ELSEVIER

Speech Communication 34 (2001) 93–114

SPEECH COMMUNICATION

www.elsevier.nl/locate/specom

Time and frequency filtering of filter-bank energies for robust HMM speech recognition

Climent Nadeu *, Dušan Macho, Javier Hernando

TALP Research Center, Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, J. Girona 1-3, Campus Nord, Edifici D5, E-08034 Barcelona, Spain

Abstract

Every speech recognition system requires a signal representation that parametrically models the temporal evolution of the speech spectral envelope. Current parameterizations involve, either explicitly or implicitly, a set of energies from frequency bands which are often distributed in a mel scale. The computation of those energies is performed in diverse ways, but it always includes smoothing of basic spectral measurements and non-linear amplitude compression. Several linear transformations are then applied to the two-dimensional time-frequency sequence of energies before entering the HMM pattern matching stage. In this paper, a recently introduced technique that consists of filtering that sequence of energies along the frequency dimension is presented, and its resulting parameters are compared with the widely used cepstral coefficients. Then, that frequency filtering transformation is jointly considered with the time filtering transformation that is used to compute dynamic parameters, showing that the flexibility of this combined (tiffing) approach can be used to design a robust set of filters. Recognition experiment results are reported which show the potential of tiffing for an enhanced and more robust HMM speech recognition. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Robust speech recognition; Time and frequency filtering; Modulation spectrum; Filter-bank energies

1. Introduction

Current speech recognition systems use a pattern matching approach (Rabiner and Juang, 1993). The classifier, which is commonly based on hidden Markov models (HMM), relies on a speech spectrum representation that must be adequate in two senses: (1) it has to carry the acoustic features of speech that are relevant for sound discrimination and (2) it has to be properly adapted to the HMM paradigm. Additionally, if the classifier has to work in adverse conditions, the speech representation has to be robust to signal degradations.

A reasonable way of representing the time evolution of the speech characteristics might be to segment the signal in proper acoustic–phonetic units and to subsequently model these units by a set of spectral parameters. Something like temporal decomposition (Atal, 1983), where the signal flow is broken according to overlapping windows and a target spectrum is associated with each one. However, for HMM-based speech recognition, an approach like that has not proven yet to be more useful than the straightforward frame-to-frame procedure, where the waveform is regularly partitioned in blocs or frames. Each frame is considered

[*] Corresponding author.

E-mail addresses: climent@talp.upc.es (C. Nadeu), dusan@talp.upc.es (D. Macho), javier@talp.upc.es (J. Hernando).

PII: S 0 1 6 7 - 6 3 9 3 (0 0) 0 0 0 4 8 - 0

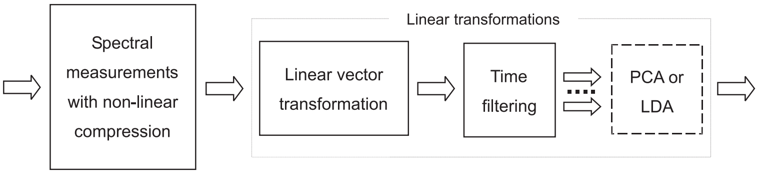

Spectral measurements with non-linear compression
Linear transformations
Linear vector transformation
Time filtering
PCA or LDA

Fig. 1. Scheme of the parameterization front-end.

a segment of a realization of a stochastic process, and it is modeled by a small set of parameters.

Fig. 1 shows the usual scheme for extracting a parametric speech spectral representation from the signal waveform in current speech recognition front-ends. After an initial signal conditioning (A/D conversion plus possible application of speech enhancement techniques, filtering, etc.), the pre-processed speech signal enters the short-time spectral estimation step, where a set of spectral measurements is carried out to obtain a parametric representation of the spectral envelope for the current frame.

Firstly, that vector of initial spectral parameters is linearly transformed and then the temporal sequence of transformed vectors is filtered to compute from it several new time sequences of vectors (dynamic feature vectors), in such a way that the resulting whole set of parameter vectors can benefit more from the HMM formalism that is used in the pattern-matching recognition stage than the initial non-transformed vector. Finally, a linear transformation (PCA or LDA) may be applied to that set of vectors in order to either obtain an uncorrelated and more compact representation or increase the discrimination capacity of the speech representation.

This paper will assume that the spectral measurements are logarithmic filter-bank energies (log FBEs), though most of the presented material is also extendible to other types of spectral estimation techniques, e.g. linear prediction. Usually, the discrete cosine transform (DCT) is used to compute from the log FBEs a set of uncorrelated parameters, the so-called mel-frequency cepstral coefficients (MFCC) or mel-cepstrum, probably the most used spectral representation in speech recognition (Davis and Mermelstein, 1980). On the other hand, orthogonal (Legendre) polynomial filters are used to compute the supplementary dynamic (delta) feature vectors for each frame (Furui, 1986). For example, the recent distributed speech recognition (DSR) standard front-end for clean speech (ETSI SQL W1007) establishes this kind of speech representation.

In this paper, we mainly intend to address several issues involved in that usual FBE-log-DCT-Legendre parameterization scheme that has traditionally been considered unquestionable. In particular, the use of cepstral parameters is discussed, and a computationally simple alternative to the DCT, called frequency filtering (FF), is presented. By performing a combination of decorrelation and liftering, FF yields good recognition performance for both clean and noisy speech. Furthermore, this new linear transformation, unlike DCT, maintains the speech parameters in the frequency domain.

In Section 2, FBEs are reviewed, focusing on their quasi-optimality in statistical terms. In Section 3, the frequency-filtered sequence of log FBEs is presented. Time filtering is considered in Section 4, where the robustness of supplementary dynamic features is discussed. In Section 5, by jointly considering *ti*me and *f*requency *f*ilter*ing* (*tiffing*) as a linear processing of the two-dimensional sequence of spectral energies, the (two-dimensional) modulation spectrum is presented as a tool for designing a robust set of filters. A few tests with the Aurora digit recognition setup and database that are currently used to develop the ETSI STQ WI008 (Pearce, 1998) DSR standard for noisy speech are also presented in that section. A comparison between frequency filtering and optimal techniques for decorrelation (PCA) and linear discrimination (LDA) is reported in Section 6. And, finally, conclusions are drawn in Section 7.

2. Non-linearly compressed filter-bank energies

In current speech recognition front-ends, the first parameterization step consists of extracting a short-time representation of the spectral envelope for each speech frame. There are many reported techniques to estimate the set of spectral parameters (Junqua and Haton, 1996; Picone, 1991), but they always combine some kind of smoothing of raw spectral measurements with non-linear operations.

2.1. Spectral smoothing

Spectral smoothing is used to remove the harmonic structure of the speech spectrum corresponding to pitch information and to reduce the variance (error) of the speech spectral envelope estimation. Additionally, an envelope representation with a small number of parameters is obtained. That operation has basically been done in two alternative ways: linear prediction (LP) analysis and spectral band energy estimation (Rabiner and Juang, 1993). The strength of the LP method arises from the fact that it matches the all-pole model of speech production. In this way, it is able to approximately separate the vocal tract response, which corresponds to the spectral envelope, from the glottal excitation.

However, the band energy parameters have become increasingly popular. They separately represent the energy at each frequency band since they result from integrating the energy values in the time-frequency area specified by the frame length and the effective bandwidth. The main reason of the usefulness of these energies is perhaps the higher flexibility of the sub-band approach with respect to the full-band approach involved in LP modeling. In fact, it offers the possibility of defining the width and shape of the bands along the frequency axis. Also, if the signal-to-noise ratio (SNR) of each band is known, the band energy representation allows to use it in straightforward ways: noise masking, spectral subtraction, etc. (Junqua and Haton, 1996).

The computation of the band energies can be performed in several ways. The classical implementation consists of a bank of filters that perform time convolution followed by wave rectification and low-pass filtering (Rabiner and Juang, 1993). Currently, the most used implementation of the filter-bank analysis operates in the frequency domain by computing a weighted average of the magnitude (or, sometimes, the spectral magnitude) of the DFT values of the windowed speech frame in each frequency band, obtaining in this way the so-called filter-bank energies (FBEs) (Davis and Mermelstein, 1980). Fig. 2 shows the sequence of operations involved in the computation of the FBEs for a given windowed speech frame; it also includes the posterior non-linear compression step from Section 2.3.

Hybrid techniques that combine filter-bank and LP analysis have also been proposed. The best known one is PLP (Perceptual LP), which applies LP modeling after FBE computation and other perceptually motivated processing steps (Hermansky, 1990). Note that, as the order of the LP analysis is usually chosen low (Hermansky, 1998), the PLP parameterization involves an additional smoothing effect. An inverse hybrid approach that performs LP before FB analysis has also been considered so far (Rahim and Juang, 1996; Hernando and Nadeu, 1997b). Logically, the first processing step of the hybrid techniques heavily determines the characteristics of the spectral estimate. A version of the inverse hybrid approach will be used in this paper when the frequency filtering technique, developed for

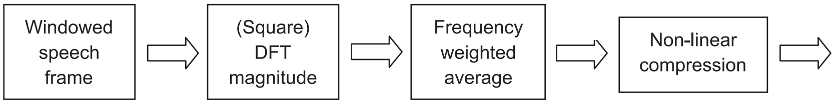


Fig. 2. Usual scheme for computing the non-linearly compressed filter-bank energies for a given frame. Sometimes, a LP modeling block is inserted at the end (like in PLP).

FBEs, will be applied to LP spectral estimates (see Section 3.5.1).

2.2. Quasi-optimality of frequency averaging

LP speech spectral estimates are well established theoretically, since they are based on the all-pole model of speech production. However, the theoretical foundations of the above-mentioned FBEs have not received much attention so far, in spite of becoming a kind of standard in speech recognition. In the following, we are going to review the quasi-optimality of the spectral estimator from which the FBEs come out.

The spectral estimator that results from computing the square magnitude of the Fourier transform of a finite-length segment of signal is called periodogram (Oppenheim and Schafer, 1989). If the set of weights used to compute the FBEs is the same for each band, those band energies can be seen as samples (non-linear sampling if a mel scale is used) of a spectral estimate that results from convolving the periodogram with the weighting function: the frequency-averaged periodogram (FAP) (Nadeu et al., 1997b).

The FAP can be viewed as belonging to the family of multiwindow (MW) spectral estimators, i.e. those that result from averaging several periodograms, each one computed with a different window. Since Thomson's introductory work (Thomson, 1982), good statistical properties have been claimed for the MW estimators that use a set of orthogonal windows which arise from Karhunen–Loève (KL) eigenequations. Unfortunately, the FAP estimator does not match the optimal KL formalism.

Nevertheless, as it was experimentally shown by the authors (Nadeu et al., 1997b), when not only variance and frequency resolution of the estimator but also time resolution are taken into account, the statistical performance of the FAP and that of the estimator arising from the MW–KL formalism are almost identical. Additionally, it can be shown (Nadeu et al., 1998) that the FAP estimator is equivalent, asymptotically and in terms of the first and the second moments, to the optimal MW–KL estimator that uses orthogonal sinusoids as windows.

2.3. Non-linear compression

To compute the speech parameters, non-linear processing is used in both axes of the spectral representation. First, the bands are often distributed in a mel scale to mimic the properties of the human auditory processing, giving less emphasis to the high-frequency bands. And, secondly, non-linear operators are used to compress the large amplitude range of spectral measurements, producing a distribution more similar to the Gaussian one.

The most used non-linear operator is the logarithm which has the additional advantage of converting a gain factor in an additive component in the feature space, which can be easily removed. Although the logarithm is perhaps the most appropriate non-linear operator for recognition of clean speech, it may no longer keep its advantage whenever additive noise is present. Other reported non-linear operators, such as the root $|E|^{\gamma}$ (Alexandre and Lockwood, 1993) and the lin-log $\log(1 + JE)$ (Hermansky and Morgan, 1994), where E denotes a spectral measurement (usually a FBE), are alternative candidates to cope with the problem of parameterizing noisy speech. Actually, both have a parameter which can be adapted to the SNR: γ (Tian and Viikki, 1999) or J (Hermansky and Morgan, 1994). Recently, both techniques were interpreted as masking procedures at spectral valleys (Hunt, 1999). A few results with the root will be presented in Section 4.2.

Alternatively, the DFT magnitude values can be non-linearly compressed before spectral smoothing (Deller et al., 1993). The band energies can be computed by filtering the logarithmically compressed DFT magnitude values along frequency and subsequently decimating them in the frequency axis to select one parameter for each band (Silverman and Dixon, 1974). The decimation may be non-uniform, implementing the mel scale (Mashao et al., 1996). An alternative implementation of the same approach is based on liftering, i.e. windowing in the cepstral domain. In fact, both filtering and liftering techniques also imply a weighted averaging in the frequency domain. On the other hand, as will be seen in Section 3, an explicit or implicit liftering is usually performed after the posterior linear transformation step indicated in Fig. 1.

2.4. Compressed FBEs assumed in this work

Unless otherwise stated, we will henceforth consider that the spectral parameters delivered by the first block of the parameterization front-end scheme from Fig. 1 are logarithmically compressed FBEs at a given number Q of frequency bands. The FBEs will be obtained by a triangular weighted average of either the square magnitude or the magnitude of the DFT values of the Hamming-windowed signal in Q mel-scaled frequency bands. However, when linear prediction is the final spectral measurement step, we can still assume the parameters are band energies; in fact, they can be computed from the prediction coefficients by a linear transformation that may consist of cascading the well-known recursive transformation to cepstral coefficients (Oppenheim and Schafer, 1989) with a DFT to obtain the parameters in the log spectral domain (as will be done in Section 3.5.1 to carry out a few recognition tests with these LP-based energies).

The vector of log FBEs,

$$\boldsymbol{S} = (S(1) \;\; S(2) \ldots S(Q))^{\mathrm{T}}, \tag{1}$$

is then linearly transformed in order that the feature vectors supplied to the pattern matching stage are better adapted to the assumptions of the HMM formalism and take more advantage from it. That vector undergoes at least two kinds of linear transformations, one in the frequency domain and the other in the time domain. In the next sections, both kinds of transformations will be discussed, and also alternatives for improving the recognition performance will be proposed.

3. Linear transformation of the parameter vector

Usual HMMs assume that the acoustic observation vectors can be modeled by Gaussian distributions with diagonal covariance matrices, i.e. they assume that the elements of those vectors are uncorrelated. As the spectral measurements are strongly correlated (e.g. the correlation coefficient of log FBEs of adjacent bands for the TI digits database and $Q = 12$ is 0.92), the parameterization front-ends require a linear transformation that obtains a set of spectral parameters that are globally decorrelated. In the following sections, the conventional approach for parameter decorrelation based on the cepstrum will be questioned, and an alternative transformation that avoids translating the spectral parameters to a non-frequency domain will be proposed.

3.1. Disadvantages of cepstral coefficients for speech recognition

Let us consider the elements of the vector $\boldsymbol{S}$ of log spectral band energies in (1) to be a sequence along the frequency index k. By approximating this sequence $S(k),\ k = 1, \ldots, Q$, with a first-order Markov model, it follows that its corresponding discrete Karhunen–Loève transform (DKLT) is almost equivalent to the data-independent discrete cosine transform (DCT), since the value of the real pole of the model is close to 1 (0.92 for the above-mentioned data base) (see, for instance, Akansu and Haddad, 1992). Due to its closeness to the optimal DKLT, the DCT is able not only to nearly decorrelate the vector of logarithmically compressed FBEs but also to sort the transformed coefficients in variance order. Then, the resulting vector is truncated to retain the highest energy coefficients. It is the mel-frequency cepstral coefficients (MFCC) representation, also called mel-cepstrum. That truncation actually represents an implicit liftering operation with a rectangular lifter that smoothes the spectral envelope represented by the frequency sequence of log FBEs $S(k)$. On the other hand, for some recognition systems employing Euclidean distances, cepstral coefficients have been weighted (explicit liftering) in order to enhance the discrimination of sounds (Paliwal, 1982; Juang et al., 1987; Tokhura, 1987; Hanson and Wakita, 1987).

The cepstral coefficients show three disadvantages for speech recognition:

1. They do not lie in the frequency domain, so lacking a frequency meaning which may be useful, especially for implementing robust techniques.

2. As most current HMMs use Gaussian distributions with diagonal covariance matrices and ML-estimated standard deviations, those HMMs cannot benefit from a cepstral weighting (liftering), since any multiplying factor that is applied to the observations does not affect the Gaussian exponent calculation.
3. They require a DCT computation.

An alternative set of speech parameters that avoids these disadvantages has been recently presented by the authors (Nadeu et al., 1995). This new parameter set is obtained with a very simple linear transformation, called frequency filtering, that will be summarized in the following section.

3.2. The frequency filtering technique

Actually, in current filter-bank analysis, the extreme bands, that would be centered around $\omega = 0$ and $\omega = \pi$, are not considered in the computed energies $S(k)$, $k = 1, \ldots, Q$. Since, in practical situations, these two extreme bands only include a very small fraction of the signal energy, we will extend the sequence $S(k)$ by appending one zero at each end, i.e.

$$\{S(0) = 0, S(1), \ldots, S(Q), S(Q+1) = 0\}. \tag{2}$$

Frequency filtering (FF) (Nadeu et al., 1995) is a transformation of that set of spectral band energies consisting of a convolution between the sequence $S(k)$, $k = 0, \ldots, Q+1$, from (2) and a given (impulse response) sequence $h(k)$ to obtain a new sequence of Q filtered parameters $F(k)$, $k = 1, \ldots, Q$, i.e.

$$F(k) = S(k) * h(k), \quad k = 1, \ldots, Q. \tag{3}$$

Notice that the filtered parameters $F(k)$ still lie in the frequency domain, and only Q values are computed. We will henceforth assume that $h(k)$ is either a first-order FIR filter or a second-order FIR filter centered around $k = 0$; in this way, only the $Q+2$ values from (2) are needed to compute $F(k)$ in (3).

Expression (3) is a linear convolution. However, the same values $F(k)$, $k = 1, \ldots, Q$, can be obtained by the circular convolution $\tilde{F}(k)$ between the sequences $\tilde{S}(k)$ and $\tilde{h}(k)$, $k = -Q \ldots, 0, \ldots, Q+1$, where $\tilde{h}(k)$ is $h(k)$ extended with zeroes, $\tilde{S}(k) = S(k)$ for $k = 0, \ldots, Q+1$, and the symmetry of the log spectrum around the zero frequency $(k = 0)$ implies that $\tilde{S}(k)$ is an even sequence, i.e. $\tilde{S}(-k) = \tilde{S}(k)$ for $k = 1, \ldots, Q$.

In the inverse DFT (cepstral) domain, that circular convolution can be expressed as the product

$$C_H(m) = C(m)H(m), \quad m = -Q, \ldots, 0, \ldots, Q+1, \tag{4}$$

where $C_H(m)$, $C(m)$ and $H(m)$ are, respectively, the inverse DFT of $\tilde{F}(k)$, $\tilde{S}(k)$ and $\tilde{h}(k)$, $k = -Q, \ldots, 0, \ldots, Q+1$. Since, according to the usual definition of cepstrum (Oppenheim and Schafer, 1989), $C(m)$ is a real valued cepstral sequence, Eq. (4) expresses liftering, i.e. weighting in the cepstral domain by $H(m)$. Notice, however, that according to (3), this liftering is implemented as a convolution in the spectral domain.

Fig. 3 illustrates the computations involved in frequency filtering using as example a filter with impulse response $h(k) = \{1, 0, -1\}$, which transfer function is

$$H(z) = z - z^{-1}. \tag{5}$$

In matrix notation,

$$\boldsymbol{F} = \boldsymbol{HS}, \tag{6}$$

where $\boldsymbol{F}$ is the vector whose components are $F(k)$ in (3), $\boldsymbol{S}$ was defined in (1), and

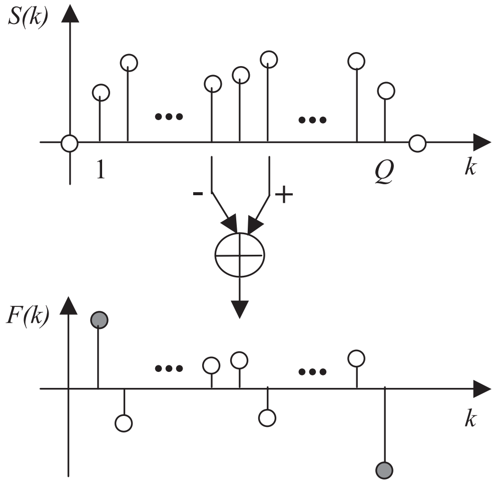


Fig. 3. Scheme of the FF computation with the filter $z - z^{-1}$.

$$\boldsymbol{H} = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & -1 & 0 \end{pmatrix}. \tag{7}$$

This filter, which will be proposed below as the usual one, is computationally simple, since for each band it only requires to subtract the log FBEs of the two adjacent bands. Note that, for this filter, the magnitudes of the shared endpoints in Fig. 3 are absolute energies. The FF technique will herewith be denoted FF2 when this second-order filter is employed.

The above filter has a zero at $z = 1$, i.e. it cancels the cepstral coefficient $c(0)$. When the filter does not possess this zero, the average value of the sequence $S(k)$ of (2) is removed before the filtering computations are performed (Nadeu et al., 1995). It is worth to note that the outputs $F(k)$, $k = 2, \dots, Q-1$, of such a derivative-type filter actually are spectral slope measures and, according to Klatt, a phonetic distance based on the spectral slope near the peaks correlates very well with perceptual data, unlike other speech characteristics such as the FBE values or the linear prediction residual (Klatt, 1982).

3.3. FF and decorrelation of FBEs

The first goal of frequency filtering is to decorrelate the parameter vector $S(k)$ like cepstral coefficients do. Assuming $E\{C_H(m)\} = 0$ for every m, it can be shown that, if the random variables $\tilde{F}(k) = F(k)$, $k = 1, \dots, Q$, are uncorrelated, the cepstral variance (*spectrum* of the sequence $\tilde{F}(k)$ $E\{|C_H(m)|^2\}$) is constant for $m = 1, \dots, Q$.

Thus, to aim at decorrelating the spectral sequence of FBEs $S(k)$, the filter $h(k)$ should be designed in such a way that its cepstral counterpart $H(m)$ equalizes the variance of cepstral coefficients $C(m)$ for $m = 1, \dots, Q$ (Nadeu et al., 1995). A first-order FIR filter that maximally equalizes the variance of cepstral coefficients can be easily obtained by least-squares modeling in the following way. Firstly, the variance is estimated by averaging over all the frames of a given database. Then, after performing a DFT, the quotient r between the values of the resulting sequence (the covariance of $S(k)$) at index 1 and index 0 is computed. Thus, the first-order FIR filter that maximally flattens the variance will be

$$H(z) = 1 - rz^{-1}. \tag{8}$$

Fig. 4 shows the estimated variance corresponding to the TI digits database (Nadeu et al., 1995) (decimated from 20 to 8 kHz and using $Q = 12$ mel-scale frequency bands) along with the inverse square magnitude of $H(m)$, that was computed following the above procedure. The resulting value of r is 0.5. The coefficients of the least-squares second-order FIR filter $z - a_1 - a_2 z^{-1}$ are $a_1 = 0.5$ and $a_2 = 0.05$, a fact that shows how a first-order filter already obtains an accurate modeling of the inverse variance. Note in Fig. 4 that the variance is zero for $m = 0$, since the average value of the sequence $S(k)$ of (3) has been removed.

3.4. FF and discriminative liftering

Decorrelation is a desired property of spectral features since diagonal covariance matrices are currently assumed in HMM recognition systems. Nevertheless, what is really relevant to the classification process is the discrimination capacity of

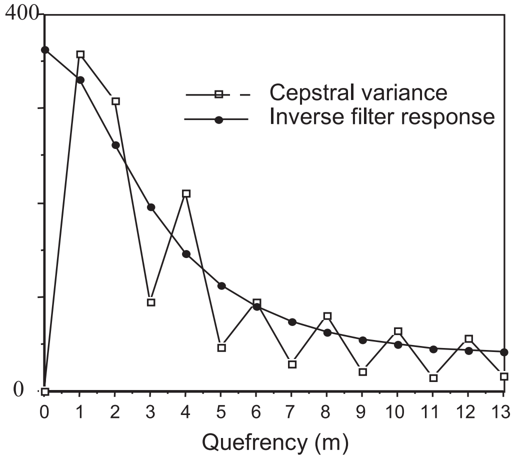


Fig. 4. Approximation of the TI digits estimated variance with the inverse square magnitude of the filter $1 - 0.5z^{-1}$.

those features. [1] Therefore, in this section, we are going to analyze the probability expressions involved in the HMM formalism, assuming that covariance matrices are diagonal and spectral parameters are log FBEs $S(k),\ k = 1, \ldots, Q$.

In continuous observation Gaussian density HMM (CDHMM), using one Gaussian with diagonal covariance matrix per state, the log probability that the given observation vector $\boldsymbol{S}$ in (1) has been generated by a given state q is

$$\log p(\boldsymbol{S}/q) = -\frac{1}{2}\sum_{k=1}^{Q}\log 2\pi\sigma^2(k) - \frac{1}{2}\sum_{k=1}^{Q}\left|\frac{S(k) - \mu(k)}{\sigma(k)}\right|^2, \tag{9}$$

where $\mu(k)$ and $\sigma^2(k)$ are, respectively, the mean and variance of the kth spectral parameter in the state q.

Note that, given the state q, the first term in (9) is constant, so thereby we will only consider the last term, which depends on the frequency sequence $S(k)$. To facilitate the reasoning, we will assume the same variance for all states (grand variance). In this way, the variance sequence is estimated over all the data. We will consider it as constant along k, i.e. $\sigma^2(k) = \sigma^2,\ k = 1, \ldots, Q$; this is reasonable, since a constant value can be obtained by a proper signal pre-emphasis.

In the following, we will express the last term in (9) in terms of the cepstral sequence $C(m)$ corresponding to $S(k)$. First, note that we can write it in terms of the even sequences $\tilde{S}(k)$ and $\tilde{\mu}(k)$, $k = -Q, \ldots, 0, \ldots, Q+1$, where $\tilde{\mu}(k)$ is formed as was $\tilde{S}(k)$ in Section 3.2. Since $\tilde{S}(0) = \tilde{S}(Q+1) = 0$ and $\tilde{S}(-k) = \tilde{S}(k) = S(k)$, the last term in (9) is proportional to

$$\sum_{k=-Q}^{Q+1}\left|\tilde{S}(k) - \tilde{\mu}(k)\right|^2. \tag{10}$$

Then, by applying the Parseval relation (Oppenheim and Schafer, 1989) it follows that that term is also proportional to

$$\sum_{m=-Q}^{Q+1}|C(m) - M(m)|^2, \tag{11}$$

where the even cepstral sequences $C(m)$ and $M(m)$, $m = -Q, \ldots, 0, \ldots, Q+1$, are, respectively, the inverse DFT of the even frequency sequences $\tilde{S}(k)$ and $\tilde{\mu}(k)$.

Expression (11) shows that, although the HMM framework uses observations lying in the spectral domain, the probability can be computed from the cepstral coefficients. Since $M(m)$ is also the mean of $C(m)$ in the state q, every cepstral coefficient $C(m)$ contributes in an additive way to the probability according to its square distance to the mean value in the state, exactly like $S(k)$ in (10). However, although the grand variance of the observations has been equalized in (10), it is not so in (11).

Consequently, if the cepstral coefficients had to contribute uniformly to the probability computation, they should be weighted in such a way that their variance was equalized. Interestingly enough, the same conclusion was reached in the last section by aiming at decorrelated FBEs.

However, an even contribution of all cepstral coefficients to the probability computation may not be well suited to recognition purposes since the various coefficients may show different levels of discrimination capacity. In particular, cepstral variance equalization may imply an excessive weighting of the high quefrency coefficients. Actually, these coefficients represent fast oscillations of the sequence of FBEs $S(k)$, which may be unreliable (Juang et al., 1987) and may not carry much useful information for phone discrimination, especially for a high number of bands Q. This problem could be avoided by choosing a small number of bands, but if Q is too small, there is not sufficient resolution in the spectral representation given by $S(k)$. By the way, notice that in the FF approach the number of bands Q is the number of transformed parameters as well, so it determines the size of the feature vectors that are delivered to the pattern matching step. A way of avoiding that constraint is to decimate the filtered sequence to retain a number of parameters M lower than Q; in this case, the filter should be designed to avoid aliasing.

[1] Herewith, we use the term discrimination in a general sense, as synonym of recognition.

Several reported works (Juang et al., 1987; Tokhura, 1987; Hanson and Wakita, 1987) have shown that a sort of inverse variance weighting of low quefrency components enhances the discrimination capacity of speech recognizers that are based on the Euclidean distance and use linear prediction cepstral coefficients (LPCC) as spectral parameters. As the variance of $C(m)$ shows a decreasing tilt along m (see Fig. 4), this kind of liftering deweights the low quefrency components. Additionally, those proposed lifters do not aim at equalizing the variance of the high quefrency components but they also deweight them. For those lifters, the non-deweighted middle quefrency components lie between $m = 6$ and $m = 8$ (for a 8 kHz sample frequency), indexes that correspond to oscillations of the spectral envelope that show between 3 and 4 peaks up to $\omega = \pi$ (4 kHz), which is the average *formant rate*.

The cepstral response $H(m)$ of the frequency filter $H(z) = z - z^{-1}$ has a sine shape, as depicted in Fig. 5, which is exactly like the lifter shape used in (Juang et al., 1987). Due to this shape, that simple data-independent filter deweights both low and high quefrencies, showing a rather good performance for a broad range of conditions (Nadeu et al., 1995), as we will see in the following sections. In particular, when Q lies between values 12 and 14, which have been found experimentally optimum for speech signals sampled at 8 kHz, $H(m)$ shows its maximum value between $m = 6$ and $m = 8$, exactly the same values that correspond to the *formant rate* mentioned in the above paragraph. Since for this particular filter the two endpoints of the filtered sequence actually are absolute energies, not differences, the full-band energy is usually neglected in the speech representation.

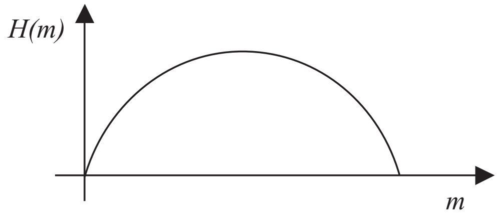


Fig. 5. Sine shape of the cepstral lifter corresponding to the frequency filter $z - z^{-1}$.

In summary, the best strategy for filtering the sequence of FBEs for HMM speech recognition using diagonal covariance matrices may not be aiming at a complete decorrelation, since the lifter shape that yields the best recognition performance probably will not coincide with the one that equalizes the variance of cepstral coefficients. Consequently, the filter has to be properly designed to balance decorrelation and discriminative liftering.

3.5. Recognition tests with static parameters

In this section, frequency-filtered parameters are empirically compared with cepstral coefficients for clean and noisy speech conditions, using the static features alone, i.e. with no addition of supplementary dynamic features, and training with clean speech. The adult portion of the TI English digit database will be used (Leonard, 1984). It consists of 112 speakers for training and 113 for testing. Each speaker utters two repetitions of the 11 single digits and 55 digit strings ranging from 2 to 7 digits.

A speech recognition system based on continuous observation Gaussian density hidden Markov models (CDHMM) (HTK software, Young et al., 1997) was used to carry out the tests. Each of the 11 left-to-right HMMs for the digit words consisted of eight effective states, and the silence model had three states. No skips were allowed and only one diagonal covariance Gaussian was employed per state.

3.5.1. Clean speech tests

First of all, let us report several digit recognition results for clean speech. Most of them were already presented in the FF introductory paper (Nadeu et al., 1995). After decimating the signals from 20 to 8 kHz sampling rate, and pre-emphasizing them with a zero at $z = 0.95$, Hamming windowed frames of 30 ms were taken every 10 ms. Then, Q mel-scale filter-bank energies were computed for each frame and logarithmically compressed. Each energy was obtained by a triangular weighting average (the one implemented in HTK) of the square magnitude of the DFT coefficients of the windowed frame in the given band. Unless otherwise stated, 12 bands ($Q=12$) are used for the FF parameters, since this value yielded the highest recognition scores in our tests. After tuning the system to MFCC, 20 frequency bands ($Q=20$) and 8 cepstral coefficients ($M=8$) were chosen as the empirically optimal parameters, though the differences with respect to $Q=20$ and $M=12$ were slight.

Using only single (isolated) digits for training and testing (according to the partition of the set of utterances between training and testing that is given with the TI database), MFCC recognition rate was 97.42% when the (non-normalized) frame energy was not included and 97.51% when it was. The result for FF with the database-independent second-order filter $z-z^{-1}$ in (5) was 98.11%. When both single and connected digit utterances were employed for training and testing, the recognition errors shown in Table 1 were obtained. Three different filters were tested: (1) the first-order least-squares variance-equalizer filter proposed in Section 3.3 (Equalizer 1); (2) the second-order least-squares variance-equalizer filter that also was already mentioned in that section (Equalizer 2); and (3) the second-order filter in (5).

Note, in Table 1, the significant improvement achieved by the frequency-filtered log FBEs with respect to conventional MFCC: 21.7% relative improvement in string error rate ($z-z^{-1}$ filter), and 28% in word error rate (first-order equalizer). The second-order equalizer yields almost exactly the same rates as those of the first-order filter, since the second coefficient is very small. The computationally inexpensive filter $z-z^{-1}$ yields the lowest string recognition error rate, although its word error rate is a little higher than those from the equalization filters, but still much lower than the MFCC one.

It is worth noting that the FF–FBE representation can improve its performance if the frequency filter is empirically optimized. Recognition percentage increased 1% using the filter $(1-0.7z^{-1})(1+0.3z)$. That filter attenuates low and high quefrencies less than the above filter $z-z^{-1}=(1-z^{-1})(1+z)$ so that it has a response closer to that of the first-order equalizer $1-0.5z^{-1}$.

When the frame energy is included as an additional parameter, the string recognition error rate for MFCC decreases down to 20.81%. Notice that the improvement is larger than for isolated digits. Conversely, the inclusion of the frame energy is not beneficial for the frequency-filtered FBEs since the error increases up to 19.45%. These results suggest that the FF–FBE parameterization somehow includes the energy information through both endpoints of the filtered frequency sequence, since their values actually are the log energies of the second and the next to last (with a minus sign) bands.

Applying principal component analysis (PCA) (or Karhunen–Loève transform) to the average-subtracted log FBEs in order to globally decorrelate them, 20.60% string error rate and 7.49% word error rate were obtained, scores worse than those of the frequency-filtered log FBEs. This result reinforces the discussion about decorrelation and discrimination presented in Section 3, since if only decorrelation were meaningful, PCA should obtain better results than any other transformation including FF.

The reasoning in Section 3.4 assumes only one Gaussian mixture per state. To validate the recognition improvement yielded by FF with respect to MFCC for a higher number of Gaussians, an experiment with eight Gaussians and using the first-order equalizer was performed. The percent-

Table 1
Percentage of connected digit recognition errors for MFCC and three different frequency-filtered parameter sets[a]

Error %	String	Word	Deletions	Substitutions	Insertions
MFCC	22.59	8.09	3.63	4.46	1.07
Equalizer 1	18.02	5.79	1.98	3.81	1.27
Equalizer 2	18.08	5.81	2.01	3.80	1.30
$z-z^{-1}$	17.69	6.00	1.97	4.03	0.91

[a] The error is measured with string error rate, word error rate, and the percentage of deletions, substitutions and insertions.

Table 2
Percentage of connected digit recognition errors for LPCC and two different frequency-filtered parameter sets

Error %	String	Word	Deletions	Substitutions	Insertions
LPCC	24.03	7.76	2.47	5.29	2.10
Equalizer 1	19.71	6.92	2.31	4.61	0.99
$z - z^{-1}$	19.67	6.90	2.21	4.60	1.08

ages of relative improvement were 16% for string recognition error and 25% for word error. Note that these relative improvement scores with respect to MFCC are not much lower than those from the one-Gaussian results from Table 1 (20% for string and 28% for word error).

Let us show a few experiments with the inverse hybrid LP-FBE approach mentioned in Section 2.1 (Nadeu et al., 1995). After computing 13 cepstral coefficients $C(m)$, $m = 0, \ldots, 12$, from a 10th-order LP analysis, they were transformed to the spectral domain using a 24-point DFT. The obtained 13 spectral values were considered as log FBEs, and thereby they were filtered as it was done with the above mel-scaled DFT-based FBEs. The value of the zero of the optimal first-order equalizer is, in this case, 0.53, quite similar to that of the previous DFT-based case. Table 2 shows the recognition results for the conventional LPCCs and two frequency-filtered LP-based log FBE. These results are not so good as those from Table 1, but FF improves again the recognition performance with respect to cepstrum.

3.5.2. Noisy speech tests

The TI single digit database has also been used to carry out experiments with additive artificial noise and SNR = 10 dB. To set this SNR value, the mean power of each utterance was computed only in the speech portion (the speech signal had been manually endpointed) and the noise power was chosen to obtain the specified SNR in that speech portion. Training is performed again with clean speech. The recognition setup is like the one used for the above clean speech tests, except that preemphasis is not performed in this case and the frame energy is not included.

First of all, results with white noise will be presented. Four techniques have been compared using the same number of (static) parameters for each one (12 parameters):

1. MFCC with $Q = 20$ bands and $M = 12$ cepstral coefficients per frame. $M = 12$ was found to be preferable to $M = 8$ for noisy speech.
2. FF with $Q = 12$ bands and the second-order difference filter in (5) (FF2).
3. A modification of FF2 that discards $F(Q)$, a parameter whose magnitude is the absolute energy of the 12th band. In this case, $Q = 13$ was used, but the number of spectral parameters $F(k)$ is still 12.
4. FF with $Q = 12$ bands and the first-order difference filter $1 - z^{-1}$ (FF1). Note that $F(Q)$ in this case is not an absolute energy.

Note in Table 3 that FF2 yields a higher recognition rate than MFCC for noisy speech, but it performs even better if $F(Q)$ is discarded, since this high-frequency energy shows a relatively low SNR. However, when $F(Q)$ is discarded, the large improvement that FF obtains for clean speech with respect to MFCC is substantially reduced. Additionally, the first-order difference filter improves even more the recognition rate for 10 dB noisy speech. However, its performance for clean speech is worse.

Also, by comparing the results in Table 3 for clean speech tests with the ones reported at the beginning of the last section, we notice that while

Table 3
Single digit recognition performance corresponding to four parameterizations: MFCC, the usual FF2 technique, the modified FF2 that discards $F(Q)$, and FF1

Recognition %	Clean	10 dB
MFCC	96.70	28.53
$z - z^{-1}$	98.03	41.85
$z - z^{-1}$ modified	97.26	50.34
$1 - z^{-1}$	96.82	57.59

MFCC performs significantly better when preemphasis is carried out, the improvement for FF is much smaller.

It is worth noting a few results with an artificially generated low-pass noise with 10 dB SNR and cut-off frequency 1.1 kHz. Actually, all the frequency filtering techniques considered in Table 3 give lower scores than MFCC for this kind of noise when only static parameters are used, e.g. whereas MFCC yields 24.59% recognition rate, the modified FF2 (the third technique in Table 3) gets 17.06%. A similar observation was already reported in (Hernando and Nadeu, 1997b) for real low-pass noise. There are two possible reasons for the degradation of FF results with noises that are concentrated in a low-pass frequency band: (1) the first frequency-filtered parameter is the absolute energy of the second band, so it is strongly corrupted by noise; and (2) a few FF parameters are affected by the step of the transition band of the noise spectrum. However, as will be discussed in Section 5, the noise stationarity will allow the time filters to attenuate these effects and, therefore, FF will obtain better recognition performance than MFCC.

Similar conclusions about the performance of FF with respect to MFCCs for speech corrupted by white noise have been recently reported by Paliwal (1999) using the same basic FF idea. In that work, he also arrives to the conclusion that the filter in (5) is a good option. However, he does not include the end parameters $F(1)$ and $F(Q)$. That is probably the reason why his results for clean speech are not as good as those presented here.

3.5.3. Conclusion

Other recognition experiments have been performed in our laboratory during the last years to assess the FF technique: for different speech recognition tasks (acoustic–phonetic decoding (Nadeu et al., 1995), and word spotting with phone units (Nadeu et al., 1996)), different noise conditions (Hernando and Nadeu, 1997b), speaker recognition (Hernando and Nadeu, 1997a), and also using features that were not obtained from an usual filter-bank but from LP modeling (Hernando and Nadeu, 1997b). More results are also presented in the following sections by using dynamic feature sets along with the static one. From the whole set of tests, it appears that FF generally offers better recognition performance than MFCC.

Summarizing, we can conclude that frequency filtering is a simple and effective operation that performs a combination of decorrelation and liftering, while still maintaining the speech parameters in the frequency domain, so avoiding the disadvantages of cepstral coefficients that were listed in Section 3.1. Note in particular that FF coefficients may be especially useful whenever their frequency localization property is convenient. For instance, to use them in a missing feature paradigm, like it is done in (de Veth et al., 1999).

3.6. Alternative combination of FF and non-linearity

In this section, we are going to break the clear separation we have maintained above between the two main parameterization blocks of the scheme in Fig. 1: spectral measurements and linear transformations. Since the frequency-filtered spectral parameters lie in the frequency domain, we can speculate about placing FF before the non-linear operator or at least performing both operations jointly. In fact, if FF was applied to the linear spectral energies E_k instead of the non-linearly compressed ones, it might better attenuate an additive white noise component. To illustrate it, let us assume that the filtered sequence of log spectral energies of (3),

$$F(k) = (\log E_k) * h(k), \tag{12}$$

is substituted by

$$G(k) = \frac{E_k * h(k)}{E_k}. \tag{13}$$

Notice that if $h(k)$ were a differentiator,

$$G(k) \underset{Q\to\infty}{\longrightarrow} F(k), \tag{14}$$

where $Q \to \infty$ means that the frequency variable becomes continuous. Note that, for the second-order filter $H(z) = z - z^{-1}$, $G(k)$ in (13) are relative spectral differences, and, although a white noise component can be removed from the numerator, its influence remains in the denominator.

A few speech recognition tests were performed with $G(k)$ and the filter $H(z) = z - z^{-1}$, for both clean speech and speech contaminated with white noise, using the TI isolated digits database and the same recognition setup as in Section 3.5.2. As can be expected, the recognition rate decreased for clean speech with respect to the FF parameters $F(k)$, from 98.03% to 97.14%. However, it noticeably increased for 10 dB noisy speech from 41.85% to 66.32%.

4. Temporal filtering

The pattern-matching formalism based on HMM assumes that each acoustic observation vector is uncorrelated with its temporal neighbors. This assumption cannot be fulfilled by the transformed vectors for the usual frame shifts (typically, 10 ms). That has been the reason to justify the inclusion of smoothed time derivatives as additional parameter vectors (they are also referred to as "dynamic" features (Furui, 1986)). Thus, not only the first-order differential parameter vector but often the second-order one are appended to the basic "static" vector (for the sake of simplicity of the explanation, we will assume in the following that the global energy, if used, and its differences are already included in the parameter vectors). These two new temporal sequences of differential vectors are computed by filtering the basic time sequence of spectral parameter vectors.

Filtering of each time sequence of spectral parameters (TSSP) has also been used for robust speech recognition with another goal: to remove its dc and slowly variant components when they are carrying undesired perturbations as linear distortion (convolutional noise, additive in the log spectral domain). That is the aim of both the cepstral mean subtraction (CMS) technique (Rosenberg et al., 1994) and the IIR filter with a pole close to 1 that is used in the so-called RASTA processing (Hermansky and Morgan, 1994).

4.1. Modulation spectrum analysis

The effect of temporal filtering (TF) can be better understood in the frequency domain. The frequency counterpart of the frame index n is the modulation frequency θ (Houtgast and Steeneken, 1985). For this reason, the TSSP spectrum has been called modulation spectrum (MS) (Greenberg and Kingsbury, 1997). From the analysis of the MS of filtered TSSP, it can be concluded that (Nadeu et al., 1994, 1997a):

1. Each dynamic TSSP emphasizes a given band of meaningful modulation frequencies. This effect is achieved with an approximate equalization of the static MS in that band.
2. The modulation frequency bands of the various TSSP (static and dynamic) are distributed along an interval of the modulation frequency axis in such a way that the function that results from adding their MS is rather flat in that interval, which is phonetically relevant and does not carry an excessive spectral estimation noise.
3. If a single dynamic vector is employed to replace (not to supplement) the static vector, a very large single digit recognition improvement was achieved by enhancing the 3–4 Hz band, which roughly corresponds to the syllable rate.

It was conjectured from these observations that the 3–4 Hz band is important for speech recognition as it is for speech intelligibility (e.g. the intelligibility study in (Arai et al., 1996)). And also that that strong improvement was possible due to the relatively low temporal spreading of unit boundaries caused by the filter when: (1) the lengths of the modeled units are large, and (2) the units to recognize appear in isolation.

Note that the assertion in point 3 contradicts the classical statement that, for clean speech, dynamic features do not perform well on their own. In fact, if only one parameter vector is employed, a dynamic feature vector may yield higher recognition rates than the static vector for clean speech, provided that the temporal filter is properly designed. This was observed in (Nadeu et al., 1997a) with both whole-word models (isolated and connected digits) and context-dependent subword models.

4.2. Temporal filters for robust speech recognition

It is not a fact under discussion that time-filtered features are less affected by convolutional noises than the static features, so that they can help the recognition system to cope with mismatches between training and testing data. However, that is not so clear for additive noises (Hanson et al., 1996). In the following, some recognition results will be presented attempting to gain an insight into the effect of time filtering on the robustness of the speech representation for real additive noises. Moreover, the role of the modulation spectrum to guide the filter design will be illustrated.

The recognition tests were performed with the TI single digit database and the same experimental setup as in Section 3.5.2, except that, for FF2, $Q = 13$. Moreover, the modified technique in Section 3.5.2 that discards the endpoint $F(Q)$ (third row of Table 3) was employed for tests with white noise. Also, a few tests were carried out applying to the FBEs the root compression with $\gamma = 0.03125$ instead of the logarithm (see Section 2.3) before performing FF.

Additionally, tests with the MFCC three-feature-set front-end standard for clean speech (ETSI SQL W1007) were also performed to have reference scores. This standard front-end uses $Q = 23$ bands and $M = 12$ cepstral coefficients. Unlike the FF front-ends used in these experiments, the standard one uses pre-emphasis filter $1 - 0.97z^{-1}$, DFT magnitude instead of square magnitude, and 25 ms window length instead of 30 ms. Additionally, the standard employs the usual orthogonal polynomial with length 7 to compute the derivative parameters, der(7), and the same kind of filter with length 5 to compute the acceleration parameters, acc(11), from the derivatives ones (so the length of the composite filter is 11).

Slepian FIR filters are used in combination with FF to have more flexibility in the filter design procedure. The following Slepian filters have been used in cascade with the equalization filter $H(z) = 1 - 0.97z^{-1}$ (see (Nadeu et al., 1997a) for the notation and the design procedure): TF1(15): $k = 1$, $W = 12$, $L = 14$; TF2(15): $k = 2$, $W = 12$, $L = 14$; TF1(8): $k = 1$, $W = 16$, $L = 7$. The number in the parenthesis is the length of the whole filter. Figs. 6(a) and (b) illustrate the simulated modulation spectra of time-filtered speech parameters, which were obtained by using the spectral response of the filter $1/(1 - 0.97z^{-1})$ as an approximation of the mean modulation spectrum (Nadeu and Juang, 1994; Nadeu et al., 1997a).

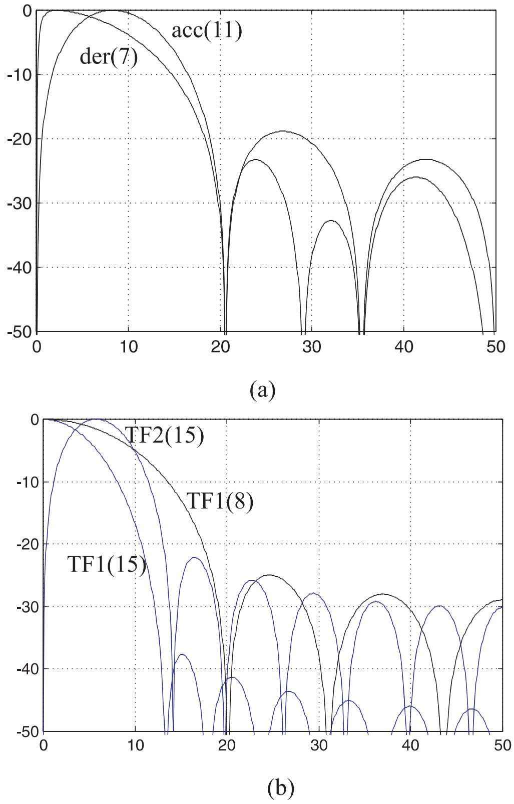


Fig. 6. Simulated modulation spectra of time-filtered feature sets for: (a) del(7) and acc(11) filters; (b) TF1(15), TF2(15) and TF1(8) filters.

Training was carried out with clean speech. For tests with noise, speech was contaminated by pub, railway station and white noises with SNR = 10 dB (both real noises were extracted from the SUN-ROM-1 noise database [2]). Fig. 7 shows recognition percentages for FF2, using the static features alone and also using one and two time-filtered feature sets. Additionally, it includes results for FF and two features sets, using the root non-linearity instead of the logarithm; and, finally, the results corresponding to the above-mentioned standard MFCC parameterization are also shown.

[2] Produced in the European Esprit-2094 project.

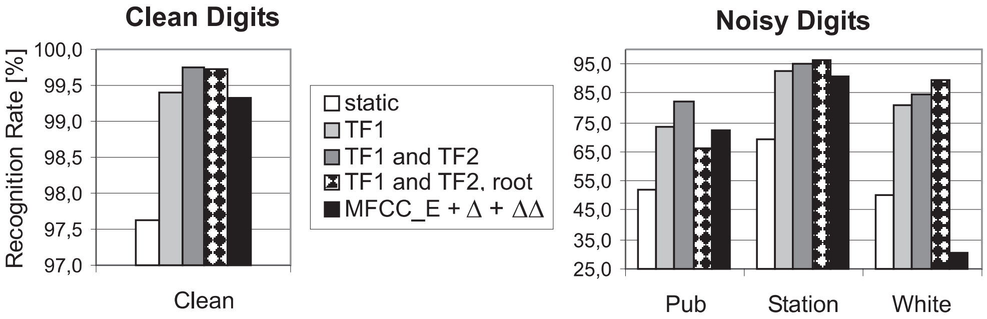


Fig. 7. Isolated digit recognition rate for FF2 using: (1) the static feature set, (2) one time-filtered feature set (TF1), (3) two time-filtered feature sets (TF1 and TF2), and (4) the root non-linearity instead of the logarithm with two features sets. Also, results from the standard three-feature set MFCC front-end are included as reference.

By comparing the first two bars of the various noise conditions (static and TF1 features), it is apparent that the time filter TF1 improves significantly the recognition rate for both clean and noisy speech. This seems to indicate that dynamic features can be less affected than static features by additive noises provided that the filter is properly designed. Furthermore, using two time-filtered feature sets instead of one, the FF2 results improve.

Notice that FF2 with only one (dynamic) feature set even yields a higher recognition rate than the standard parameterization with three MFCC feature sets for both clean and noisy speech. However, if CMS is employed in the standard front-end, its results increase noticeably, especially for noisy speech, but still remain below the FF2 results with two feature sets. For example, for white noise, which is the noise condition that undergoes the largest improvement, the recognition percentage increases from 30.78% to 64.83%.

Using the empirically optimized exponent value in the root compression, good results have been obtained for the station and white noises, and bad results in the case of pub noise, which is speech-like. Actually, the root compression leads to spectra with more dynamic range than log spectra at peaks, so it may enhance the high-power narrow-band components of the pub noise.

When the temporal filter TF1(8) is used instead of TF1(15), the recognition rate does not change for clean speech and diminish for the station and white noises, but it noticeably increases for pub noise (from 73.76% to 84.67%). In Fig. 6(b), it can be observed that the modulation spectrum from the filter TF1(8) shows a broader band than the one from the filter TF1(15). Moreover, both filters TF1(15) and TF2(15) together cover a modulation frequency band similar to the TF1(8) one. Apparently, that inclusion of higher modulation frequencies with respect to the filter TF1(15) alone accounts for the better performance with pub noise since: (1) when using both TF1(15) and TF2(15) feature sets, the recognition rate of pub noisy speech also increases significantly up to 82.01%, as shown in Fig. 7; and (2) a test with two FF feature sets using del(7) and acc(11) temporal filters, which show a broader band than TF1(15) and TF2(15) (see Fig. 6(a)), yields even a higher recognition rate for pub noise, 85.15%.

5. Tiffing (time and frequency filtering)

Let us consider the two-dimensional (2-D) sequence of log FBEs $S(k, n)$, where the index k denotes the frequency band and the index n denotes the time frame. In the above sections, we have presented filtering as being separately performed in the dimensions k and n. However, the effect of filtering is remarkedly similar in both dimensions. In fact, time and frequency filters show similar characteristics since both perform a kind of smoothed derivative. Concretely, both the frequency filter and the time filters used in this work can be viewed as the combination of three operations (Nadeu et al., 1996): (1) removal (or severe attenuation) of the average value; (2) approximate variance or power equalization in the transform domain (quefrency for k, or modulation frequency for n) with a first-order high-pass FIR filter; and (3) smoothing of the resulting sequence with a low-pass filter that shapes the (equalized) band. Additionally, the effects of both kinds of filters are not orthogonal; for example, the dc component of the 2-D time-frequency sequence $S(k, n)$ may be removed by both filters.

On the other hand, frequency-filtered log FBEs seem more able to benefit from temporal filtering than cepstral coefficients. Actually, in the tested cases where MFCC outperforms FF with only static parameters, the use of dynamic parameters reverse the comparison result. There are two sets of experiments that account for that. In Section 3.5.2, for low-pass noise, whereas MFCC yielded 24.59% recognition rate, the modified FF2 technique obtained 17.06%. However, using the temporal filter TF1(15) defined in Section 4.2, the scores for the one feature set are 75.94% for MFCC and 77.14% for the modified FF2 technique. The other set of experiments will be reported in Section 5.4.

These observations lead us to think that there is something of a synergy effect between both types of filtering operations. Consequently, in this section we are going to consider both types of filters together as applied to a 2-D frequency-time sequence. Therefore, the 2-D modulation spectrum (2D-MS) (Macho and Nadeu, 1998), can be helpful for designing and analyzing them.

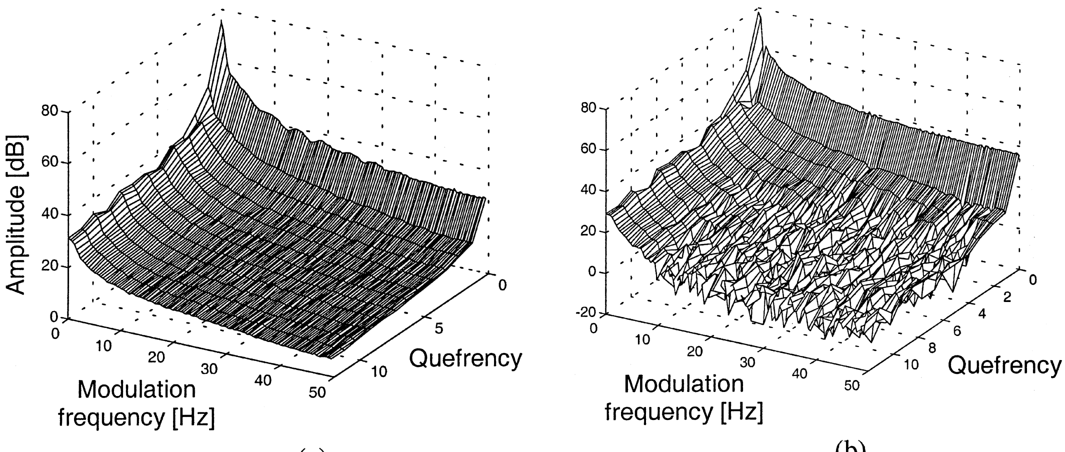
5.1. The two-dimensional modulation spectrum (2D-MS)

The 2D-MS $T(m, \theta)$ is estimated from the 2-D sequence of log FBEs $S(k, n)$ by computing and averaging over a speech database the power 2-D transform function $|C(m,\theta)|^2$, which is computed as

$$S(k,n)\xrightarrow{\mathrm{IDFT}_k}c(m,n)\xrightarrow{\mathrm{FT}_n}C(m,\theta)\xrightarrow{|.|^2}|C(m,\theta)|^2, \qquad (15)$$

i.e. the function $C(m, \theta)$ is obtained with an inverse DFT to translate $S(k, n)$ from the frequency domain k to the quefrency domain m followed by a Fourier transform of the resulting sequence $c(m, n)$ from the time domain n to the modulation frequency domain θ. Fig. 8(a) shows the 2D-MS obtained from the isolated TIdigits database.

The 2D-MS can be used to investigate which are the most important quefrency-modulation frequency (QMF) regions for speech recognition, and also to observe the mismatch between clean



(a) (b)

Fig. 8. (a) 2-D modulation spectrum computed over the whole adult portion of the TI isolated digit database; (b) mismatch in the 2-D modulation spectrum between clean and noisy speech for SNR = 10 dB.

speech and speech contaminated by noise. The mismatch is defined as the difference

$$\left|C_{\text{noisy}}(m,\theta) - C_{\text{clean}}(m,\theta)\right|^2 \quad \text{for all } m \text{ and } \theta. \tag{16}$$

Note in Fig. 8(b) that, for additive white noise, the largest mismatch between the 2D-MS of clean and noisy speech is located at the low QMF region.

5.2. 2D-MS-assisted design of the time and frequency filters for robust speech recognition

The 2-D modulation spectrum can be used to guide the design of time and frequency filters. As observed in Fig. 8(a), $T(m, \theta)$ decreases along both axes so the highest energy is located at the low QMF region. This region represents the slowest oscillations of the 2-D sequence $S(k, n)$, i.e. the spectral tilt (in k) and the long-term time changes (in n). However, the most discriminative information probably lies in the alternation of peaks (formants) and valleys in the spectral domain, and in the alternation of stationary and transitional segments of speech in the time domain, which are represented by higher QMF components.

For noisy speech, there is an additional reason to select higher QMF components: the largest mismatch is situated at low QMF, being its maximum at the (0, 0) point. Using time and frequency filtering, we aim to balance the enhancement of discriminatively important 2-D regions with the attenuation of 2-D regions that are strongly affected by noise. Each feature set can emphasize its own QMF region so that a different time filter and a different frequency filter are employed for each region.

A 2D-MS region with potentially high discrimination capacity was investigated in (Macho et al., 1999b). The quotient between the between-class and the within-class variances of the 2-D modulation spectrum, that was used for that purpose, achieved high values around $\theta = 3$–4 Hz and $m = 3$–4 for the TI isolated digits database (see Fig. 9).

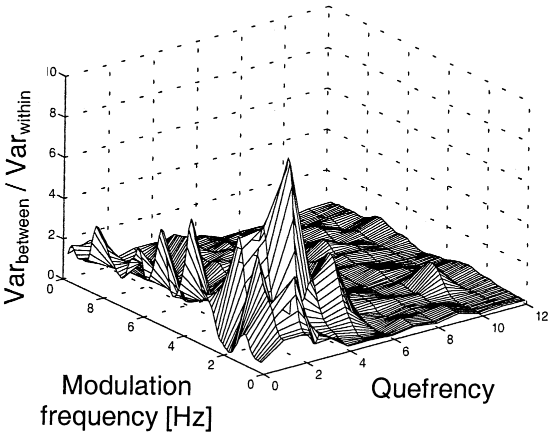


Fig. 9. The quotient Variance$_{\text{between}}$/Variance$_{\text{within}}$ for isolated digits.

Several recognition tests were performed for the TI single digits database and 10 dB white noise. The experimental set-up was like the one in Section 4.2. Three feature sets, obtained with different time filters, were tested: static (no filter), TF1-filtered and TF2-filtered. TF1 and TF2 are the same two time filters of length 15 defined in Section 4.2. The approximated modulation frequency bands where the time-filtered parameters show maximum energy are, respectively, 0–3 Hz and 2–9 Hz. Two different frequency filters defined in Section 3.5.2 were employed: FF1 $(1 - z^{-1})$ and FF2 $(z - z^{-1})$. The latter was used with $Q = 13$ and discarding $F(Q)$, like in Section 4.2. The former is a first-order filter, so it does not include the high-frequency absolute energy, only the low-frequency one: $F(1)$.

As can be observed from the results shown in Fig. 10 for noisy digits, the preferred frequency filter for static and TF1-filtered parameters is FF1, the one that attenuates low quefrencies. However, for TF2-filtered parameters, which have a band located at higher modulation frequencies, the frequency filter that yielded the best rate was FF2, which emphasizes middle quefrencies. Thus, as shown in the bottom part of Fig. 10, the best results are achieved using a different frequency filter for each time filter (black bar).

5.3. Tiffing versus cepstral-time matrices

A 2-D cepstrum representation can be computed by applying a 2-D discrete Fourier transform to a spectral-time matrix (Pai and Wang,

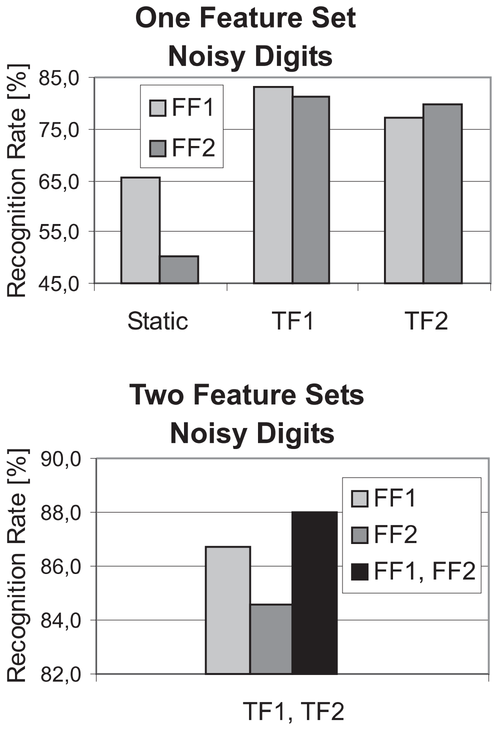


Fig. 10. Recognition rates for clean and noisy (SNR = 10 dB) digits for different time and frequency filters. The black bar corresponds to the pair of time-frequency filters TF1–FF1 and TF2–FF2.

1992). The cepstral-time matrix (CTM) representation (Vaseghi et al., 1993) is a running 2-D cepstrum where the spectral-time matrix is built with L parameter vectors around the current one. So the CTM $C_{\mathrm{CTM}}(m, i)$ is computed by applying a 2-D DCT to L stacked adjacent log FBE vectors. Since the 2-D DCT can be decomposed in two 1-D DCTs, the CTM computation can be performed in the following way:

$$S(k,n) \overset{\mathrm{DCT}_k}{\rightarrow} c_{\mathrm{DCT}}(m,n) \overset{\mathrm{DCT}_n}{\rightarrow} C_{\mathrm{CTM}}(m,i). \qquad (17)$$

Thus, by applying the first DCT to the frequency index k, a 2-D time sequence of cepstral coefficients is obtained. The second DCT transforms a time sequence of L stacked cepstral vectors to the modulation frequency domain. A band in the modulation frequency domain can be associated to each DCT basis sequence, with a central frequency that depends upon both the number of cepstral vectors L and the frame rate. Consequently, a modulation frequency can be assigned to each column of a CTM. Analogously, each row of a CTM corresponds to a different value of quefrency index m.

Typically, only a low-indexed sub-matrix of CTM is used for recognition. In this way, it is an alternative to tiffing for selecting a QMF region. Actually, CTMs can, similarly to tiffing, benefit from the conclusions that can be drawn from the analysis performed on the 2D-MS.

Recently, both tiffing parameters and CTMs were comparatively tested for the TI single digit database (Macho et al., 1999b). It is worth noting that CTM works particularly well with a small number of parameters for this task and clean speech. Although the differences in recognition rate between both approaches were not large, tiffing parameters showed a consistently better performance for both clean speech and speech contaminated with real additive noises.

5.4. Recognition tests with the Aurora database and recognition setup

To conclude this section, let us mention a very recent result corresponding to the clean and noisy connected TI digit recognition task proposed by the Aurora project to standardize a robust front-end for distributed speech recognition (ETSI STQ WI008) (Pearce, 1998). Four different types of noises are used in the recognition task: hall (Noise 1), babble (Noise 2), train (Noise 3) and car (Noise 4). Instead of training from clean speech, like in all the results reported above, training is performed in this task in a multicondition way (i.e. the training corpus contains both clean and noisy speech signals, for various noise conditions). The set of test utterances corresponding to a given type of noise is tested with both clean speech and six different SNRs (20, 15, 10, 5, 0 and −5 dB), but noise conditions 0 and −5 dB are not used for training. An average word recognition accuracy for noisy speech is computed by considering all the test conditions for noisy speech except −5 dB.

In the HTK (Young et al., 1997) recognition back-end established by the Aurora project that is used in our tests, digit HMMs have 16 states and 3

Gaussians per state. A short pause model is used in addition to the silence model. The MFCC recognition system is the standard front-end for clean speech (ETSI SQL W1007) already presented in Section 4.2. The tiffing front-end differs from it in the three pre-processing steps already mentioned in that section. FF2 was used with $Q = 13$, so both parameterizations use the same number of parameters (MFCC includes 12 cepstral coefficients plus energy).

Table 4 shows the average word recognition accuracy scores for noisy speech tests and clean speech tests. First of all, FF was tested along with MFCC by using static features alone, i.e. with no addition of supplementary dynamic features. The average accuracy score for noisy conditions for MFCC is better than for FF. However, when three feature sets are used for both MFCC and FF, and the same time filters der(7) and acc(11) of the standard parameterization, which were presented in Section 4.2, are also used for both MFCC and FF, the average accuracy score is higher for FF (third and fourth rows in Table 4). At the beginning of Section 5, it was already noted that static FF parameters alone do not perform so well as MFCC ones for low-pass noise, but the opposite occurs also when dynamic parameters are included.

The last row in Table 4 shows the average speech recognition accuracy results for the tiffing parameters, i.e. when the time filters are also changed. For that purpose, we used two Slepian-type filters that had been empirically optimized to recognize TI connected digits with models trained with clean digit utterances (Macho et al., 1999a). They are Slepian-type filters: the filter TF1(8) already presented in Section 4.2, and the filter TF2(10) whose specifying parameters are $k = 2$, $W = 16$ and $L = 9$.

Table 4
Average performances for the TI connected digit Aurora task for DSR standardization

Recognition accuracy	Average clean	Average noisy
MFCC – static set	95.39	76.24
FF – static set	95.54	74.88
MFCC – 3 feature sets	98.57	86.34
FF – 3 feature sets	98.91	87.07
Tiffing – 3 feature sets	98.94	87.68

The relative improvements of average accuracy rates are meaningful: 25.8% and 9.8% for clean and noisy recognition, respectively, in comparison to the standard front-end. The detailed results for both the standard parameterization and the tiffing one are presented in Table 5. As specified by the Aurora project, the average score in the right bottom corner of the table is computed over all types of noise but only for SNR conditions from 20 dB to 0 dB. Notice that tiffing outperforms in average terms the usual mel-cepstrum representation for every kind of noise and SNR. Furthermore, tiffing performs significantly better than the MFCC standard front-end when the tested conditions are not seen during training (i.e., for the SNRs equal to 0 dB and −5 dB) for noises 1, 3 and 4 in Table 5. However, for the speech-like noise (noise 2), tiffing works better in matched conditions.

5.5. Conclusion: advantage of time and frequency filtering

When extracting conclusions about the recognition performance of spectral parameter post-processing techniques, we have to carefully distinguish between the different recognition tasks and the various types of modeled phonetic units. In this paper, results have mostly been reported from experiments with clean and noisy speech for: (1) isolated digit recognition, using whole word models and clean speech training; and (2) connected digit recognition, using whole word models and for either clean speech training or multicondition training (Aurora task).

In our experimental work, the tiffing technique has rarely yielded worse results than a standard and tuned MFCC parameterization with dynamic parameters, neither for clean nor for noisy speech tests. However, large vocabulary continuous speech recognition tests should also be performed to assess the tiffing technique in a more complete way. In a recognition task like that, in which subword units are involved, the time filters must be shorter, so that there exists less freedom to design them than using whole-word models.

Table 5
Recognition accuracy for the Aurora task

Recognition accuracy	Noise 1	Noise 2	Noise 3	Noise 4	Average
(a) Standard MFCC					
Clean	98.65	98.49	98.48	98.64	98.57
20 dB	97.60	96.67	98.03	98.43	97.68
15 dB	96.16	93.80	97.26	98.12	96.34
10 dB	92.85	86.40	94.78	97.59	92.91
5 dB	83.11	70.25	88.22	94.72	84.08
0 dB	47.31	49.27	66.51	79.67	60.69
−5 dB	18.61	31.68	30.63	47.24	32.04
Average	83.41	79.28	88.96	93.71	86.34
(b) Tiffing					
Clean	98.93	98.88	98.72	99.23	98.94
20 dB	97.57	96.89	97.85	98.89	97.80
15 dB	95.95	95.22	96.96	98.21	96.59
10 dB	92.08	89.30	94.75	97.28	93.35
5 dB	82.78	74.09	87.35	93.80	84.51
0 dB	60.73	48.79	71.67	83.52	66.18
−5 dB	27.36	26.03	37.73	53.84	36.24
Average	85.82	80.86	89.72	94.34	87.68

In this work, filter design has been based on an experimental approach; statistically optimal designs can be alternatively pursued, either based on a linear discriminant analysis approach (Avendaño et al., 1996; Hermansky, 1998) or on a maximum likelihood approach (Pachès-Leal et al., 1999).

6. Optimal transformations of the whole set of features: PCA and LDA

Dynamic parameter vectors are correlated between them and with the static vector (Ljolje, 1994). Therefore, the whole set of parameters may still be linearly transformed to either reduce the number of features or to improve the recognition performance. That linear transformation has also been applied so far to a set of several adjacent static vectors.

Principal component analysis (PCA) (or the KL transform) (Fukunaga, 1990) uses a data-dependent matrix to decorrelate any of the two sets and sort the transformed parameters in terms of variance, so allowing a dimensionality reduction of the final feature vector.

Alternatively, we may apply a linear transformation that provides an increased class separability. LDA (Fukunaga, 1990) has been used so far for this purpose (Hunt and Lefèbvre, 1989). Most of the reported works that propose an LDA technique consider HMM states as classes, despite that the classes should ideally be linguistic units. LDA assumes that the data in each class can be modeled by a single Gaussian distribution that shares its covariance matrix with all the other classes. Therefore, the LDA transformation matrix has to be calculated carefully, and it should be retrained for every new acoustic modeling.

Those two linear transformations were compared along with MFCC and frequency-filtered log FBEs (Batlle et al., 1998). Using an acoustic–phonetic decoding task on the TIMIT database, it was observed that PCA and LDA can improve the recognition rate and reduce the number of required features. Both transformations showed their potential better when applied to the whole set of parameters, i.e. including the dynamic ones; in fact, using only the current static vector, FF and LDA yielded a similar recognition score. Additionally, it was also observed that LDA and FF outperformed the others for the same number of final features. Note that FF, which was applied with the data-independent filter $z - z^{-1}$, does not require either a matrix training nor a class definition.

7. Conclusions

In this paper, the linear transformations of nonlinearly compressed spectral band energies that implement current speech recognition systems have been reviewed and discussed, and a few recent contributions related to them have been presented.

According to what has been reported in this paper, frequency-filtered FBEs are a robust speech representation which, unlike cepstral coefficients, maintains the frequency meaning, so being a real alternative to them. Also, we have observed how the recognition performance can benefit from an appropriate time filter design, and that an additional advantage can be obtained from jointly considering time and frequency filters.

Acknowledgements

The authors would like to thank E. Batlle, F. Galindo, J. Marí, J.B. Mariño, P. Pachès and J. Padrell for their helpful comments and discussions. And also to express their gratitude to J. Marí for his cooperation to obtain the results mentioned in Section 3.6. This work has been supported by CICYT, under projects TIC98-0683 and TIC98-0423-C06-01.

References

Akansu, A.N., Haddad, R.A., 1992. Multirresolution Signal Decomposition. Academic Press, New York.

Alexandre, P., Lockwood, P., 1993. Root cepstral analysis: A unified view. Application to speech processing in car noise environments. Speech Communication 12 (3), 277–288.

Arai, T., Pavel, M., Hermansky, H., Avendaño, C., 1996. Intelligibility of speech with filtered time trajectories of spectral envelopes. In: Proc. ICSLP, pp. 2490–2493.

Atal, B., 1983. Efficient coding of LPC parameters by temporal decomposition. In: Proc. ICASSP, pp. 81–84.

Avendaño, C., vanVuuren, S., Hermansky, H., 1996. Data based filter design for RASTA-like channel normalization in ASR. In: Proc. ICSLP, pp. 2087–2090.

Batlle, E., Nadeu, C., Fonollosa, J.A.R., 1998. Feature decorrelation methods in speech recognition. A comparative study. In: Proc. ICSLP, Vol. 3, pp. 951–954.

Davis, S.B., Mermelstein, P., 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoust. Speech Signal Process. ASSP-28 (4), 357–366.

de Veth, J., De Wet, F., Cranen, B., Boves, L., 1999. Missing features theory in ASR: make sure you miss the right type of features. In: Proceedings of Workshop on Robust Methods for Speech Recognition in Adverse Conditions, pp. 231–234.

Deller, J.R., Proakis, J.G., Hansen, J.H.L., 1993. Discrete-time Processing of Speech Signals. Macmillan, New York.

ETSI SQL W1007, http://webapp.etsi.org/WorkProgram/Report_WorkItem.asp? WKI_ID = 6400.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition. Academic Press, New York.

Furui, S., 1986. Speaker-independent isolated word recognition using dynamic features of speech spectrum. IEEE Trans. ASSP 34 (1), 52–59.

Greenberg, S., Kingsbury, B.E.D., 1997. The modulation spectrogram: in pursuit of an invariant representation of speech. In: Proc. ICASSP, Vol. 3, pp. 1647–1650.

Hanson, B.A., Wakita, H., 1987. Spectral slope distance measures with linear prediction analysis for word recognition in noise. IEEE Trans. Acoust. Speech Signal Process. ASSP-35 (7), 968–973.

Hanson, B.A., Applebaum, T.H., Junqua, J.C., 1996. Spectral dynamics for speech recognition under adverse conditions. In: Lee, C.H., Soong, F.K. (Eds.), Advanced Topics in Automatic Speech and Speaker Recognition. Kluwer Academic Publishers, Dordrecht.

Hermansky, H., 1990. Perceptual linear predictive (PLP) analysis of speech. J. Acoust. Soc. Amer. 87 (4), 1738–1752.

Hermansky, H., 1998. Should recognizers have ears? Speech Communication 25, 3–27.

Hermansky, H., Morgan, N., 1994. RASTA processing of speech. IEEE Trans. SAP 2 (4), 1–12.

Hernando, J., Nadeu, C., 1997a. CDHMM speaker recognition by means of frequency filtering of filter-bank energies. In: Proc. Eurospeech, Vol. 5, pp. 2363–2366.

Hernando, J., Nadeu, C., 1997b. Robust speech parameters located in the frequency domain. In: Proc. Eurospeech, Vol. 1, pp. 417–420.

Houtgast, T., Steeneken, H.J.M., 1985. A review of the MTF concept in room acoustics and its use for estimating speech intelligibility in auditoria. J. Acoust. Soc. Amer. 77 (3), 1069–1077.

Hunt, M.J., 1999. Spectral signal processing for ASR. In: Proc. Workshop ASRU.

Hunt, M.J., Lefèbvre, C., 1989. A comparison of several acoustic representations for speech recognition with degraded and undegraded speech. In: Proc. ICASSP, Vol. S1, pp. 262–265.

Juang, B.H., Rabiner, L.R., Wilpon, J.G., 1987. On the use of bandpass liftering in speech recognition. IEEE Trans. Acoust. Speech Signal Process. ASSP-35 (7), 947–954.

Junqua, J.-C., Haton, J-P. (Ed.), 1996. Robustness in Automatic Speech Recognition. Kluwer Academic Publishers, New York, pp. 1996.

Klatt, D.H., 1982. Prediction of perceived phonetic distance from critical band spectra: A first step. In: Proc. ICASSP, pp. 1278–1281.

Leonard, R.G., 1984. A database for speaker-independent digit recognition. In: Proc. ICASSP, Vol. 3, pp. 42–45.

Ljolje, A., 1994. The importance of cepstral parameter correlations in speech recognition. Computer Speech and Language 8, 223–232.

Macho, D., Nadeu, C., 1998. On the interaction between time and frequency filtering of speech parameters for robust speech recognition. In: Proc. ICSLP, pp. 1487–1490.

Macho, D., Nadeu, C., Hernando, J., Padrell, J., 1999a. Time and frequency filtering for speech recognition in real noise conditions. In: Proceedings of the Workshop on Robust Methods for Speech Recognition in Adverse Conditions, Tampere, pp. 111–114.

Macho, D., Nadeu, C., Jancovic, P., Rozinaj, G., Hernando, J., 1999b. Comparison of time and frequency filtering and cepstral-time matrix approaches in ASR. In: Proc. Eurospeech, Vol. 1, pp. 77–80.

Mashao, D.J., Gotoh, Y., Silverman, H.F., 1996. Analysis of LPC/DFT features for HMM-based alphadigit recognizer. IEEE Signal Process. Lett. 3 (4), 103–106.

Nadeu, C., Juang, B.H., 1994. Filtering of spectral parameters for speech recognition. In: Proc. ICSLP, pp. 1927–1930.

Nadeu, C., Hernando, J., Gorricho, M., 1995. On the decorrelation of filter-bank energies in speech recognition. In: Proc. Eurospeech, pp. 1381–1384.

Nadeu, C., Mariño, J.B., Hernando, J., Nogueiras, A., 1996. Frequency and time filtering of filter-bank energies for HMM speech recognition. In: Proc. ICSLP, pp. 430–433.

Nadeu, C., Pachès-Leal, P., Juang, B.H., 1997a. Filtering the time sequence of spectral parameters for speech recognition. Speech Communication 22, 315–322.

Nadeu, C., Padrell, J., Esquerra, I., 1997b. Frequency averaging: an useful multiwindow spectral analysis approach. In: Proc. ICASSP, pp. 3953–3956.

Nadeu, C., Galindo, F., Padrell, J., 1998. On frequency averaging for spectral analysis in speech recognition. Proc. ICSLP 3, 1071–1074.

Oppenheim, A.V., Schafer, R.W., 1989. Discrete-Time Signal Processing. Prentice-Hall, Englewood Cliffs, NJ.

Pachès-Leal, P., Rose, R.C., Nadeu, C., 1999. Optimization algorithms for estimating modulation spectrum domain filters. In: Proc. Eurospeech, Vol. 1, pp. 89–92.

Pai, H-F., Wang, H-C., 1992. A study of two-dimensional cepstrum approach for speech recognition. Computer Speech and Language 6, 361–375.

Paliwal, K.K., 1982. On the performance of the quefrency-weighted cepstral coefficients in vowel recognition. Speech Communication 1 (2), 151–154.

Paliwal, K.K., 1999. Decorrelated and liftered filter-bank energies for robust speech recognition. In: Proc. Eurospeech, Vol. 1, pp. 85–88.

Pearce, D., 1998. Experimental framework for the performance evaluation of distributed speech recognition front-ends. Aurora project report, Version 1.

Picone, J.W., 1991. Signal modeling techniques in speech recognition. Proc. IEEE 79 (4), 1214–1247.

Rabiner, L., Juang, B.H., 1993. Fundamentals of Speech Recognition. Prentice-Hall, Englewood Cliffs, NJ.

Rahim, M.G., Juang, B.H., 1996. Signal bias removal by maximum likelihood estimation for robust telephone speech recognition. IEEE Trans. SAP 4 (1), 19–30.

Rosenberg, A.E., Lee, C.-H., Soong, F., 1994. Cepstral channel normalization techniques for HMM-based speaker verification. In: Proc. ICSLP, pp. 1835–1839.

Silverman, H.F., Dixon, N.R., 1974. A parametrically controlled spectral analysis system for speech. IEEE Trans. Acoust. Speech Signal Process. ASSP-22 (5), 362–381.

Thomson, D.J., 1982. Spectrum estimation and harmonic analysis. Proc. IEEE 70 (9), 1055–1096.

Tian, J., Viikki, O., 1999. Generalized cepstral analysis for speech recognition in noise. In: Proc. Eurospeech, pp. 87–90.

Tokhura, Y., 1987. A weighted cepstral distance measure for speech recognition. IEEE Trans. Acoust. Speech Signal Process. ASSP-35 (10), 1414–1422.

Vaseghi, S.V., Conner, P.N., Milner, B.P., 1993. Speech modeling using cepstral-time feature matrices in hidden Markov models. IEE Proc. I 140 (5), 317–320.

Young, S., Odell, J., Ollason, D., Valtchev, V., Woodland, P., 1997. Hidden Markov Model Toolkit V2.1 reference manual. Technical report, Speech group. Cambridge University Department, March 1997.