 Introduction

One goal in many speech analysis applications is to follow fast variations in the fundamental frequency ( $F_0$ ) of a signal. Many pitch detection algorithms (PDAs) analyze a speech signal by partitioning it into segments and calculating the respective fundamental frequencies (short-term analysis) [1]. The length of the segments (frames) limits the minimum frequency or the maximum period to be determined. Autocorrelation functions and average magnitude difference functions are widely used methods to detect periodicity. Typically, two periods are necessary to detect a periodic pattern [1, 2], e.g., to detect frequencies down to 50 Hz, the segment in real-time PDAs has to contain a history of 40 ms. This results in a time delay of the estimates in real-time applications, i.e., a discrepancy appears between the calculated and the actual fundamental frequency. In addition, using long segments leads to a situation where changes of the fundamental frequency within the segment are lost [3]. An ideal algorithm should meet the requirements of both a low minimum detectable frequency and a small lag.

In this paper, we present the concept and evaluation of an adaptive autocorrelation (AAC) algorithm that was devised by Zierhofer (personal communication) as a development within the context of novel stimulation strategies in the field of cochlear implants. An early

version of the algorithm and some preliminary results were already presented in [4]. The AAC algorithm uses an autocorrelation of a part of the speech signal and the shifted speech signal to calculate an estimate of the fundamental frequency. After an estimate is obtained, the algorithm restarts and repeats frequency estimation consecutively.

As will be shown in the following sections, the AAC algorithm shows a performance comparable to other algorithms, but the minimum required length of the segment is only once the maximum expected period. Compared with short-term analysis methods, the time delay between the estimates and the actual fundamental frequency is, thus, reduced; the algorithm responds faster to frequency changes, and estimates are calculated whenever a fundamental period is determined. These factors are advantageous in real-time tracking of the fundamental frequency in applications which require low latencies.

The present work is organized as follows. First, the principle of operation of the AAC algorithm is described in detail. In section “Methods,” the applied procedure of evaluation is introduced. The dependencies on the segment length and the time constant are shown in section “Characteristics.” In section “Comparison,” we evaluate detection lag and performance of the AAC algorithm on the basis of three reference algorithms. In section “Noise robustness,” the performance of the AAC algorithm in the presence of noise is analyzed.

\* Correspondence: Michael.Staudacher@uibk.ac.at  
Institute of Mechatronics, University of Innsbruck, Technikerstrasse 13, 6020 Innsbruck, Austria

## 2 Algorithm

In this section, we describe the principle of operation of the AAC algorithm. The goal of the algorithm is to estimate the fundamental frequency of a sampled speech signal  $x$ . This is achieved by calculating autocorrelations between the signal  $x$  and a segment  $s$  of itself. We choose the segment to be the first  $M_s$  points of the speech signal, with  $M_s = f_s/F_l$ . Here,  $F_l$  is the lowest frequency that can be resolved. The length of the segment is  $T_s = M_s/f_s$  in seconds.

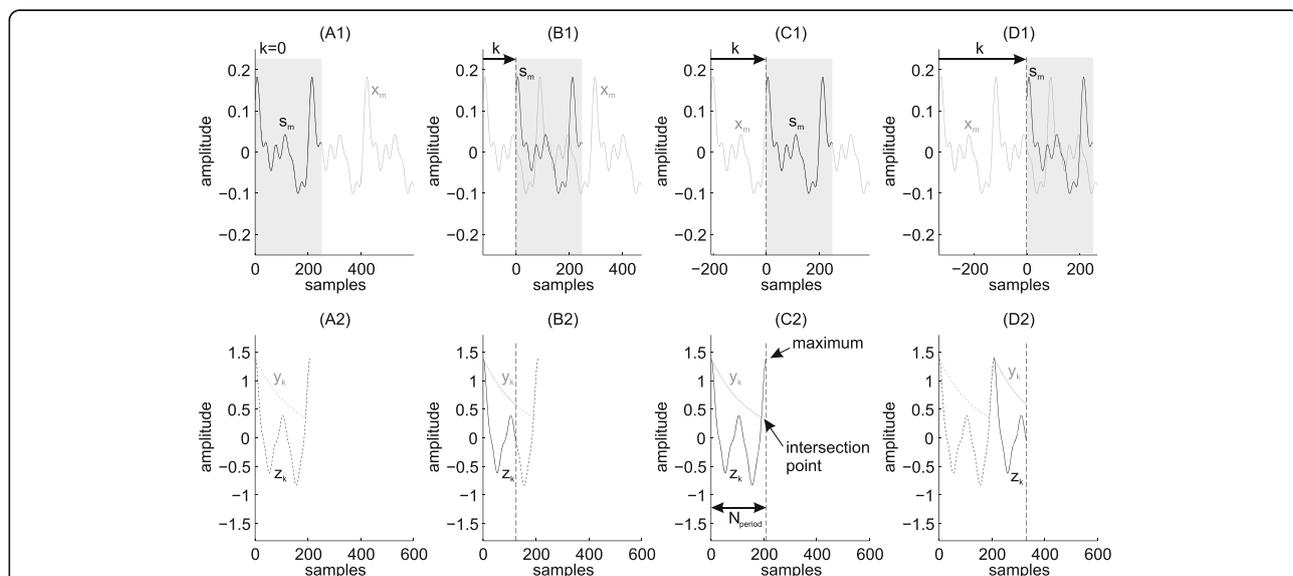
The fundamental frequency of the speech signal can be estimated by finding maxima of an autocorrelation function of the sampled signal  $x_m$  and the segment  $s_m$ , given by

$$z_k = \sum_{m=1}^{M_s} s_m x_{m+k} \tag{1}$$

The function  $z_k$  has maxima at values for  $k$  where the segment and the shifted signal have their best matches, i.e., at  $k = 0$ , and then at integer multiples of the period associated with the fundamental frequency in the signal. Since  $z_k$  has, in general, additional maxima due to higher frequency components of the signal, determining maxima of the correlation function alone is not sufficient. In order to identify the fundamental frequency, we introduce a peak detector function  $y_k = z_{k_0} \exp(-(k - k_0)/f_s \tau)$ . With

increasing  $k$ , every maximum of  $z_k$  up to the point where  $z_k$  and  $y_k$  intersect is ignored. The subsequent maximum is then used for the fundamental frequency estimation. The starting point  $k_0$  of the exponential function is chosen to be the value where the slope of  $z_k$  becomes more negative than the slope of  $y_k$  to prevent high-frequency false estimations. This could happen if the correlation function falls off more slowly than the peak detector function following a fundamental frequency estimation. A suitable value of the time constant  $\tau$  depends on the length of the segment and the desired range of frequencies. Empirical results from speech databases show suitable values for  $\tau$  to be in the order of 6 to 10 ms.

The principle of operation is explained with the help of a periodic signal as illustrated in Fig. 1. It shows exemplary steps of the algorithm in sequential order from left to right. The top row shows the pinned signal segment highlighted as a shaded area and the periodic signal that is shifted with respect to the segment. The bottom row shows the progression of the functions  $z_k$  and  $y_k$ . As a visual guideline, only parts of those functions necessary for the fundamental frequency estimation at the respective algorithm snapshots are shown as solid lines, and those values that do not contribute to the estimation at those moments, i.e., future values and values that have already led to an estimation are shown as dotted lines.



**Fig. 1** Principle of operation of the AAC algorithm in four steps. *Top row*: content  $s_m$  of the segment as a black line on gray background and signal  $x_m$  in gray. *Bottom row*: correlation signal  $z_k$  in black and peak detector function  $y_k$  in gray. Solid lines show parts of the functions necessary for fundamental frequency estimation in the respective figure. Dotted lines indicate future values and values that have already led to an estimate. *Column A*: segment  $s_m$  is chosen as a slice of the signal  $x_m$  with length  $M_s$  starting at the first sample. *Column B*: the shift between both signals is increased, and the correlation signal  $z_k$  is calculated following Eq. (1). *Column C*: the shift between  $s_m$  and  $x_m$  is further increased to the point where both signals are qualitatively similar, and the signal  $z_k$  exhibits a maximum. The displacement of the segment to the signal provides a measure for the period. *Column D*: after having detected a maximum in  $z_k$ , the content of the segment is replaced with a slice of the signal  $x_m$  starting right after the detected peak, and fundamental frequency detection restarts

Column (A) depicts the starting point of the algorithm, with the signal segment and the signal overlapping. Further progression of the algorithm is shown in column (B), leading to the fundamental frequency determination step in column (C), where the first maximum of the correlation function  $z_k$  following its intersection with the peak detector function  $y_k$  yields the estimate for the period  $N_{\text{period}}$  (in samples), and hence the fundamental frequency  $F_0 = f_s/N_{\text{period}}$  of the signal. Each time a new  $F_0$  estimate is obtained, the algorithm resets using the signal starting from that estimation point as the new signal. It proceeds as depicted in column (D) for subsequent fundamental frequency estimations. As a side note, if  $k$  exceeds the maximum expected period  $T_{\text{max}} = 1/F_l$ , the algorithm returns the most recent valid estimate and restarts with a new segment.

The number of time steps until an estimate is obtained is adaptive, i.e., the estimates delivered by the algorithm are in general not equally spaced in time. The same segment is used as long as no maximum in  $z_k$  is detected. Since the correlation of a segment with a shifted section of the signal is calculated, the required length of the segment is only once the maximum expected period. Thus, the algorithm only needs the length of the segment plus the actual period of voice for fundamental frequency calculation. For Fig. 1, a periodic function of a fundamental frequency of 96 Hz has been used, together with the assumption of the signal being known one segment length ahead in the positive time direction. In a real-time application, the correlation signal  $z_k$  would be delayed by one segment length.

Figure 2 shows an example where the algorithm is applied to a real speech signal which, in general, is not strictly periodic. The figure on the left shows the raw speech data (top), the result of the correlation and peak detector function (center), and the calculated  $F_0$  estimates (bottom). Every vertical gray line in the center

plot indicates a completed  $F_0$  estimation. Respective estimates are depicted in the bottom row. The figure on the right shows a detailed view of the part of the correlation and peak detector function marked with a box in the center plot of the left figure. As can be seen, an  $F_0$  estimate is returned when a peak in the correlation is detected, which is indicated by the vertical dashed line. The algorithm then restarts with a new signal segment. In general, the change of the segment leads to discontinuities in  $z_k$ , e.g., when the signal is not periodic.

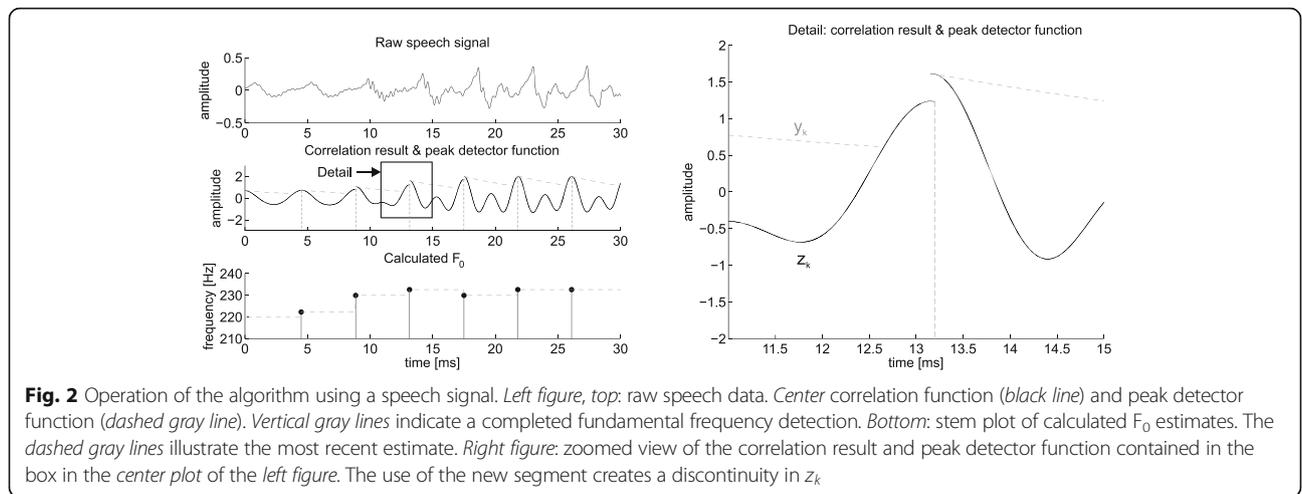
### 3 Methods

In this section, we introduce the used datasets, the applied reference algorithms, the method of evaluation, and the settings for the following investigations.

#### 3.1 Speech databases

The Edinburgh Evaluation Database (DB1) [5] contains one male and one female speaker saying 50 English sentences. The total duration is about 7 min. The Keele Speech Database (DB2) [6] consists of five male and five female speakers reading the “North Wind story” in English. The duration per speaker and story is about 30 s. The PTDB-TUG Database (DB3) [7] provides 4720 recorded sentences of 20 English native speakers.

All databases consist of speech data files and reference fundamental frequency values which are used to evaluate the calculated frequency estimates. These reference values are created with the help of a PDA analyzing a laryngograph signal recorded in parallel with the speech signal. The databases further contain information about the location of voiced and unvoiced parts, which allows us to focus only on the voiced parts of the signals. In the special case of voiced/unvoiced detection for cochlear implants, unvoiced parts contain a very small amount of energy and hence do not lead to a stimulation of any electrodes. The reference fundamental frequency values



are sampled at 100 Hz; thus, the estimates of the AAC algorithm have also been extracted from the calculated fundamental frequency vector at intervals of 10 ms for direct comparison of the AAC with reference algorithms.

### 3.2 Reference algorithms

We compare the AAC algorithm to a well-known and established set of algorithms with respect to gross error and detection lag. The used algorithms are an autocorrelation algorithm [8], the frequency domain-based subharmonic-to-harmonic ratio procedure (SHRP) [9, 10], and the respective author's publicly available implementation of the YIN algorithm [11, 12]. In contrast to the AAC algorithm, these algorithms are short-term analysis methods which calculate estimates in equidistant, non-adaptive steps, e.g., every tenth millisecond as in the present work.

The choice of comparison methods is inspired by the similarity of those algorithms to the AAC and by the fact that they have already been compared with another wide variety of algorithms. All of these algorithms are correlation-based and use a fixed segment size in their basic implementation. This implies that the number of calculation steps in these algorithms is not dependent on the form or frequency of the signal. Since this leads to a predictable calculation time for the estimates, we believe these algorithms to be well-suited for real-time tracking of the fundamental frequency.

Other implementations addressing the problem of pitch detection with low time lag can be found, e.g., in the field of audio to MIDI conversion [13], where  $F_0$  is estimated from a given set of partials for musical instruments based on a probabilistic model or in an instantaneous implementation of the RAPT framework [14] which has a higher inherent time lag compared with typical lags of the AAC or our used reference algorithms. A different approach is parametric pitch estimation routines (e.g., [15]) which assume a predetermined model for the estimation of the signal under investigation.

In addition to comparing the gross errors of the algorithms in terms of the fundamental frequency, we also want to measure the detection lag, a quantity that is a fidelity indicator for real-time fundamental frequency tracking. We use artificially generated signals with a controlled fundamental frequency progression and define the detection lag to be the time between the occurrence of a fundamental frequency in the signal and its detection by any of the algorithms. There are more sophisticated algorithms like SWIPE [16, 17] that can produce a smaller overall detection lag upon processing a full audio signal and uses varying segment sizes as well as advanced optimization techniques. However, in this paper, we focus on the comparison with fixed

segment size algorithms in their basic forms with few optimizations for better comparability and with a signal that is preprocessed in the same way for each algorithm.

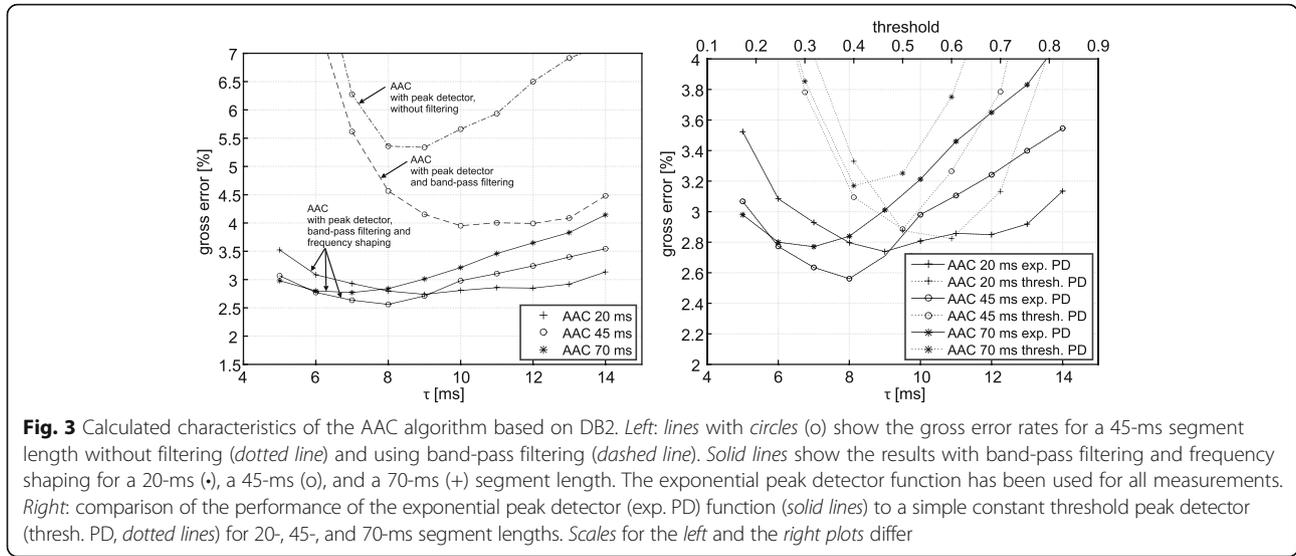
### 3.3 Settings

We investigated three scenarios. In scenario one, we applied all algorithms to the raw speech data. In scenario two, the datasets were band-pass filtered using a sixth-order Butterworth filter with a lower corner frequency of 50 Hz, an upper corner frequency of 500 Hz. This frequency range contains all typical fundamental frequencies of the human voice. In scenario three, frequency shaping was used in addition to the band-pass filter. This frequency shaping is a simple low-pass filtering above 50 Hz and attenuates higher frequency components with 6 dB/oct. To measure the performance of pitch detection algorithms, the gross error is calculated. This error is a measure of how often a deviation of more than 20 % from the reference fundamental frequency occurs and is frequently used in the literature [11, 18, 19]. Consistent with other studies [11, 19], this investigation only considered voiced parts of the speech signals, while unvoiced sequences were not taken into account.

## 4 Characteristics

In this section, we study the behavior of the AAC algorithm as a function of the independent parameters, segment length  $T_s$  and time constant  $\tau$  of the peak detector, and the influence of filtering and frequency shaping. In addition, we compare the performance of the AAC algorithm when implemented with the exponential peak detector function as described above to an implementation with a constant threshold method. The goal here is to identify a set of parameters which are optimized for one (test) database (here DB2) and will then be used for all other datasets. The reason for this approach is that for arbitrary speech databases or in realistic conversation situations, it is not feasible to determine the optimal parameters a priori since no reference data is available in real time.

In the left part of Fig. 3, the gross error is plotted as a function of the time constant  $\tau$  of the exponentially decaying peak detector function for different values of the used segment length  $T_s$ . The solid lines show a comparison of the results of the full AAC algorithm with band-pass filtering and frequency shaping for a 20-, 45-, and 70-ms segment. As can be seen in the figure, the gross error depends on both the time constant and the segment length, and a minimal value is found for a segment length of  $T_s = 45$  ms and a time constant  $\tau = 8$  ms for DB2. The obtained gross error is relatively stable as a function of  $T_s$  and  $\tau$  close to the optimal value which



indicates that using parameters specifically optimized for a distinct database should not critically deteriorate the performance of the algorithm when applied to other databases or in situations where an optimization of parameters is impracticable.

The left graph of Fig. 3 also shows the impact of frequency shaping and band-pass filtering of the input signal on the gross error. The effects are demonstrated for a segment length of  $T_s = 45$  ms. As can be seen from the figure, applying band-pass filtering significantly reduces the gross error rate (dashed line) in comparison to using the raw data (dotted line). This error rate is further significantly decreased by additionally adding frequency shaping (solid lines).

In the right part of Fig. 3, the performance of the exponential peak detector function (solid line) is compared with a standard constant threshold method (dotted line). Both methods have been implemented in the AAC algorithm with  $T_s = 20, 45,$  and  $70$  ms and optimized in terms of the decay time  $\tau$  for the exponential case and

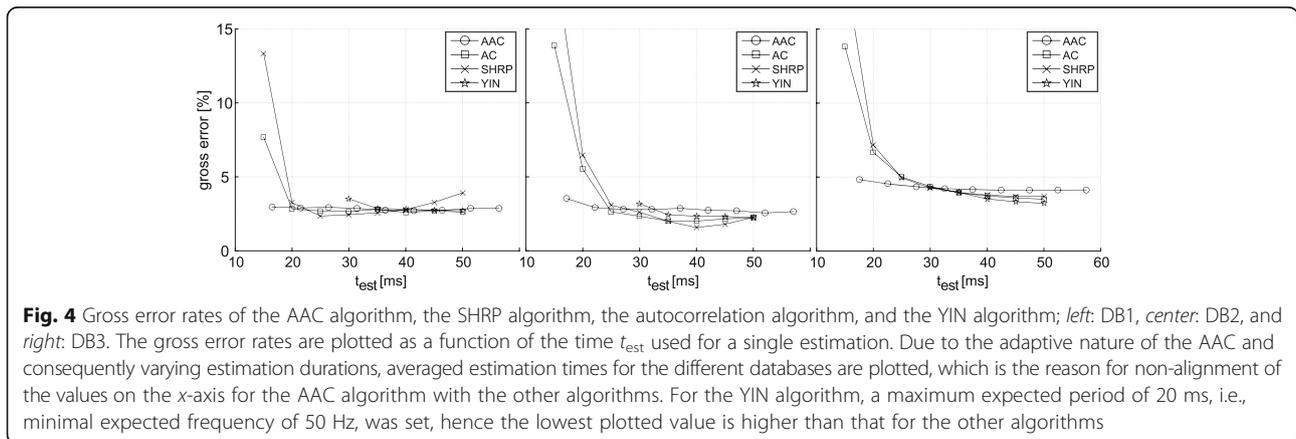
the cut-off threshold for the constant method. We find that the exponentially decaying peak detector function yields better results when tuning to the respective optimal value. Therefore, we will use the exponential peak detector function for all the following measurements.

### 5 Comparison

In this section, we compare the performance and the detection lag of the AAC and the reference algorithms. In Fig. 4, the gross error rates are plotted as functions of the number of samples used for a single estimation in the respective algorithms, expressed in milliseconds. The figure shows the results for DB1 (left), DB2 (center), and DB3 (right). For direct comparison, all algorithms use the same pre-filtering of the data, i.e., band-pass filtering and frequency shaping.

#### 5.1 Gross error

As the evaluation of the three databases in Fig. 4 shows, the AAC algorithm has a low error rate for a wide range



of estimation durations. In general, the error rates increase for very short estimation times due to a lack of data. For the YIN algorithm, the integration window size is decoupled from the maximum expected period [11], which we choose to be 20 ms, corresponding to 50 Hz. There are no data points for the YIN below 30 ms, because for an estimate, YIN requires the maximum expected period plus the minimum integration window size, chosen as 10 ms, the same as for the AAC algorithm. For the SHRP and the autocorrelation algorithm, the maximum expected period is linked directly with the amount of data used for one estimation. Therefore, the error rate rises significantly when the minimum expected frequency exceeds typical fundamental frequencies in the speech signal, as can be seen for all databases in Fig. 4. Since the estimation duration  $t_{est}$  for the AAC algorithm depends on the estimated frequency, averaged values are plotted in Fig. 4. For this reason, the AAC data points do not align with the other algorithms on the  $x$ -axis.

A more quantitative comparison of the performance of the different algorithms requires some statistical considerations. For the assessment of significant findings, statistical multiple comparison methods have to be used. The gross error does not conform to a normal distribution, and we find that the variances vary substantially for the different algorithms. Hence, requirements for the application of an ANOVA or  $t$  test comparison are not fulfilled. In our speech databases, we find a high variability in the gross error between different speech files but a low variability in the performance of the different algorithms on the individual speech files. Therefore, we use a non-parametric paired multiple comparison test, specifically the statistical Friedman test.

We have performed the statistical comparisons for all databases and three representative estimation times  $t_{est}$  chosen as 20, 30, and 45 ms. The evaluation of the databases DB1 and DB2 shows a similar performance of the different algorithms in most of the cases as can also be seen in Fig. 4. Some significant differences can be found for DB1: at 30 ms, the error rate of YIN is higher than that of the other algorithms and the error rate of AAC is higher compared with SHRP, while at 45 ms, the SHRP performs worse than the autocorrelation (AC) and YIN algorithms. For DB2, there are no statistically significant differences except for a lower error rate of the SHRP compared with the AAC at 45 ms.

DB3 consists of a considerably higher number of speech samples compared with DB1 and DB2. This allows to identify statistical significances with a much higher accuracy. For this reason, we explicitly present the values of the mean gross errors for the different algorithms and estimation times in Table 1 and report on

**Table 1** Mean gross error rates for DB3

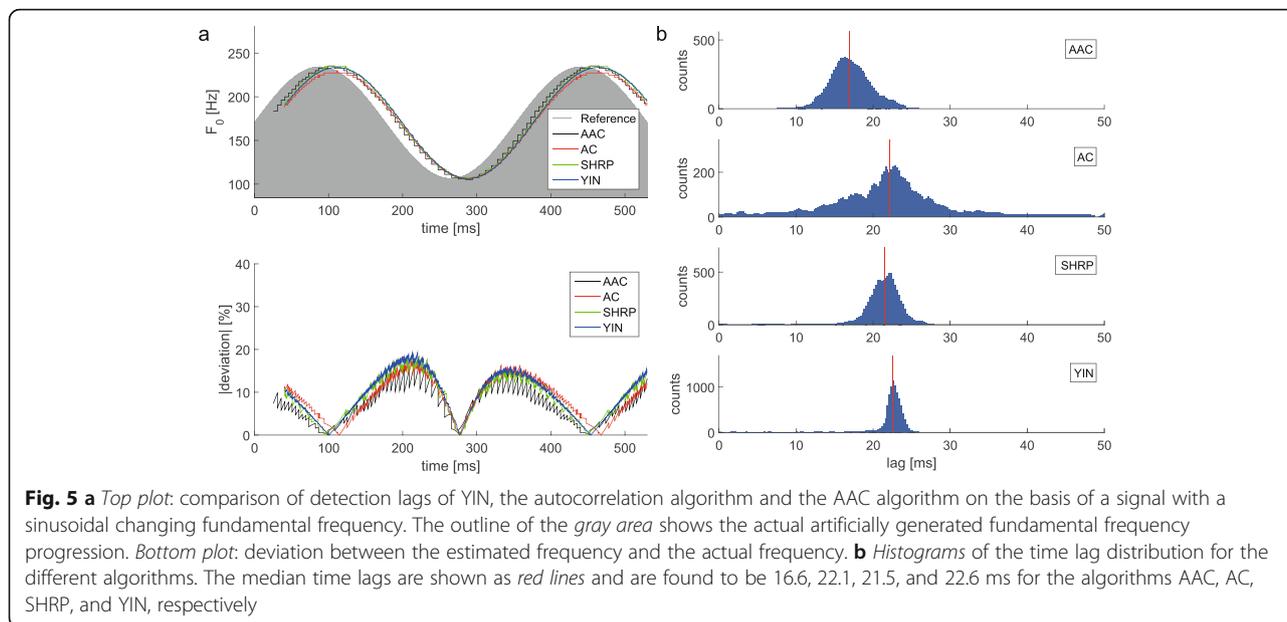
ms	AAC	AC	SHRP	YIN
20	4.62	6.67	7.15	
30	4.27	4.37	4.26	4.32
45	4.12	3.6	3.7	3.3

the found significances in the text. For the short estimation time  $t_{est} = 20$  ms, the AAC outperforms the other algorithms with a significant difference in the mean gross error as the adaptive nature of the AAC grants low error rates even when the algorithm uses fewer samples for each estimate. Even though the almost indistinguishable means of the gross error at 30 ms suggest a similar behavior for all algorithms, the algorithms perform statistically significantly different due to the large size of the database. At 45 ms, we find two significantly different groups; the AC and YIN yield better results than the AAC and SHRP.

**5.2 Detection lag**

In the following evaluation, we compare the detection lags of the different algorithms. For accurately calculating time lags, we use artificial input signals, as an accurate knowledge of the reference frequency at every point in time is essential. While in the used speech databases, the reference frequencies are only available on a relatively coarse time grid of 10 ms, the advantage of an artificial signal is that the reference is known at arbitrary points in time. The predetermination of the frequency progression allows to systematically probe a larger frequency range and in addition has the advantage that gross errors can be avoided for the comparison of lags.

Specifically, we use an input signal with sinusoidal frequency modulation (FM) of the fundamental frequency and higher harmonics, as shown in Fig. 5a. For calculating an estimate of the fundamental frequency at any time, the algorithms are only given causally accessible samples of the signal, as would be the case in real-time tracking of  $F_0$ . For Fig. 5, the minimum resolved frequency is set to 50 Hz for every algorithm and the reference algorithms estimate the fundamental frequency every millisecond. As can be seen qualitatively in the top plot of Fig. 5a, the AAC algorithm tracks the fundamental frequency of the signal with less detection lag and is able to follow frequency variations faster than the reference algorithms. Using smaller segments lowers the detection lag for all algorithms, but it also raises the minimum detectable frequency. The bottom plot of Fig. 5a shows the deviation between the estimated frequency and the actual frequency. The plotted deviations of the reference algorithms are smoother



due to the more frequent estimation rates in contrast to the adaptive estimation times returned by the bare AAC algorithm.

A statistical comparison requires a more quantitative approach of the time lags obtained for the different algorithms. In Fig. 5b, the distribution of the time lags and the corresponding median values (red lines) are shown for the same artificial signal as in Fig. 5a. By first optically inspecting the results, we find that the AAC algorithm yields a smaller time lag compared with the other algorithms.

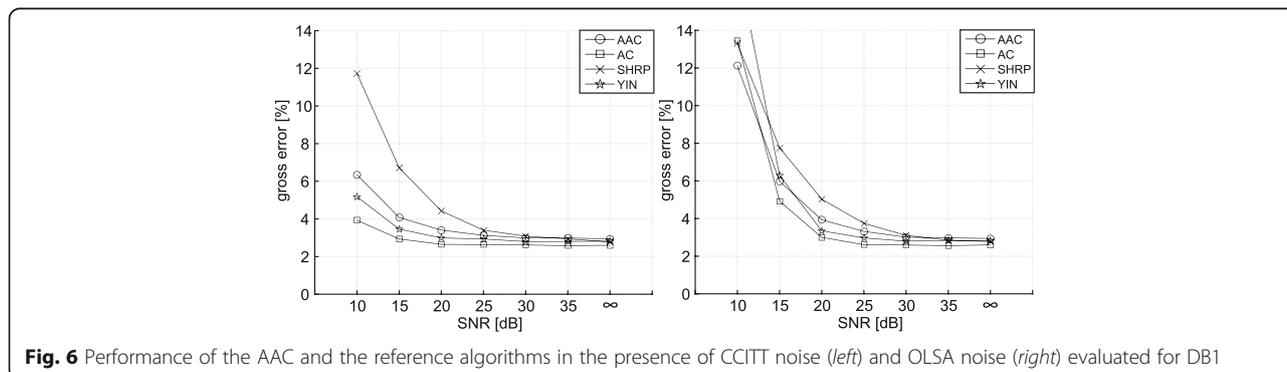
Statistically significant differences are again tested with multiple comparison tests. The lags do not conform to a normal distribution and show substantially different variances for different algorithms. Thus, for the non-parametric comparison between the four unpaired groups, we use the Kruskal-Wallis one-way analysis of variance test. The differences between all the groups are found to be highly significant, even for the

groups with comparable median values, which is again owed to the large size of the dataset.

### 6 Noise robustness

Fundamental frequency detection in real applications is complicated by noise. In this section, the AAC algorithm is evaluated with DB1 and added noise. Used sources of noise are Comité Consultatif International Téléphonique et Télégraphique (CCITT) noise and Oldenburg sentence test (OLSA) noise, i.e., speech-shaped noise created by an averaging procedure over sentences in the German OLSA speech test [20], which are commonly used in the field of cochlear implants for evaluating speech perception.

In accordance with the preceding investigations, the minimum expected frequency for the algorithms in the following evaluation is set to 50 Hz. As the evaluation for both noise sources in Fig. 6 shows, all algorithms have a similar sensitivity to noise for large signal-to-



noise ratios. Using the statistical Friedman test again, we determine that the performance of the algorithms deteriorates significantly for an SNR below 20 dB in the presence of CCITT noise for all algorithms except SHRP which becomes significantly worse for an SNR below 25 dB already. In contrast, the performance drops more severely in the presence of OLSA noise, specifically already below 30 dB SNR for the SHRP and below 25 dB for the AAC and YIN, while the AC algorithm is robust for an SNR up to 20 dB. This difference between the two noise sources can be attributed to the OLSA noise having significant parts of the spectrum within the  $F_0$  range of interest.

## 7 Conclusions

We have shown that the AAC algorithm presented in this paper, when compared with other well-established algorithms, can find fundamental frequency estimates with a similarly low error rate. An advantage over other short-term analysis methods is that fewer data points can be used for an estimation of the fundamental frequency. This is a direct result of the adaptivity of the algorithm, and leads to a small detection lag, which allows for analyzing and extracting information from signals with a rapidly varying fundamental frequency. This feature of fast fundamental frequency estimation is advantageous in the field of real-time analysis of speech signals. Together with the simple implementation of the algorithm, this opens new possibilities in the development of applications in the area of real-time voice recognition and speech signal processing, e.g., in cochlear implants.

### Competing interests

The authors declare that they have no competing interests.

Received: 8 December 2015 Accepted: 18 October 2016

Published online: 24 October 2016

### References

1. W Hess, *Pitch determination of speech signals: algorithms and devices* (Springer, Berlin, New York, 1983)
2. J Harrington, S Cassidy, *Techniques in speech acoustics* (Springer, Netherlands, 1999)
3. PG McLeod, Dissertation, fast, accurate pitch detection tools for music analysis, University of Otago, 2008
4. M Staudacher, V Steixner, A Griessner, C Zierhofer, in Tagungsband zur OEGBMT-Jahrestagung 2014: Grundfrequenzbestimmung in Sprachsignalen durch adaptive Kreuzkorrelation, ed. by Department für Biomedizinische Informatik und Mechatronik, OEGBMT-Jahrestagung 2014, Hall in Tirol, September 2014, Lecture Notes in Biomedical Computer Science and Mechatronics, vol. 4, p. 22
5. P Bagshaw, Evaluating pitch determination algorithms, <http://www.cstr.ed.ac.uk/research/projects/fda/>. Accessed 21 Oct 2016.
6. F Plante, G Meyer, W Ainsworth, A pitch extraction reference database. Paper presented at the 4th European Conference on Speech Communication and Technology, EUROSPEECH 1995, Madrid, 18–21 September 1995
7. G Pirker, M Wohlmayr, S Petrik, F Pernkopf, A pitch tracking corpus with evaluation on multipitch tracking scenario, INTERSPEECH 2011. Paper

- presented at the 12th Annual Conference of the International Speech Communication Association, Florence, 27–31 August 2011
8. N Seo, Project: pitch detection, <http://note.sonots.com/SciSoftware/Pitch.html>. Accessed 21 Oct 2016.
  9. S Xuejing, Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio. Paper presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Orlando, 13–17 May 2002
  10. S Xuejing, Pitch determination algorithm (2002), <http://www.mathworks.com/matlabcentral/fileexchange/1230-pitch-determination-algorithm/content/shrp.m>. Accessed 21 Oct 2016.
  11. A De Cheveigne, H Kawahara, YIN, a fundamental frequency estimator for speech and music. *J. Acoust. Soc. Am.* **111**(4), 1917–1930 (2002). doi:10.1121/1.1458024
  12. A De Cheveigne, YIN pitch estimator, <http://audition.ens.fr/adc/sw/yin.zip>. Accessed 21 Oct 2016.
  13. O Derrien, A very low latency pitch tracker for audio to MIDI conversion. Paper presented at the 17th Int. Conference on Digital Audio Effects, Erlangen, 1–5 September 2014
  14. E Azarov, M Vashkevich, A Petrovsky, in Proceedings of the 20th European Signal Processing Conference (EUSIPCO): instantaneous pitch estimation based on RAPT framework, Bucharest, August 2012, pp. 2787–2791
  15. B Doval, X Rodet, Estimation of fundamental frequency of musical sound signals. Paper presented at the International Conference on Acoustics, Speech and Signal Processing (ICASSP-91), Toronto, 14–17 April 1991
  16. A Camacho, JG Harris, A sawtooth waveform inspired pitch estimator for speech and music. *J. Acoust. Soc. Am.* **124**(3), 1638–1652 (2008). doi:10.1121/1.2951592
  17. S Calderón, G Alvarado, A Camacho, AUD-SWIPE-P: A parallelization of the AUD-SWIPE pitch estimation algorithm using multiple processes and threads. Paper presented at the Conference on Parallel and Distributed Computing and Networks - PDCN 2013, Innsbruck, 11–13 February 2013
  18. PC Bagshaw, SM Hiller, MA Jack, in EUROSPEECH '93: enhanced pitch tracking and the processing of  $F_0$  contours for computer aided intonation teaching, European Conference on Speech Communication and Technology, Berlin, September 1993, pp. 1003–1006
  19. A Bánhalmi, K Kovács, A Kocsor, T László, Fundamental frequency estimation by least-squares harmonic model fitting. Paper presented at the 9th European Conference on Speech Communication and Technology, INTERSPEECH 2005 - Eurospeech, Lisbon, 4–8 September 2005
  20. K Wagener, T Brand, B Kollmeier, Entwicklung und Evaluation eines Satztests für die deutsche Sprache I: design des Oldenburger Satztests. *Zeitschrift für Audiologie/Audiological Acoustics* **38**, 4–15 (1999)

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)