Feature extraction based on pitch-synchronous averaging for robust speech recognition

Juan A. Morales-Cordovilla, Antonio M. Peinado, *Senior Member, IEEE,* Victoria Sánchez, *Member, IEEE,* and José A. González

***Abstract*—In this paper we propose two estimators for the autocorrelation sequence of a periodic signal in additive noise. Both estimators are formulated employing tables which contain all the possible products of sample pairs in a speech signal frame. The first estimator is based on a pitch-synchronous averaging. This estimator is statistically analyzed and we show that the SNR can be increased up to a factor equal to the number of available periods. The second estimator is similar to the former one but it avoids the use of those sample products more likely affected by noise. We prove that, under certain conditions, this estimator can remove the effect of an additive noise in a statistical sense. Both estimators are employed to extract mel frequency cepstral coefficients (MFCC) as features for robust speech recognition. Although these estimators are initially conceived for voiced speech frames, we extend their application to unvoiced sounds in order to obtain a coherent feature extractor. The experimental results show the superiority of the proposed approach over other MFCC-based front-ends such as the higher-lag autocorrelation spectrum estimation (HASE), which also employs the idea of avoiding those autocorrelation coefficients more likely affected by noise.**

***Index Terms*—Robust speech recognition, autocorrelation estimation, mel frequency cepstral coefficients, pitch-synchronous analysis, acoustic noise, AMFCC.**

I. Introduction

It is well-known that the performance of an automatic speech recognition (ASR) system diminishes when a mismatch between training and testing conditions appears. One of the main causes of performance degradation is the additive noise that may appear in many practical situations. There are a large number of different solutions to alleviate this problem [1]. We can identify two main classes of techniques for noise-robust ASR. First, we have those techniques which try to compensate in some way the mismatch due to noise. These are the compensation techniques which can operate either over the recognition features or over the acoustic models. The compensation is carried out by using some type of knowledge about the noise. In the second class we have those robust feature extraction techniques which do not require any explicit knowledge about the noise. Examples of this type of techniques can be found in [2]–[8]. Both classes of techniques are not exclusive and could be jointly applied [9].

In this work, we are interested in the second class. In particular, we will focus on those signal analysis techniques which extract the recognition features from an estimate of the signal autocorrelation. The autocorrelation domain has several advantages when signals are corrupted by additive noise. These advantages are based on "reasonable" assumptions: a) the noise is not correlated with the speech signal, and b) the noise is random and white enough. The first assumption means that the additive condition is maintained since the autocorrelation of the sum of two uncorrelated signals is the sum of their autocorrelations. The second assumption involves that the noise autocorrelation only has significant values for the lower lags.

The first techniques which employed this alternative domain for speech recognition used the one-sided autocorrelation (OSA), that is, the causal part of the autocorrelation, in order to obtain linear prediction cepstral coefficients (LPCC) [10]. The OSA sequence is employed since it provides spectral estimators less sensitive to broadband noise [11]. A first example of this type of analysis is the short-time modified coherence (SMC) method [10], which owes its name to the specific autocorrelation estimate employed. This estimate is used to build a set of normal equations which are solved in order to obtain the corresponding LPC coefficients and, then, the LPCC features. Hernando *et al* [11] propose an OSA-LPC technique which considers that the OSA sequence can be modeled as an AR process with the same poles as the original signal. Then, its autocorrelation is employed to obtain LPCC features (after an LPC analysis). A more recent example, which outperforms the former ones, is the HASE (higher-lag autocorrelation spectrum estimation) technique [12]. It also employs the OSA sequence, but in this case it is simply FFT-transformed in order to obtain a spectral estimate from which MFCC coefficients can be extracted. All these methods reduce the contribution of the lower-lag autocorrelation coefficients since these are the ones more affected by the additive noise, according to our second assumption. While SMC and OSA-LPC do this through windowing, the HASE method explicitly fixes them to zero up to a certain lag and applies a window to the remaining lags.

In this paper we propose a feature extraction methodology which also provides autocorrelation-based MFCC (AMFCC) coefficients. However, instead of employing the OSA sequence, we will propose two novel estimators for the whole autocorrelation (causal and non-causal parts). These estimators are initially developed for voiced speech frames, although they are later extended to unvoiced sounds, and make an explicit use of the signal pitch in order to increase robustness against noise. The idea of reducing the effect of additive noise on cyclostationary signals by employing the pitch has been previously studied in [6], [13], [7], [8] or [14]. The first estimator proposed in this work is derived from an averaging of the

IEEE Trans. 2011 v.. The authors are with the Department of Teoría de la Señal Telemática y Comunicaciones, Universidad de Granada, 18071-Granada, Spain, e-mail: {jamc,amp,victoria,joseangl}@ugr.es. Tel.: +34-958240031. Fax: +34-958242740.

different pitch periods included in a speech frame. This kind of pitch-synchronous analysis has been successfully applied to speech enhancement [13]. The second estimator is based on the former one, although it incorporates the aforementioned idea that the noise mainly affects the lower-lag autocorrelation coefficients. However, instead of eliminating them as HASE, it can still estimate them through a selection process that we will refer to as *sifting*. The sifting process eliminates from the estimator those sample products more likely affected by the noise, but it exploits the signal periodicity to provide estimates of the corresponding autocorrelation coefficients.

As mentioned before, the proposed techniques involve an explicit knowledge of the pitch period. Although this fact means that a pitch extractor is required, we must take into account that there exist a number of simple pitch extraction methods and that some standardized front-ends already include pitch extraction [16], [18]. In addition, it must be pointed out that our proposals are not exclusive, and could be combined with active noise reduction techniques such as Wiener filtering or spectral subtraction.

The paper is organized as follows. The next section is devoted to AMFCC features. In sections III and IV we develop the proposed autocorrelation estimators and include the corresponding statistical interpretations. Section V explains the implementation details of the feature extraction and section VI gives details of the experimental setup, and shows our results. The paper concludes with a summary and some future extensions of this work.

II. Autocorrelation-based Mel Frequency Cepstral Analysis

In our work we have taken the ETSI front-end (FE) [15] as starting point for MFCC feature extraction. In order to obtain a set of MFCC features, the following procedure is commonly applied: estimation of the spectrum by means of a Fourier transformation of the windowed signal, application of a triangular Mel filterbank to the estimated spectrum, and, finally, application of the DCT to the log-outputs of the filterbank. The spectrum estimate can vary from one particular front-end to another. Thus, FE employs an amplitude spectrum obtained from the modulus of the FFT transform (applied to the Hamming-windowed signal), while the ETSI advanced front-end (AFE) [17] uses the square of that modulus, which corresponds to a power spectral estimate (the periodogram of the windowed signal).

AMFCC feature extraction also follows the procedure described above except for the spectrum estimate, which is obtained from the autocorrelation instead of directly from the signal. Then, the first step in an AMFCC front-end is the estimation of the signal autocorrelation. We can obtain different AMFCC features by applying different estimators. The most common estimator of the autocorrelation of a signal frame $x(n)$ $(n = 0, \ldots, N-1)$ is the biased one, which is defined as,

$$\hat{r}_x(k) = \frac{1}{N} \sum_{n=k}^{N-1} x(n)x(n-k) \quad (0 \leq k < N) \tag{1}$$

This definition is completed for negative lags with $\hat{r}_x(k) = \hat{r}_x(-k)$ $(k = -1, \ldots, -(N-1))$. For the sake of simplicity, we will neglect the negative part of the autocorrelation, just keeping in mind that $\hat{r}_x(k)$ is an even function. It is easily derived that,

$$E[\hat{r}_x(k)] = w_B^N(k) r_x(k) \tag{2}$$

where $w_B^N(k) = 1 - |k|/N$ $(|k| < N)$ is the Barlett window of radius N. Since the Fourier transform of the biased autocorrelation results in the periodogram, then the corresponding AMFCC features are equivalent to those provided by the AFE front-end.

The HASE method also uses the biased autocorrelation estimator, but, as previously mentioned, it does not obtain the spectral estimate from the whole autocorrelation but from the OSA sequence. Also, it explicitly fixes to zero the first coefficients of OSA, that is, those more affected by additive noise, and it applies a double dynamic range (DDR) Hamming window to the remaining higher-lags [12]. This window provides a dynamic range of 86 dB, suitable for speech power spectra. Finally, the Fourier transform modulus of the resulting autocorrelation yields the required spectral estimate.

III. Autocorrelation estimation by averaging

A. Pitch-synchronous signal averaging

Let $x(n)$ $(n = 0, \ldots, N-1)$ be a noisy signal corresponding to a voiced speech frame with a pitch period T. We will assume that $x(n)$ is the superposition of a periodic signal $p(n)$ and a distortion signal $d(n)$,

$$x(n) = p(n) + d(n) \quad (n = 0, \ldots, N-1) \tag{3}$$

The distortion signal accounts for non-periodic components and, mainly, additive acoustic noise. If we neglect the non-periodic components, then the original clean signal mainly corresponds to the periodic signal $p(n)$. Our goal will be the estimation of this *clean* signal $p(n)$. This can be accomplished by averaging the different pitch periods [13],

$$\begin{aligned} z(n) &= \frac{1}{N_p(\underline{n})} \sum_{i=0}^{N_p(\underline{n})-1} x(iT + \underline{n}) \\ &= p(n) + \frac{1}{N_p(\underline{n})} \sum_{i=0}^{N_p(\underline{n})-1} d(iT + \underline{n}) \end{aligned} \tag{4}$$

where $\underline{n}$ $(\underline{n} = 0, 1, \ldots, T-1)$ is the remainder of n/T, and $N_p(\underline{n})$ is the number of available periods for $\underline{n}$, that is, the number of samples in $x(n)$ placed at position $\underline{n}$. Note that, for every $\underline{n}$, the number of available periods $N_p(\underline{n})$ can differ. The results presented in this work take into account this fact. However, for the sake of simplicity, we will assume in the subsequent theoretical developments that $N_p(\underline{n}) = N_p$ is a constant and, therefore, $N = TN_p$. That is, we consider that the signal has an integer number of periods.

Assuming that we have a sufficient number of periods and that $d(n)$ is a zero-mean signal (without periodicity of period T), we see that the averaged signal $z(n)$ is a periodic signal and an estimate of the unknown clean periodic signal $p(n)$, that is, $z(n) \approx p(n)$ $(n = 0, \ldots, N-1)$. Therefore,

the autocorrelation obtained for $z(n)$ is also an estimate of the unknown autocorrelation $r_p(k)$. If we estimate the autocorrelation of $z(n)$ with the biased estimator of equation (1), the following is obtained,

$$\hat{r}_z(k) = \bar{r}_x(k) = \frac{1}{N}\sum_{n=k}^{N-1} z(n)z(n-k) \quad (0 \le k < N) \quad (5)$$

The notation $\hat{r}_z(k) = \bar{r}_x(k)$ is introduced to make it clear that this estimator is implicitly employing samples of the noisy signal $x(n)$. We will refer to (5) as averaging estimator.

Figure 1 shows three periodograms of a voiced segment corresponding to the clean signal, the signal contaminated with an additive 0-dB white noise, and the averaged signal obtained from the contaminated one. It is clear that the pitch-based averaging processing diminishes the effect of the noise. We can also observe that the averaging process tends to preserve the spectrum at the pitch harmonics, just where the SNR of the contaminated signal is higher, and reduces the noise at intermediate frequencies. In fact, the averaging process increases the SNR by a factor N_p if the noise fulfills certain conditions [13]. This is analyzed later in subsection III-C.

B. Table-based formulation of the autocorrelation estimators

We will introduce a new notation which allows us to express the averaging estimator in terms of symmetric tables. This new framework will facilitate the statistical interpretation of the different estimators in the next subsection. Also, it will be used in the next section to obtain an improved estimate of the clean signal autocorrelation.

Let us define the symmetric table of sample products as,

$$\pi_x(n,m) = x(n)x(m) \quad (n,m = 0,\ldots,N-1) \quad (6)$$

Then, we can consider that the biased autocorrelation estimate $\hat{r}_x(k)$ corresponds to the sum of the elements contained in the kth diagonal of the table,

$$\hat{r}_x(k) = \frac{1}{N}\sum_{n=k}^{N-1} \pi_x(n,n-k) \quad (k = 0,\ldots,N-1) \quad (7)$$

This notation also allows us to compute the averaging autocorrelation estimate $\bar{r}_x(k)$ (equation (5)) as,

$$\begin{aligned}
\bar{r}_x(k) &= \frac{1}{N}\sum_{n=k}^{N-1} \pi_z(n,n-k) \\
&= \frac{1}{N}\sum_{n=k}^{N-1} \frac{1}{N_p^2}\sum_{i=0}^{N_p-1}\sum_{j=0}^{N_p-1} x(iT+\underline{n})x(jT+\underline{n-k}) \\
&= \frac{1}{N}\sum_{n=k}^{N-1} \frac{1}{N_p^2}\sum_{i=0}^{N_p-1}\sum_{j=0}^{N_p-1} \pi(iT+\underline{n}, jT+\underline{n-k}) \\
&= \frac{1}{N}\sum_{n=k}^{N-1} \bar{\pi}_x(n,n-k) \quad (k = 0,\ldots,N-1) \quad (8)
\end{aligned}$$

where we have defined,

$$\bar{\pi}_x(n,m) = \frac{1}{N_p^2}\sum_{i=0}^{N_p-1}\sum_{j=0}^{N_p-1} \pi_x(iT+\underline{n}, jT+\underline{m}) \quad (n,m = 0,1,\ldots,N-1) \quad (9)$$

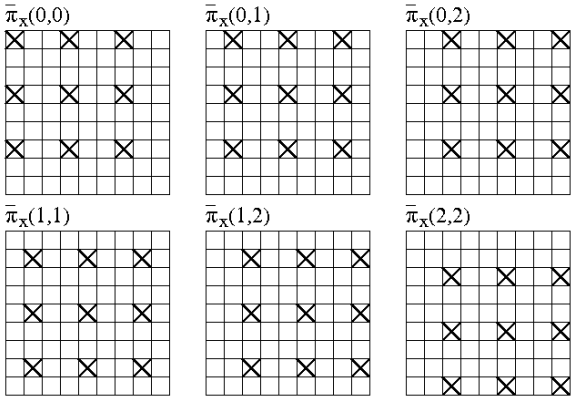


Fig. 2: Computation of elements $\bar{\pi}_x(0,0)$, $\bar{\pi}_x(0,1)$, $\bar{\pi}_x(0,2)$, $\bar{\pi}_x(1,1)$, $\bar{\pi}_x(1,2)$ and $\bar{\pi}_x(2,2)$ from elements of table $\pi_x(n,m)$ for a signal with pitch $T = 3$ and $N_p = 3$ periods.

Table $\bar{\pi}_x(n,m)$ is symmetric and periodic with period (T,T), so that $\bar{\pi}_x(n,m) = \bar{\pi}_x(\underline{n},\underline{m})$ for all (n,m). We can observe that equation (8) has the same form as the biased estimator of equation (7), although substituting table $\pi_x(n,m)$ by $\bar{\pi}_x(n,m)$. That is, every autocorrelation coefficient $\bar{r}_x(k)$ can be obtained from this new table by summing the elements of its k-th diagonal.

The elements of one period of table $\bar{\pi}_x(n,m)$ (that is, $n,m = 0,1,\ldots,T-1$) are obtained from the original product table $\pi_x(n,m)$ by averaging the elements of a periodic grid shifted by (n,m). This is illustrated in the example of figure 2. This example considers the case of a signal with pitch $T = 3$ and $N_p = 3$ periods. Then, table $\bar{\pi}_x(n,m)$ has period $(3,3)$ and, taking into account its symmetry, contains only 6 different elements ($\bar{\pi}_x(0,0)$, $\bar{\pi}_x(0,1)$, $\bar{\pi}_x(0,2)$, $\bar{\pi}_x(1,1)$, $\bar{\pi}_x(1,2)$ and $\bar{\pi}_x(2,2)$). The elements of the original table $\pi_x(n,m)$ required for the computation of the six different elements in $\bar{\pi}_x(n,m)$ are marked with "x" in the figure.

C. Statistical interpretation of the averaging estimator

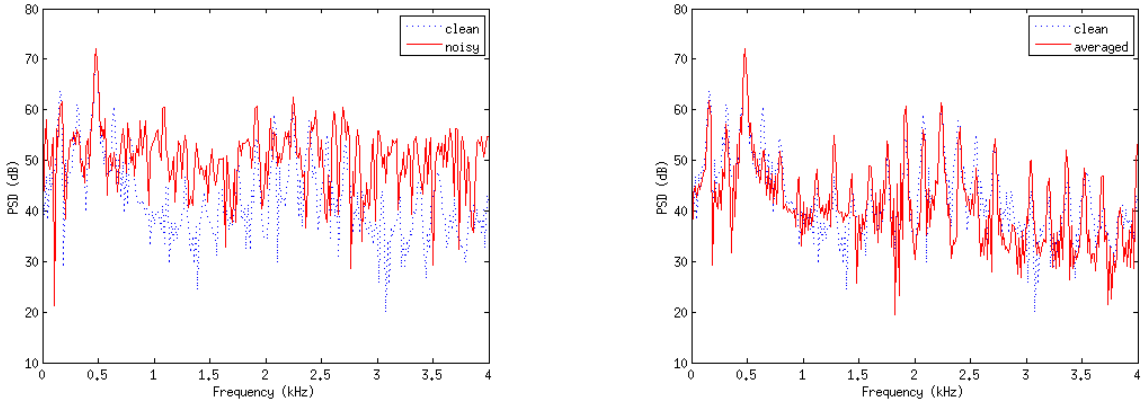
We will consider now that our input signal and the distortion signal correspond to two random processes, $x(n)$ and $d(n)$, respectively. Then, we can express the expected values of the biased and averaged autocorrelation estimators in terms of the autocorrelations $r_p(k)$ and $r_d(k)$. For the biased estimator (equation (1)), it is directly derived from (2) that,

$$E[\hat{r}_x(k)] = w_B^N(k)\,(r_p(k) + r_d(k)) \quad (10)$$

For the averaging estimator, we obtain from equation (8) that,

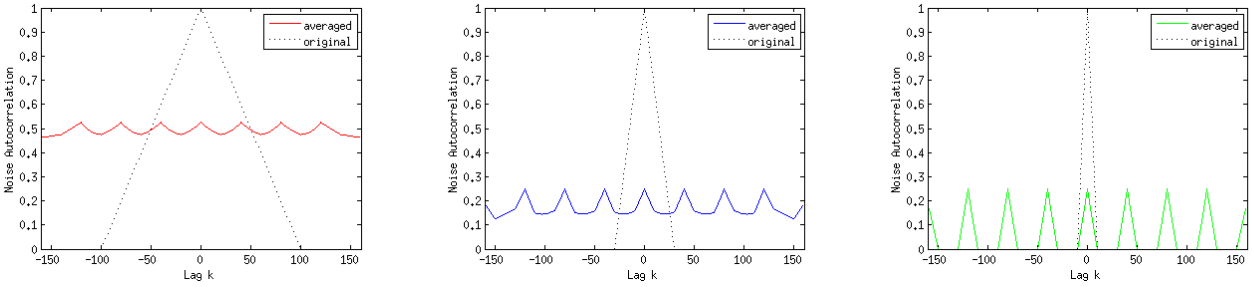
$$E[\bar{r}_x(k)] = \frac{w_B^N(k)}{N-k}\sum_{n=k}^{N-1} E[\bar{\pi}_x(n,n-k)] \quad (11)$$

The development of this expression is carried out in appendix A, and the solution is given by equation (29) and the auxiliary functions defined in equations (20), (24) and (26). Comparing equations (10) and (29), we can see that both estimators differ in the treatment of the distortion autocorrelation $r_d(k)$. While the biased estimator keeps it unaltered, the averaging one



(a) (b)

Fig. 1: Periodogram of a voiced signal: a) clean and contaminated with a 0 dB noise, and b) clean and averaged.



(a) (b) (c)

Fig. 3: Examples of noise autocorrelation transformation with the averaged estimator considering a period $T = 40$ and $N_p = 4$ periods. Three different autocorrelation widths δ ($r_d(k) = 0$ for $|k| \geq \delta$) are considered: a) $\delta = 100 > T$, b) $\delta = 30 > T/2$, c) $\delta = 10 < T/2$.

reduces its power although it generates a series of images. This is illustrated in figures 3a, 3b and 3c. These figures show three examples of how the averaging estimator transforms the noise autocorrelation. In these three cases it is supposed that the noise autocorrelation is non zero only in an interval $|k| < \delta$. We can see that the distortion power is always diminished, although more significantly (by a factor N_p) in the case that $\delta < T$, that is, when the noise does not have significant long-term correlations (figures 3b and 3c). It is particularly interesting the case of figure 3c ($\delta < T/2$), since the transformed autocorrelation consists of a series of images of the original autocorrelation which appear repeated every T lags and scaled by a factor $1/N_p$.

From the previous discussion, we see that the averaging technique always reduces the distortion component power, although it is more effective for noises without long-term correlations. In fact, we can expect that this will be a typical situation when speech is contaminated with acoustic noise. In the next section we will further exploit this fact to obtain an improved averaging estimator.

It is interesting to point out that the averaging technique developed in this section can be alternatively interpreted under a framework based on the use of sample permutations in which the signal samples are reordered through a permutation $b(n)$, so that a permuted signal is obtained as $z(n) = x(b(n))$ ($n = 0, 1, \ldots, N-1$) [19]. An important property of this technique is that if the applied permutation is random enough, then the signal spectrum is whitened. In our case we must preserve the periodic component and, therefore, the applied permutations must be periodic, that is,

$$b(n) = b_{\underline{n}}(\overline{n})T + \underline{n} \tag{12}$$

where $\overline{n} = \lfloor n/T \rfloor$ and $b_{\underline{n}}(i)$ ($i = 0, 1, \ldots, N_p - 1$) is a random permutation. In this case, only the permutation of samples with the same position $\underline{n}$ inside a period is allowed, so that any periodicity with period T is preserved. It can be proved that for a given signal $x(n)$, the correlation between two permuted versions of $x(n)$, $z_1(n) = x(b_1(n))$ and $z_2(m) = x(b_2(m))$, considering $b_1(n)$ and $b_2(m)$ two random variables with uniform distribution, is given by,

$$E[z_1(n)z_2(m)] = E[\bar{\pi}_x(n, m)]$$

Then, from equation (8) we can write,

$$E[\bar{r}_x(k)] = \frac{1}{N} \sum_{n=k}^{N-1} E[z_1(n)z_2(n-k)]$$

This result provides us with an alternative interpretation of the pitch-synchronous signal averaging by means of which the noise would be first (to some extent) whitened with two different periodic permutations and, then, reduced with the cross-correlation computation (the cross-correlation of two white noises is zero).

Also, it can be proved that, in the same way as random permutations tend to preserve the signal spectrum at frequency $\omega = 0$ [19], periodic permutations do the same at the pitch harmonics ($\omega = k/T;\ k = 0, 1, \ldots$) as it is observed in figure 1b.

IV. Sifting of sample product tables

As previously mentioned, random additive noises tend to affect more the lower lag autocorrelations. Then, the first approach to diminish the noise effect is to eliminate these autocorrelations with the argument that the short-term correlations required for recognition are still present around the subsequent pitch repetitions (lags $k = nT$, $n = 1, 2, \ldots$). This is the main argument of the HASE technique. The problem of this strategy is that the estimation of the higher lags is worse than that of the lower ones since there are less samples for it.

In this section we present an alternative based on signal averaging which does not require to remove the lower autocorrelation lags. In order to do this, we will start from the pitch-synchronous averaging of the previous section, which provides the autocorrelation estimate $\bar{r}_x(k)$ given by equation (8). Our proposal consists in modifying the definition of the table elements $\bar{\pi}_x(n, m)$ required by (8) so that we avoid the use of those elements of the sample product table $\pi_x(n, m)$ more affected by noise. As pointed out in the introduction section, if we consider that the noise affects more the lower autocorrelation lags (let us say $|k| < \delta$), the neglected elements will be those closer to the main diagonal of $\pi_x(n, m)$. We will refer to this procedure as *sifting*. The resulting autocorrelation estimate is obtained as,

$$\tilde{r}_x(k) = \frac{1}{N} \sum_{n=k}^{N-1} \tilde{\pi}_x(n, n-k) \qquad (k = 0, \ldots, N-1) \quad (13)$$

where the new table $\tilde{\pi}_x(n, m)$ $(n, m = 0, \ldots, N-1)$ is obtained from table $\bar{\pi}_x(n, m)$ (see equation (9)) by sifting the elements $\pi_x(n, m)$, that is,

$$\tilde{\pi}_x(n, m) = \frac{1}{N_\delta(\underline{n}, \underline{m})} \sum_{(i,j) \in S_\delta(\underline{n}, \underline{m})} \pi_x(iT + \underline{n}, jT + \underline{m}) \quad (14)$$

where δ is the sifting interval and $N_\delta(\underline{n}, \underline{m})$ is the number of pairs (i, j) $(i, j = 0, \ldots, N_p - 1)$ belonging to set $S_\delta(\underline{n}, \underline{m})$. In turn, this set is defined by,

$$S_\delta(\underline{n}, \underline{m}) = \{(i, j) : |(i - j)T + \underline{n} - \underline{m}| \geq \delta\} \quad (15)$$

The restriction $(i, j) \in S_\delta(\underline{n}, \underline{m})$ ensures that only sample products $\pi_x(n, m)$ placed on diagonals at a distance δ or higher from the main diagonal are used to obtain $\tilde{\pi}_x(n, m)$. This new table is again symmetric and periodic with period (T, T).

The sifting process is illustrated in figure 4. The elements eliminated in table $\pi(n, m)$ are marked in gray. We can see

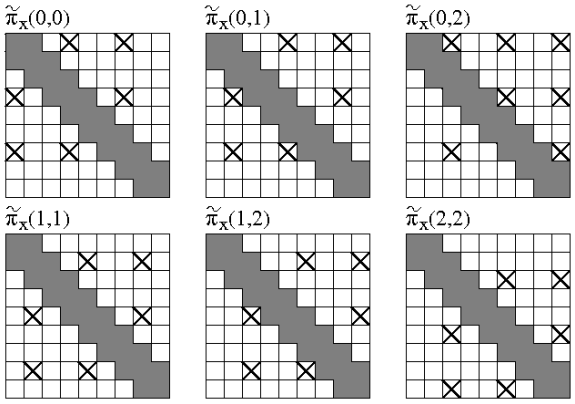


Fig. 4: Computation of elements $\tilde{\pi}_x(0, 0)$, $\tilde{\pi}_x(0, 1)$, $\tilde{\pi}_x(0, 2)$, $\tilde{\pi}_x(1, 1)$, $\tilde{\pi}_x(1, 2)$ and $\tilde{\pi}_x(2, 2)$ from the sifted table $\pi_x(n, m)$ (sifting interval $\delta = 2$) for a signal with pitch $T = 3$ and $N_p = 3$ periods.

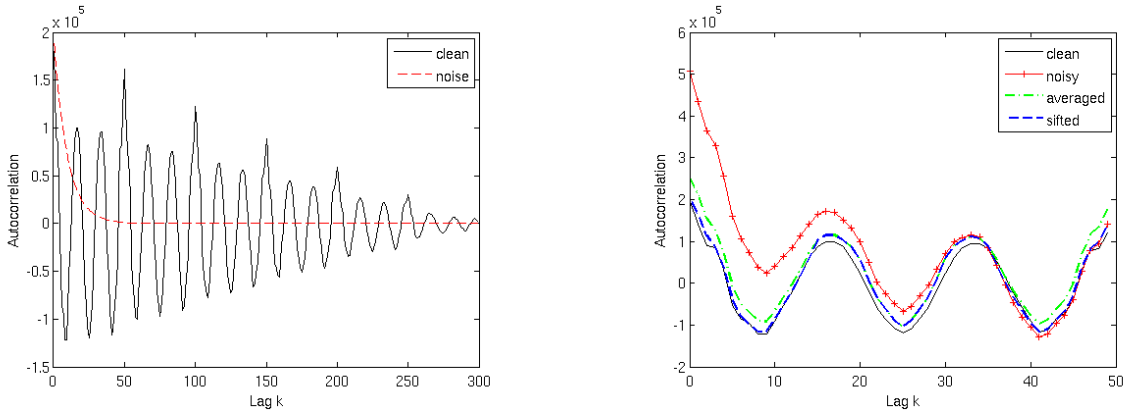
that, unlike HASE and despite of the applied sifting, we can still compute the whole table $\tilde{\pi}_x(n, m)$ for all (n, m) and, therefore, we can obtain estimates of the autocorrelation for all k. In fact, in a typical case, the frame size N will be much greater than the sifting interval δ, so that there will still be a large number of sample products to compute every $\tilde{\pi}_x(n, m)$.

In order to illustrate the effect of sifting over the product tables, the three autocorrelation estimators developed so far (biased, averaging, sifting) are compared in figure 5 when applied to a voiced speech segment. Figure 5a shows the biased estimate of the clean signal and the autocorrelation of an AR(1) process employed to contaminate the clean signal at 0 dB.

The sifting interval is chosen so that the AR(1) autocorrelation has decreased until a 20% of its peak at lag $k = 0$ ($\delta = 15$ in this case).

Figure 5b shows the biased autocorrelation estimates of the clean and contaminated signals, as well as the averaging and sifting estimates obtained from the noisy signal (only for $0 \leq k < T$ given the periodicity of the averaging and sifting estimators). It is clear that the averaging and sifting estimates are much closer to the clean autocorrelation. Furthermore, we observe that the sifting estimator obtains even better results than the averaging one for $|k| < \delta$ and $T - \delta \leq |k| < T$. These two ranges are especially relevant for applications like speech recognition since they represent the short-time correlations and the spectral envelope. We will show in the next subsection that sifting can provide better results than averaging in these ranges of the autocorrelation lag and, also, that both estimators coincide in the range $\delta \leq |k| < T - \delta$.

We still have the problem of selecting a suitable value for δ. In principle, we could say that a reasonable value is that from which the noise autocorrelation is small enough, that is, $r_d(k) \approx 0$ for $|k| \geq \delta$. We will see in the experimental results section that the answer is not so straightforward since we also have to take into account that as the value of δ gets higher, more signal information is wasted. Therefore, we will have to



(a) (b)

Fig. 5: Examples of autocorrelation estimates: a) biased autocorrelation of the voiced clean signal ($T = 50$) and Yule-Walker autocorrelation of a 0-dB AR(1) contaminating noise, and b) biased autocorrelation of the clean signal and biased, averaged and sifted ($\delta = 15$) estimates obtained from the noisy signal.

achieve a compromise between both facts.

A. *Statistical interpretation of the sifting estimator*

We carry out now a statistical analysis similar to that of subsection III-C. As shown in appendix A, the expected values of the averaging and sifting estimators are obtained in the same way, although employing two different auxiliary functions $\bar{s}_d(j)$ and $\tilde{s}_d(j)$, respectively (see equations (20) and (30)). Function $\bar{s}_d(j)$ consists of a series of $2N_p - 1$ images (shifted every T lags) of the original noise autocorrelation $r_d(k)$, although the function is only defined in the interval $[-(T-1), T-1]$. For simplicity, we will assume now that the noise autocorrelation is not null only in a specific range $|k| < \delta$ (δ will be taken as sifting interval), since this is the main argument of the sifting technique. Furthermore, we assume that $\delta < T/2$ (as it is depicted in figure 3c). In this case, function $\bar{s}_d(j)$ only has the contribution of images $l = -1, 0, +1$ since it is only defined in the interval $[-(T-1), T-1]$. Then, it can be simplified as,

$$\bar{s}_d(j) = \frac{N_p - 1}{N_p^2} r_d(j-T) + \frac{1}{N_p} r_d(j) + \frac{N_p - 1}{N_p^2} r_d(j+T) \quad (16)$$

Figure 6 shows the three images of $r_d(k)$ (which correspond to the three terms of equation (16)).

For the sifting estimator, we have to consider function $\tilde{s}_d(j)$ (equation (30)) instead of $\bar{s}_d(j)$. This is a sifted version of $\bar{s}_d(j)$ which only includes those terms of (16) ($l = -1, 0, +1$) belonging to set $L_\delta(j)$ (defined in equation (31)). In order to compute this new function $\tilde{s}_d(j)$, we will distinguish three different ranges for variable j by taking into account both the definition of $L_\delta(j)$ and the form of the original $\bar{s}_d(j)$ function depicted in figure 6. We will consider only $0 \leq j < T$, although the result can be straightforwardly extended to $|j|$ ($j \in [-(T-1), T-1]$) given that $\bar{s}_d(j)$ is an even function. The three cases are:

1) Case $0 \leq j < \delta$. The image of $r_d(j)$ corresponding to $l = 0$ is not included ($l = 0 \notin L_\delta(j)$). Then, $\tilde{s}_d(j) = 0$.

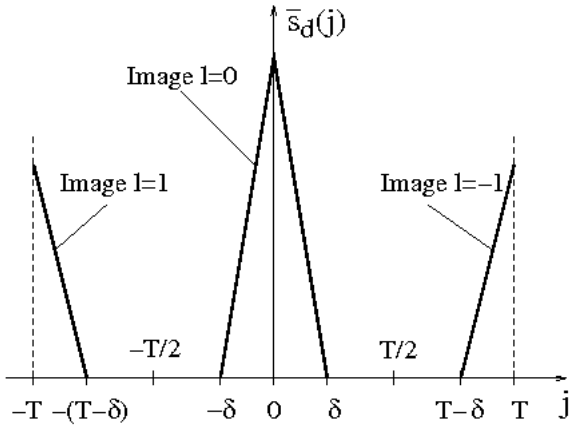


Fig. 6: Example of $\bar{s}_d(j)$ function when $r_d(k) = 0$ for $|k| \geq \delta$ and $\delta < T/2$.

2) Case $\delta \leq j \leq T - \delta$. The three images $l = -1, 0, +1$ are used, so $\tilde{s}_d(j) = \bar{s}_d(j) = 0$.
3) Case $T - \delta \leq j < T$. The image of $r_d(j)$ corresponding to $l = -1$ is not included ($l = -1 \notin L_\delta(j)$). Then, $\tilde{s}_d(j) = 0$.

Therefore, we have that $\tilde{s}_d(j) = 0$ for all $j \in [-(T-1), T-1]$ and, considering equation (29), we can conclude that,

$$E[\tilde{r}_x(k)] = w_B^N(k) r_p(k) \quad (17)$$

Then, the influence of the noise is completely removed in a statistical sense. In other words, if we neglect the Barlett window, we can say that the sifted estimator is an unbiased estimator of the periodic signal autocorrelation $r_p(k)$.

In a real situation, we can only expect that the confinement condition ($r_d(k) = 0$ for $|k| > \delta$) will be accomplished only approximately. It must be considered that even in the case that this condition is clearly violated, we can still expect that the noise autocorrelation will be more significant for the lower

lags, so that the proposed sifting will still be useful. Another implementation issue is how to select the sifting interval δ. This is addressed in section VI.

V. Implementation of the AMFCC-based recognition system

A. AMFCC feature extraction

Our AMFCC feature extractor is based on the Aurora/ETSI xFE (extended FE) front-end [16]. xFE extracts the same features as the FE front-end, although it incorporates a pitch extractor (intended for a possible synthesis of the signal from the speech features), which is also employed for the computation of the averaging and sifting autocorrelation estimates (see the following subsection). In our experiments we also consider a modified xFE in which the FFT spectrum is squared (we will use the notation xFE^2 to reflect this fact). Therefore, a power spectral estimate is employed, as in all the AMFCC tested techniques. In order to compute the autocorrelation estimates required for AMFCC feature extraction, we employ an 8 kHz sampling rate. The signal is offset-compensated and pre-emphasized and, then, segmented into frames of length 256 (32 ms) and shifted 80 samples (10 ms). Front-ends xFE and xFE^2 have been modified to also use these pre-processing parameters.

Except HASE, which uses the OSA sequence, all the autocorrelation estimates (biased, averaging and sifting) are obtained for the whole lag interval $[-255, 255]$ (511 autocorrelation coefficients), which are windowed with a DDR window (centered on the 0th lag) since it provides a suitable dynamic range for speech power spectral estimation. It must be pointed out that the considered autocorrelation estimates do not have noticeable differences in computational cost, except for the required pitch extraction. The spectrum is obtained through a 512-point FFT which is decimated to 256 points for compatibility with the xFE filterbank. The rest of the feature extraction process coincides with that of xFE. Then, a set of 13 MFCC coefficients (including the 0th-order one) is obtained. Although the log-Energy is typically included in the feature vector and is more commonly employed than the 0th-order MFCC coefficient, we use this last one in our AMFCC feature vectors since it is naturally provided by our feature extractor and, in fact, also provides a frame energy measure. From the resulting 13 static features, delta and delta-delta features are derived (as in xFE), obtaining a feature vector with 39 components. Finally, cepstral mean normalization (CMN) is applied to all the tested front-ends.

B. Pitch computation

The pitch extractor employed is that of the xFE front-end, although we have added a post-processing for smoothing. The xFE pitch extractor provides a pitch period value as well as a class label indicating one of these four voicing classes: nonspeech, unvoiced, mixed-voiced and voiced. For our purposes, the two first classes are grouped into one single unvoiced class, and the other two into one single voiced class. In the case of an unvoiced frame, xFE assigns a null pitch.

Then, all this voicing information is submitted to a two-stage smoothing process:

1) The voicing class is smoothed through a mode filter which selects the most frequent class in an interval of length 15 (heuristically determined) centered in the current frame. If this mode filter does not modify the voicing class initially assigned, the pitch value initially assigned is also kept. However, if the mode filtering modifies the voicing class initially assigned to a given frame, two possibilities must be considered. First, if a voiced frame is relabeled as unvoiced, then its pitch is fixed to zero. On the contrary, if an unvoiced frame is relabeled as voiced, its pitch is kept to zero (although recomputed in the next stage).
2) It must be taken into account that the pitch value obtained in the previous stage for voiced frames could be erroneous due to noise or, also, because it is zero. These errors must be detected and mitigated. Thus, an error is detected if the pitch value is out of the interval $[0.625T_{aver}, 1.6T_{aver}]$, where T_{aver} is the average pitch of the whole utterance (this average excludes all zero values). This means that we only allow pitch values in the range given by the average pitch frequency ± 4 tones [21]. In case of error, the new assigned (estimated) pitch corresponds to the lag where the signal biased autocorrelation is maximum. The search for this maximum is restricted to a range of ± 2 tones around the instantaneous average pitch $\bar{T}_t$ [21], that is, to the interval $[0.80\bar{T}_t, 1.25\bar{T}_t]$.
   When an error is detected, this average pitch is initialized to $\bar{T}_0 = T_{aver}$. If the error situation persists in the following time steps ($t > 0$), the following recursion is employed,

$$\bar{T}_t = (1-\alpha)T_{t-1} + \alpha\bar{T}_{t-1} \qquad (t > 0), \qquad (18)$$

   where T_{t-1} is the pitch value estimated at the previous time step. The filtering weight α was determined through a preliminary experiment performed over five training sentences of the Aurora-2 database (described in section VI). The experiment consisted of obtaining the mean square error (MSE) between the clean pitch contours and the estimated ones for the same utterances contaminated with different Aurora-2 noises at 0 dB. The minimum MSE value was obtained for $\alpha = 0.7$.

C. Application to unvoiced frames

The averaging and sifting techniques have been conceived and developed for voiced frames. However, in ASR we will also have to consider unvoiced sounds, as well as silence. We see now how our proposed estimators can also be employed in these two cases.

The case of silence is quite simple. Ideally, silence has a zero autocorrelation. Then, the observed autocorrelation mainly corresponds to additive acoustic noise. Therefore, if we assign a fictitious and suitable pitch value in order to apply the averaging or the sifting estimator, then the noise autocorrelation is transformed as it was shown either in

subsection III-C (for the averaging estimator) or in IV-A (for the sifting estimator), where we saw that the noise contribution can be reduced or even completely removed.

The case of an unvoiced sound is more complicated. While the autocorrelation of a voiced sound is little altered by averaging and sifting due to its periodicity, the autocorrelation of an unvoiced sound does not have this property. Furthermore, it can be expected that it will have its highest and most significant values in the lower lag section, just the same as the additive noise. A first possibility for the analysis of unvoiced sounds could be the application of a standard feature extraction (as in xFE), reserving averaging (or sifting) to voiced frames. However, although this strategy could work in clean conditions, it would result in a heterogeneous sequence of feature vectors in the presence of additive noise. Thus, frames treated with averaging/sifting (voiced and silence) would have been de-noised but the rest are still noisy. In the experimental results section (VI-B) we will see that it is better to follow the same strategy as with silence, that is, to use a fictitious and suitable pitch and apply our estimators without any other modification. This strategy is also followed in the HASE technique, and it is supported by two different facts. First, it provides the same type of features for all types of sounds, avoiding a possible mismatch due to noise. Second, as mentioned in sections III.A and III.C (and shown in figure 1), the averaging technique tends to preserve the spectrum at the pitch harmonics, so that the resulting autocorrelation estimate still contains information about the original signal.

VI. Experimental framework and results

A. Recognition system and task

The experiments have been carried out with the Aurora-2 database and its corresponding experimental setup [20]. The Aurora-2 database is based on the TI-Digits database (connected digits) decimated to 8 kHz. We also use the speech recognizer provided by Aurora, which uses eleven 16-state continuous word HMM models (except silence and pause, that have 3 and 1 states, respectively), with 3 Gaussians per state (except silence, with 6 Gaussians per state, and pause, which is the central state of silence). Training is performed with 8440 clean sentences and test is carried out over three sets (A, B and C). Test set A contains 4 subsets (1001 sentences each) contaminated with four different types of additive noise (subway, babble, car and exhibition) at different SNRs (clean, 20, 15, 10, 5, 0 and -5 dB). For every SNR, the word accuracy (WAcc) is obtained by averaging the WAcc values of the four subsets. A mean word accuracy is computed by averaging the results obtained for all the SNRs excluding those of clean and -5 dB [20]. The experiments with set B are exactly the same ones except for the use of a different set of noises (restaurant, street, airport and train station). Set C includes two subsets (contaminated with subway and street) over a different channel (MIRS channel). Training and testing are always carried out with the same feature extractor.

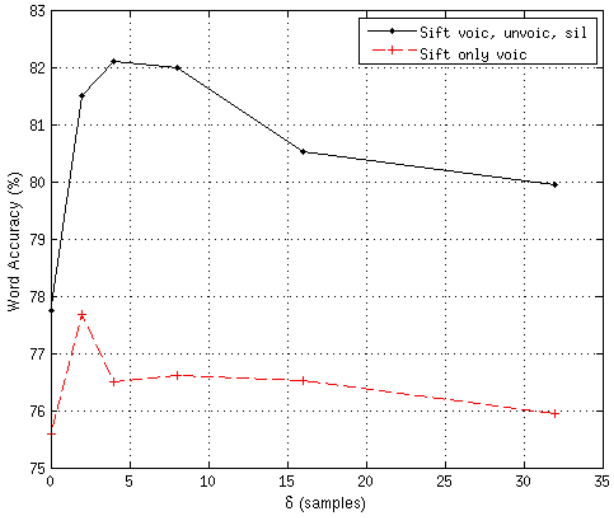


Fig. 7: Mean word accuracy of Set-A versus the sifting interval δ for two cases of averaging/sifting: applied only to voiced frames (+) and to all types of frames (●).

δ (samples)	0	2	4	8	16	32
WAcc(%)	77.94	78.75	80.05	80.44	79.65	79.30

TABLE I: Mean word accuracy for the whole Aurora-2 database versus the sifting interval δ (averaging/sifting applied to all frames).

B. Search for the sifting interval and application to unvoiced sounds

For the sake of simplicity, we will focus now on the search for a suitable fixed sifting interval δ. The use of a variable (dynamic) sifting interval is discussed later in this paper. In order to determine a suitable fixed δ value, there are two facts that must be taken into account.

First, we can expect that the noise autocorrelation has a decreasing shape. However, in the case of a real noise we cannot expect to find a specific lag from which the noise autocorrelation is null. Thus, in principle, the best choice would be to select δ as large as possible. On the other hand, it is recommendable to take δ as small as possible in order to avoid the loss of useful information about the signal. Then, we will have to achieve a compromise between these two effects in order to optimize the performance of our estimators.

First, we only evaluate the use of different sifting intervals under additive noise (only test set A is considered), since averaging and sifting have been developed for these types of distortion. The resulting mean WAcc plots are shown in figure 7. Following the discussion of section V.C, we apply first averaging and sifting only to voiced frames, while AMFCC features (obtained from the biased autocorrelation) are extracted for the unvoiced ones (*Sift only voic* plot). Second, all types of frames (voiced, unvoiced and silence) undergo our averaging/sifting processing (*Sift voic, unvoic, sil* plot). In this case, the fictitious pitch assigned to unvoiced frames is $T = 55$ samples (6.9 ms), which corresponds to the

average human pitch (preliminary experiments showed us that this is not a critical parameter of the system). We can see that, in general, sifting is more beneficial than a simple averaging ($\delta = 0$). Also, it provides a better performance when applied to all types of frames than when applied only to the voiced ones. This last result confirms the assertion made in section V.C about the benefits of applying averaging and sifting to all types of frames, since it is preferable to apply a noise reduction mechanism to unvoiced frames than doing nothing. This is specially true for silence frames (as also mentioned in section V.C) since they only contain noise. We also observe in the figure that the extension of averaging/sifting to unvoiced sounds and silence requires the use of a slightly higher sifting interval (from $\delta = 2$ to $4 - 8$).

Although the averaging and sifting estimators have been specifically developed for additive noise, convolutive distortion may also degrade speech signals. The robustness of averaging/sifting under convolutive noise can be assessed in table I, where the whole Aurora-2 database (including set C) has been employed. Again, it is observed that sifting introduces substantial improvements and that the best performance is obtained for $\delta = 8$. We will use this value in the following experiments. The need for a slightly higher δ can be explained by the autocorrelation spread caused by the channel filtering (a narrower frequency band involves a wider autocorrelation).

C. Final results and comparisons

Now, we can analyze the performance of the proposed feature extraction methods and compare them with other methods. For this comparison, we consider the following front-ends as references: xFE, xFE^2 and HASE. HASE explicitly fixes to zero the first 16 autocorrelation coefficients. These reference front-ends employ log-Energy as energy feature. Then, we have the AMFCC front-ends derived from the three considered autocorrelation estimators: AMFCC-Bias (AMFCC from the biased autocorrelation), AMFCC-Aver (AMFCC obtained from averaging), AMFCC-Sift (AMFCC obtained from averaging plus sifting with $\delta = 8$) and AMFCC-Aver/Sift-Ideal (the same as AMFCC-Aver or AMFCC-Sift but with pitch extracted without noise). These two last front-ends are introduced in order to show the upper limit of averaging/sifting. All methods apply CMN.

Word accuracy results for different SNR values are given in table II. Let us pay attention first to the last column (mean WAcc). In general, AMFCC features are more robust than regular MFCCs. The best results are achieved by AMFCC-Aver (5.23% better than HASE) and AMFCC-Sift (7.73% better than HASE), what confirms the robustness of the proposed pitch-synchronous feature extraction methods and, in particular, of the sifting approach. The results provided by AMFCC-Aver-Ideal and AMFCC-Sift-Ideal indicate that our proposals could provide further improvements with a more robust pitch extractor. The behavior of the different front-ends versus the SNR is also shown in figure 8. It can be observed that averaging and sifting (especially this last one) clearly overcome HASE in the middle range SNRs (15 to 0 dB). This superiority only ceases in the case of -5 dB since an

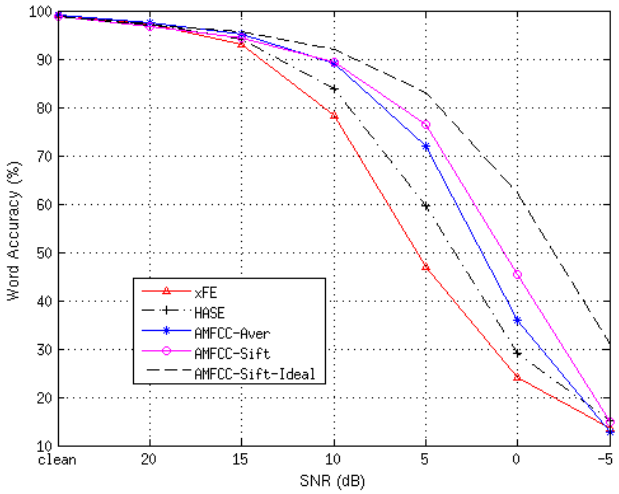


Fig. 8: Word Accuracy versus SNR for the tested front-ends.

appropriate pitch estimation in this case is obviously a difficult task. We must also mention the case of the clean condition. Here, the averaging technique provides a performance similar to that of xFE and HASE, but sifting introduces a slight degradation. This behavior can be easily understood if we take into account that sifting, although quite effective against additive noise, is also an information loss process, so under no-noise conditions there is nothing to win and part of our data is wasted.

Finally, table I also shows the results obtained with the ETSI advanced front-end (AFE) [17]. AFE provides a high performance reference although it must be taken into account that it combines several techniques and an explicit noise estimation, while averaging or sifting only require a pitch estimate.

In order to better assess the improvements achieved, table III compares the mean WAcc values of the three best front-ends (HASE, AMFCC-Aver and AMFCC-Sift) for the different additive noises employed for testing. It is observed that AMFCC-Aver and AMFCC-Sift clearly overcome HASE in all cases and, also, that AMFCC-Sift is always better than AMFCC-Aver except for *Restaurant* and *Airport*. There are several reasons why AMFCC-Sift can perform worse than AMFCC-Aver (a non suitable sifting interval, errors in pitch extraction, ...). In particular, in the case of *Airport* with perfect pitch extraction, AMFCC-Sift can outperform AMFCC-Aver in 0.77% (with $\delta = 2$).

In the case of *Restaurant* with perfect pitch extraction, AMFCC-Sift also reduces its distance (up to 0.56% with $\delta = 4$) to AMFCC-Aver. These results point out the need to improve our pitch extractor and to apply sifting with a dynamic value of δ (that is, a suitable δ for each instant). In order to estimate how a better pitch estimator and a dynamic δ can improve the performance, table III also shows two new experiments named AMFCC-Sift ($\delta = Ideal$) and AMFCC-Sift-Ideal ($\delta = Ideal$), where the best possible δ for each testing sentence is applied with estimated and ideal pitch,

Technique	Clean	20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB	Mean (20-0 dB)
xFE	99.07	97.21	92.90	78.37	47.04	24.05	13.57	67.91
xFE^2	98.96	97.13	93.29	80.84	52.61	26.16	15.06	70.01
HASE	99.10	97.24	94.01	83.70	59.50	29.11	15.05	72.71
AMFCC-Bias	99.09	97.63	94.79	84.91	57.65	27.89	14.63	72.57
AMFCC-Aver	99.02	97.40	95.11	89.09	72.04	36.05	12.79	77.94
AMFCC-Sift ($\delta = 8$)	98.80	96.63	94.45	89.36	76.39	45.37	14.84	80.44
AMFCC-Aver-Ideal	99.02	97.67	95.99	91.52	79.98	53.37	24.00	83.71
AMFCC-Sift-Ideal ($\delta = 8$)	98.80	97.06	95.58	91.93	83.06	62.29	30.87	85.99
AFE	99.07	97.73	96.14	91.91	82.35	60.19	29.02	85.66

TABLE II: WAcc results obtained by the different front-ends tested with Aurora-2 (Set A, B and C) for different SNR values.

Technique	WM	MM	HM	Mean
xFE	84.03	62.15	37.85	61.34
HASE	85.91	64.69	43.34	64.65
AMFCC-Sift ($\delta = 8$)	76.80	50.14	39.11	55.35
AMFCC-Sift-Ideal ($\delta = 8$)	84.52	71.47	61.44	72.48

TABLE IV: WAcc results obtained by the different front-ends for Aurora-3 Danish (real noise).

respectively. In these experiments, we have employed the same acoustic models as for AMFCC-Sift ($\delta = 8$) and the optimal testing δ value is selected among (0, 2, 4, 8, 16 y 32). The improvements with respect to a fixed value ($\delta = 8$) are quite significant.

Finally, table IV shows the results obtained over a real noise database (Aurora-3, Danish). This database contains in-car speech recorded with different microphones placed at different places, which provides 3 different degradation conditions: well-matched (WM), medium mismatch (MM) and high mismatch (HM). More details about this database and the configuration of the back-end employed in this case can be found in [22]. We can observe that AMFCC-Sift requires a good pitch estimator in order to improve the HASE results. In this case, AMFCC-Sift(-Ideal) obtains around a 18% WAcc improvement over HASE for the high mismatch condition. Again, it can be expected that a dynamic δ will also contribute to improve these results.

VII. Summary and future work

In this paper we have proposed two robust autocorrelation estimators for periodic signals which are based on a pitch-synchronous signal averaging. They have been applied to the problem of feature extraction when recognizing speech contaminated by additive noise. The proposed estimators are inspired by the commonly used biased estimator although formulated through a sample product table. The basic idea of our proposal is to use a pitch-synchronous averaging in order to obtain new estimators by appropriate transformations of this table. The first estimator, which is referred to as averaging estimator, is the one directly derived from the mentioned averaging. We have shown that this estimator preserves the autocorrelation of voiced signals while tends to reduce the power of the noise. We have seen that under certain reasonable conditions, the noise power is reduced by a factor equal to the number of pitch periods, which is equivalent to an SNR increase by the same factor. Unlike other autocorrelation estimators employed in speech processing, as the ones employed by SMC, OSA-LPC or HASE, the averaging process does not involve the removal or weighting of the lower-lag autocorrelations, so it provides an estimate of the whole autocorrelation sequence of the clean signal.

The second estimator is based on a modification of the former one which sifts the original sample product table so that those sample products which are presumably more sensitive to noise are discarded. This is referred to as sifting estimator. We have shown that, in a statistical sense and under certain conditions, this estimator can totally remove the noise while preserves the periodic autocorrelation of the clean signal.

Although the proposed techniques are initially conceived only for voiced sounds, we have also successfully extended them to all kinds of sounds. Our experimental results have shown the clear superiority of our feature extractor when compared with standard front-ends like FE/xFE and with a robust technique like HASE. We have also shown that the feature-extractors derived from the proposed autocorrelation estimators still have a large margin for improvement. This improvement depends on the development of pitch extractors more robust against noise.

The different techniques and results obtained along this paper also suggest some directions to explore in future work. Thus, we think that the extension of the pitch-synchronous analysis to unvoiced sounds should be improved in order to preserve more information about the original signal. Also, we must point out that the proposed autocorrelation estimators involve a fixed feature extraction, but they could be made more adaptive. Thus, for example, the best sifting interval could be adaptively determined according to the type of additive noise. Furthermore, we think that sifting should be applied not only to products placed in the central diagonals of the sample product table, but also to any product meaningfully affected by noise.

Appendix A
Expected values of the averaging and sifting estimators

This appendix is devoted to obtaining the expected value of the averaging autocorrelation estimator (equation (11)). Then, the obtained solution is extended to the sifting estimator.

In order to develop equation (11), we have to compute first the expected value of $\bar{\pi}(n, m)$ (equation (9)) considering that

Technique	Set A				Set B				Set C		Mean (20-0 dB)
	Subw	Babb	Car	Exhi	Rest	Stre	Airp	Trai	Subw MIRS	Stre MIRS	
HASE	71.79	73.82	70.44	68.58	75.55	73.90	76.54	74.22	71.21	72.83	72.89
AMFCC-Aver	78.40	79.84	76.92	75.83	80.48	78.68	79.90	78.10	74.95	76.26	77.94
AMFCC-Sift ($\delta = 8$)	83.92	81.99	81.28	80.80	77.62	82.47	79.84	80.60	76.84	79.05	80.44
AMFCC-Sift ($\delta = Ideal$)	89.07	87.49	86.68	86.88	85.03	88.07	85.92	86.03	85.17	85.96	86.63
AMFCC-Sift-Ideal ($\delta = Ideal$)	93.40	92.10	91.44	90.49	91.06	92.28	91.11	92.49	91.43	91.40	91.72

TABLE III: WAcc results obtained by the different front-ends for the different noises of Aurora-2.

$x(n)$ is a stationary random process,

$$\begin{aligned} E[\bar{\pi}_x(n,m)] &= \frac{1}{N_p^2}\sum_{i,j=0}^{N_p-1} E[\pi_x(iT+\underline{n}, jT+\underline{m})] \\ &= \frac{1}{N_p^2}\sum_{i,j=0}^{N_p-1} r_x\left((i-j)T+(\underline{n}-\underline{m})\right) \\ &= \frac{1}{N_p^2}\sum_{l=-(N_p-1)}^{N_p-1} (N_p-|l|) r_x\left(lT+(\underline{n}-\underline{m})\right) \end{aligned} \quad (19)$$

If we define the following even function,

$$\bar{s}_x(j) = \frac{1}{N_p^2}\sum_{l=-(N_p-1)}^{N_p-1} (N_p-|l|) r_x(lT+j) \quad (j=-(T-1),\ldots,T-1) \quad (20)$$

then we have that,

$$E[\bar{\pi}_x(n,m)] = \bar{s}_x(\underline{n}-\underline{m}) \quad (21)$$

Thus, the expected value of equation (11) becomes,

$$E[\bar{r}_x(k)] = \frac{w_B^N(k)}{N-k}\sum_{n=k}^{N-1} \bar{s}_x(\underline{n}-\underline{n-k}) \quad (22)$$

We can consider here two possible cases:

1) Case $\underline{n} \geq \underline{n-k}$. Then, $\underline{n-k} = \underline{n} - \underline{k}$ and the elements of the kth diagonal of table $E[\bar{\pi}_x(n,m)]$ can be expressed as,

$$E[\bar{\pi}_x(n,n-k)] = \bar{s}_x(\underline{k}) \quad (23)$$

The number of elements contained in this diagonal is,

$$N_1(k) = (N_p - \overline{k})(T - \underline{k}) \quad (24)$$

2) Case $\underline{n} < \underline{n-k}$. Then, $\underline{n-k} = \underline{n} - \underline{k} + T$ and the elements of the kth diagonal of table $E[\bar{\pi}_x(n,m)]$ can be expressed as,

$$E[\bar{\pi}_x(n,n-k)] = \bar{s}_x(\underline{k}-T) \quad (25)$$

The number of elements contained in this diagonal is,

$$N_2(k) = (N_p - \overline{k} - 1)\underline{k} \quad (26)$$

It can be easily shown that,

$$N_1(k) + N_2(k) = N - k \quad (27)$$

Finally, we can express,

$$E[\bar{r}_x(k)] = w_B^N(k)\frac{N_1(k)\bar{s}_x(\underline{k}) + N_2(k)\bar{s}_x(\underline{k}-T)}{N-k} \quad (28)$$

When $x(n)$ is a periodic signal of period T, we can easily see that $\bar{s}_x(j) = r_x(j)$ $(j = -(T-1),\ldots,T-1)$ and, also, that $\bar{s}_x(\underline{k}-T) = r_x(\underline{k})$ given the periodicity of $r_x(k)$. Therefore, $E[\bar{r}_x(k)] = w_B^N(k) r_x(k)$. In fact, there is no randomness in this case, so $\bar{r}_x(k) = w_B^N(k) r_x(k)$.

When $x(n)$ is the sum of a periodic signal $p(n)$ and a stationary random process $d(n)$ (not correlated with $p(n)$), then the expected value (11) becomes,

$$E[\bar{r}_x(k)] = w_B^N(k)\cdot \cdot\left(r_p(k) + \frac{N_1(k)\bar{s}_d(\underline{k}) + N_2(k)\bar{s}_d(\underline{k}-T)}{N-k}\right) \quad (29)$$

The expected value of the sifting estimator can be obtained in the same way. All the above expressions can be employed although function $\bar{s}_x(j)$ must be substituted by its sifted version, that is,

$$\tilde{s}_x(j) = \frac{1}{N_\delta(j)}\sum_{l\in L_\delta(j)} (N_p-|l|) r_x(lT+j) \quad (j=-(T-1),\ldots,T-1) \quad (30)$$

where,

$$L_\delta(j) = \{l \in [-(N_p-1), N_p-1] : |lT+j| \geq \delta\} \quad (31)$$

and $N_\delta(j)$ is the number of elements in set $L_\delta(j)$.

ACKNOWLEDGMENT

This work has been supported by the Spanish MEC/FEDER project TEC2007-66600.

REFERENCES

[1] A. M. Peinado and J. C. Segura, *Speech Recognition over Digital Channels*, Wiley, July 2006.
[2] H. Hermansky, "Perceptual Linear Predictive (PLP) Analysis of Speech," *J. Acoust. Society of America*, vol. 87, no. 4, pp. 1738-1752, 1990.
[3] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. on Speech and Audio Proc.*, vol. 2, no. 4, pp. 578-589, Oct. 1994.
[4] O. Ghitza, "Temporal non-place information in the auditory nerve firing patterns as a front-end for speech recognition in a noisy environment," *Computer, Speech and Language*, vol. 1, pp. 109-130, 1986.
[5] K.K. Paliwal and M.M. Sondhi, "Recognition of noisy speech using cumulant-based linear prediction analysis," *Proc. ICASSP*, pp. 429-432, May 1991.
[6] K.K. Paliwal and Y. Sagisaka, "Cyclic autocorrelation-based linear prediction analysis of speech," *Proc. EUROSPEECH*, pp. 279-282, 1997.
[7] N. Ma, P. Green, J. Barker and A. Coy, "Exploiting correlogram structure for robust speech recognition with multiple speech sources," *Speech Communication*, vol. 49, no. 12, pp. 874-891, Dec. 2007.

[8] M. Seltzer, J. Droppo, and A. Acero, "A harmonic-model based front end for robust speech recognition," *Proc. of Eurospeech*, 2003.
[9] J. Barker, M. Cooke and P. Green, "Robust ASR based on clean speech models: An evaluation of missing data techniques for connected digit recognition in noise," *Proc. EUROSPEECH*, pp. 213-216, 2001.
[10] D. Mansour and B.H. Juang, "The Short-Time Modified Coherence Representation and Noisy Speech Recognition," *IEEE Trans. Audio Speech and Signal Processing*, vol. 37, pp. 795-804, 1989.
[11] J. Hernando and C. Nadeu, "Linear prediction of the one-sided autocorrelation sequence for noisy speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 1, pp.80-84, 1997.
[12] B. Shannon and K.K. Paliwal, "Feature extraction from higher-lag autocorrelation coefficients for robust speech recognition," *Speech Communication*, vol. 48, no. 1, pp. 1458-1485, Jan. 2006.
[13] Y. Kuroiwa and T. Shimamura, "An improvement of LPC based on noise reduction using pitch synchronous addition," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, pp. 122-125, 1999.
[14] L. Buera, J. Droppo and A. Acero, "Speech enhancement using a pitch predictive mode," *Proc. of the IEEE International Conference on Acoustics Speech and Signal Processing*, 2008.
[15] ETSI ES 201 108 v1.1.3. Distributed Speech Recognition; Front-end Feature Extraction Algorithm; Compression Algorithms. April 2003.
[16] ETSI ES 202 211 v1.1.1. Distributed speech recognition; Extended front-end feature extraction algorithm; Compression algorithms; Back-end speech reconstruction algorithm. July 2001
[17] ETSI ES 202 050 v1.1.3. Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithms. July 2003.
[18] ETSI ES 202 212 v1.1.1. Distributed speech recognition; Extended advanced front-end feature extraction algorithm; Compression algorithms; Back-end speech reconstruction. August 2003.
[19] B. Lacaze, D. Roviras, "Effect of random permutations applied to random sequences and related applications," *Signal Processing*, vol. 81, pp. 821-831, 2002.
[20] D. Pearce and H. Hirsch, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," *Proc. of ICSLP'2000*, vol. 4, pp. 29-32, October 2000.
[21] H.F. Olson, *Music, physics and engineering*, Dover Publications, Second Edition, 1967.
[22] Aurora Project Database: Subset of SpeechDat-Car - Danish Database. *European Language Resources Association (ELRA)*, 2001.

Juan A. Morales-Cordovilla received the M.Sc. degree in electronic engineering from the University of Granada (UGR), Spain, in 2006. He is currently working towards the Ph.D. degree on robust feature extraction for speech recognition at UGR. Since 2007, he has been with the Research Group on Signals, Networking and Communications (GSTC), Department of Signal Theory, Networking and Communications, UGR, under a research grant. His research interests are in signal processing for noise-robust automatic speech recognition and computational auditory scene analysis.

Antonio M. Peinado (M'95-SM'05) received the M.S. and the Ph.D. degrees in physic sciences from the University of Granada, Spain, in 1987 and 1994, respectively. Since 1988, he has been working with the GSTC research group of the University of Granada, where has leaded or participated in several research projects related with speech recognition, coding and transmission. In 1989, he was a Consultant in the Speech Research Department, AT&T Bell Labs, USA. Since 1996 he is Associate Professor at the Dpt. of Signal Theory, Networking and Communications in Granada. He is author of numerous publications including the book *Speech Recognition Over Digital Channels* (Wiley), and has served as reviewer for several international journals and conferences. His research interests are in distributed and robust speech recognition, speech and audio coding and transmission, and ultrasound signal processing.

Victoria Sánchez (M'95) received the M.S. and the Ph.D. degrees from the University of Granada, Granada, Spain, in 1988 and 1995, respectively.

In 1988, she joined the Research Group on Signals, Networking and Communications (GSTC) of the University of Granada. During 1991, she was visiting with the Electrical Engineering Department, University of Sherbrooke, Canada. Since 1997, she is an Associate Professor at the department of Signal Theory, Networking and Communications of the University of Granada. Her research interests include robust speech and audio coding and transmission and speech recognition. She has served as reviewer for several international journals and conferences.

José A. González received the M.Sc. degree in computing science from the University of Granada (UGR), Spain, in 2006. He is currently working towards the Ph.D. degree on statistical methods for robust speech recognition at UGR. Since 2007, he has been with the Research Group on Signals, Networking and Communications (GSTC), Department of Signal Theory, Networking and Communications, UGR, under a research grant. His research interests are in robust speech recognition and coding and signal processing.