Cepstral Vector Normalization Based on Stereo Data for Robust Speech Recognition

Luis Buera, Eduardo Lleida, *Member, IEEE*, Antonio Miguel, Alfonso Ortega, and Óscar Saz

***Abstract*—In this paper, a set of feature vector normalization methods based on the minimum mean square error (MMSE) criterion and stereo data is presented. They include multi-environment model-based linear normalization (MEMLIN), polynomial MEMLIN (P-MEMLIN), multi-environment model-based histogram normalization (MEMHIN), and phoneme-dependent MEMLIN (PD-MEMLIN). Those methods model clean and noisy feature vector spaces using Gaussian mixture models (GMMs). The objective of the methods is to learn a transformation between clean and noisy feature vectors associated with each pair of clean and noisy model Gaussians. The direct approach to learn the transformation is by using stereo data; that is, noisy feature vectors and the corresponding clean feature vectors. In this paper, however, a nonstereo data based training procedure, is presented. The transformations can be modeled just like a bias vector (MEMLIN), or by using a first-order polynomial (P-MEMLIN) or a nonlinear function based on histogram equalization (MEMHIN). Further improvements are obtained by using phoneme-dependent bias vector transformation (PD-MEMLIN). In PD-MEMLIN, the clean and noisy feature vector spaces are split into several phonemes, and each of them is modeled as a GMM. Those methods achieve significant word error rate improvements over others that are based on similar targets. The experimental results using the SpeechDat Car database show an average improvement in word error rate greater than 68% in all cases compared to the baseline when using the original clean acoustic models, and up to 83% when training acoustic models on the new normalized feature space.**

***Index Terms*—Feature vector normalization, Gaussian mixture models (GMMs), minimum mean square error (MMSE), robust speech recognition.**

I. Introduction

When training and testing acoustic conditions differ, the accuracy of speech recognition systems rapidly degrades. To compensate for the effects of additive and convolutional noises, which are the main cause of the mismatch between training and recognition spaces, robustness techniques have been developed along the following two main lines of research:

- acoustic model adaptation methods, which map acoustic models from training space to recognition space;
- feature vector adaptation/normalization methods, which map recognition space feature vectors to the training space.

Some of the techniques can be combined to generate hybrid solutions, which are effective under certain conditions [1], [2]. The choice of a robustness technique depends on the characteristics of the application in each situation. In general, acoustic model adaptation methods produce the best results [3] because they can model the uncertainty caused by the noise statistics. Well-known successful acoustic model adaptation methods include maximum *a posteriori* (MAP) [4], maximum-likelihood linear regression (MLLR) [5], parallel model combination (PMC) [6], and vector Taylor series (VTS) [7]. However, these methods require more data and computing time than do feature vector adaptation/normalization methods.

Feature vector adaptation/normalization methods fall into one of three main classes [8]: high-pass filtering, model-based techniques and empirical compensation.

High-pass filtering contains methods such as cepstral mean normalization (CMN) [9], [10] and relative spectral amplitude (RASTA) processing, [11]. Although the results produced by those methods are limited individually, some of them, particularly CMN, are included in almost every speech recognition systems because they use simple and effective procedures.

Model-based methods assume that a mismatch between training and recognition spaces can be represented by a structural model of environmental degradation. The parameters of the structural model are estimated and applied to the appropriate inverse operation to compensate the recognition signal. Examples of model-based methods are VTS [7], codeword dependent cepstral normalization (CDCN) [12], minimum mean square error log spectral amplitude estimator (MMSE-LSA) [13], and spectral subtraction (SS) [14].

Empirical compensation methods that use direct cepstral comparisons are entirely data driven. Typically, they require stereo data, but sometimes "blind" approaches are used [7]. Empirical compensation methods need a training phase where some transformations are estimated by computing the frame-by-frame differences between the vectors representing speech in the clean and noisy environments (stereo data). Algorithms used in that approach include multivariate Gaussian-based cepstral normalization (RATZ) [7], stereo-based piecewise linear compensation for environments (SPLICE) [15], and probabilistic optimum filtering (POF) [16].

Independently of the feature vector normalization method, several algorithms assume a prior probability density function (pdf) for the estimation variable. In those cases, a Bayesian estimator can be used to estimate the clean feature vector. The most commonly used criterion is to minimize the mean square error (MSE), and the optimal estimator for this criterion, minimum mean square error (MMSE), is the mean of the posterior pdf. Many different methods, such as CDCN, VTS, RATZ, and SPLICE use the MMSE estimator to compute the estimated clean feature vector.

Manuscript received March 20, 2006; revised August 18, 2006. This work was supported by the MEC of the Spanish government under Project TIN 2005-08660-C04-01. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Timothy J. Hazen.

The authors are with the University of Zaragoza, 50009 Zaragoza, Spain (e-mail: lbuera@unizar.es; lleida@unizar.es; amiguel@unizar.es; ortega@unizar.es; oskarsaz@unizar.es).

Digital Object Identifier 10.1109/TASL.2006.885244

This paper focuses on empirical feature vector normalization based on stereo data and the MMSE estimator. Some methods, such as VTS, CDCN, POF, and RATZ, assume that the clean feature space can be modeled using a Gaussian mixture model (GMM). However, although the uncertainty between clean and normalized feature vectors is reduced, a mismatch is generated in the estimation of the *a posteriori* probability of the clean model Gaussian, given the noisy feature vector [7] in the normalization. To avoid the problem and maintaining the uncertainty improvement, some other algorithms, e.g., SPLICE, model the noisy space using a GMM. In general, noisy space modeling produces better results than does clean space modeling; however, both modeling methods still have high uncertainty when learning transformations because they model only the clean or the noisy space.

To improve the results obtained using state-of-the-art empirical feature normalization methods, we propose several solutions based on the joint modeling of clean and noisy space. We present multi-environment model-based linear normalization (MEMLIN) [17], which splits noisy space into several basic environments and models each basic noisy and clean feature spaces using GMMs.

Most empirical feature vector normalization methods compute a bias vector transformation for each clean model Gaussian, e.g., RATZ, each noisy model Gaussian, e.g., SPLICE, or each pair of clean and noisy model Gaussians, e.g., MEMLIN. In this work, we propose several approximations to modify the simple bias correction term used in MEMLIN. A first-order polynomial transformation, polynomial multi-environment model-based linear normalization (P-MEMLIN) addresses the use of a different slope and bias term for each pair of clean and noisy model Gaussians. A nonlinear transformation, multi-environment model-based histogram normalization (MEMHIN) [18] addresses the use of a histogram equalization for each pair of clean and noisy model Gaussians. Those two new methods can compensate for the effects of the noise over the means and the variance of the feature vectors.

To reduce the uncertainty between the new normalized feature vectors and the acoustic models, we propose a phoneme-dependent multi-environment model-based linear normalization (PD-MEMLIN) [19] in which the clean and noisy spaces are split into phonemes that are modeled using GMMs. The bias vector transformation is defined for the pair of clean and noisy model Gaussians of each phoneme.

In many acoustic environments and training databases, stereo data are unavailable. To overcome the limitation of the need for stereo data, a nonstereo data training algorithm that uses only noisy feature vectors is proposed. That "blind" technique is applied over the PD-MEMLIN method, [20].

Although these new methods attempt to map the noisy feature vectors to the clean space, the transformation is not perfect; therefore, there remains a mismatch between clean space and the new normalized space. To compensate for that mismatch, we propose to adapt the acoustic models to the new normalized space.

To compare the performance of the proposed methods in a real and dynamic environment, experiments were carried out using the Spanish SpeechDat Car database [21]. Car noise characteristics depend on driving conditions [9], [22], and the Lombard [23] effect can be important; consequently, speech recognition in cars is a difficult task that can generate valid results with which to compare the different techniques.

This paper is organized as follows: In Section II, the noise effects and the basic MMSE-based feature vector normalization methods are detailed. In Section III, the Spanish SpeechDat Car database and the results from the different state-of-the-art MMSE-based feature vector normalization techniques, CMN, RATZ, SPLICE, and MEMLIN are explained. In Section IV, P-MEMLIN, MEMHIN, PD-MEMLIN and "blind" PD-MEMLIN are described, and the results of these methods are presented. Finally, a discussion and the conclusions are presented in Section V.

II. Noise Effects and Basic MMSE-Based Feature Vector Normalization Methods

We assume a general, simplified approximation of speech signal degradation based on additive noise and convolutional noise [12]. In this case, the noisy signal in the mel frequency cepstral coefficient (MFCC) domain y_t can be modeled as

$$y_t = x_t + f(x_t, n_t, h_t) \tag{1}$$

where t is the time frame index, x_t is the clean MFCC vector, n_t is the additive noise MFCC vector, and h_t is the corresponding convolutional noise MFCC vector. The random nature of the additive and convolutional noises results in one to many mapping between clean and noisy feature spaces: a given clean feature vector can generate different noisy feature vectors, and vice versa, which creates an uncertainty.

Fig. 1 shows the scattergrams and histograms for the first MFCC coefficient in non-silence frames for clean and noisy feature vectors from different degradation conditions. Note that the uncertainty between clean and noisy coefficients always exists, even when controlled convolutional noise only is considered [Fig. 1(a)]. The convolutional noise mainly shifts the mean of the coefficients, whereas additive noise [Fig. 1(b)] modifies the pdf, reducing the variance of the coefficients. In the same way, the real car environment [Fig. 1(c)] modifies the mean and variance, jointly.

To compensate for noise effects, there are several kinds of feature vector normalization methods (Section I), but we focus on empirical methods based on the MMSE criterion. Therefore, given the noisy feature vector y_t, the estimated clean feature vector $\hat{x}_t$ is obtained by using the MMSE criterion as

$$\hat{x}_t = E[x|y_t] = \int_X x p(x|y_t) dx \tag{2}$$

where x is the clean feature vector, and $p(x|y_t)$ is the pdf of x given y_t. The way $p(x|y_t)$ and x are approximated determines the different MMSE-based feature vector normalization methods.

A. Basic MMSE-Based Feature Vector Normalization Methods

There are mainly three basic feature vector normalization methods based on the MMSE criterion that have been used extensively: CMN, which is a very simple method, RATZ [7], and

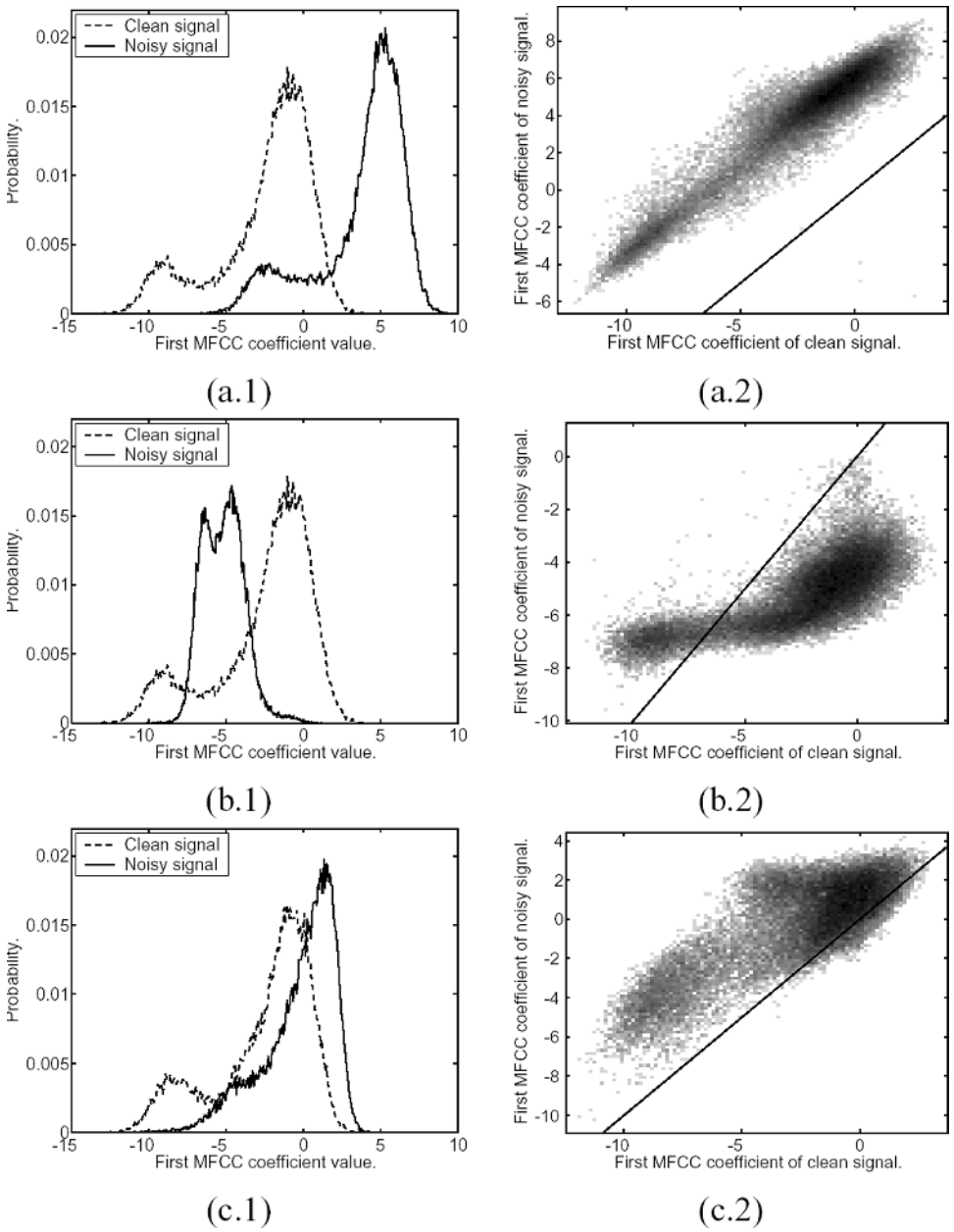


Fig. 1. Scattergrams and histograms for the first MFCC coefficient in non-silence frames between clean (x-axis) and contaminated (y-axis) in several degraded conditions. (a) Controlled convolutional noise when the filter response is longer than the Hamming window used in the computing of MFCC. (b) Additive car noise with 0-dB SNR. (c) Real car condition. The line in the scattergrams represents the function $x = y$.

SPLICE [15]. In the CMN method, no assumptions are made in estimating $p(x|y_t)$, and the clean feature vector x is approximated as $x \approx \Psi(y_t, r) = y_t - r$, where r is a bias vector transformation between y_t and x. With that approximation, for CMN, (2) becomes

$$\hat{x}_t = \int_X (y_t - r)p(x|y_t)dx = y_t - r. \tag{3}$$

To estimate the bias vector transformation, r, the mean square error, ξ, is defined and minimized with respect to r

$$\begin{aligned} r &= \arg\min_r(\xi) = \arg\min_r \left(E\left[(\hat{x}_t - x_t)^2\right]\right) \\ &= E[y_t] - E[x_t] \end{aligned} \tag{4}$$

where $E[\bullet]$ is the corresponding mean. In some cases, the mean of the clean feature vectors is removed before training the acoustic models, and then the bias vector transformation for CMN is computed as $r = E[y_t]$. Actually, the basic CMN algorithm, or an extension of it, is considered a standard and it is used in almost every speech recognition systems because of the low computing time and satisfactory results.

To improve the CMN approximation, RATZ makes two assumptions. The first assumption consists of modeling the clean space using a GMM

$$p(x) = \sum_{s_x} p(x|s_x)p(s_x) \tag{5}$$

$$p(x|s_x) = \mathcal{N}(x; \mu_{s_x}, \Sigma_{s_x}) \tag{6}$$

where μ_{s_x}, Σ_{s_x}, and $p(s_x)$ are the mean, the diagonal covariance matrix, and the *a priori* probability associated with the clean model Gaussian s_x. The second assumption for RATZ is to approximate the clean feature vector as $x \approx \Psi(y_t, r_{s_x}) = y_t - r_{s_x}$, where r_{s_x} is a bias vector transformation between y_t and x for the the clean model Gaussian, s_x. The estimation of r_{s_x} is included in [7]. With the two assumptions, RATZ makes (2) become

$$\hat{x}_t \approx \int_X \sum_{s_x} (y_t - r_{s_x}) p(x, s_x|y_t) dx = y_t - \sum_{s_x} r_{s_x} p(s_x|y_t) \tag{7}$$

where $p(s_x|y_t)$ is the *a posteriori* probability of the clean model Gaussian s_x, given the noisy feature vector y_t, and it can be computed using (5) and (6) assuming an additive effect of the noise in the MFCC domain [7].

Although RATZ can improve the performance concerning CMN because it models better the clean space, in the normalization, the estimation of $p(s_x|y_t)$ can produce a mismatch. To avoid it, SPLICE proposes to model the noisy space instead of the clean one using GMM

$$p(y_t) = \sum_{s_y} p(y_t|s_y) p(s_y) \tag{8}$$

$$p(y_t|s_y) = \mathcal{N}(y_t; \mu_{s_y}, \Sigma_{s_y}) \tag{9}$$

where s_y denotes the corresponding Gaussian of the noisy model, μ_{s_y}, Σ_{s_y}, and $p(s_y)$ are the mean vector, the diagonal covariance matrix, and the *a priori* probability associated with s_y. At the same time, the clean feature vector x is approximated as $x \approx \Psi(y_t, r_{s_y}) = y_t - r_{s_y}$, where r_{s_y} is a bias vector transformation between y_t and x for the noisy model Gaussian, s_y. The estimation of r_{s_y} is evaluated in [15]. Therefore, SPLICE transforms (2) into

$$\hat{x}_t \approx \int_X \sum_{s_y} (y_t - r_{s_y}) p(x, s_y|y_t) dx = y_t - \sum_{s_y} r_{s_y} p(s_y|y_t) \tag{10}$$

where $p(s_y|y_t)$ is the *a posteriori* probability of the noisy model Gaussian, s_y, given the noisy feature vector, y_t, computed using (8) and (9).

The bias vector transformations of RATZ and SPLICE depend on the environment. So, to consider several acoustic conditions, RATZ and SPLICE multi-environment methods have been developed: interpolated RATZ (IRATZ) [7] and SPLICE with environmental model selection [15]. In those methods, noisy space is split into several basic environments concerning similar acoustic properties [signal-to-noise ratio, (SNR), spectral shape], and the bias vector transformations are computed independently for each basic environment. The final resulting transformation is computed as a weighted sum of all of the basic environment bias vector transformations (soft decision), or using only the most probable basic environment bias vector transformations (hard decision).

B. Multi-Environment Model-Based Linear Normalization (MEMLIN)

MEMLIN proposes a general MMSE-based framework by providing a GMM modeling of the clean and noisy spaces. Noisy space is divided in a combination of basic acoustic environments. Therefore, a bias vector transformation is associated with each pair of Gaussians from the clean and the noisy basic environment spaces.

1) MEMLIN Approximations: In MEMLIN, three approaches are used.

- Noisy space is divided into a combination of several basic environments e, and the noisy feature vectors y_t are modeled as a GMM for each basic environment

$$p_e(y_t) = \sum_{s_y^e} p\left(y_t|s_y^e\right) p\left(s_y^e\right) \tag{11}$$

$$p\left(y_t|s_y^e\right) = \mathcal{N}\left(y_t; \mu_{s_y^e}, \Sigma_{s_y^e}\right) \tag{12}$$

  where s_y^e denotes the corresponding Gaussian of the noisy model for the e basic environment, $\mu_{s_y^e}$, $\Sigma_{s_y^e}$, and $p(s_y^e)$ are the mean vector, the diagonal covariance matrix, and the *a priori* probability associated with s_y^e.
- Clean feature vectors are modeled using a GMM: expressions (5) and (6).
- Clean feature vectors can be approximated as a linear function of the noisy feature vector, which depends on the basic environment and the clean and noisy model Gaussians: $x \approx \Psi(y_t, s_x, s_y^e) = y_t - r_{s_x, s_y^e}$, where r_{s_x, s_y^e} is a bias vector transformation between noisy and clean feature vectors for each pair of Gaussians, s_x and s_y^e.

2) MEMLIN Enhancement: With those approximations, MEMLIN transforms (2) into

$$\begin{aligned}\hat{x}_t &= y_t - \int_X \sum_e \sum_{s_y^e} \sum_{s_x} r_{s_x, s_y^e} p\left(x, s_x, e, s_y^e|y_t\right) dx \\ &= y_t - \sum_e \sum_{s_y^e} \sum_{s_x} r_{s_x, s_y^e} \\ &\quad \times p(e|y_t) p\left(s_y^e|y_t, e\right) p\left(s_x|y_t, e, s_y^e\right)\end{aligned} \tag{13}$$

where $p(e|y_t)$ is the *a posteriori* probability of the basic environment; $p(s_y^e|y_t, e)$ is the *a posteriori* probability of the noisy model Gaussian, s_y^e, given the feature vector, y_t, and the basic environment, e. Those two terms are computed for each frame applying (11) and (12). Finally, the cross-probability model $p\left(s_x|y_t, e, s_y^e\right)$ is the probability of the clean model Gaussian s_x, given the feature vector y_t, the basic environment e, and the noisy model Gaussian s_y^e. That term, along with the bias vector transformation, r_{s_x, s_y^e}, is estimated in a training phase using stereo data.

The *a posteriori* probability of the basic environment $p(e|y_t)$ is computed recursively by applying (11) and (12) as

$$p(e|y_t) = \beta \cdot p(e|y_{t-1}) + (1 - \beta) \frac{p_e(y_t)}{\sum_e p_e(y_t)} \tag{14}$$

where β is the memory constant ($0 \leq \beta \leq 1$), and $p(e|y_0)$ is considered to be uniformly distributed over all the basic environments. Considering the defined acoustic environments, β has to be close to 1 due to the succession of the basic environments along the time is not very fast. So, in this paper, β has been set to 0.98. The *a posteriori* probability of the noisy model Gaussian, given the feature vector, y_t, and the basic environment, e, $p\left(s_y^e|y_t, e\right)$ can be computed considering (11) and (12) as

$$p\left(s_y^e|y_t, e\right) = \frac{p\left(y_t|s_y^e\right) p\left(s_y^e\right)}{\sum\limits_{s_{\hat{y}}^e} p\left(y_t|s_{\hat{y}}^e\right) p\left(s_{\hat{y}}^e\right)}. \tag{15}$$

3) MEMLIN Training: Given a stereo data corpus for each basic environment $(X_e, Y_e) = \left\{\left(x_1^e, y_1^e\right); \ldots; \left(x_{t_e}^e, y_{t_e}^e\right); \ldots; \left(x_{T_e}^e, y_{T_e}^e\right)\right\}$, with $t_e \in [1, T_e]$, the bias vector transformation, r_{s_x, s_y^e}, is estimated by minimizing the defined mean weighted square error, ξ_{s_x, s_y^e}, with respect to r_{s_x, s_y^e}

$$\begin{aligned}\xi_{s_x, s_y^e} &= \sum_{t_e} p\left(s_x|x_{t_e}^e, e\right) p\left(s_y^e|y_{t_e}^e, e\right) \\ &\quad \times \left(x_{t_e}^e - y_{t_e}^e + r_{s_x, s_y^e}\right)^2 \qquad (16)\\ r_{s_x, s_y^e} &= \arg\min_{r_{s_x, s_y^e}} \left(\xi_{s_x, s_y^e}\right) \\ &= \frac{\sum\limits_{t_e} p\left(s_x|x_{t_e}^e, e\right) p\left(s_y^e|y_{t_e}^e, e\right) \left(y_{t_e}^e - x_{t_e}^e\right)}{\sum\limits_{t_e} p\left(s_x|x_{t_e}^e, e\right) p\left(s_y^e|y_{t_e}^e, e\right)} \qquad (17)\end{aligned}$$

where $p\left(s_x|x_{t_e}^e, e\right)$ is the *a posteriori* probability of the clean model Gaussian, s_x, given the clean feature vector, $x_{t_e}^e$, and the basic environment, e. It can be estimated by applying (5) and (6)

$$p(s_x|x_{t_e}^e, e) = \frac{p\left(x_{t_e}^e|s_x\right) p(s_x)}{\sum\limits_{s_x} p\left(x_{t_e}^e|s_x\right) p(s_x)}. \tag{18}$$

The cross-probability model, $p\left(s_x|y_t, e, s_y^e\right)$, is simplified by avoiding the time dependence given by the noisy feature vector, y_t, $\left(p\left(s_x|y_t, e, s_y^e\right) \simeq p\left(s_x|s_y^e, e\right)\right)$. The term $p\left(s_x|s_y^e, e\right)$ can be estimated by using relative frequency, a hard solution, or using (11), (12), (5), and (6), soft decision. Therefore, the corresponding expression for the hard decision is

$$p\left(s_x|s_y^e, e\right) = \frac{C_N\left(s_x|s_y^e\right)}{N_{s_y^e}} \tag{19}$$

where $C_N\left(s_x|s_y^e\right)$ is the count number of times that the most probable pair of Gaussians is s_x and s_y^e for all pairs of stereo training data of the e basic environment, and $N_{s_y^e}$ is the count number of times that the most probable Gaussian for noisy training feature vectors is s_y^e for the e basic environment.

The estimation of the cross-probability model using the soft decision is

$$p\left(s_x|s_y^e, e\right) = \frac{\sum\limits_{t_e} p\left(x_{t_e}|s_x\right) p\left(y_{t_e}|s_y^e\right) p(s_x) p\left(s_y^e\right)}{\sum\limits_{t_e}\sum\limits_{s_x} p(x_{t_e}|s_x) p\left(y_{t_e}|s_y^e\right) p(s_x) p\left(s_y^e\right)}. \tag{20}$$

When there are enough data to estimate the cross-probability model, both solutions, hard and soft, obtain similar results: no significant changes in recognition were obtained in this case. However, when there are not enough data, the soft option provides a more consistent solution. The hard solution was used in all the experiments carried out with MEMLIN in this work.

In summary, MEMLIN associates a bias vector transformation to each pair of noisy and clean Gaussians. So, comparing against RATZ or SPLICE, which define a bias vector transformation from a Gaussian to the whole noisy or clean space, the mapping space associated to each MEMLIN transformation is more enclosed, having a less uncertainty region. Therefore, given an appropriate cross-probability model, MEMLIN is expected to outperform RATZ or SPLICE performances.

III. SpeechDat Car Database and Results Using Basic Feature Vector Normalization MMSE-Based Methods

To compare the performance of the basic multi-environment MMSE-based feature vector normalization methods (IRATZ, SPLICE with environment model selection, and MEMLIN) in a real, dynamic, and complex environment, a set of experiments were performed using the Spanish SpeechDat Car database [21]. Seven basic environments were defined as follows:

E1: car stopped, motor running;
E2: town traffic, closed windows, and climatizer off (silent conditions);
E3: town traffic and noisy conditions (windows open, and/or climatizer on);
E4: low speed, rough road, and silent conditions;
E5: low speed, rough road, and noisy conditions;
E6: high speed, good road, and silent conditions;
E7: high speed, good road, and noisy conditions.

In this study, two channels of the database recorded simultaneously (stereo data) have been used: A clean signal from a CLose talK channel (CLK), which was recorded using a Shure SM-10A microphone, and a noisy signal from a hands-free channel (HF), which was recorded using a Peiker ME15/V520-1 microphone located on the ceiling in front of the driver. HF signals are used in recognition tasks.

For speech recognition, the feature vector is composed of the 12 MFCCs, first and second derivatives and the delta energy, giving a final feature vector of 37 coefficients computed every 10 ms using a 25-ms Hamming window. On the other hand, in this paper, the feature vector normalization methods are applied to the 12 MFCCs and log energy only, whereas the derivatives are computed over the normalized static coefficients.

The recognition task is isolated and continuous digits recognition. Word acoustic models are built from a set of 674 left and right context-dependent and 25 context-independent units. Each unit is modeled by one-state continuous density HMMs with 16 Gaussians. In addition, two silence models for long and interword silences are considered. Each phoneme is modeled by the left contextual unit, the incontextual unit and right contextual unit. So, for example, the word acoustic model for Spanish digit "dos" ("two") can be obtained by the concatenation of the following units: /#⟨d/ /d/ /d⟩o/ /d⟨o/ /o/ /o⟩s/ /o⟨s/ /s/ /s⟩#/, where # is the silence unit, / < / is the left context-dependent unit, //

TABLE I
NUMBER OF UTTERANCES AND WORDS FOR TRAINING AND TESTING CORPORA USED IN ALL THE EXPERIMENTS

	E1	E2	E3	E4	E5	E6	E7	Total
# utterances train	3,393	3,122	2,356	2,106	2,550	2,038	543	16,108
# utterances test	199	223	136	152	200	120	56	1,086
# words train	10,542	9,652	7,160	6,517	7,908	6,265	1,673	49,717
# words test	1,049	1,166	715	798	1,049	630	294	5,701

TABLE II
WER BASELINE RESULTS, IN PERCENT, FROM THE DIFFERENT BASIC ENVIRONMENTS ($E1, \ldots, E7$) WHEN CLEAN (CLK IN THE TRAIN COLUMN) OR NOISY (HF IN THE TRAIN COLUMN) ACOUSTIC MODELS ARE APPLIED. HF† INDICATES THAT SPECIFIC ACOUSTIC MODELS FOR EACH BASIC ENVIRONMENT ARE TRAINED. "TEST" REFERS TO THE RECOGNIZED DATA, EITHER AS CLEAN (CLK) OR NOISY (HF)

Train	Test	E1	E2	E3	E4	E5	E6	E7	MWER (%)
CLK	CLK	0.38	2.06	1.40	0.50	0.57	0.16	0.00	0.86
CLK	HF	4.29	11.06	11.61	14.79	14.49	11.27	20.07	11.52
HF	HF	1.72	5.49	3.08	4.01	4.86	3.33	5.78	3.95
HF†	HF	1.24	4.37	2.10	3.38	3.63	1.11	3.40	2.82

is the context-independent unit, and finally $/ > /$ is the right context-dependent unit.

A training corpus for each basic environment is used for training acoustic models and learning the corresponding bias vector transformations and the cross-probability models (16 108 utterances for all basic environments and different tasks: isolated and continuous digits, spelling, dates, commands and names). The testing corpus is composed of 1086 utterances for all basic environments, and different speakers from the training corpus. The composition of the training and testing corpora is explained in detail in Table I, where it is included the number of utterances and words for each basic environment. No voice activity detector (VAD) is applied in any case.

The word error rate (WER) baseline results for each basic environment are presented in Table II, where MWER is the Mean WER, which is computed proportionally to the number of words in each basic environment (see Table I). The CMN method is applied to testing and training data. "Train" column refers to the signals used to obtain the corresponding acoustic HMMs; if they are trained with all clean training utterances, the column is marked CLK, and if the column is marked HF, the acoustic models are trained with all noisy training utterances. HF† indicates that specific acoustic HMMs for each basic environment are applied in the recognition task (environment match condition). "Test" column indicates which signals are using for recognition: clean, CLK, or noisy, HF.

Table II shows the effect of real car conditions, which produces a significant increase in WER in all of the basic environments, (Train CLK, Test HF), concerning the rates for clean signal, (Train CLK, Test CLK). When acoustic models are retrained using all basic environments, (Train HF) MWER decreases considerably. Finally, the lowest MWER when the noisy signal is used for recognition is obtained for environment match condition, (Train HF†): 2.82%.

Fig. 2 shows the mean improvement in WER (MIMP) in percent for each of the multi-environment basic feature vector normalization methods based on MMSE (IRATZ, SPLICE with environmental model selection, SPLICE MS, and MEMLIN). A

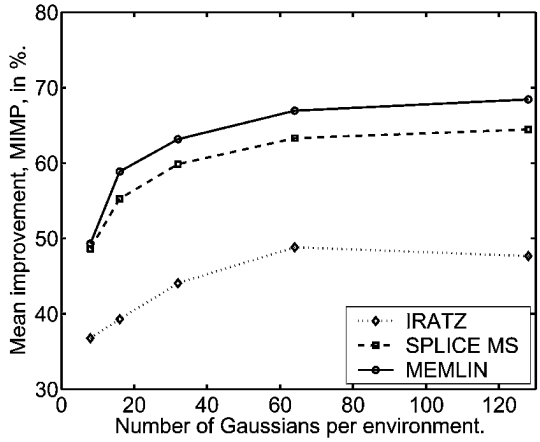


Fig. 2. Mean improvement in WER for interpolated RATZ (IRATZ), SPLICE with environmental model selection (SPLICE MS) and MEMLIN.

100% MIMP would be obtained when the MWER is the same as in clean conditions. So, given a MWER, the corresponding MIMP will be

$$\mathrm{MIMP} = \frac{100(\mathrm{MWER} - \mathrm{MWER}_{\mathrm{CLK-HF}})}{\mathrm{MWER}_{\mathrm{CLK-CLK}} - \mathrm{MWER}_{\mathrm{CLK-HF}}} \tag{21}$$

where $\mathrm{MWER}_{\mathrm{CLK-CLK}}$ is the mean WER obtained with clean conditions (0.86 in this case), and $\mathrm{MWER}_{\mathrm{CLK-HF}}$ is the baseline (11.53). In order to compare all the methods, the MIMP has been depicted with respect to the number of Gaussians per basic environment, because it gives an idea of the computing cost. The SPLICE MS method always produces better results than does RATZ, which is because of the assumption of the noisy model when the *a posteriori* probability of a clean model Gaussian, given the noisy feature vector is computed [7]. On the other hand, the MEMLIN algorithm improves the results based on SPLICE MS for any number of Gaussians per basic environment due to the projection space associated to a bias vector transformation in MEMLIN is smaller than SPLICE MS, being the transformations more specific. To obtain more specific transformations in MEMLIN, the number of them associated to a noisy or clean model Gaussian is higher than in SPLICE MS or IRATZ, but the computing cost in the normalization process is almost the same.

Fig. 3(a) and (b) shows the comparative histograms and scattergrams between clean and noisy and normalized first MFCC coefficients in non-silence frames from E4 basic environment. The normalized coefficients are obtained using MEMLIN with 128 Gaussians per basic environment. Although all terms in SPLICE MS normalization are obtained directly using the noisy GMM, the corresponding histograms and scattergrams are visually similar to MEMLIN ones. However, MEMLIN histograms and scattergrams can be improved considerably if the cross-probability model is estimated properly (it will be considered in the Section V). It can be observed that the normalized signal histogram is close to the clean signal one, although there is still a considerable uncertainty between clean and normalized coefficients [Fig. 3(b.2)]. The peak that appears in the normalized signal histogram [Fig. 3(b.1)] is due to the transformation of noisy feature vectors towards

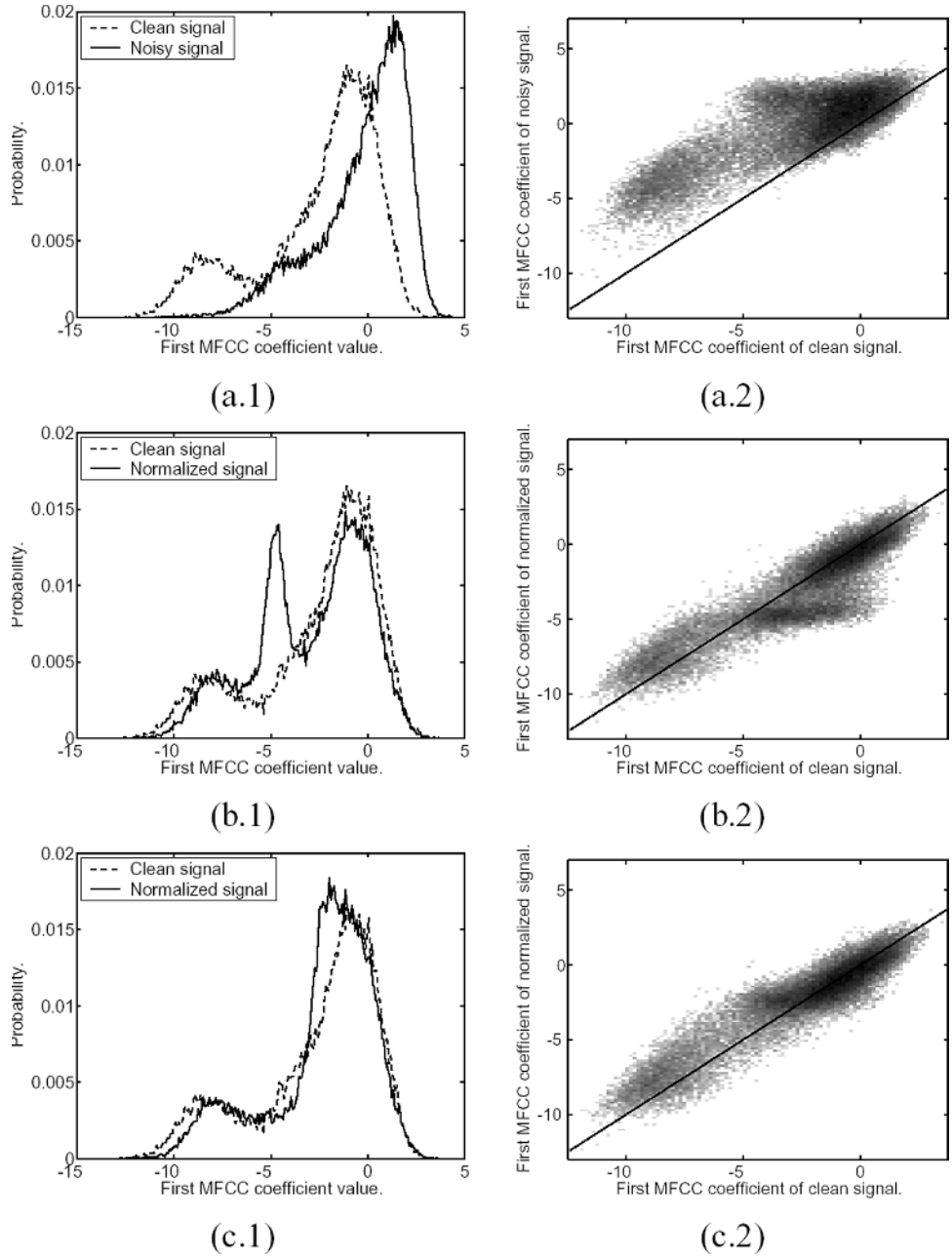


Fig. 3. Scattergrams and histograms between the first MFCC coefficient in non-silence frames for clean (x-axis) and contaminated (a) or normalized (b) using MEMLIN with 128 Gaussians per basic environment (y-axis) signals for the E4 basic environment. Also, the scattergram and histogram when the transformations of MEMLIN with 128 Gaussians are computed only with non-silence frames are presented (c). The line in the scattergrams represents the function $x = y$.

the clean silence. This problem can be solved if an efficient VAD were used in training and during the normalization. To confirm this, the noisy signals were normalized with the transformations and the cross-probability models for MEMLIN with 128 Gaussians trained only with the non-silence frames. Fig. 3(c) presents the scattergram and histogram between the first MFCC coefficients in non-silence frames for clean and normalized with this new training condition signals. It can be observed that the peak disappears.

The most representative results from each of the methods are summarized in Table III, indicating the number of Gaussians per basic environment (♯ Gaussian) required to obtain the best corresponding values.

TABLE III
BEST MWER AND MEAN IMPROVEMENT IN WER (MIMP) IN PERCENT FROM INTERPOLATED RATZ, SPLICE WITH ENVIRONMENTAL MODEL SELECTION, SPLICE MS, AND MEMLIN, WITH THE REQUIRED NUMBER OF GAUSSIANS PER BASIC ENVIRONMENT INDICATED

	♯ Gaussian	MWER (%)	MIMP (%)
Interpolated RATZ	64	6.32	48.82
SPLICE-MS	128	4.65	64.46
MEMLIN	128	4.16	69.09

IV. IMPROVEMENTS OVER MEMLIN

There are two important approximations in MEMLIN expressions that can affect the final performance of the method.

One is the selection of the linear model for x associated with a pair of Gaussians that has an independent term only $\left(x \approx \Psi\left(y_t, s_x, s_y^e\right) = y_t - r_{s_x, s_{y|}^e}\right)$. That model compensates for the mean shift, but not for the modification of the variance. The second approximation involves treating all of the sounds in the same way. So, there is always a bias vector transformation which maps from a noisy model Gaussian to every clean model one and it can produce, for example, that several non-silence noisy feature vectors are mapped towards the clean silence. To overcome these approximations, we consider different solutions. To develop a more realistic model for x, we use a modification of $\Psi\left(y_t, s_x, s_y^e\right)$ and define two novel multi-environment feature vector normalization methods based on MMSE: P-MEMLIN, which uses a complete first order polynomial approximation, and MEMHIN, which assumes a nonlinear model.

Although the MEMLIN algorithm achieves significant improvements over other basic MMSE-based feature vector methods, the variance of the error between clean and noisy feature vectors can be reduced if more specific bias vector transformations are estimated. To do that, we propose learning phoneme-dependent bias vector transformations. That modification of the MEMLIN is called the PD-MEMLIN.

A. $\Psi\left(y_t, s_x, s_y^e\right)$ Modifications: P-MEMLIN and MEMHIN

1) P-MEMLIN: The novel model $\Psi\left(y_t, s_x, s_y^e\right)$ for polynomial MEMLIN is

$$x(i) \approx \Psi\left(y_t, s_x, s_y^e\right)(i) = a_{s_x, s_{y|}^e}(i) y_t(i) - b_{s_x, s_{y|}^e}(i) \quad (22)$$

where i is the coefficient index, and $a_{s_x, s_{y|}^e}$ and $b_{s_x, s_y e}$ are the slope and the independent terms of the model, respectively. Both depend on the basic environment and on clean and noisy model Gaussians. Note that it is assumed that the feature coefficients are independent. Using P-MEMLIN, (2) becomes

$$\hat{x}_t(i) = \sum_e \sum_{s_{y|}^e} \sum_{s_x} \left(a_{s_x, s_{y|}^e}(i) y_t(i) - b_{s_x, s_{y|}^e}(i)\right) p(e|y_t) \times p(s_y^e|y_t, e) p\left(s_x|y_t, e, s_y^e\right) \ \forall\, i. \quad (23)$$

The only modification to MEMLIN is $\Psi\left(y_t, s_x, s_y^e\right)$; therefore, the expressions of $p(e|y_t)$, $p\left(s_y^e|y_t, e\right)$, and $p\left(s_x|y_t, e, s_y^e\right)$ are estimated as (14), (15), and (19) or (20), respectively. On the other hand, $a_{s_x, s_{y|}^e}(i)$ and $b_{s_x, s_{y|}^e}(i)$ are computed in the training phase using stereo data

$$a_{s_x, s_{y|}^e}(i) = \frac{\sigma^x_{s_x, s_{y|}^e}(i)}{\sigma^y_{s_x, s_{y|}^e}(i)} \quad (24)$$

$$b_{s_x, s_{y|}^e}(i) = \frac{\sigma^x_{s_x, s_{y|}^e}(i)}{\sigma^y_{s_x, s_{y|}^e}(i)} \mu^y_{s_x, s_{y|}^e}(i) - \mu^x_{s_x, s_{y|}^e}(i) \quad (25)$$

where $\sigma^x_{s_x, s_{y|}^e}(i)$ and $\sigma^y_{s_x, s_{y|}^e}(i)$ are the ith coefficients of the standard deviations of clean and noisy feature vectors, respectively, associated with the pair of Gaussians s_x and s_y^e. $\mu^x_{s_x, s_{y|}^e}(i)$ and $\mu^y_{s_x, s_{y|}^e}(i)$ are the ith coefficients of the means of clean and noisy feature vectors associated with s_x and s_y^e. They are computed as follows, where z can be x or y

$$\mu^z_{s_x, s_{y|}^e}(i) = \frac{\sum_{t_e} p(s_x|x_{t_e}) p(s_y^e|y_{t_e}) z_{t_e}(i)}{\sum_{t_e} p(s_x|x_{t_e}) p\left(s_y^e|y_{t_e}\right)} \quad (26)$$

$$\sigma^z_{s_x, s_{y|}^e}(i) = \sqrt{\frac{\sum_{t_e} p(s_x|x_{t_e}) p\left(s_y^e|y_{t_e}\right) \left(z_{t_e}(i) - \mu^z_{s_x, s_{y|}^e}(i)\right)^2}{\sum_{t_e} p(s_x|x_{t_e}) p\left(s_y^e|y_{t_e}\right)}}. \quad (27)$$

Note that if the standard deviation terms are equal $\left(\sigma^x_{s_x, s_{y|}^e}(i) = \sigma^y_{s_x, s_{y|}^e}(i),\ \forall\, i\right)$, the algorithm expressions are the same as those in the MEMLIN.

2) MEMHIN: Although P-MEMLIN uses a first-order polynomial to compensate for the variance transformations, sometimes noise can produce a more complex modification of clean and noisy feature pdfs associated with a pair of Gaussians. In that case, the linear approximation for $\Psi\left(y_t, s_x, s_y^e\right)$ of MEMLIN or P-MEMLIN is not the best option; therefore, we propose a nonlinear model based on histogram equalization. The new model is expressed as

$$\Psi\left(y_t, s_x, s_y^e\right) = C^{-1}_{x, s_x, s_{y|}^e}\left(C_{y, s_x, s_{y|}^e}(y_t)\right) \quad (28)$$

where $C_{x, s_x, s_{y|}^e}$ is the clean feature vector cumulative probability associated with s_x and s_y^e Gaussians, and $C^{-1}_{x, s_x, s_{y|}^e}$ is the reciprocal function. $C_{y, s_x, s_{y|}^e}$ is the noisy feature vector cumulative probability associated with s_x and s_y^e Gaussians. For MEMHIN, (2) takes the following expression:

$$\hat{x}_t = \sum_e \sum_{s_x^{pl_i}} \sum_{s_{y|}^{e\text{-}pl_i}} C^{-1}_{x, s_x, s_{y|}^e}\left(C_{y, s_x, s_{y|}^e}(y_t)\right) \times p(e|y_t) p\left(s_y^e|y_t, e\right) p\left(s_x|y_t, e, s_y^e\right). \quad (29)$$

The only difference between MEMLIN and MEMHIN is $\Psi\left(y_t, s_x, s_y^e\right)$; therefore, the probabilities $p(e|y_t)$, $p\left(s_y^e|y_t, e\right)$, and $p\left(s_x|y_t, e, s_y^e\right)$ are estimated following (14), (15), and (19) or (20), respectively. To compute $C_{x, s_x, s_{y|}^e}$ and $C_{y, s_x, s_{y|}^e}$, the n band histograms associated with s_x and s_y^e for each component of the noisy and clean feature vectors are obtained in the training phase, assuming that the components are independent. To estimate the histograms for each pair of Gaussians, the components of the feature vectors are weighted by the product of the *a posteriori* probabilities $p\left(s_x|x^e_{t_e}, e\right)$, and $p\left(s_y^e|y^e_{t_e}, e\right)$. $C_{x, s_x, s_{y|}^e}$ and $C_{y, s_x, s_{y|}^e}$ are computed by cumulating the bands of the corresponding histograms.

3) Results From $\Psi\left(y_t, s_x, s_y^e\right)$ Modifications: To compare the results using P-MEMLIN and MEMHIN with those based on MEMLIN, the experiments described in Section III were repeated.

P-MEMLIN and MEMHIN provide significant improvement over MEMLIN when few Gaussians are considered (33.87% of MIMP for MEMLIN with four Gaussians per basic environ-

TABLE IV
MWER AND MIMP IN PERCENT FOR MEMLIN AND MEMHIN FOR 8, 16, AND 32 GAUSSIANS PER BASIC ENVIRONMENT WITH 5-dB SNR ADDITIVE NOISE

	MWER (%)	MIMP (%)
MEMLIN 8-8	8.15	25.52
MEMLIN 16-16	7.71	30.01
MEMLIN 32-32	7.06	36.63
MEMHIN 8-8	6.93	37.92
MEMHIN 16-16	6.55	41.88
MEMHIN 32-32	6.37	43.64

ment and 39.12% and 37.82% of MIMP for PMEMLIN and MEMHIN); however, if the algorithms are evaluated using more than eight Gaussians per basic environment, the mean results are very similar among the three methods. That performance results from the compensation of the variance of the feature vectors, which is more important when the number of Gaussians used for representing the space is reduced. As the number of Gaussians decrease, the space data modeled by each Gaussian increase and the transformation is more affected by the variance deviation between clean and noisy space. In those situations, a more complex model of x $\left(\Psi\left(y_t, s_x, s_y^e\right)\right)$ produces significant improvements. Although the methods behave similarly when there are more than eight Gaussians, we carried out experiments using controlled additive noise [18], which demonstrated important improvements by using MEMHIN compared to MEMLIN. MEMHIN is better able to compensate for the modifications of the variance in feature vectors caused by additive noise. To confirm that, additive car noise was added to clean signals of the Spanish SpeechDat Car database. Table IV shows some of the results from MEMLIN and MEMHIN with additive car noise of 5 dB of SNR, and clean and noisy GMM of 8, 16, and 32 Gaussians to model the clean and the basic environments.

B. Phoneme-Based Transformations

1) PD-MEMLIN: To obtain a more specific set of transformations, trying to reduce the uncertainty between the normalized feature vectors and the acoustic models, we developed PD-MEMLIN. In PD-MEMLIN, noisy space is divided into a combination of basic acoustic environments as MEMLIN and each one is split into phonemes, which are modeled as a GMM. The clean space is also divided in phonemes and each one of them is modeled as a GMM. Therefore, a bias vector transformation is associated with each pair of Gaussians from the same phoneme of the clean and noisy basic environment spaces.

- *PD-MEMLIN approximations:* In PD-MEMLIN, three approximations are considered.
  First approximation: noisy space is split into several basic environments e. The noisy feature vectors associated with the different phonemes ph of each basic environment are modeled as a GMM

$$p_{e,ph}(y_t) = \sum_{s_{y_j}^{e,ph}} p\left(y_t|s_y^{e,ph}\right) p\left(s_y^{e,ph}\right) \tag{30}$$

$$p(y_t|s_y^{e,ph}) = \mathcal{N}\left(y_t; \mu_{s_{y_j}^{e,ph}}, \Sigma_{s_{y_j}^{e,ph}}\right) \tag{31}$$

  where $s_y^{e,ph}$ denotes the Gaussian that corresponds to phoneme ph and basic environment e; $\mu_{s_{y_j}^{e,ph}}$, $\Sigma_{s_{y_j}^{e,ph}}$, and $p\left(s_y^{e,ph}\right)$ are the mean vector, the diagonal covariance matrix, and the *a priori* probability associated with $s_y^{e,ph}$.
  Second approximation: the clean feature vectors of each phoneme are modeled as a GMM

$$p_{ph}(x) = \sum_{s_x^{ph}} p\left(x|s_x^{ph}\right) p\left(s_x^{ph}\right) \tag{32}$$

$$p\left(x|s_x^{ph}\right) = \mathcal{N}\left(x; \mu_{s_x^{ph}}, \Sigma_{s_x^{ph}}\right) \tag{33}$$

  where s_x^{ph} denotes the Gaussian that corresponds to phoneme ph; $\mu_{s_x^{ph}}$, $\Sigma_{s_x^{ph}}$, and $p(s_x^{ph})$ are the mean, the diagonal covariance matrix, and the *a priori* probability associated with s_x^{ph}.
  Third approximation: PD-MEMLIN assumes that a clean feature vector can be approximated by a linear function that depends on the basic environment and the phoneme-dependent Gaussians of the clean and noisy models: $x \approx \Psi(y_t, s_x^{ph}, s_y^{e,ph}) = y_t - r_{s_x^{ph}, s_{y_j}^{e,ph}}$, where $r_{s_x^{ph}, s_{y_j}^{e,ph}}$ is a bias vector transformation between the clean and noisy feature vectors of each pair of Gaussians of the same phoneme, s_x^{ph} and $s_y^{e,ph}$.
- *PD-MEMLIN enhancement:* With those approximations, PD-MEMLIN transforms (2) into

$$\hat{x}_t = y_t - \sum_e \sum_{ph} \sum_{s_{y_j}^{e,ph}} \sum_{s_x^{ph}} r_{s_x^{ph}, s_{y_j}^{e,ph}} p(e|y_t) p(ph|y_t, e) \times p\left(s_y^{e,ph}|y_t, e, ph\right) p\left(s_x^{ph}|y_t, e, ph, s_y^e\right) \tag{34}$$

  where $p(e|y_t)$ is the *a posteriori* probability of the basic environment; $p(ph|y_t, e)$ is the *a posteriori* probability of the phoneme, given the noisy feature vector, y_t, and the basic environment, e; $p\left(s_y^{e,ph}|y_t, e, ph\right)$ is the *a posteriori* probability of the phoneme-dependent Gaussian of the noisy model, $s_y^{e,ph}$, given the noisy feature vector, y_t, the basic environment, e, and the phoneme, ph. Finally, $p\left(s_x^{ph}|y_t, e, ph, s_y^{e,ph}\right)$ is the cross-probability between the phoneme-dependent Gaussians of the clean and noisy models, given the noisy feature vector, y_t, the basic environment, e, and the phoneme, ph. That term and the bias vector transformation $r_{s_x^{ph}, s_{y_j}^{e,ph}}$, are estimated using stereo data in the training phase.
  The *a posteriori* probability of the basic environment, $p(e|y_t)$, is computed iteratively by applying (30) and (31) as the same way as (14) considering all the phonemes.
  The *a posteriori* probability of the phoneme ph, given the noisy feature vector, y_t, and the basic environment, e, $p(ph|y_t, e)$, can be computed using (30) and (31)

$$p(ph|y_t, e) = \frac{p_{e,ph}(y_t)}{\sum_{ph} p_{e,ph}(y_t)}. \tag{35}$$

  The *a posteriori* probability of the phoneme-dependent Gaussian of the noisy model, $s_y^{e,ph}$, given the noisy feature vector, y_t, the basic environment, e, and the phoneme, ph, $p\left(s_y^{e,ph}|y_t, e, ph\right)$, is computed using (30) and (31) as the same way as (15) considering the different phonemes.

- *PD-MEMLIN training:* Using clean training feature vectors, a forced Viterbi segmentation in phonemes is used to get a stereo data corpus for each basic environment and phoneme $(X_{e,ph}, Y_{e,ph}) = \left\{\left(x_1^{e,ph}, y_1^{e,ph}\right); \ldots; \left(x_{t_{e,ph}}^{e,ph}, y_{t_{e,ph}}^{e,ph}\right); \ldots; \left(x_{T_{e,ph}}^{e,ph}, y_{T_{e,ph}}^{e,ph}\right)\right\}$, with $t_{e,ph} \in [1, T_{e,ph}]$. The bias vector transformation, $r_{s_x^{ph}, s_y^{e,ph}}$, is estimated by minimizing the defined mean weighted square error, $\xi_{s_x^{ph}, s_y^{e,ph}}$, with respect to $r_{s_x^{ph}, s_y^{e,ph}}$, as shown by (36) and (37) at the bottom of the page, where $p\left(s_x^{ph} | x_{t_{e,ph}}^{e,ph}, e, ph\right)$ is the *a posteriori* probability of the phoneme-dependent Gaussian of the clean model, s_x^{ph}, given the clean feature vector, $x_{t_{e,ph}}^{e,ph}$, the basic environment, e, and the phoneme, ph. It can be computed by applying (32) and (33)

$$p\left(s_x^{ph} | x_{t_{e,ph}}^{e,ph}, e, ph\right) = \frac{p\left(x_{t_{e,ph}}^{e,ph} | s_x^{ph}\right) p\left(s_x^{ph}\right)}{\sum\limits_{s_x^{ph}} p\left(x_{t_{e,ph}}^{e,ph} | s_x^{ph}\right) p\left(s_x^{ph}\right)}. \quad (38)$$

The cross-probability between the phoneme-dependent Gaussians of the clean and noisy models is simplified by avoiding the time dependence given by the noisy feature vector y_t, $p\left(s_x^{ph} | y_t, e, s_y^{e,ph}, ph\right) \simeq p\left(s_x^{ph} | e, s_y^{e,ph}, ph\right)$. There are two ways to compute $p\left(s_x^{ph} | e, s_y^{e,ph}, ph\right)$: using relative frequency (hard solution), which expression is

$$p\left(s_x^{ph} | y_t, e, s_y^{e,ph}, ph\right) \simeq p\left(s_x^{ph} | s_y^{e,ph}, e, ph\right) = \frac{C_N\left(s_x^{ph} | s_y^{e,ph}\right)}{N_{s_y^{e,ph}}} \quad (39)$$

where $C_N\left(s_x^{ph} | s_y^{e,ph}\right)$ is the count number of times that the most probable pair of Gaussians is s_x^{ph}, and $s_y^{e,ph}$ for all pairs of stereo training data of e basic environment and ph phoneme, and $N_{s_y^{e,ph}}$ is the count number of times that the most probable Gaussian for noisy training feature vectors is $s_y^{e,ph}$ for e basic environment and ph phoneme.

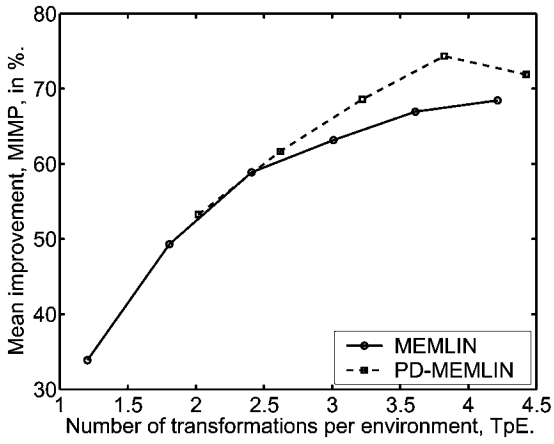


Fig. 4. Mean improvement in WER of MEMLIN and PD-MEMLIN.

The soft solution can be obtained using (30), (31), (32), and (33) as shown by (40) at the bottom of the page.

Since it is possible that some phonemes do not have associated enough data, all the experiments were carried out applying the soft solution.

2) Results From PD-MEMLIN: The same experiments defined in Section III were performed again, normalizing the noisy feature vectors with PD-MEMLIN. Bias vector transformations were obtained for all the 25 Spanish phonemes and the silence. Although only some of the phonemes would be necessary in this task, all of them were included in the normalization process. Fig. 4 presents the mean improvement in WER in percent of PD-MEMLIN comparing to MEMLIN. To make a fair comparison between two methods, the results have been plotted as a

$$\begin{aligned}\xi_{s_x^{ph}, s_y^{e,ph}} &= \sum_{t_{e,ph}} p\left(s_x^{ph} | x_{t_{e,ph}}^{e,ph}, e, ph\right) p\left(s_y^{e,ph} | y_{t_{e,ph}}^{e,ph}, e, ph\right) \\ &\quad \times \left(x_{t_{e,ph}}^{e,ph} - y_{t_{e,ph}}^{e,ph} + r_{s_x^{ph}, s_y^{e,ph}}\right)^2 \quad (36)\\ r_{s_x^{ph}, s_y^{e,ph}} &= \arg\min_{r_{s_x^{ph}, s_y^{e,ph}}} \left(\xi_{s_x^{ph}, s_y^{e,ph}}\right) \\ &= \frac{\sum\limits_{t_{e,ph}} p\left(s_x^{ph} | x_{t_{e,ph}}^{e,ph}, e, ph\right) p\left(s_y^{e,ph} | y_{t_{e,ph}}^{e,ph}, e, ph\right) \left(y_{t_{e,ph}}^{e,ph} - x_{t_{e,ph}}^{e,ph}\right)}{\sum\limits_{t_{e,ph}} p\left(s_x^{ph} | x_{t_{e,ph}}^{e,ph}, e, ph\right) p\left(s_y^{e,ph} | y_{t_{e,ph}}^{e,ph}, e, ph\right)} \quad (37)\end{aligned}$$

$$\begin{aligned}p\left(s_x^{ph} | y_t, e, ph, s_y^{e,ph}\right) &\simeq p\left(s_x^{ph} | s_y^{e,ph}, e\right) \\ &= \frac{\sum\limits_{t_{e,ph}} p\left(x_{t_{e,ph}}^{e,ph} | s_x^{ph}\right) p\left(y_{t_{e,ph}}^{e,ph} | s_y^{e,ph}\right) p\left(s_x^{ph}\right) p\left(s_y^{e,ph}\right)}{\sum\limits_{t_{e,ph}} \sum\limits_{s_x^{ph}} p\left(x_{t_{e,ph}}^{e,ph} | s_x^{ph}\right) p\left(y_{t_{e,ph}}^{e,ph} | s_y^{e,ph}\right) p\left(s_x^{ph}\right) p\left(s_y^{e,ph}\right)}. \quad (40)\end{aligned}$$

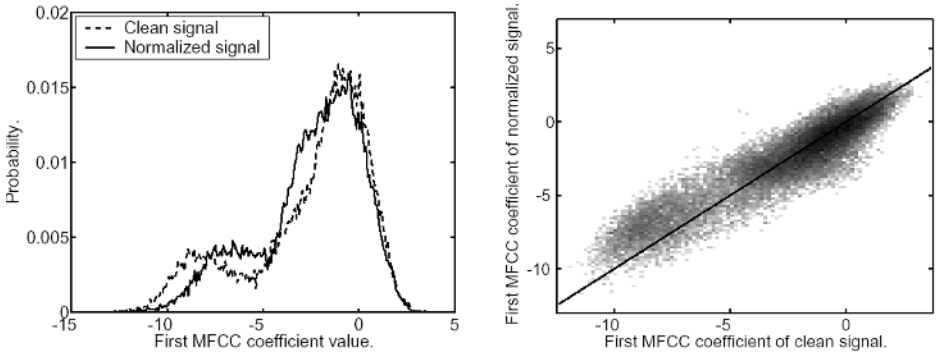


Fig. 5. Scattergram and histogram between the first MFCC coefficient in non-silence frames for clean (x-axis) and normalized using PD-MEMLIN with 16 Gaussians per phoneme (y-axis) signals from the E4 basic environment. The line in the scattergram represents the function $x = y$.

TABLE V
MWER AND MIMP IN PERCENT OF THE DIFFERENT BASIC ENVIRONMENTS FOR THE CLK AND NORMALIZED ONE WITH KNOWN PD-MEMLIN WITH 16 GAUSSIANS PER BASIC ENVIRONMENT

	E1	E2	E3	E4	E5	E6	E7	MWER (%)	MIMP (%)
CLK	0.38	2.06	1.40	0.50	0.57	0.16	0.00	0.86	–
KPD-MEMLIN 16-16	0.58	2.23	1.81	1.00	0.57	0.16	0.00	1.05	98.19

function of the number of Transformations per basic Environment (*TpE*), which each method has to compute for each frame in normalization, in $\log_{10}$

$$TpE = \log_{10}\left(n_{s_y^{ph}} n_{s_x^{ph}} n_{ph}\right) \tag{41}$$

where $n_{s_y^{ph}}$ and $n_{s_x^{ph}}$ are the number of noisy and clean model Gaussians for ph phoneme, respectively, and n_{ph} is the number of phonemes ($n_{ph} = 1$, for MEMLIN). In this paper, all of the phonemes have the same number of clean and noisy model Gaussians per basic environment: 2, 4, 8, 16 or 32.

The results show that PD-MEMLIN makes significant improvements relative to MEMLIN, specially when more than four Gaussians per phoneme are used ($TpE = 2.62$). Fig. 5 shows the histogram and scattergram of the first MFCC coefficient in non-silence frames for clean and normalized data using PD-MEMLIN with 16 Gaussians per basic environment for the E4 basic environment. From this figure, we can conclude that the transformations proposed by PD-MEMLIN solve the problem of mapping the noisy feature vectors towards the clean silence as in MEMLIN. PD-MEMLIN reduces the mapping space at the level of the phonemes, adapting in a better way the bias vector transformations to the acoustic models.

To estimate the limit of the PD-MEMLIN approximation, a new experiment was performed. Each frame was normalized using only the bias vector transformations of the "correct" phoneme $\hat{ph}$, which is obtained using a forced Viterbi segmentation in phonemes on the clean testing feature vectors. That pseudomethod is called known PD-MEMLIN (KPD-MEMLIN), and (34) is transformed into (42)

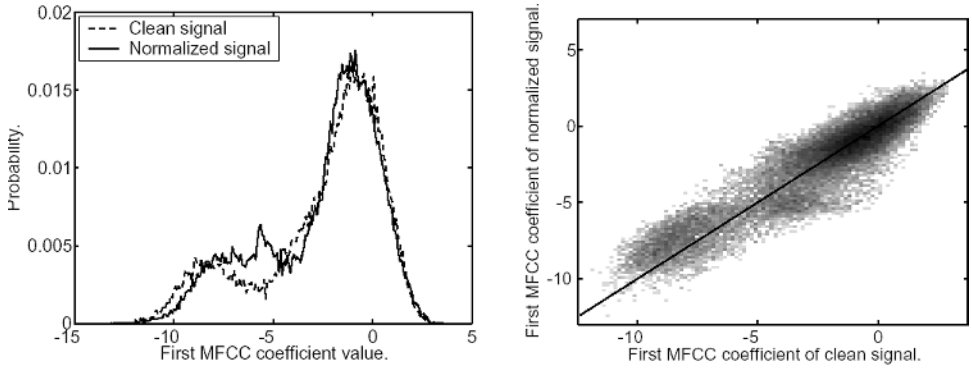
$$\hat{x}_t = y_t - \sum_e \sum_{s_y^{e,\hat{ph}}} \sum_{s_x^{\hat{ph}}} r_{s_x^{\hat{ph}}, s_y^{e,\hat{ph}}} p(e|y_t) \times p\left(s_y^{e,\hat{ph}}|y_t, e, \hat{ph}\right) p\left(s_x^{\hat{ph}}|y_t, e, \hat{ph}, s_y^e\right). \tag{42}$$

Table V shows the results for clean signal (CLK) and KPD-MEMLIN with 16 Gaussians per phoneme. The scattergram and the histogram between the first MFCC coefficient in non-silence frames for clean and normalized using KPD-MEMLIN with 16 Gaussians per phoneme are presented in Fig. 6.

Table V and Fig. 6 indicate an improvement of almost 100%, while the uncertainty between clean and normalized feature vectors using KPD-MEMLIN is not reduced significantly. Therefore, the phoneme-dependent normalization maps the noisy feature vectors inside the own uncertainty of the phonemes, which are modeled by the acoustic models. This fact can be confirmed by computing the mean correct phoneme (MCP) recognition rate. For this purpose, the correct phoneme sequence is the one obtained by forced Viterbi segmentation over clean signal using the clean acoustic models. For each normalized feature vector, the most probable phoneme is obtained using the clean phoneme-dependent GMMs. The MCP rate is computed as the rate of correct phonemes over all the testing utterances. Table VI shows the MCP rates for PD-MEMLIN and KPD-MEMLIN with 16 Gaussians per phoneme and for all of the basic environments. KPD-MEMLIN matches the frames with the correct phoneme much better than does PD-MEMLIN, increasing the average MCP more than 10%.

From Tables V and VI, we conclude that the proposed transformations associated to the different phonemes are consistent, because the feature vectors are mapped from the noisy space to the space associated to the forced clean phonemes. Therefore, it provides a future line of research which consists on estimating in a better way the *a posteriori* probability of the phoneme, ph, given the noisy feature vector, y_t, and the basic environment, e, $p(ph|y_t, e)$.

3) "Blind" PD-MEMLIN: In many cases, stereo data are not available; therefore, an iterative "blind" training procedure is needed. As PD-MEMLIN results are better than any other



(a.1) (a.2)

Fig. 6. Scattergram and histogram between the first MFCC coefficient in non-silence frames for clean signals (x-axis) and signals normalized using KPD-MEMLIN with 16 Gaussians per phoneme (y-axis) from the E4 basic environment. The line in the scattergram represents the function $x = y$.

TABLE VI
MCP RECOGNITION RATE IN PERCENT FOR NORMALIZED NON-SILENCE SIGNALS USING PD-MEMLIN AND KNOWN PD-MEMLIN WITH 16 GAUSSIANS PER PHONEME IN EACH OF THE SEVEN BASIC ENVIRONMENTS

MCP (%)	E1	E2	E3	E4	E5	E6	E7	Mean
PD-MEMLIN 16-16	32.64	31.23	30.38	32.54	32.04	34.14	31.21	32.03
KPD-MEMLIN 16-16	37.68	40.15	39.87	43.06	45.15	48.35	50.28	42.42

considered feature vector normalization method, we propose a "blind" training procedure for PD-MEMLIN. The expressions for MEMLIN can be obtained directly from the "blind" PD-MEMLIN ones.

Let us assume that the noisy training feature vectors and the phoneme-dependent clean and noisy GMMs are available. So, the problem is to estimate the cross-probability between the phoneme-dependent Gaussians of the clean and noisy models, $p\left(s_x^{ph}|s_y^{e,ph},e,ph\right)$, and the bias vector transformation, $r_{s_x^{ph},s_y^{e,ph}}$, without the clean part of the training stereo data. The proposed iterative "blind" training procedure consists of an initialization and an iterative process.

In the initialization, $p_0\left(s_x^{ph}|s_y^{e,ph},e,ph\right)$ and $r_{0,s_x^{ph},s_y^{e,ph}}$ are obtained. $p_0\left(s_x^{ph}|s_y^{e,ph},e,ph\right)$ is estimated using a modified Kullback–Liebler distance [24], which gives a similarity measure of s_x^{ph} and $s_y^{e,ph}$ without considering the effects of the noise. For initialization purposes, we assume that the noise modifies mainly the mean vectors of the Gaussian models. So, the similarity between Gaussians is computed in terms of the *a priori* probabilities and the diagonal covariance matrices of the corresponding Gaussians. Thus, given the phoneme-dependent Gaussians of the clean and noisy models, $s_y^{e,ph}$ and s_x^{ph}, the modified Kullback–Liebler distance $d_{\mathrm{KL}}\left(s_y^{e,ph},s_x^{ph}\right)$ can be computed as follows:

$$\begin{aligned} d_{\mathrm{KL}}&\left(s_y^{e,ph},s_x^{ph}\right) \\ &= \frac{p\left(s_y^{e,ph}\right)}{2} \\ &\quad\times\sum_i\left[log\left(\frac{\Sigma_{s_x^{ph}}(i,i)}{\Sigma_{s_y^{e,ph}}(i,i)}\right)+\frac{\Sigma_{s_y^{e,ph}}(i,i)}{\Sigma_{s_x^{ph}}(i,i)}-1\right] \\ &\quad+p\left(s_y^{e,ph}\right)\log\left(\frac{p\left(s_y^{e,ph}\right)}{p\left(s_x^{ph}\right)}\right) \end{aligned} \tag{43}$$

where $\Sigma_{s_x^{ph}}(i,i)$ and $\Sigma_{s_y^{e,ph}}(i,i)$ are the ith term of the diagonal covariance matrices of the s_x^{ph} and the $s_y^{e,ph}$ Gaussians.

The modified Kullback–Liebler distance is not symmetric, and it is not proportional to the likelihood; therefore, a pseudolikelihood $pl_{\mathrm{KL}}\left(s_y^{e,ph},s_x^{ph}\right)$ is defined

$$pl_{\mathrm{KL}}\left(s_y^{e,ph},s_x^{ph}\right)=\frac{1}{d_{\mathrm{KL}}\left(s_y^{e,ph},s_x^{ph}\right)+d_{\mathrm{KL}}\left(s_x^{ph},s_y^{e,ph}\right)}. \tag{44}$$

Finally, $p_0\left(s_x^{ph}|s_y^{e,ph},e,ph\right)$ is estimated as

$$p_0(s_x^{ph}|s_y^{e,ph},e,ph)=\frac{pl_{\mathrm{KL}}\left(s_y^{e,ph},s_x^{ph}\right)}{\sum\limits_{s_x^{ph}}pl_{\mathrm{KL}}\left(s_y^{e,ph},s_x^{ph}\right)}. \tag{45}$$

On the other hand, $r_{0,s_x^{ph},s_y^{e,ph}}$ is obtained replacing $x_{t_{e,ph}^{e,ph}}$ with $\mu_{s_x^{ph}}$ in (37)

$$r_{0,s_x^{ph},s_y^{e,ph}}=\frac{\sum\limits_{t_{e,ph}}p\left(s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},e,ph\right)\left(y_{t_{e,ph}}^{e,ph}-\mu_{s_x^{ph}}\right)}{\sum\limits_{t_{e,ph}}p\left(s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},e,ph\right)}. \tag{46}$$

A very simple recognition experiment with phoneme acoustic models was carried out, normalizing the noisy signal with $p_0\left(s_x^{ph}|s_y^{e,ph},e,ph\right)$ and $r_{0,s_x^{ph},s_y^{e,ph}}$ and four Gaussians per phoneme-dependent GMM. The mean improvement in WER over the seven basic environments was 20.2%.

Once $r_{0,s_x^{ph},s_y^{e,ph}}$ is computed, $r_{s_x^{ph},s_y^{e,ph}}$ can be estimated iteratively by the EM [25] algorithm in a similar way as [7] (see Appendix I). In this case, the corresponding expression for the nth iteration $r_{n,s_x^{ph},s_y^{e,ph}}$ with $n>0$, is shown in (47) and (48) at the bottom of the next page.

The same simple recognition experiment with phoneme acoustic models was performed, normalizing the noisy signal with $p_0\left(s_x^{ph}|s_y^{e,ph},e,ph\right)$ and $r_{n,s_x^{ph},s_y^{e,ph}}$ and four Gaussians per phoneme-dependent GMM. The mean improvement in WER in this case was 41.03% if $n=1$ and 46.90% if $n=10$.

To improve the estimation of $p_0\left(s_x^{ph}|s_y^{e,ph},e,ph\right)$ pseudostereo data are obtained normalizing the noisy training feature vectors with KPD-MEMLIN. In this case, the phoneme

associated with each noisy training feature vector $\hat{ph}$ is estimated using a forced Viterbi segmentation of noisy training utterances. With the pseudostereo data, $(\hat{X}_{e,ph}, Y_{e,ph}) = \left\{\left(\hat{x}_1^{e,ph}, y_1^{e,ph}\right); \ldots; \left(\hat{x}_{t_{e,ph}}^{e,ph}, y_{t_{e,ph}}^{e,ph}\right); \ldots; \left(\hat{x}_{T_{e,ph}}^{e,ph}, y_{T_{e,ph}}^{e,ph}\right)\right\}$, where $\hat{x}_{t_e^{ph}}^{e,ph}$ is the normalized feature vector of $y_{t_e^{ph}}^{e,ph}$, a new iteration for $p\left(s_x^{ph}|s_y^{e,ph}, e, ph\right)$ can be estimated using (39) or (40).

Using one iteration of the EM algorithm to estimate $r_{s_x^{ph}, s_y^{e,ph}}$ and another one to compute $p\left(s_x^{ph}|s_y^{e,ph}, e, ph\right)$ with pseudostereo data, the mean improvement in WER with phoneme acoustic models and four Gaussians per phoneme-dependent GMM was 50.23%. As the use of pseudostereo data produces significant improvements, they can also be used to adjust the estimation of $r_{s_x^{ph}, s_y^{e,ph}}$ using (37). So, if $p\left(s_x^{ph}|s_y^{e,ph}, e, ph\right)$ is estimated with three iterations and the first iteration of $r_{s_x^{ph}, s_y^{e,ph}}$ with the EM algorithm, $r_{1, s_x^{ph}, s_y^{e,ph}}$ is tuned with two additionally iterations with pseudostereo data, the mean improvement in WER with phoneme acoustic models and four Gaussians per phoneme-dependent GMM was 58.68%. These results show that the use of the EM algorithm and the pseudostereo data jointly produces important improvements. "Blind" MEMLIN training procedure can be developed in the same way as PD-MEMLIN, avoiding the phoneme dependence.

4) Results From "Blind" PD-MEMLIN: The experiments in Section III were performed again, using "blind" PD-MEMLIN. The mean improvement in WER in percent of "blind" PD-MEMLIN compared to PD-MEMLIN and MEMLIN is presented in Fig. 7. Three iterations with pseudostereo data were needed for $p\left(s_x^{ph}|s_y^{e,ph}, e, ph\right)$, and $r_{s_x^{ph}, s_y^{e,ph}}$ was estimated with two iterations with pseudostereo data, once $r_{1, s_x^{ph}, s_y^{e,ph}}$ had been computed with the EM algorithm.

The results show that "blind" PD-MEMLIN is able to produce improvements that are very similar to MEMLIN ones for all the TpE.

The most representative results from each of the improvement methods over MEMLIN are summarized in Table VII, indicating the TpE required to obtain the best corresponding values. It can be observed that PD-MEMLIN obtains the best improvement with the smallest TpE.

V. Discussion and Conclusion

In this paper, some basic methods of feature vector normalization based on MMSE estimator and stereo data, such as RATZ, SPLICE, and our proposed technique MEMLIN, have been explained and compared using real car noise conditions from the SpeechDat Car database. With respect to RATZ and SPLICE, MEMLIN proposes modeling clean and noisy spaces with GMMs, learning a bias vector transformation for each pair of Gaussians (one for the clean GMM and the other one for the noisy GMM). MEMLIN produces results that are significantly better than those obtained using other methods. MEMLIN produces a mean improvement in WER of 69.09%, far away from 48.82% of Interpolated RATZ and better than 64.46% of SPLICE with environmental model selection.

Further improvements have been considered using first-order polynomial and a nonlinear function for each pair of Gaussians instead of the bias vector transformation in MEMLIN. The new methods are called P-MEMLIN and MEMHIN, respectively. Both methods compensate for the effects of the noise in the mean and the variance of the feature vectors. The results show an

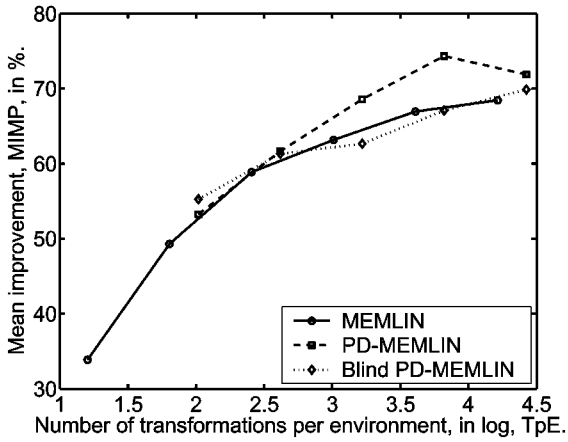


Fig. 7. Mean improvement in WER of MEMLIN, PD-MEMLIN, and "blind" PD-MEMLIN.

TABLE VII
Best MWER and MIMP in Percent From MEMLIN, MEMHIN, P-MEMLIN, PD-MEMLIN, and "Blind" PD-MEMLIN, With The Required TpE Indicated

	TpE	MWER (%)	MIMP (%)
MEMLIN	4.21	4.16	69.09
MEMHIN	4.21	4.26	68.09
P-MEMLIN	4.21	4.14	69.24
PD-MEMLIN	3.82	3.60	74.32
"Blind" PD-MEMLIN	4.42	4.07	69.88

$$r_{n,s_x^{ph}, s_y^{e,ph}} = \frac{\sum_{t_{e,ph}} p\left(s_y^{e,ph}|y_{t_{e,ph}}^{e,ph}, e, ph\right) p\left(s_x^{ph}|y_{t_{e,ph}}^{e,ph}, s_y^{e,ph}, n-1\right)\left(y_{t_{e,ph}}^{e,ph} - \mu_{s_x^{ph}}\right)}{\sum_{t_{e,ph}} p(s_y^{e,ph}|y_{t_{e,ph}}^{e,ph}, e, ph) p\left(s_x^{ph}|y_{t_{e,ph}}^{e,ph}, s_y^{e,ph}, n-1\right)} \tag{47}$$

$$p\left(s_x^{ph}|y_{t_{e,ph}}^{e,ph}, s_y^{e,ph}, n-1\right) = \frac{N(y_{t_{e,ph}}; \mu_{s_x^{ph}} + r_{n-1,s_x^{ph}, s_y^{e,ph}}, \Sigma_{s_y^{e,ph}}) p\left(s_y^{e,ph}\right)}{\sum_{s_x^{ph}} N\left(y_{t_{e,ph}}; \mu_{s_x^{ph}} + r_{n-1,s_x^{ph}, s_y^{e,ph}}, \Sigma_{s_y^{e,ph}}\right) p\left(s_y^{e,ph}\right)} \tag{48}$$

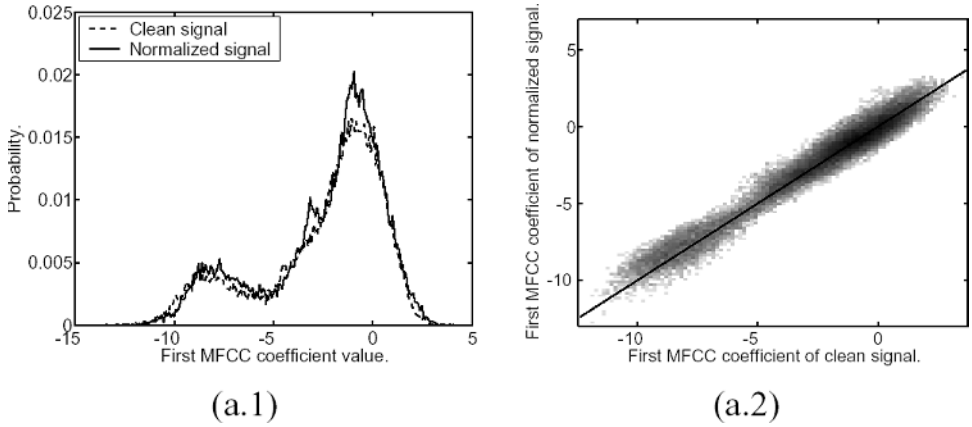


Fig. 8. Scattergram and histogram between the first MFCC coefficient in non-silence frames for clean (x-axis) and normalized using MEMLIN with 128 Gaussians per basic environment, where $p\left(s_x|s_y^e, e\right)$ is computed with clean signal as $p(s_x|x_t)$. The line in the scattergram represents the function $x = y$.

TABLE VIII
MEAN MWER AND MIMP IN PERCENT FROM THE METHODS MEMLIN, MEMHIN, PD-MEMLIN, AND "BLIND" PD-MEMLIN WITH ML-ADAPTED ACOUSTIC MODELS TO THE NORMALIZED SPACE. THE NUMBER OF GAUSSIANS PER BASIC ENVIRONMENT ARE INDICATED BESIDE THE NAME OF EACH NORMALIZATION METHOD

	MWER (%)	MIMP (%)
CLK	0.86	–
HF	3.95	71.05
† HF	2.82	81.59
MEMLIN 128-128 + ML	3.70	73.33
MEMHIN 128-128 + ML	3.40	76.15
PD-MEMLIN 32-32 + ML	3.09	79.10
"Blind" PD-MEMLIN 32-32 + ML	2.72	82.55

improvement concerning MEMLIN when less number of Gaussians is used. When the number of Gaussians increases, the improvements in WER are very similar (68.09% and 69.24% for MEMHIN and P-MEMLIN with 128 Gaussians, respectively).

MEMLIN, P-MEMLIN and MEMHIN allow mapping from any noisy GMM Gaussian towards any clean GMM Gaussian. PD-MEMLIN has been developed to constrain the mapping space in terms of the acoustic models. So, noisy and clean spaces are split into phonemes and the transformations are only possible between Gaussians of the same phoneme. The improvement in WER of PD-MEMLIN is 74.32%. As in many cases stereo data are not available, a "blind" training procedure has been developed to estimate the needed variables for PD-MEMLIN without the clean part of the stereo data. The improvement in WER in this case reaches 69.88%, which is even better than MEMLIN. Furthermore, It can be observed that a perfect estimation of the *a posteriori* probability of the phoneme, given the noisy feature vector and the basic environment, in PD-MEMLIN (KPD-MEMLIN) can generate almost a 100% of improvement in WER, while the uncertainty between the clean feature vectors and the normalized ones is not reduced significantly.

Although the transformation is not perfect, the normalized feature vectors define a new normalized space more homogeneous than the noisy one. So, new acoustic models can be retrained with the normalized training data. The MWER and MIMP results are presented in Table VIII. It can be observed that the results for all techniques are better than the ones obtained with noisy acoustic models HF, and in some cases very similar to use specific acoustic models for each environment †HF.

In all the presented techniques, the estimation of the cross-probability model term, $p(s_x|y_t, e, s_y)$, in MEMLIN, P-MEMLIN and MEMHIN, and $p\left(s_x^{ph}|y_t, e, s_y^{e,ph}\right)$, in PD-MEMLIN, has a hugh impact on the final performance. A simple experiment approximating the cross-probability model using the clean feature vectors gives an improvement in WER close to 100%, and reducing dramatically the uncertainty between the clean feature vectors and the normalized ones, as shown in the Fig. 8. These results open a new line of future work, improving the estimation of the cross-probability model and the *a priori* probability of the phoneme, given the noisy feature vector and the basic environment.

APPENDIX
ESTIMATION OF $r_{s_x^{ph}, s_y^{e,ph}}$ BY THE EM ALGORITHM FOR "BLIND" PD-MEMLIN

We consider a set of noisy labeled feature vectors associated to a basic environment e, and a phoneme, ph, $Y_e^{ph} = \left\{y_1^{e,ph}; \ldots; y_{t_{e,ph}}^{e,ph}; \ldots; y_{T_{e,ph}}^{e,ph}\right\}$, with $t_{e,ph} \in [1, T_{e,ph}]$. Noisy and clean feature vectors are modeled with GMMs: (30), (31), (32), and (33). We assume that the pdf of the noisy feature vectors, given s_x^{ph}, $s_y^{e,ph}$, the e basic environment and the ph phoneme is

$$p\left(y_{t_{e,ph}}^{e,ph}|s_x^{ph}, s_y^{e,ph}, e, ph\right) = N\left(y_{t_{e,ph}}^{e,ph}; \mu_{s_x^{ph}} + r_{s_x^{ph}, s_y^{e,ph}}, \Sigma_{s_y^{e,ph}}\right). \quad \text{(A.1)}$$

The log-likelihood function, $L(Y_e^{ph})$ is

$$L\left(Y_e^{ph}\right) = \sum_{t_{e,ph}} \log \sum_{s_y^{e,ph}} \sum_{s_x^{ph}} p\left(s_x^{ph}, s_y^{e,ph}|e, ph\right) \times N\left(y_{t_{e,ph}}^{e,ph}; \mu_{s_x^{ph}} + r_{s_x^{ph}, s_y^{e,ph}}, \Sigma_{s_y^{e,ph}}\right) \quad \text{(A.2)}$$

where $p(s_x^{ph}, s_y^{e,ph}|e, ph)$ is the joint probability of the pair of Gaussians, given the basic environment e and the phoneme ph.

The auxiliary function $Q(\phi,\phi_{\text{new}})_e^{ph}$, with $\phi=\left\{r_{s_x^{ph},s_y^{e,ph}}\right\}$ is defined as

$$Q(\phi,\phi_{\text{new}})_e^{ph}=\sum_{t_{e,ph}}\sum_{s_y^{e,ph}}\sum_{s_x^{ph}}p\left(s_x^{ph},s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},\phi,e,ph\right)\times\log\left(p\left(y_{t_{e,ph}}^{e,ph},s_x^{ph},s_y^{e,ph}|\phi_{\text{new}},e,ph\right)\right). \quad \text{(A.3)}$$

Defining $\Omega=y_{t_{e,ph}}^{e,ph}-\mu_{s_x^{ph}}-r_{\text{new},s_x^{ph},s_y^{e,ph}}$, (A.3) is transformed into

$$\begin{aligned}&Q(\phi,\phi_{\text{new}})_e^{ph}\\&\quad=\text{constant}+\sum_{t_{e,ph}}\sum_{s_y^{e,ph}}\sum_{s_x^{ph}}\\&\qquad\times p\left(s_x^{ph},s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},\phi,e,ph\right)\\&\qquad\times\left(-\frac{1}{2}\log\left|\Sigma_{s_y^{e,ph}}\right|-\frac{1}{2}\Omega^T\Sigma_{s_y^{e,ph}}\Omega\right).\end{aligned} \quad \text{(A.4)}$$

The value of $r_{\text{new},s_x^{ph},s_y^{e,ph}}$ is obtained by taking derivatives and setting it equal to zero

$$\begin{aligned}&r_{\text{new},s_x^{ph},s_y^{e,ph}}\\&\quad=\frac{\delta\left(Q(\phi,\phi_{\text{new}})_e^{ph}\right)}{\delta\left(r_{\text{new},s_x^{ph},s_y^{e,ph}}\right)}\\&\quad=\sum_{t_{e,ph}}p\left(s_x^{ph},s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},\phi,e,ph\right)\\&\qquad\times\Sigma_{s_y^{e,ph}}\left(y_{t_{e,ph}}^{e,ph}-\mu_{s_x^{ph}}-r_{\text{new},s_x^{ph},s_y^{e,ph}}\right)=0\end{aligned} \quad \text{(A.5)}$$

$$\begin{aligned}&r_{\text{new},s_x^{ph},s_y^{e,ph}}\\&\quad=\frac{\sum_{t_{e,ph}}p\left(s_x^{ph},s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},\phi,e,ph\right)\left(y_{t_{e,ph}}^{e,ph}-\mu_{s_x^{ph}}\right)}{\sum_{t_{e,ph}}p\left(s_x^{ph},s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},\phi,e,ph\right)}\end{aligned} \quad \text{(A.6)}$$

where $p\left(s_x^{ph},s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},\phi,e,ph\right)$ can be computed as follows:

$$\begin{aligned}&p\left(s_x^{ph},s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},\phi,e,ph\right)\\&\quad\simeq p\left(s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},e,ph\right)\\&\qquad\times p\left(s_x^{ph}|y_{t_{e,ph}}^{e,ph},s_y^{e,ph},\phi,e,ph\right)\end{aligned} \quad \text{(A.7)}$$

$$\begin{aligned}&p\left(s_x^{ph},s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},\phi,e,ph\right)\\&\quad\simeq p\left(s_y^{e,ph}|y_{t_{e,ph}}^{e,ph},e,ph\right)\\&\qquad\times\frac{p\left(s_y^{e,ph}\right)p\left(y_{t_{e,ph}}^{e,ph}|s_x^{ph},s_y^{e,ph},\phi\right)}{\sum_{s_x^{ph}}p\left(s_y^{e,ph}\right)p\left(y_{t_{e,ph}}^{e,ph}|s_x^{ph},s_y^{e,ph},\phi\right)}.\end{aligned} \quad \text{(A.8)}$$

REFERENCES

[1] A. Sankar and C. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 3, pp. 190–202, May 1996.

[2] J. A. Nolazco-Flores and S. J. Young, "Continuous speech recognition in noise using spectral subtraction and HMM adaptation," in *Proc. ICASSP*, Adelaide, Australia, Apr. 1994, vol. 1, pp. 409–412.

[3] L. Neumeyer and M. Weintraub, "Robust speech recognition in noise using adaptation and mapping techniques," in *Proc. ICASSP*, Detroit, MI, May 1995, vol. 1, pp. 141–144.

[4] J.-L. Gauvain and C.-H. Lee, "Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 291–298, Apr. 1994.

[5] C. Leggeter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Comput. Speech Lang.*, vol. 9, pp. 171–185, 1995.

[6] M. J. F. Gales and S. J. Young, "Robust speech recognition in additive and convolutional noise using parallel model combination," *Comput. Speech Lang.*, vol. 9, pp. 289–307, 1995.

[7] P. Moreno, "Speech recognition in noisy environments," Ph.D. dissertation, Elect. Comput. Eng. Dep., Carnegie Mellon Univ., Pittsburgh, PA, 1996.

[8] R. M. Stern, B. Raj, and P. J. Moreno, "Compensation for environmental degradation in automatic speech recognition," in *ESCA Tutorial Res. Workshop Robust Speech Recognition for Unknown Commun. Channels.*, Pont-au-Mousson, France, Apr. 1997, pp. 33–42.

[9] N. Hanai and R. M. Stern, "Robust speech recognition in the automobile," in *Proc. ICSLP*, Yokohama, Japan, Sep. 1994, pp. 1339–1342.

[10] U. Yapanel, X. Zhang, and J. Hansen, "High performance digit recognition in real car environments," in *Proc. ICSLP*, Denver, CO, Sep. 2002, pp. 793–796.

[11] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 578–589, Oct. 1994.

[12] A. Acero, "Acoustical and environmental robustness in automatic speech recognition," Ph.D. dissertation, Elect. Comput. Eng. Dept., Carnegie Mellon Univ., Pittsburgh, PA, 1990.

[13] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square log-spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 2, pp. 443–445, Apr. 1985.

[14] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-27, no. 2, pp. 113–120, Apr. 1979.

[15] J. Droppo, L. Deng, and A. Acero, "Evaluation of the SPLICE algorithm on the AURORA2 database," in *Proc. Eurospeech*, Aalborg, Denmark, 2001, pp. 217–220.

[16] L. Neumeyer and M. Weintraub, "Probabilistic optimum filtering for robust speech recognition," in *Proc. ICASSP*, Adelaide, Australia, Apr. 1994, vol. 1, pp. 417–420.

[17] L. Buera, E. Lleida, A. Miguel, and A. Ortega, "Multi-environment models based linear normalization for robust speech recognition in car conditions," in *Proc. ICASSP*, Montreal, QC, Canada, May 2004, vol. 1, pp. 1013–1016.

[18] ——, "Multi-environment models based linear normalization for robust speech recognition," in *Proc. SPECOM*, Saint Petersburg, Russia, 2004, pp. 174–180.

[19] ——, "Robust speech recognition in cars using phoneme dependent multi-environment linear normalization," in *Proc. Interspeech*, Lisbon, Portugal, 2005, pp. 381–384.

[20] ——, "Recent advances in PD-MEMLIN for speech recognition in car conditions," in *Proc. ASRU*, San Juan, PR, Nov. 2005, pp. 180–185.

[21] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, S. Euler, and J. Allen, "Speechdat-car: A large speech database for automotive environments," in *Proc. LREC*, Athens, Greece, 2000, vol. 2, pp. 895–900.

[22] C. Mokbel and G. Chollet, "Automatic word recognition in cars," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 5, pp. 346–356, Sep. 1995.

[23] E. Lombard, "Le signe de l'elevation de la voix," in *Ann. Maladies Oreille, Larynx, Nez, Pharynx*, 1911, vol. 37, pp. 101–119.

[24] S. Kullback and R. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, pp. 79–87, 1951.

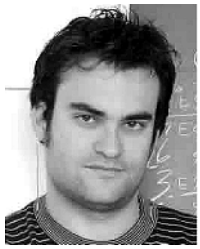
[25] A. P. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from imcomplete data via the em algorithm," *J. R. Statist. Soc.*, vol. 9, no. 1, pp. 1–37, 1977.

Luis Buera was born in Lleida, Spain, in 1978. He received the M.Sc. degree in telecommunication engineering from the University of Zaragoza (UZ), Zaragoza, Spain, in 2002. He is currently working towards the Ph.D. degree at UZ based on feature vector normalization techniques for robust speech recognition.

From 2002 to 2006, he was with the Communications Technologies Group, Department of Electronic Engineering and Communications, UZ, under a research grant.

Eduardo Lleida (M'89) was born in Spain in 1961. He received the M.Sc. degree in telecommunication engineering and the Ph.D. degree in signal processing from the Universitat Politècnica de Catalunya (UPC), Spain, in 1985 and 1990, respectively.

From 1986 to 1988, he was involved in his doctoral work at the Department of Signal Theory and Communications at UPC, Spain. From 1989 to 1990, he worked as an Assistant Professor, and from 1991 to 1993, he worked as an Associate Professor in the Department of Signal Theory and Communications, UPC, Spain. From February 1995 to January 1996, he was with AT&T Bell Laboratories, Murray Hill, NJ, as a Consultant in speech recognition. Currently, he is an Associate Professor of Signal Theory and Communications, Department of Electronic Engineering and Communications, University of Zaragoza, Zaragoza, Spain, where he is heading a research team in speech recognition and signal processing. He has been managing several speech-related projects in Spain. He has coauthored more than 100 technical papers in the field of speech and speaker recognition, speech enhancement and recognition in adverse acoustic enviroments, acoustic modeling, confidence measures, and spoken dialogue systems.



Antonio Miguel was born in Zaragoza, Spain, in 1977. He received the M.Sc. degree in telecommunication engineering from the University of Zaragoza (UZ), Zaragoza, Spain, in 2001. He is currently working towards the Ph.D. degree at UZ.

From 2000 to 2006, he was with the Communications Technologies Group, Department of Electronic Engineering and Communications, UZ, under a research grant. Since 2006, he has been an Assistant Professor in the same department. His research interest lies in the field of aucoustic modeling for ASR.

Alfonso Ortega was born in Teruel, Spain, in 1976. He received the M.Sc. degree in telecommunication engineering and the Ph.D. degree from the University of Zaragoza (UZ), Zaragoza, Spain, in 2000 and 2005, respectively.

In 1999 he joined, under a research grant, the Communications Technologies Group, UZ, where he has been an Assistant Professor since 2001. He is also involved as a Researcher with the Aragon Institute of Engineering Research (I3A). His research interest lies in the signal processing field applied to speech technologies.

Óscar Saz was born in Zaragoza, Spain, in 1980. He received the M.Sc. degree in telecommunication engineering from the University of Zaragoza (UZ), Zaragoza, Spain, in 2004. He is currently working towards the Ph.D. degree from UZ.

From 2004 to 2006, he has been with the Communications Technologies Group, Department of Electronic Engineering and Communications, UZ, under a research grant. His research interests are in the field of speaker adaptation and personalization of ASR systems, especially oriented to users with pathological speech.