TR - H - 195

Speech Fundamental Frequency Estimation

Alain de Cheveigné

1996. 5. 27
(1996. 3.21 受付)

ATR人間情報通信研究所
〒619-02 京都府相楽郡精華町光台2-2 ☎ 0774-95-1011

ATR Human Information Processing Research Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan
Telephone: +81-774-95-1011
Facsimile: +81-774-95-1008

© ㈱ATR人間情報通信研究所

Speech Fundamental Frequency Estimation

Alain de Cheveigné

Abstract

Several methods for voiced speech Fundamental Frequency (F_0) estimation were implemented and evaluated on a database of speech recorded together with a laryngograph signal. A first approach was based on an algorithm originally designed for concurrent speech (two-voice) F_0 estimation. The idea was that, by modeling the speech signal as the sum of two periodic signals, the algorithm would deal with common causes of F_0 estimation failure: strong harmonics or subharmonics (diplophony), changes in amplitude and spectrum (modeled by the algorithm as a local beat pattern), periodic interference (hum, interfering speech, reverberation), etc.. The approach turned out to be less effective than expected and was abandoned. The second approach, based on a careful error analysis of the classic AMDF algorithm, resulted in several new schemes to avoid errors. Combined, these schemes reduced errors over the database in a ratio of 3.5 for a male voice and 9 for a female voice, and allowed the algorithm to outperform the standard ESPS get_f0 algorithm by a factor of about 2.

1 Introduction

Fundamental frequency estimation is an old and elusive problem that has inspired much effort and many ingenious ideas. Hess's treatise on the question [29] covers the period up to 1983, but many schemes have been proposed since [38, 31, 22, 5, 34, 1, 2, 3, 6, 4, 7, 8, 9, 10, 11, 12, 16, 17, 18, 20, 23, 25, 26, 27, 30, 33, 36, 37, 40, 41, 42, 45, 19, 32, 35].

In this section we review some recent ideas in the field of F_0 estimation, and discuss a few points that need clarifying. In the next section we present our methodology and database, and derive some useful statistics about signal characteristics that correlate with F_0 estimation errors. We then describe a number of improvements to the classic AMDF (Average Magnitude Difference Function) algorithm [46], and evaluate each one using a laryngograph-labeled database. The final version of the algorithm is compared with other F_0 estimation schemes.

1.1 Approaches to F_0 estimation

A rich variety of ideas for F_0 estimation continues to be published each year. Unfortunately many "new" ideas are fundamentally equivalent to earlier ones, and run into the same difficulties. Attempts such as those of Ney [40], Hess [29], and more recently Doval [19], to classify schemes and clarify their interrelations are most welcome.

Most methods model the signal as a periodic function of time. Supplied with a truly periodic input (with a period in the range that they were designed to process), almost all will function perfectly. Some notable exceptions are:

- low-pass filtering, that only works if a fundamental frequency component is present,
- the cepstrum method, that assumes that the *spectrum* is periodic (often true for speech, not true in general),
- inverse filtering, that seeks a unique epoch per period.

Apart from particular weaknesses such as these, all methods are faced with the same fundamental difficulty due to the imperfect periodicity of signals such as speech. This was well analyzed in the theses of Doval [19] and Geoffrois [24]. For any periodic function there is a countably infinite set of strictly positive numbers T such that:

$$\forall t, s(t) = s(t+T) \tag{1}$$

The set allows a smallest element T_0 that is the mathematical period. This notion is well defined in the case of perfect periodicity, but not so in the case of *approximate* periodicity. The problem is not so much the definition of approximate periodicity: that is easily done by relaxing some of the constraints in Eq. 1, either restricting the condition to some particular range of t, or else replacing equality by some form of

"similarity", or possibly even allowing the "constant" pseudo-period T to vary within limits over the range of t:

$$\forall t \in range, \quad s(t) \sim s(t + T_t) \tag{2}$$

The set of Ts for which this is true, and therefore its smallest element, depends on the parameters that define approximate periodicity. A slight change may cause a big jump of the pseudo-period: algorithms based on this definition are not robust. If the definition is made more severe, the jump is towards a multiple of the "true" period. If it is relaxed, the jump may be to a divisor corresponding to a strong harmonic. These "subharmonic" and "superharmonic" errors (often inaccurately called "octave" errors) are thus a consequence of the ambiguity of the definition of the "period" of an imperfectly periodic signal.

In practice, the usual course is to search for a design or parameters that give the best tradeoff between the two types of error. It is usually possible to adjust an estimation algorithm to reduce the occurrence rate of one type of error, usually at the expense of the other. Sometimes the bias is implicit, as in the gradual tapering of the autocorrelation function (see item 3 of Section 1.3). In other cases it may depend on a large set of parameters that allow for endless adjustment and "tinkering". Any insight that can reduce the need for blind adjustment is welcome.

The problem may be viewed in terms of projection in a vector space of functions. The notion of "approximate" periodicity is akin to the projection of the signal on a subspace of perfectly periodic signals. The precise projection (and therefore the period estimate) depends on details of the distance that serves to define the closest member of the subspace [19]. Various methods differ in this respect, each searching for a distance that will somehow prove more reliable than others.

A different course is to replace the set of periodic functions by a larger set of signals that are *approximately* periodic (but whose period may nevertheless be defined unambiguously). The set might contain frequency- or amplitude-modulated periodic signals (with certain constraints on the modulation), or sounds produced by a variable source and a constant filter (or vice-versa), or typical notes of a set of musical instruments, etc. [19]. The idea is that the signal to estimate will be better matched by a well defined quasi-periodic function than by a periodic signal. This supposes that certain assumptions can be made about the class of signals to be estimated. These can take the form of a priori knowledge (for example of speech production), or they may be learned. The probabilistic approach of Doval [19] is probably the most sophisticated and well-argumented example of the learning paradigm, but there are other examples [5, 32]. The risk is of course that assumptions (learned or otherwise) may prove to be wrong for some class of signals, for which performance will be degraded rather than improved.

Geoffrois [24] argues that, given the inherent difficulty of F_0 estimation, one should concentrate instead on the more realistic goal of estimating the *variation* of F_0 with time. F_0 changes are perceptually salient, and may be correlated with intonative features [39]. Nevertheless, there remain applications for which an absolute F_0 estimation is useful.

A powerful strategy is to combine evidence over time, on the assumption that F_0 varies without discontinuous jumps. For example several candidates may be selected for each frame, and a path traced among them using a DP-matching or HMM algorithm [40, 47, 24, 19]. Among the possible criteria for choosing a canditate estimate are F_0 proximity with the previous estimate, "plausibility" scores for each candidate, and information about the direction and rate of F_0 change derived from a F_0-change estimation algorithm [24, 39]. The success of such an algorithm depends on:

- The presence of the correct estimate among candidates,
- The quality of "plausibility" information and continuity constraints that favor the correct estimate,
- The lack of similar support for rival candidates.

There is a risk that the algorithm may lock on to an incorrect path at some point, and continue to enforce it thereafter. Other problems are that the behavior is rather difficult to predict and analyze, and that the extra parameters encourage more tinkering. As far as possible, efforts should be devoted to improving the basic estimate, before handing it over to correction algorithms.

1.2 Error factors

Well-known factors that cause F_0 estimation errors are:

- Changes in F_0, overall amplitude and spectral shape due to articulation, that cause successive periods to be different, so the repetition of a motive is no longer evident and the assumptions of stationarity or slow variation used by processing schemes are violated.
- Strong harmonics that masquerade as a fundamental, causing "octave" (more accurately: "harmonic") errors.
- The fact that period-to period similarity also occurs over period multiples, causing "sub-octave" (more precisely: "sub-harmonic", or "super-period") errors.

In some cases vocal fold vibration itself is irregular (diplophony, creaky voice, etc.), in which case F_0 is difficult to define [27, 21]. Later on we discuss statistics that relate the F_0 error rate to various characteristics of the speech signal (amplitude, amplitude change, F_0, F_0 change, spectral change).

1.3 Windows

It is worthwhile clarifying a few points concerning windows.

1. Size and shape are usually chosen to obtain spectral peaks of suitable shape and width. There is however another important consideration. If a short-term

transform such as STFT, autocorrelation, AMDF, etc. is applied to a quasi-periodic signal, the integration can be seen as a low-pass filter that removes most of the energy at the fundamental and harmonics, leaving only the "DC" component representing the quantity to be estimated. This smoothing allows the estimate to be sampled reliably. If the window is too short the estimate fluctuates during the period, and the sampled estimate depends arbitrarily on the sampling phase. For effective smoothing, a square window must be at least as long as the longest period expected, and most other windows must be at least twice as long. The longer the stabler, with due regard to the conflicting constraints of tracking pertinent signal variations.

2. A consequence of the previous remark is that variable-window analysis techniques that have been proposed for F_0 estimation [31, 22, 35] are likely to give unstable estimates. For example, if the autocorrelation function is calculated with a window size proportional to lag, the low-lag part of the function is the result of integration with a relatively short window. Given a periodic signal to estimate, the portion of the ACF with lag shorter than the period is insufficiently smoothed, and therefore unstable. Since F_0 estimation relies on comparison between different parts of the ACF, it too will be unreliable.

3. The usual definition of the autocorrelation function

$$acf(\tau) = \frac{1}{n} \sum_{i=1}^{n-\tau} s_i s_{i+\tau}$$

is unfortunate for two reasons. One is that the integration time is short when the lag τ is large, which makes the estimate unstable as explained previously. The other is that the overall decrease of $acf(\tau)$ with τ causes a bias towards shorter period estimates. This is sometimes cited as an advantage, as it avoids locking to subharmonics. We suggest that such a bias, if useful, had better be applied explicitely.

In conclusion, a square integration window should be at least as long as the longest anticipated period. Other shapes may need to be at least twice as long. Window size should not vary, either explicitely or implicitely due to the limits of the total analysis window (for example if the ACF is calculated by FFT). Lag-domain functions such as ACF and AMDF are calculated up the longest expected period, and the calculation thus involves a total duration of two to three periods, which is the figure usually cited as a minimum window size.

2 Methods

We wished to evaluate various F_0 estimation algorithms. The task was to estimate F_0 of speech in a range of 50 to 800 Hz (4 octaves). Each algorithm or variant was evaluated quantitatively on a database of speech recorded with a laryngograph signal. Evaluation was differential: the effect of each "improvement" was assessed by comparing error rates with and without it, and an "absolute" measure of overall performance was made by comparison with a single well-known baseline algorithm (AMDF)[1].

2.1 Database

The evaluation database consisted of speech and laryngograph data recorded together. It is easier and more reliable to estimate F_0 from the laryngograph (lx) track than from the speech waveform (for example by hand-labelling of pitch markers), although problems related to aperiodic phonation (fry, diplophony, etc.) still remain. As there is no single definition of F_0 in that case, such portions were marked as unvoiced in the database.

Speakers were one japanese female (FHS) and one british male (NC). Speech was recorded in quiet conditions, with a microphone placed relatively close to the speaker's mouth. Speech and laryngograph (lx) signals were digitized with 16 bit resolution at a sampling rate of 16 kHz. Their alignment was verified by calculating the cross-correlation between the half-wave rectified differentiated lx, and the twice-differentiated speech waveform. From the position of the cross-correlation peak, the delay between lx and speech (due to propagation time from glottis to microphone) was judged to be between 0.56 and 1.0 ms.

For practical reasons, only a subset of the available data for each speaker was used. Within each file only the voiced portions (according to a laryngograph-based criterion) and at least 50 ms of speech on both sides were retained. For each speaker a subset of 100 files was selected. For NC (male) this represented 353s of speech, of which 153s were voiced. For FHS (female) it represented 398s of speech, of which 272s were voiced. The selected files were those that caused the largest number of gross errors for the baseline algorithm, in proportion to voiced duration.

The average F_0 was 230 Hz for FHS and 99 Hz for NC. Appendix B gives more detailed statistics in the form of histograms, including histograms that relate errors made by the baseline algorithm to signal characteristics: F_0, F_0 change, amplitude, amplitude change and spectral change.

2.2 Laryngograph-based F_0 estimate

The laryngograph measures the resistance of body tissues between two electrodes applied to the speaker's throat, on both sides of the larynx. The resistance varies during the glottal cycle, and falls sharply when the glottal folds meet. Movements

[1]This process would be greatly enhanced if a standard freely sharable database were available.

of the subject's body or articulators also cause large resistance variations on a slower scale. These are partly eliminated by a feedback loop that acts like a high-pass filter, but there remains a large "DC" component in the laryngograph signal that must be eliminated. The laryngograph waveform was differentiated to obtain a series of pulses coinciding with glottal closure. Spurious pulses were eliminated by an algorithm detailed in Appendix A, and the F_0 was calculated as the inverse of the spacing between pulses. The laryngograph F_0 estimate is authoritative, so it must be prepared with care.

From the lx signal was also derived a voicing decision based on glottal pulse regularity. Roughly speaking, at least three regularly-spaced pulses were required for a speech portion to be declared "voiced". Aperiodic phonation such as fry or diplophony was classified as "unvoiced". We have no clear definition of F_0 to propose in that case and it would be unfair to expect an F_0 estimation algorithm to do any better. The criterion is rather severe in terms of speech production (a reasonable alternative would be to say that any glottal pulse, even isolated, is a form of "voicing"). However it is essential that we eliminate all possibility of error of our authoritative F_0 algorithm. Voicing detection was automatic, and parameters were adjusted to remain "on the safe side". One must keep in mind that the database may therefore be a bit biased on the "easy" side.

2.3 Error rating

Error rates for waveform-based estimation were measured by comparison with the laryngograph-based estimate. Errors were counted according to the following rules:

- If the speech period estimate was within 20% of the lx period estimate, it was considered correct.
- If it was within 20% of a lx period multiple, a *subharmonic error* was counted.
- Otherwise a *gross error* was counted.

A deviation greater than 20% corresponds to the definition of a "gross pitch error" used by Bagshaw [3] and is equivalent to the 0.25 oct criterion of Krubsack and Niederjohn [34]. It corresponds to the 1ms criterion of Rabiner et al. [43] at an F_0 of 200 Hz. We ignored "fine pitch errors", as any of a number of techniques can be used to refine accuracy once a coarse estimate is available. Instead we concentrated our efforts on deriving this coarse estimate reliably, which is the major difficulty of F_0 estimation.

It is customary to pool "gross" and "subharmonic" errors together. Nevertheless we count subharmonic errors apart, as they reflect the basic equivalence of periods and superperiods rather than a malfunction of the algorithm. The techniques that allow to avoid them are different from the techniques that avoid other types of error.

2.4 Baseline speech F_0 estimation algorithm

The classic AMDF (Average Magnitude Difference Function) [46] was chosen for the baseline algorithm. The AMDF is defined as:

$$A_i(\tau) = \sum_{k=i}^{i+L} |s_k - s_{k-\tau}|$$

where i is the analysis index, L is the length of the (square) integration window, and τ is the lag. The absolute value of sample-to-sample differences corresponds to a city-block distance, but other choices are possible. For example the squared difference corresponds to Euclidean distance, and in this case the "AMDF" is closely related to the autocorrelation function [40]. Given our conventions, $A_i(\tau)$ is the distance between a fixed window extending to the right of analysis point i, and a sliding window shifted τ samples to the left (past).

The period is estimated as the absolute minimum of the AMDF within a range of allowable period values:

$$P_0 = argmin(A_i)_{P_{min}<i<P_{max}}$$

The values of P_{min} and P_{max} were chosen to allow a range of F_0s of 50 Hz to 800 Hz. The window length was 20ms, or one period of the lowest expected F_0.

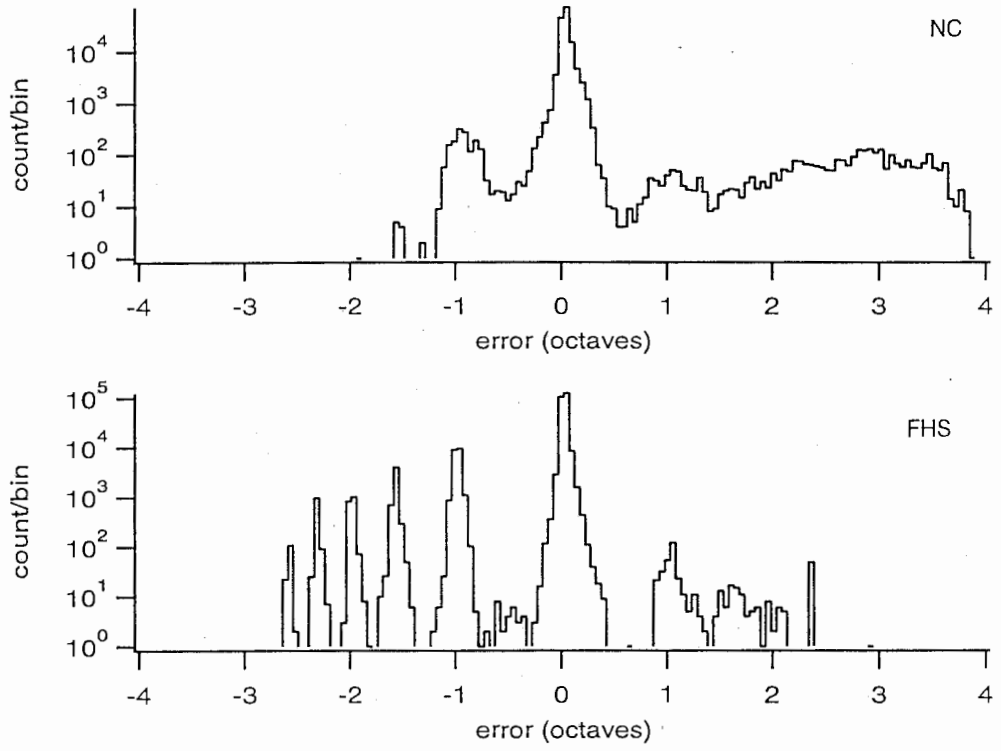

Fig. 1 *Histograms of deviation of the baseline algorithm F_0 estimate from the correct value, for male speaker NC (top) and female speaker FHS (bottom). Note the logarithmic ordinate.*

Gross error rate for NC was 5.4%, and subharmonic error rate was 0.98%. Gross error rate for FHS was 0.27%, and subharmonic error rate was 10.1%. The error distributions for NC and FHS are very different (Fig. 1). One explanation is that the different F_0 distributions relative to the search range allow different kinds of error. For example, the pitch range of NC being low, there is little room for subharmonic errors.

2.5 Time alignment of baseline and larygograph estimates

To produce a period estimate, AMDF integrates information over a certain portion of the waveform, whereas the laryngograph F_0 estimator measures the delay between two pulses. How should these estimates be time-aligned? For the lx estimate, our convention is to associate with each point of an interpulse interval the period that separates them (if it is short enough to be within range). For the AMDF, the period estimate is associated with each analysis point (left edge of the fixed window). The estimate of each AMDF analysis point is compared to the lx estimate for the interval where it is situated. This rough alignment can be improved by introducing a time shift between the estimates. Fig 2 shows how the error rate varies with alignment for both speakers for a modified version of AMDF (Section 4.3). The minimum occurs when the speech is delayed relative to the lx by 5 to 10ms. For each algorithm tested, it is important to perform a similar check of time alignment.

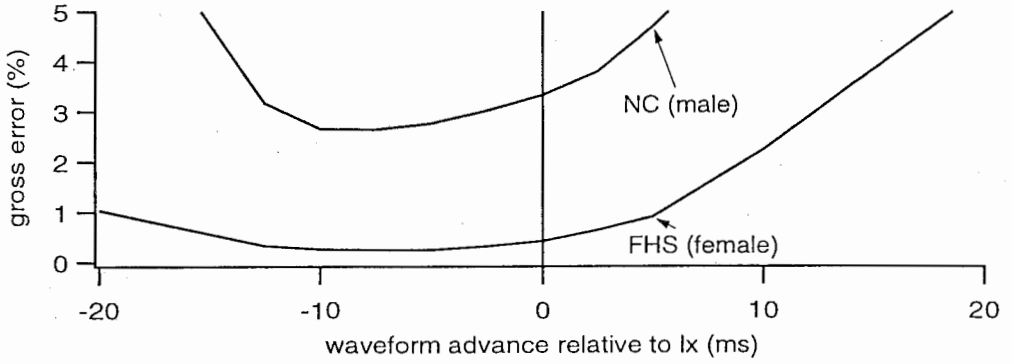

Fig. 2 *Error rate as a function of the temporal alignment between laryngograph-based and waveform-based F_0 estimate.*

3 DDF algorithm

This algorithm was meant to be the main object of our work. It turned out to be disappointing, and was finally abandoned. We describe it nevertheless.

The Double Difference Function (DDF) algorithm was designed for F_0 estimation of two concurrent voices. It is a two-parameter extension of the Average Magnitude Difference Function (AMDF) algorithm. Speech, modeled as the sum of two periodic functions, is fed to two cascaded time-domain comb filters. The parameter space is searched for a pair of lags (within a certain range) that minimizes filter output. This pair constitutes the estimate of the fundamental periods of the two voices [13].

The idea is to apply this algorithm, designed for 2 voices, to the task of estimating the F_0 of a single voice. The rationale is the following. By modeling a single voice as the sum of two periodic signals, the algorithm should be able to cope with a number of phenomena that make F_0 estimation difficult:

- A strong harmonic, modeled by one of the voice estimates.
- A subharmonic (diplophony, etc.), modeled by one of the voice estimates.
- A change in amplitude or spectrum, modeled locally as a beat pattern between two signals of similar period.
- Harmonic interference (voice, hum, computer noise, reverberation, etc.)

An additional motivation comes from evidence that the auditory system might employ a mechanism similar to time-domain comb-filtering to segregate concurrent harmonic sounds [15]. This suggests the interesting hypothesis that the same mechanism might involved in pitch perception of approximately periodic sources such as speech.

The algorithm produces two estimates, one of which must be chosen using a post-processing algorithm such as mentioned in the introduction. Our key assuption was that one of the two estimates would be correct with a greater probability than if we took the best and second-best candidate of a single-voice algorithm, as would normally be used by a post-processing algorithm. It turned out that this assumption was false. Defining error rate as the proportion of frames for which *both* estimates were incorrect, the error rate was 4.5% for NC (male) and 5.4% for FHS (female) for the DDF algorithm. For the best and second best estimate of the ordinary AMDF algorithm, the figures were very similar: 5.7% for NC and 5.4% for FHS. Thus, contrary to expectations, the DDF algorithm did *not* guarantee at least one correct estimate among the two it provided.

A possible explanation of this fiasco is the following. The two-period signal model is indeed a better match to voiced speech than the one-periodic signal model, but the set of allowable period pairs is not sufficiently constrained. Take for example a periodic signal that is the sum of two components (for example harmonics 3 and 4). The algorithm can choose the fundamental period, but also any combination of multiples of the periods the two components, many of which lead to incorrect F_0 estimates.

This simple example demonstrates how the algorithm can fail to find the period of a perfectly periodic signal.

Since the DDF algorithm involves exhaustive search and is extremely time consuming we decided, given its lack of clear advantage in terms of performance, to abandon it.

4 Improving AMDF

First of all we examine the pattern of errors made by standard AMDF, searching for the causes. Then we explore a number of schemes to address these causes. Each scheme is implemented in isolation and compared to standard AMDF. Finally, schemes are combined to obtain an F_0 estimation method that is considerably more reliable than standard AMDF.

4.1 Error analysis

The histograms of errors committed by the baseline algorithm (Appendix B) reveal factors that are correlated with error rates. Correlation does not imply cause, but the insight that the histograms provide is nevertheless useful.

Errors are common when signal level is low (Figs. 5, 10). Explanations are:

- At the end of an utterance the level falls while phonation often becomes irregular.
- Certain articulatory events cause both a fall in level and a rapid spectral change that interferes with F_0 estimation.
- Analysis of low-level portions is easily contaminated by neighboring high-level portions.

Errors are common when there is a *change* in level, at least for the male speaker NC (Fig. 7). The relation is here likely to be causal, as a level change makes period-to-period comparisons less good. In addition, a change in level introduces an estimate bias in methods such as AMDF or autocorrelation that involve a sliding window, because the norm over the sliding window varies with lag. For increasing level, AMDF is biased toward short periods and autocorrelation towards long periods. For decreasing level the bias is in the opposite direction. For NC, errors are more common for a decrease in level than an increase, probably because decreases are associated with the end of an utterance, where phonation becomes irregular.

Errors are common when F_0 is low (Figs. 6, 11). This may be partly a consequence of the limits of the search range that put a constraint on the possible errors. A second possible explanation is that F_0 tends to fall at the end of an utterance where phonation is irregular. A third is that period-to-period similarity tends to be less good for long periods.

Errors are common when there is a *change* in F_0 (Fig. 8, 13). The explanation is that the period changes during the analysis window, and the algorithm has difficulty estimating this moving target. This seems to be particularly the case for the female speaker FHS (Fig. 13).

Finally, errors are associated with spectral changes (estimated in terms of period-to-period cepstral distance), but this relation is not very strong (Figs. 9, 14).

4.2 Amplitude normalization 1

This scheme aims to compensate the estimation bias that occurs when the sliding window shifts to higher- or lower-level signal portions. The AMDF (defined in Section 2.4) is divided by the average of the norm over fixed and sliding windows:

$$A'_i(\tau) = 2A_i(\tau) / \sum_{k=i}^{i+L} (|s_k| + |s_{k-\tau}|)$$

Gross error rate was 3.3% (baseline: 5.4%) for NC and 0.25% (baseline: 0.27%) for FHS. Subharmonic error rates were practically unchanged.

4.3 Amplitude normalization 2

This variant involves amplitude-normalization of the waveform followed by standard AMDF. This form of normalization compensates not only for bias, as the previous scheme, but also for other consequences of amplitude change. Each waveform sample is divided by the norm over a window centered on that sample:

$$s'_i = s_i / \sum_{k=i-L/2}^{i+L/2} |s_i|$$

Window size L is a parameter. If too large, fast changes are not properly compensated. If too small, the modulation of the fine structure within the period is attenuated. L was given the same size as the AMDF integration window (20 ms).

Gross error rate was 2.5% (baseline 5.4%) for NC and 0.39% (baseline: 0.27%) for FHS. Subharmonic errors were increased (1.9% vs 0.98% for NC; 12.9% vs 10.1% for FHS), as a consequence of the greater waveform similarity at long lags.

4.4 Amplitude normalization 3 (Barry Verco's idea)

This idea was originally proposed by Barry Vercoe (MIT). An AMDF-like function is defined as:

$$A_i(\tau) = \sum_{k=i}^{i+L} d(2s_k - (s_{k-\tau} + s_{k+\tau}))$$

Each sample is compared to the average of samples situated τ samples on both side. If the amplitude change is linear, the average of samples one period in the past and one period in the future should be equal to the current sample, so the effect of the amplitude change should be canceled. The scheme might also reduce the effect of period-to-period timbre differences, if one assumes that speech follows a locally linear trajectory in timbre space.

Gross error rate was 4.6% for NC (baseline 5.4%) and 0.23% for FHS (baseline: 0.27%). Subharmonic errors were somewhat reduced (0.47% vs 0.98% for NC; 8.2% vs 10.1% for FHS). Overall, the performance of this elegant scheme is disappointing. It is possible that it is handicapped by the relatively large analysis window needed by windows sliding in opposite directions.

4.4.1 Split window AMDF 1

The previous three schemes addressed amplitude change. This scheme addresses F_0 change, that Figs. 8 and 13 show is highly correlated with F_0 estimation errors. Fig. 3 shows that the F_0 may change by up to 0.05 octave (approximately 5%) over a duration of 10 ms (one half the window size). It is easy to understand that such a large period change makes period-to-period comparison more difficult. A shorter integration window alleviates this problem, at the expense of stability of analysis. Some authors have proposed to preprocess the waveform by a time warp [40, 44]. The drawback is that that the warp distorts the fine structure of the period together with the period itself, and thus is not well adapted to the hypothesis of an F_0 change with constant (or uncorrelated) spectral shape.

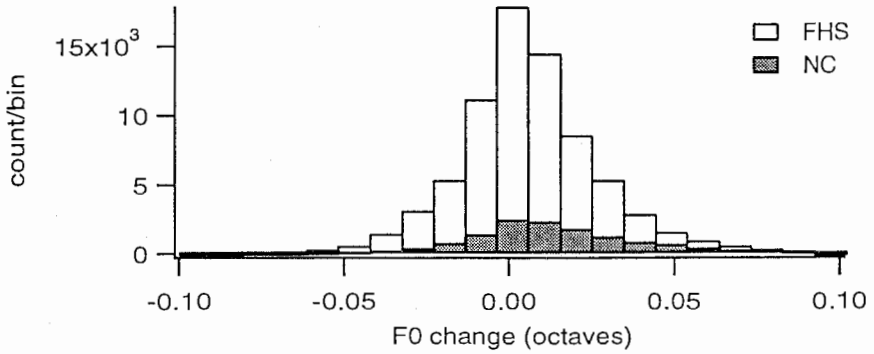

Fig. 3 *Histogram of F_0 change over 10 ms. Open: speaker FHS, filled: speaker NC.*

One crude solution is to split the window into two portions, and allow a small difference in lag to be introduced between both halves. The optimum difference is found by exhaustive search within a $\pm 5\%$ range.

Gross error rate was 3.6% (baseline: 5.4%) for NC, and 0.25% (baseline: 0.27%) for FHS.

4.5 Split window AMDF 2

A more sophisticated scheme is to introduce a linear warp in the AMDF calculation, and choose the warp factor ("angle") that gives the deepest minimum. For speed, the integration window is divided into 16 16-sample slices. For each slice a partial AMDF is calculated. Then all 16 partial AMDFs are summed, with a certain warp factor along the lag axis. Nine warps are allowed, spanning the range -5 to +5 octaves/s. The warp produces fractionary indices. Array lookup with a fractionnary index may be implemented either by linear interpolation between array values for adjacent indices immediately inferior and superior to the fractionnary index, or more simply (and faster) by taking their min. The latter scheme also produces lower error rates.

Gross error rate was 3.8% (baseline 5.4%) for NC and 0.27% (baseline 0.27%) for FHS. Despite the greater sophistication, error rates were no better than for the previous scheme.

4.6 Mean-normalized AMDF

The AMDF has a dip at zero lag, often followed by secondary "spurious" dips related to strong harmonics. These may happen to be deeper than the dip at the period, because of short-term correlations that enhance similarity at short duration, causing a "too-high" error. The problem is acute if the search range includes short periods, or if some form of bias is introduced to avoid subharmonic errors.

This problem may be addressed by dividing the AMDF function at each lag by the mean of the AMDF over shorter lags [13]:

$$A'_i(\tau) = \tau A_i(\tau) / \sum_{k=1}^{\tau} A_i(k), \quad A'_0 = 1$$

This new function starts at 1 rather than zero, and the first dips are de-emphasized.

Gross error rate was 2.1% (baseline: 5.4%) for NC, and 0.23% (baseline: 0.27%) for FHS. Subharmonic errors were little affected.

4.7 First minimum below threshold

Up till now we have ignored subharmonic errors. In practice they are a nuisance, and all the more so as any bias introduced to avoid them them may result in more errors of other kinds. Contrary to gross errors, subharmonic errors tend to occur when the waveform is highly periodic. Adjacent periods may happen to be slightly less similar than non-adjacent periods, because of limited sampling resolution, a noise burst, etc.. Taking the global minimum thus leads to choosing a subharmonic. A simple scheme to avoid this is to choose the *first* dip that falls below some threshold. If no dip falls below threshold, the absolute minimum is used. For this to work properly, the AMDF must first somehow be normalized, and for that reason it is convenient to combine this scheme with the previous one (mean-normalized AMDF).

Combining the previous and present schemes (with a threshold of 0.4), the subharmonic error rate fell from 0.99% to 0.73% for NC, and from 10.1% to 0.92% for FHS (a more than 10-fold reduction!). The gross error rate remained practically unchanged.

4.8 Euclidean vs city-block distance

The city-block distance (sum of absolute differences) of AMDF takes into account waveform differences in proportion to their size. In contrast, the more common Euclidean distance (sum of squares) gives more importance to large localized differences than to smaller distributed differences:

$$A_i(\tau) = \sum_{k=i}^{i+L} (s_k - s_{k-\tau})^2$$

With Euclidiean distance, the gross error rate was 5.2 % (baseline 5.4 %) for NC and 0.30 % (baseline: 0.27%) for FHS. Subharmonic error rates were not affected.

One may choose instead to de-emphasize large local differences by taking the square root:

$$A_i(\tau) = \sum_{k=i}^{i+L} \sqrt{|s_k - s_{k-\tau}|}$$

Gross error rate was 5.4% (baseline 5.4%) for NC , and 0.28% (baseline: 0.27%) for FHS. The fourth root produced similar results. Neither square nor root offers a significant advantage over the absolute difference.

4.9 Sampling rate

Limited sampling rate degrades the period-to-period match if the signal period is not a multiple of the sampling period. This is likely to to be particularly troublesome if F_0 is high. A solution is to upsample the signal before calculation.

With an upsampling factor of 2, the gross error rate was 4.6% (baseline 5.4%) for NC and 0.27% (baseline 0.27%) for FHS. Contrary to expectations, the improvement in rate was greatest for NC, despite the fact that FHS has a smaller average period. The subharmonic error rate for FHS was however reduced: 6.8% (baseline: 10.1%).

Computation time varies in principle with the square of the upsampling factor, but in practice sparse calculation techniques (see below) allow the dependency to be linear.

A simple hack that gives results similar to upsampling is to set the difference $|s_i - s_{i-\tau}|$ to zero everywhere it changes sign as a function of τ. With this hack, gross error rates were 4.6% (baseline 5.4%) for NC and 0.25% (baseline 0.27%) for FHS.

4.10 LPC residual

The LPC inverse filter has been proposed as a preprocessor for F0 estimation. The residual of a 16th order LPC analysis was calculated (using SpeechTools "lpc_run" and "residual" programs), and substituted for the speech waveform.

The gross error rate was 7.5% (baseline 5.4%) for NC and 0.27% (baseline 0.27%) for FHS. Subharmonic error rates were also somewhat increased: 2.0% for NC (baseline 0.99%) and 14.6% for FHS (baseline 10.1%). LPC inverse filtering is not effective.

4.11 Putting it all together

In the previous paragraph the schemes were tested in isolation. Here they are combined: speech waveform amplitude normalization, split window, mean-normalization, and choice of the first minimum below threshold.

The gross error rate was 1.1% (baseline 5.4%) for NC and 0.24% (baseline 0.25%) for FHS. The subharmonic error rate was 0.73% for NC (baseline 0.99%) and 1% for FHS (baseline 10.1%). Together, the schemes reduced the total error rate (gross + subharmonic) by a factor of 3.5 for NC, and 9 for FHS. The main parameters are

integration window size (20 ms) and threshold (0.4). Little effort was invested in tuning them, so it is unlikely that the method was tailored to our database.

4.12 Comparison with a standard algorithm

The ESPS program get_f0 was run on the same data, with default parameters except for maximum F_0 (800 instead of 550) and frame interval (1ms instead of 10 ms).

Gross error rate was 3.1% for NC, and 0.26% for FHS. Subharmonic error rate was 0.22% for NC, and 14.6% for FHS. The total error rates for standard AMDF, ESPS and "improved" AMDF are summarized in the following table:

	standard AMDF	ESPS get_f0	"improved" AMDF
NC	6.4%	3.3%	1.8%
FHS	10.8%	2.8%	1.2%

On our database the improved AMDF made about half as many errors as the standard ESPS get_f0 program.

5 Conclusion

Analysis of errors made by the classic AMDF F_0 estimation algorithm showed the importance of non-stationarities such as changes in level and F_0. A scheme involving the use of a two-voice F_0 estimation algorithm to deal with these and other sources of error proved disappointing. A more pragmatic approach that involving modifications of AMDF to address each source of error was more successful. Error rates were reduced by a factor of 3.5 to 9 relative to standard AMDF, and were smaller by a factor of about 2 than error rates of the standard ESPS get_fo algorithm. All evaluations used a database of male and female speech recorded together with a laryngograph track.

6 Acknowledgements

This research was conducted within the framework of a collaboration agreement between ATR Human Information Processing Research Laboratories and the Centre National de la Recherche Scientifique. I thank ATR for its kind hospitality, and the CNRS for leave of absence.

Hideki Kawahara offered useful advice and kindly set up laryngograph equipment. Mary Beckman offered the use of the databases she recorded, and Nick Campbell arranged access to several ATR-ITL databases, including data he recorded himself. Yoshinoru Sagisaka and Norio Higuchi kindly agreed to make a subset of the database freely available as a reference to allow future research to make meaningful comparisons between methods.

A Laryngograph-based F_0 estimation

Our evaluation scheme relies critically on the reliability of the reference F_0 estimate derived from the laryngograph signal. The lx F_0 estimation algorithm is therefore described in some detail.

Objectives are: (1) Automatic estimation with no manual correction, to save time and avoid inconsistent estimation criteria between different parts of the database, (2) No unvoiced portions labelled incorrectly as voiced, and no estimation errors within voiced portions, to avoid causing the speech-signal estimation algorithm to be unjustly penalized on these portions, (3) Few voiced->unvoiced labelling errors (these are more tolerable, but there should be as few of them as possible to avoid eliminating "difficult" portions from the database), (4) Good accuracy of the estimate at all times, (5) Practically unlimited F_0 range.

Smoothing, median filtering and postprocessing in general are rejected, as they tend to be incompatible with objective (2).

Reliable F_0 estimation from the lx is easier than from the speech waveform, but it is not trivial. The lx signal tends to have a strong, slow-varying "DC" component that must be filtered out by high-pass filtering, and this tends to reinforce noise components. The amplitude of the voice component may vary with the electrode position relative to the vocal folds, or the quality of skin contact, and these vary with articulation or movements of the speaker. The signal contains many "events", unrelated to vocal phonation, and these must be ignored when estimating the F_0.

Criteria for selecting voice-related events are strength and regulatity. Voice pulses tend to be strong, but they are sometimes weak, particularly at the beginning and end of phonation. Conversely, spurious pulses are for the most part weak, but a good proportion are strong. Voice pulses tend to occur in regularly spaced series, but: (a) Spacing changes with F_0, (b) The series may contain intervening spurious pulses that complicate estimation of the regularity. Specifically, regularity of first-order interpulse intervals is not sufficient and one must consider higher-order intervals. This introduces the well know problem of distinguishing between the period and superperiods. (c) Voiced series may be short (as short as a single pulse!). Phonation often ends with a few pulses that are essentially aperiodic. Aperiodic phonation may also last longer (diplophony, creaky voice). (d) Spurious pulses may appear from time to time to be regular.

The lx estimation algorithm proceeds in several steps, some of which are illustrated in Fig. 4:

1. The lx signal is mildly smoothed by convolution with a square window (0.2 to 0.5 ms).

2. It is negated and differentiated.

3. The result is half-wave rectified, and each peak (positive-going excursion) is replaced by a single pulse, equal to the sum of samples in the peak and placed at its center of gravity.

4. Any pulse occuring within A seconds of another pulse, and weaker than that pulse is eliminated. A is set to $P_{min}/2$, where P_{min} is the minimum expected fundamental period.

5. Any pulse weaker than B times one of its immediate neighbors is removed. B is set to 0.2.

6. The regularity of the position of each pulse is tested by a criterion related to autocorrelation. First, an autocorrelation function is calculated based on the pulse. Then, the average is taken of similar functions calculated based on neighboring pulses. The product of the two is taken and summed. The square of the average function is also taken and summed. If the ratio of the two falls below C, then the pulse is eliminated. C is set to 10.

7. Every pulse belongs to three triplets of consecutive pulses. For every pulse, the regularity of the three triplets is determined. If all three triplets are irregular, the pulse is eliminated (conversely, if a pulse eliminated in step 6 passes the test, it is rehabilitated). A triplet is regular if the intervals it defines differ by less than D. D is set to 10%.

8. Pulse groups are defined as groups of pulses separated by intervals shorter than P_{max}, delimited by intervals greater than P_{max}. The average pulse height in each group is calculated and compared to the maximum average over all groups. If the ratio is below threshold E, the entire group is eliminated. In the calculation of the average, the pulse count in the denominator is increased by F, to give an extra handicap to groups containing few spikes. E is set to 0.05, and F to 10.

9. Voice groups are defined as groups of consecutive spikes, regularly spaced as defined in 7, and starting and ending by a pulse that belongs to only one regular triplet. Speech is defined as voiced within a voiced group, and the fundamental period at each instant is defined as the interval between the previous and following pulse.

Remarks: Step 1 serves to attenuate high frequency noise and reduce the number of spurious pulses produced in steps 2-3. Step 2 emphasizes glottal closure, and step 3 produces a schematic representation indicating the instant and amplitude of each glottal closure pulse. Step 4 removes pulses that cannot possibly represent phonation-related glottis closures within F_0 limits. Step 5 further eliminates spurious pulses, particularly within and at the edge of voice groups. Step 6 applies a regularity criterion based on all-order intervals. This criterion is biased against weak pulses. Step 7 applies another regularity criterion based on first-order intervals. It would ineffective if step 6 had not eliminated most spurious pulses within voice groups. Step 8 gets rid of groups of small pulses, mainly due to high-frequency noise, that occur outside voice groups. They may be regularly spaced by chance or as a result of the elimination of irregular pulses. Amplitude is the surest criterion to get rid of them.

The criteria may eliminate genuine glottal closure pulses if they occur in short groups of weak or irregular pulses, or if they are isolated and don't form a regular spacing with at least two other pulses. Three isolated pulses may survive if they are strong and form regular intervals. An abrupt F_0 step is allowed and won't cause a voicing boundary. However an F_0 "impulse" (due to a single oversized or undersized interval) will trigger a voicing boundary.

Overall, this lx-based F_0 estimation algorithm satisfies all our objectives, in particular objective (2).

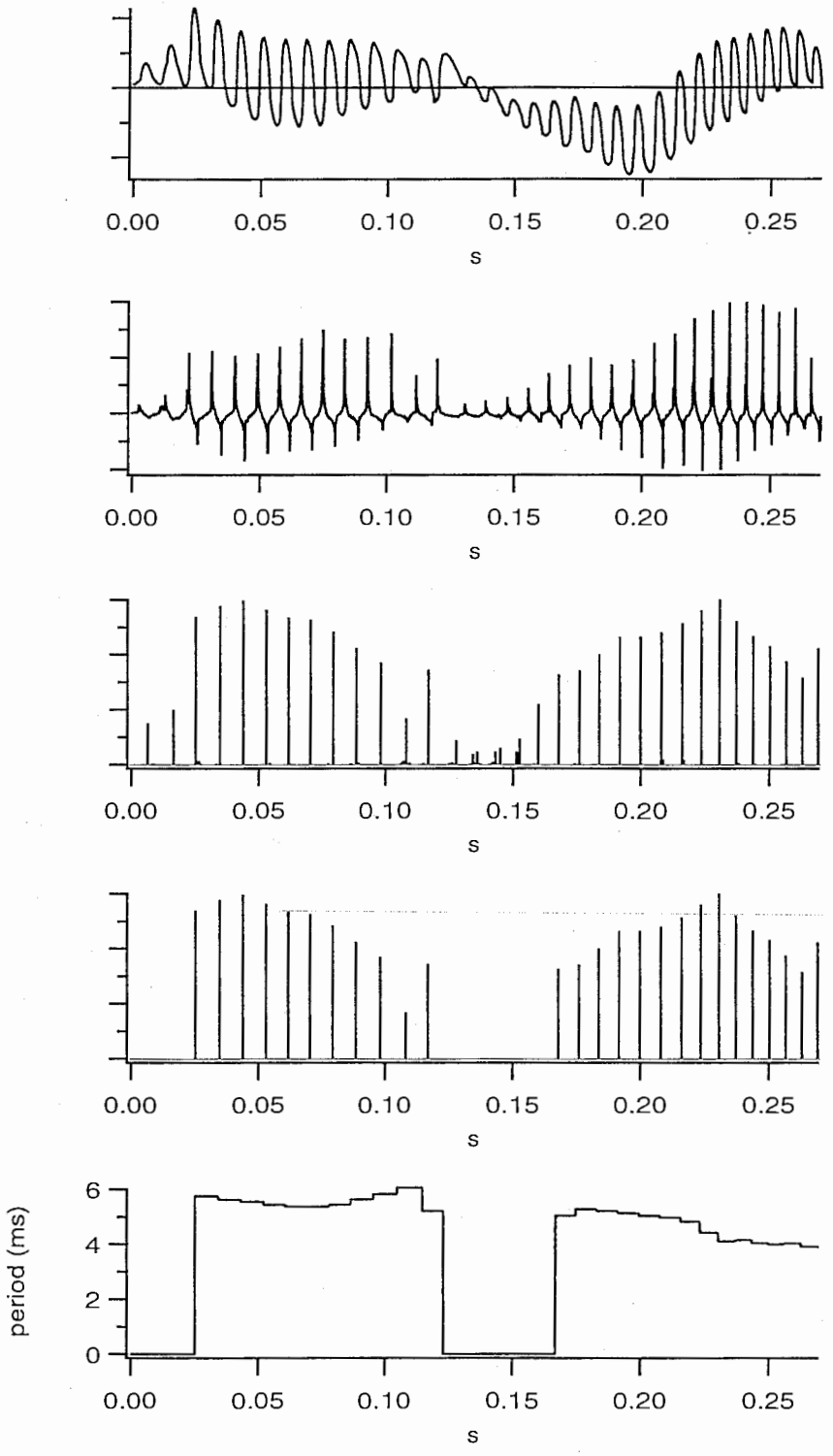

Fig. 4 *Steps in processing the laryngograph signal (lx). From top to bottom: lx, differentiated lx, pulse train representing center-of-gravity of positive excursions, selected pulses, period plot.*

B Error statistics

The histograms in this section summarize the distribution of signal characteristics (level, F0, level change, F0 change, spectral change) over the database, and show how the number of errors made by the baseline waveform F0 estimation algorithm covaries with them. Figures 5 to 9 are for the male speaker NC, figures 10 to 14 are for the female speaker FHS.

Common to both NC and FHS is the tendency for errors to be more common when either F_0 or level is low. This correlation may be due to the fact that phonation is unstable at the end of breath groups, when both level and F_0 fall. Also common is the tendency of errors to occur when there is a large period-to-period change in F_0 (Figs. 8 and 13). Note that "F_0 change" may reflect irregular phonation in addition to genuine frequency sweeps. For both NC and FHS the correlation between error probability and period-to-period spectral change (measured as RMS cepstrum change) is weak. This is somewhat unexpected.

NC and FHS differ on one account: error probability depends on level change for NC (Fig. 7) but not for FHS (Fig. 12).

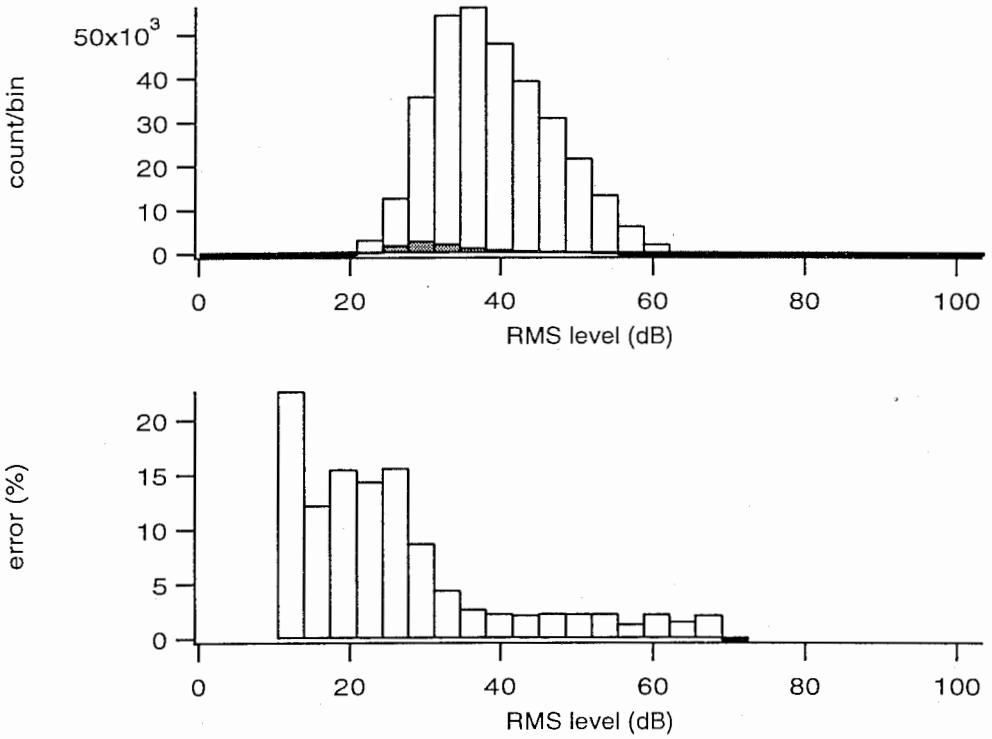

Fig. 5 *Top, open bars: distribution of frames as a function of RMS level; filled bars: distribution of errors. Bottom: error probability. Speaker NC.*

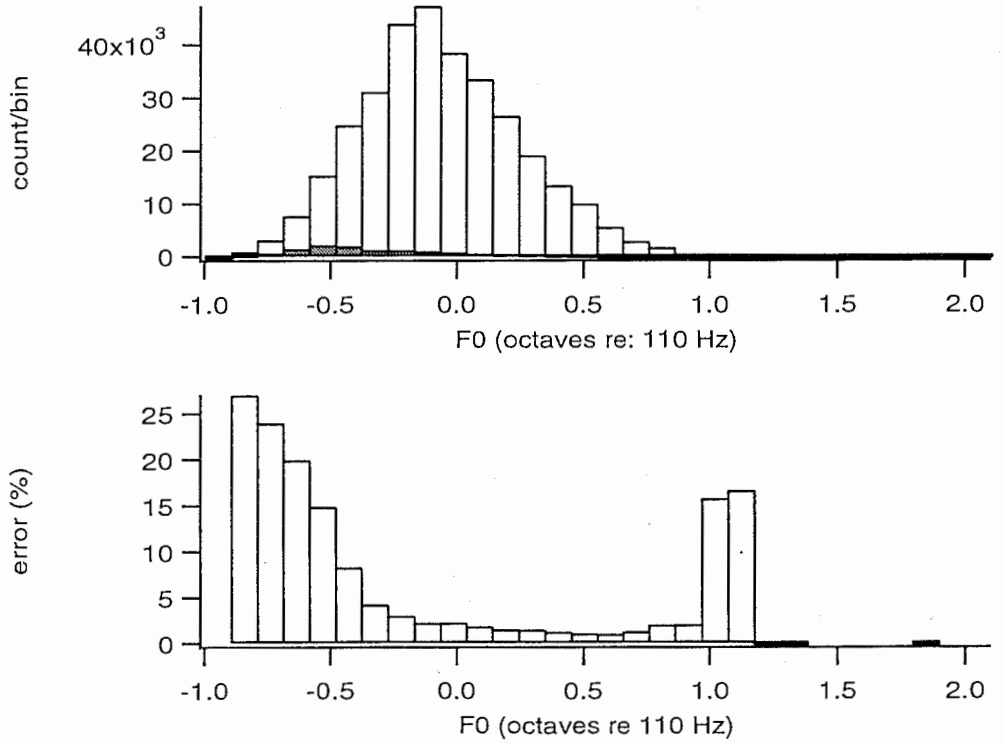

Fig. 6 *Top, open bars: distribution of frames as a function of F_0 value; filled bars: distribution of errors. Bottom: error probability. Speaker NC.*

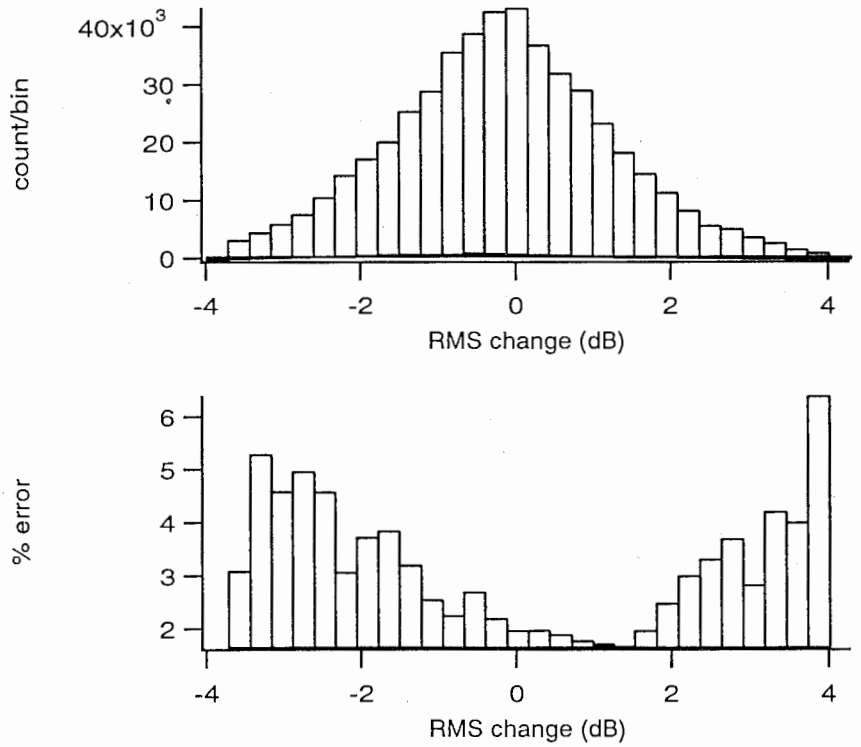

Fig. 7 *Top, open bars: distribution of frames as a function of change in RMS level; filled bars: distribution of errors. Bottom: error probability. Speaker NC.*

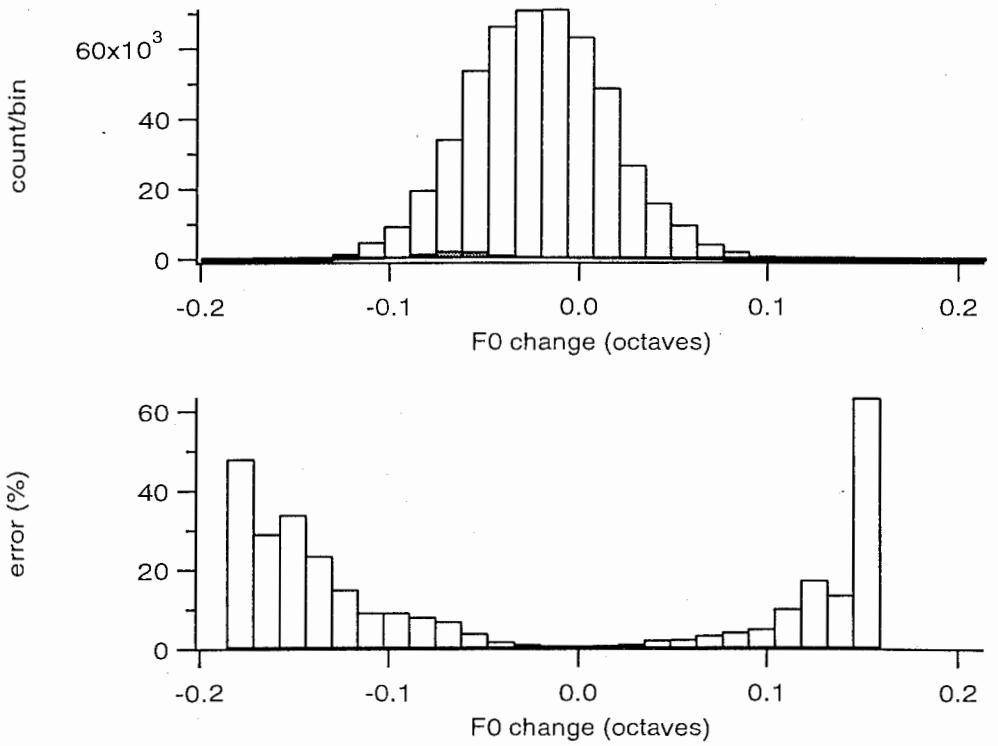

Fig. 8 *Top, open bars: distribution of frames as a function of change F_0; filled bars: distribution of errors. Bottom: error probability. Speaker NC.*

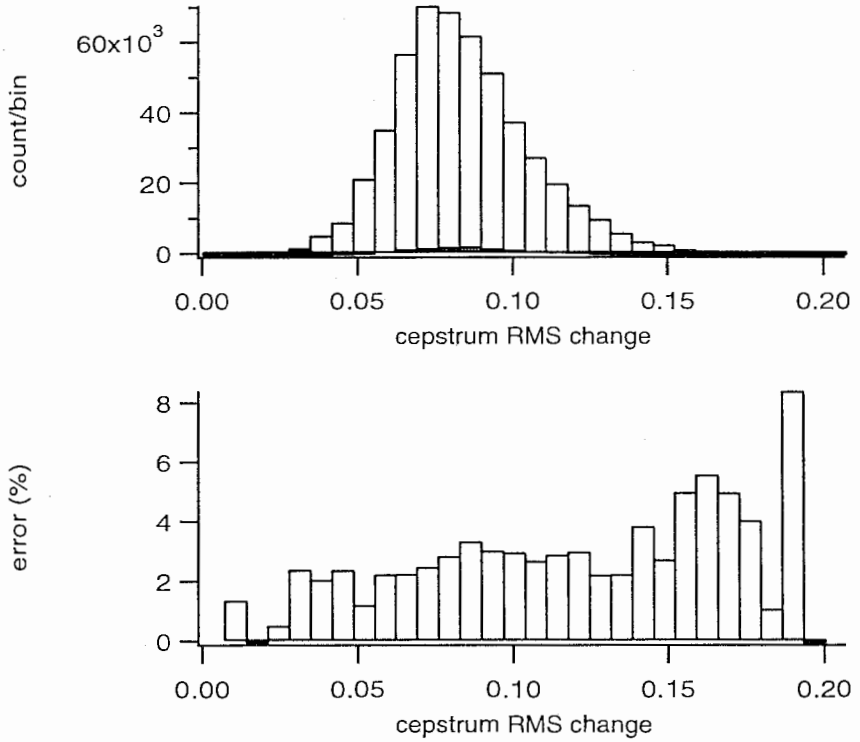

Fig. 9 *Top, open bars: distribution of frames as a function of period-to-period spectral change (RMS cepstrum distance); filled bars: distribution of errors. Bottom: error probability. Speaker NC.*

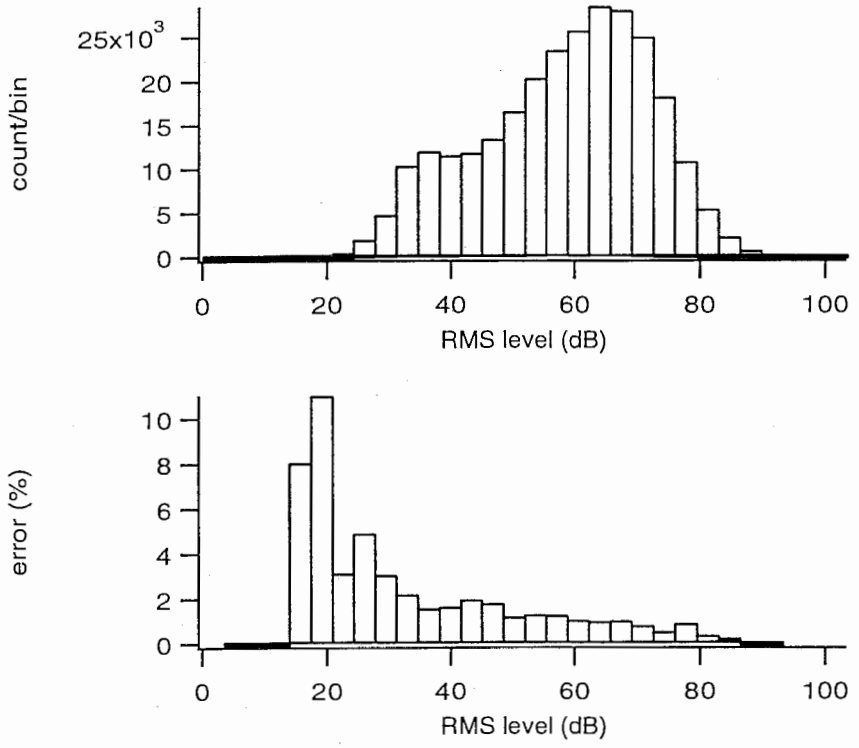

Fig. 10 *Top: distribution of frames as a function of level. Bottom: error probability. Speaker FHS.*

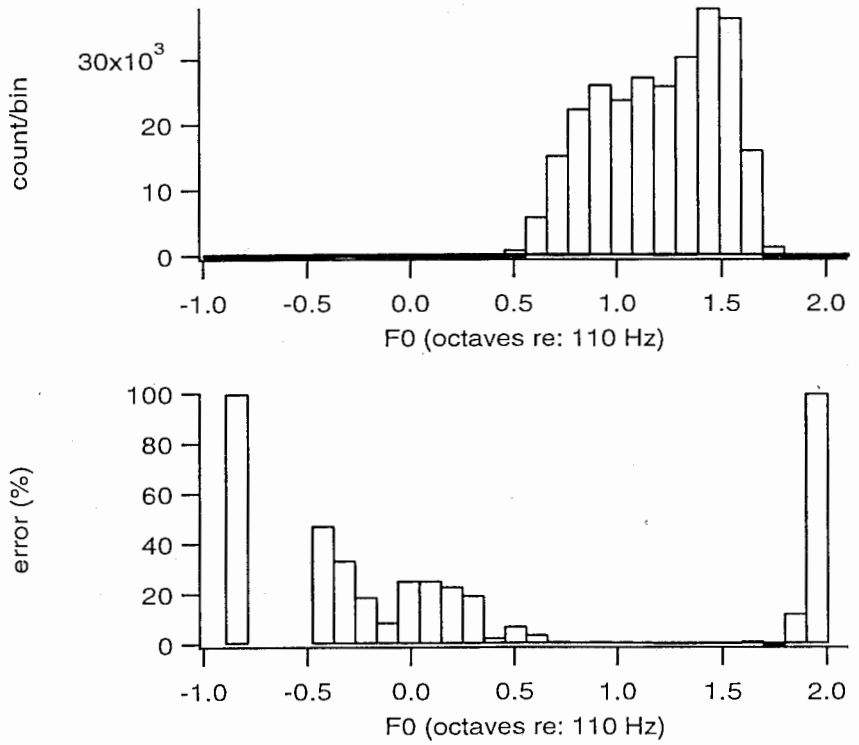

Fig. 11 *Top: distribution of frames as a function of F_0. Bottom: error probability. Speaker FHS.*

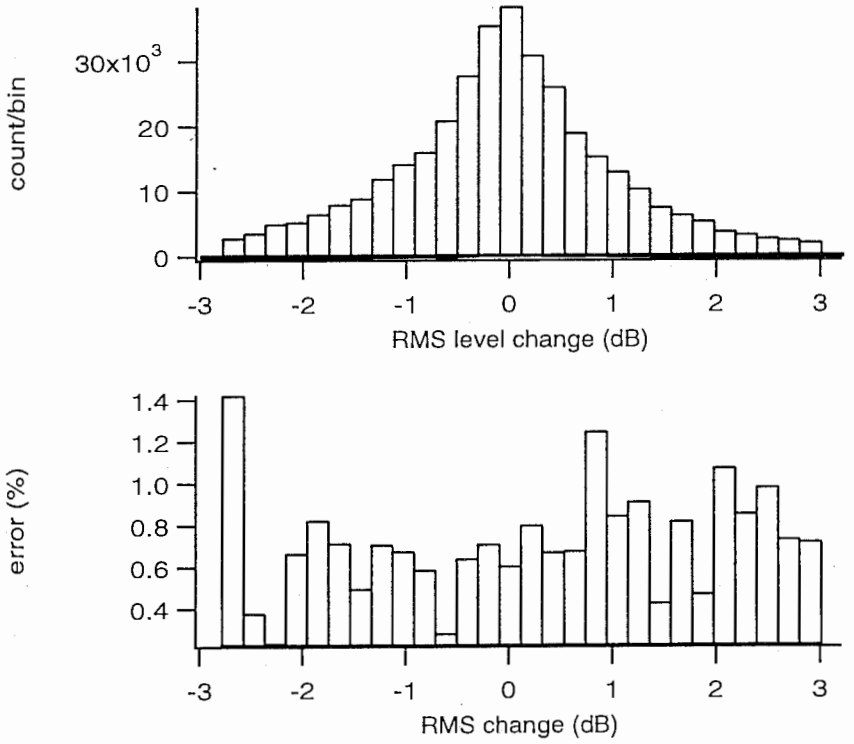

Fig. 12 *Top: distribution of frames as a function of period-to-period level change. Bottom: error probability. Speaker FHS.*

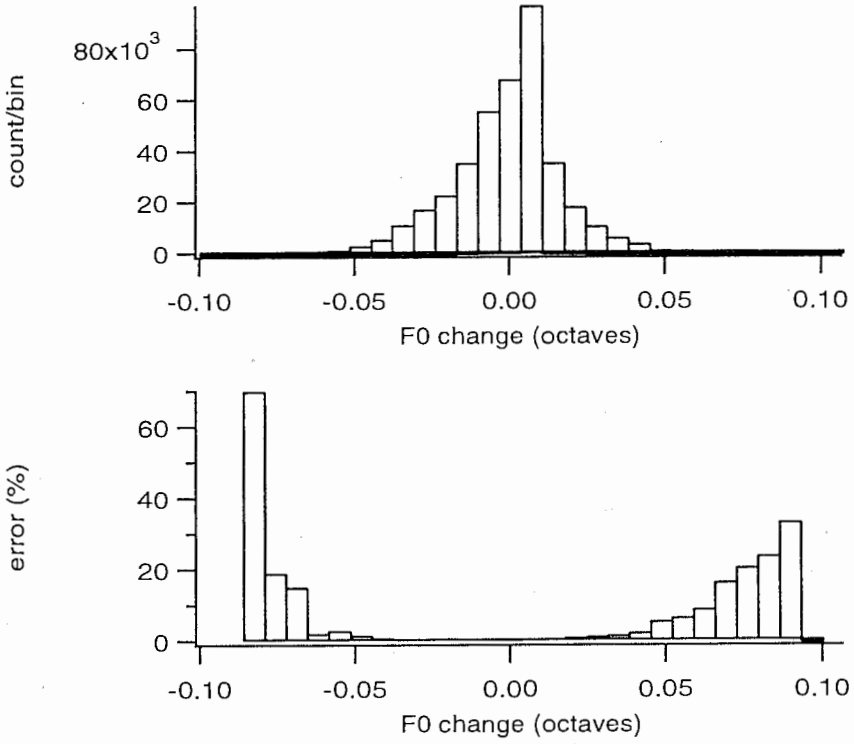

Fig. 13 *Top: distribution of frames as a function of period-to-period F_0 change. Bottom: error probability. Speaker FHS.*

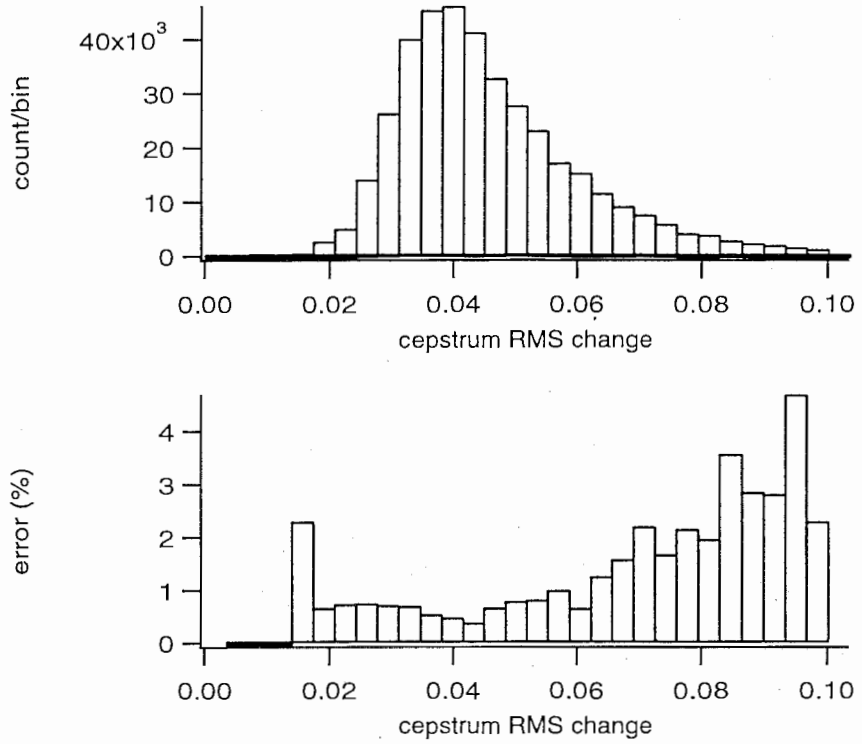

Fig. 14 *Top: distribution of frames as a function of period-to-period spectral change (RMS cepstrum difference). Bottom: error probability. Speaker FHS.*

C Speed

Both AMDF and DDF involve exhaustive search and are therefore time consuming. If W is the size of the integration window in samples, L is the maximum lag range, and R is the frame rate in Hz, AMDF requires a number of operations on the order of W x L x R, while DDF requires operations on the order of W x L x L x R.

This section reviews several schemes that allow calculation time to be considerably reduced.

C.1 Running calculation

If the frame period in samples ($F_p = F_s/R$) is less than the window size, then some calculations are performed several times on the same samples. For example, if the window size is 320 samples, and the frame period 16 samples, then the same calculations are repeated 20 times. An obvious solution is to store the results. At each analysis index i, a partial AMDF calculation is performed:

$$a_i(\tau) = \sum_{k=i}^{i+Fp} |s_k - s_{k-\tau}|$$

This partial AMDF is stored in a list of partial AMDFs, long enough to span the size of one window. The AMDF may then be calculated as:

$$A_i(\tau) = \sum_{j=i}^{i+W/Fp} a_{i-j\tau}$$

eventually with some form of window weighting. If no weighting is used (square window), a running AMDF may be maintained and updated:

$$A_i(\tau) = A_{i-1} + a_i\tau - a_{i-W/Fp}\tau$$

This scheme reduces calculations by a factor of W/F_p, in our case 16. We applied it systematically. Computation time no longer depends on the frame rate, so there is no reason not to choose the rate most adequate for the application.

The scheme requires extra space to store the list of partial AMDFs. To avoid this requirement one may perform a running integration with exponential decay [22, 31]. This is equivalent to applying an exponentially-shaped window with a tail towards negative time.

D Sparse calculation

It is usually beneficial to apply a mild degree of low-pass filtering to the waveform. In that case the waveform varies little over a span of a few samples, and one may perform the AMDF calculation using a sparse summation:

$$A_i(\tau) = \sum_{j=0}^{W/n} |s_{i+jn} - s_{i+jn-\tau}|$$

We applied a 16-point smoothing to the waveform and performed calculations at intervals of 8 samples, resulting in a speed-up factor close to 8. Sparse calculation increases somewhat the error rates. Note that sparse calculation is not equivalent to down-sampling the waveform: calculations are still performed at sample intervals along the lag dimension.

E Sum-to-criterion

Both AMDF and DDF search for a global minimum. If an upper bound on the minimum value is known, the summation (over the window) at each search point may be interrupted as soon as that bound is exceeded. For example if it is known that the "depth" of the minimum is less than 10% of the average AMDF value, calculations may be reduced by a factor close to 10 (depending on the statistics of the terms of the summation). The upper bound is updated each time a calculation produces a smaller value. The scheme is most effective if the starting value of the search parameter produces an AMDF value close to the minimum. The estimate of the previous frame is a good starting point. With this scheme, computation time depends on the waveform: computation is faster if the waveform is nearly periodic.

This scheme was not implemented, as it is difficult to concile with some of the other schemes. It might be of use in a final implementation.

F Search-to-criterion

The mean-normalized AMDF scheme proposed earlier chose the first minimum that fell under a certain threshold. Calculations need not continue after that minimum is found. This saves computation in a proportion that depends on the degree of periodicity (a below- threshold minimum must exist) and the F_0 (the minimum is found faster for a short period).

This scheme was not implemented.

G Multi-level search

There are various ways in which computation may be saved by splitting the search into levels of increasing resolution. For example the DDF algorithm may be performed iteratively, by searching one lag dimension at a time (in other words, by performing AMDF on the waveform filtered by a comb-filter tuned to the previous period estimate).

Multi-level search explores a smaller proportion of the search space than exhaustive search, and may therefore miss a global minimum. For this reason, and because of the extra complexity involved, multi-level search was not used.

References

[1] Abe, T., T. Kobayashi, et al. (1995). Harmonics tracking and pitch extraction based on instantaneous frequency. IEEE-ICASSP: 756-759.

[2] Bagshaw, P. C. (1993). "An investigation of acoustic events related to sentential stress and pitch accents, in English." Speech Comm. 13: 333-342.

[3] Bagshaw, P. C., S. M. Hiller, et al. (1993). Enhanced pitch tracking and the processing of F_0 contours for computer and intonation teaching. European Conf. on Speech Comm. (Eurospeech): 1003-1006.

[4] Barbe, T. and M. T. Janot-Giorgetti (1989). "Evaluation d'un détecteur de F_0 sur des voix normales et sur des voix pathologiques." Bulletin du laboratoire de la communication parlée (Grenoble, France) 3: 231-243.

[5] Barnard, E., R. A. Cole, et al. (1991). "Pitch detection with a neural-net classifier." IEEE Trans. Sig. Proc. 39: 298-307.

[6] Chazam, D., Y. Stettiner, et al. (1993). Optimal multi-pitch estimation using the EM algorithm for co-channel speech separation. ICASSP II: 728-731.

[7] Chilton, E. and B. G. Evans (1988). The spectral autocorrelation applied to the linear prediction residual of speech for robust pitch detection. IEEE-ICASSP: 358-361.

[8] Cohen, J. R. (1989). "Application of an auditory model to speech recognition." JASA 85: 2623-2629.

[9] Cohen, M. A., S. Grossberg, et al. (1995). "A spectral network model of pitch perception." JASA 98: 862-879.

[10] de Cheveigne, A. (1990). Experiments in pitch extraction, ATR Interpreting Telephony Res. Labs technical report: 39p.

[11] de Cheveigne, A. (1991). A mixed speech F_0 estimation algorithm. ESCA (Eurospeech)-91., Genova: 445-448.

[12] de Cheveigne, A. (1991). Speech F_0 extraction based on Licklider's pitch perception model. ICPhS 4: 218-221.

[13] de Cheveigné, A. (1993). "Separation of concurrent harmonic sounds: Fundamental frequency estimation and a time-domain cancellation model of auditory processing," J. Acoust. Soc. Am., 93, 3271-3290.

[14] de Cheveigné, A., McAdams, S., Laroche, J. and Rosenberg, M. (1995). "Identification of concurrent harmonic and inharmonic vowels: A test of the theory of harmonic cancellation and enhancement," J. Acoust. Soc. Am., 97, 3736-3748.

[15] de Cheveigné, A. (1996). "Concurrent vowel segregation III: a neural time-domain model of harmonic interference cancellation," J. Acoust. Soc. Am., in preparation.

[16] Denbigh, P. N. and J. Zhao (1992). "Pitch extraction and separation of overlapping speech." Speech Comm. 11: 119-125.

[17] Dologlou, I. and G. Carayannis (1990). "Pitch detection based on zero-phase filtering." Speech Comm. 8: 309-318.

[18] Doval, B. and X. Rodet (1991). Estimation of fundamental frequency of musical sound signals, IEEE-ICASSP

[19] Doval, B. (1994), "Estimation de la fréquence fondamentale des signaux sonores," Université Pierre et Marie Curie, unpublished doctoral dissertation.

[20] Duifhuis, H., L. F. Willems, et al. (1982). "Measurement of pitch in speech: an implementation of Goldstein's theory of pitch perception." JASA(71): 1568-1580.

[21] Fujimura, O. (1968). "An approximation to voice aperiodicity." IEEE Trans. Audio and Electroacoustics 16: 68-72.

[22] Fujisaki, H., K. Hirose, et al. (1989). "A method for pitch extraction of speech with reduced errors due to analysis frame positioning." IECE Transactions SP-89(319) 1-8 (in Japanese).

[23] Fujisaki, H., K. Hirose, et al. (1987). A new system for reliable pitch extraction of speech. IEEE ICASSP: 2422-2425.

[24] Geoffrois, E. (1995). Extraction robuste de paramètres prosodiques pour la reconnaissance de la parole, Orsay - Paris 13.

[25] Gu, Y. H. and W. M. G. van Bokhoven (1991). Cochannel speech separation using frequency bin non-linear adaptive filtering. IEEE-ICASSP: 949-952.

[26] Harris, J. D. and D. Nelson (1993). Glottal pulse alignment in voiced speech for pitch determination. IEEE ICASSP II: 519-522.

[27] Hedelin, P. and D. Huber (1990). Pitch period determination of aperiodic speech signals. IEEE-ICASSP: 361-364.

[28] Hermes, D. J. (1988). "Measurement of pitch by subharmonic summation." JASA 83: 257-264.

[29] Hess, W. (1983). Pitch determination of speech signals. Berlin, Springer-Verlag.

[30] Hess, W. J. (1992). Pitch and voicing determination. Advances in speech signal processing. Sadaoka Furui and M. M. Sohndi. New York, Marcel Dekker: 3-48.

[31] Hirose, K., H. Fujisaki, et al. (1992). A scheme for pitch extraction of speech using autocorrelation function with frame length proportional to the time lag. IEEE-ICASSP I: 149-152.

[32] Howard, I. (1991), "Speech fundamental period estimation using pattern classification," University of London, unpublished doctoral dissertation.

[33] Kroon, P. and B. S. Atal (1990). Pitch predictors with high temporal resolution. ICASSP-90: 661-664.

[34] Krubsack, D. A. and R. J. Niederjohn (1991). "An autocorrelation pitch detector and voicing decision with confidence measures developed for noise-corrupted speech." IEEE Trans. Sig. Proc. 39: 319-329.

[35] Medan, Y., Yair, E., and Chazan, D. (1991). "Super resolution pitch determination of speech signals," IEEE ASSP 39,40-48.

[36] Meyer, G. F. and W. A. Ainsworth (1993). Vowel pitch period extraction by models of neurones in the mammalian brain-stem. Eurospeech: 2029-2032.

[37] Montrésor, S. and M. Baudry (1993). Pitch estimation of speech signal with the wavelet transform. European Conf. on Speech Comm. and Technology (Eurospeech): 2017-2020.

[38] Moreno, A. and J. A. R. Fonollosa (1992). Pitch determination of noisy speech using higher order statistics. IEEE-ICASSP I: 133-136.

[39] Neuburg, E. P. (1988). On estimating rate of change of pitch. IEEE-ICASSP: 355-357.

[40] Ney, H. (1982). "A time warping approach to fundamental period estimation." IEEE Trans. SMC 12: 383-388.

[41] Nguyen, T. D., J. B. Ferguson, et al. (1988). A geometric approach to real time pitch detection. IEEE-ICASSP: 362-365.

[42] Ohmura, H. (1994). Fine pitch contour extraction by voice fundamental wave filtering method. IEEE-ICASSP II: 189-192.

[43] Rabiner, L. R., M. J. Cheng, et al. (1976). "A comparative performance study of several pitch detection algorithms." IEEE Trans. ASSP 24: 399-418.

[44] Ramalho, M. A., and Mammone, R. J. (1994). "A new speech enhancement technique with application to speaker identification.", Proc. IEEE-ICASSP, 29-32.

[45] Rheem, J. Y., M. Bae, et al. (1993). A spectral amdf method for pitch extraction of noise-corrupted speech. European Conf. on Speech Comm. and Technology (Eurospeech): 2021-2024.

[46] Ross, M. J., H. L. Shaffer, et al. (1974). "Average magnitude difference function pitch extractor." IEEE Trans. ASSP 22: 353-362.

[47] Yang, G. and H. Leich (1993). A reliable postprocessor for pitch determination algorithms. Eurospeech: 2025-2028.