INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY

CAMPUS MONTERREY

División de Tecnologías de Información y Electrónica

Robust Automatic Speech Recognition Employing Phoneme-Dependent Multi-Environment Enhanced Models based LInear Normalization

Presentada como requisito parcial para obtener el grado de

Maestría en Ciencias en Ingeniería Electrónica con Especialidad en Sistemas Electrónicos.

Por

Igmar Hernández Ochoa

Monterrey, N.L., Diciembre de 2006

Robust Automatic Speech Recognition Employing Phoneme-Dependent Multi-Environment Enhanced Models based LInear Normalization

por

Ing. Igmar Hernández Ochoa

Tesis

Presentada al Programa de Graduados de la

División de Tecnologías de Información y Electrónica

como requisito parcial para obtener el grado académico de

Maestro en Ciencias

especialidad en

Sistemas Electrónicos.

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Diciembre de 2006

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

División de Tecnologías de Información y Electrónica

Programa de Graduados

Los miembros del comité de tesis recomendamos que la presente tesis de Igmar Hernández Ochoa sea aceptada como requisito parcial para obtener el grado académico de **Maestro en Ciencias**, especialidad en:

Sistemas Electrónicos.

Comité de tesis:

Juan Arturo Nolazco Flores, Ph.D.
Asesor de la tesis

Ing. Luis Ricardo Salgado Garza
Sinodal

José Ramón Rodríguez Cruz, Ph.D.
Sinodal

Graciano Dieck Asad, Ph.D.
Director del Programa de Graduados

Diciembre de 2006

Dedico esta tésis a mis padres, los cuales me dieron el principal recurso para llegar hasta aquí, la vida. Además me brindaron todo su apoyo, me animaron y me convencieron de que yo era capaz de culminar con éxito este nuevo capítulo en mi vida. A mis hermanos y sus familias, que me dieron la fuerza necesaria para poder continuar cada vez que nos veíamos. Gracias por apoyarme aquí en Monterrey y desde Chihuahua. Y a Sara, por haber tenido la paciencia y comprensión durante todo este largo periodo.

Reconocimientos

A mi asesor Dr. Juan Arturo Nolazco Flores, por brindarme la oportunidad de aportar un granito de arena al área de investigación, por su apoyo y por compartir sus conocimientos.

A mis sinodales Ing. Luis Ricardo Salgado Garza y Dr. Juan Ramón Rodríguez Cruz, por haber sido unos excelentes maestros durante mi estancia en el tec, por su ayuda y acertados comentarios.

A Leibny Paola, por haber salvado mi vida en toda ocasión, por estar disponible en todo momento y por nunca decirme que no. Además por mostrarme cual era mi límite y demostrarme que era posible llegar más lejos.

A mis amigos del departamento Octaviano, Margarita, Isai, Ulises, Enrique, Tambor, Tere, Brenda, Alfonso, Raúl y Adriana, por todos esos momentos de alegría y tranquilidad que me brindaron.

A mis roomies Enrique Manuel Valenzuela Phillips y José Luis Montes Marrero, por que cuidaron la casa por mi durante todo mi último semestre. También por su apoyo, preocupación y ayuda que me han brindado.

A mi amigo Diego Valenzuela y su esposa Giselle Akel por todo lo que compartimos antes de su partida.

Igmar Hernández Ochoa

Instituto Tecnológico y de Estudios Superiores de Monterrey
Diciembre 2006

Robust Automatic Speech Recognition Employing Phoneme-Dependent Multi-Environment Enhanced Models based LInear Normalization

Igmar Hernández Ochoa, M.C.
Instituto Tecnológico y de Estudios Superiores de Monterrey, 2006

Asesor de la tesis: Juan Arturo Nolazco Flores, Ph.D.

This work shows a robust normalization technique by cascading a speech enhancement method followed by a feature vector normalization algorithm. An efficient scheme used to provide speech enhancement is the Spectral Subtraction algorithm, which reduces the effect of additive noise by performing a subtraction of noise spectrum estimate over the complete speech spectrum. On the other hand, a new and promising technique known as PD-MEMLIN (Phoneme-Dependent Multi-Enviroment Models based LInear Normalization) has also shown to be effective. PD-MEMLIN is an empirical feature vector normalization which models clean and noisy spaces by Gaussian Mixture Models (GMMs), and estimates the different compensation linear transformation to be performed to clean the signal. In this work the integration of both approaches is proposed. The final design is called PD-MEEMLIN (Phoneme-Dependent Multi-Enviroment Enhanced Models based LInear Normalization), which confirms and improves the effectivness of both approaches. The results obtained show that in very high degraded speech (between -5dB and 0dB) PD-MEEMLIN outperforms the SS by a range between 11.4% and 34.5%,for PD-MEMLIN by a range between 11.7% and 24.84%, and for SPLICE by a range between 6.04% and 22.23%. Furthemore, in moderate SNR, i.e. 15 or 20 dB, PD-MEEMLIN is as good as PD-MEMLIN and SS techniques.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Automatic Speech Recognition (ASR) is a field that has evolved due to the human interest on finding an artificial method capable to understand what it is spoken. The speech, itself, is a natural human process in two ways: producing and decoding. However, it is very diffcult for a machine to perform those tasks. The research on Automatic Speech Recognition (ASR) tries to find a suitable method for a computer to identify words uttered by a person, and also performs an identification of the complete expressed sentence.

The state of the art of ASR shows that the main problems are in the speech variability among users, in the utterance properties (spontaneous speech, coarticulation, etc), vocabulary extension and complexity, and environmental conditions. From all of them, the environmental conditions such as noise and robust achievement were chosen as the main theme of this thesis. The speech recognition process when working in real conditions it is affected by the additive and convolutional noise. Despite this types of noise, the disturbed speech signal should be recognized by the ASR system. This is the main reason of the increasing development or improvement of robust speech recognition techniques.

The development of robust speech recognition systems plays a key rule in real environment applications. This was noticed several decades ago, when Spectral Subtraction (SS) [1], and other techiques were developed. Nowadays, the development of robust ASR systems still remains as an issue, and new methods have been developed, for example SPLICE (State Based Piecewise Linear Compensation for Environments) [2], PMC (Parallel Model Combination) [3], RATZ (Multivariate Gaussian Based Cepstral Normalization) [4], ALGONQUIN [6] and RASTA (The Relative Spectral Technique) [7].

When the training environment is different to the testing environment, it is necesary to make some adjustments to the signal to compensate the mismatch. The algorithms that try to solve this problem can be grouped in two main categories: model adaptation and feature compensation. In speech model adaptation techniques, such as PMC [3], the speech model is adapted to the noisy conditions. In feature vector normalization techniques such as MEMLIN (Multi-Environment Models Based LInear Normalization) [10] and PD-MEMLIN (Phoneme-Dependent Multi-Environment Models Based LInear Normalization) [11], clean and noisy speech spaces are modeled by Gaussian Mixture Models (GMMs).

This research follows the above trends and proposes to make a combination of techniques in order to reduce the noise effects. The architecture philosophy expressed in [8] clearly shows this integration. The core scheme composed of a Continuous Spectral Subtraction (CSS) and a Parallel Model Combination (PMC) is employed [3]. Persuing the same idea, a combination of the SS method and PD-MEMLIN [10] is presented in this document.

1.1 Applications of Speech Robustness

The noise suppresion has applications in virtually any field of communications (channel equalization) and fields like pattern analysis or data forecasting.

Nowadays, a common application is the reduction of background acoustic noise in cellular phones. A very natural situation would be the hands-free operation of a cellular phone while a person is driving. The fact that the vehicle is moving provides a contaminated environment with the engine noise, the wind and the traffic on the road. Another application is the communication between passengers inside a car. In this situtation, the same factors that disturbed the speech signal are present. The driver needs to turn the head to make possible that the passengers can hear.

In these applications, the general purpose is to maintain good recognition accuracy even with the quality degradation of the speech signal. It is important to note that that these situations demand an even greater degree of environmental robustness. That is the reason why many approaches have been developed to address the problem described by the applications.

These two applications are merly given as examples, because they are daily things for most of people. Giving a better description of the approach of the research developed.

1.2 Problem Definition

When a speech signal is employed in high-noise environment conditions, the noise becomes an important factor to deal with. There are some levels of noise that are tolerated, but when the SNR is between 5dB and -5dB the environment is highly degraded. In these conditions the ASR system has problems to identify the elements in the speech signal that it is analyzed.

1.3 Objective

The SS method has shown to obtain improvements in the ASR tasks in noisy conditions. This techinque is based on the noise spectrum estimation, which is subtracted from the original noisy signal. On the other hand, the PD-MEMLIN compensates the mismatch between the noisy and clean signals. Hence, the main objective of this research is to demonstrate that a better performance can be obtained by the integration of the SS and PD-MEMLIN, working in highly degraded environments.

1.4 Justification

Noisy speech is present in almost every real condition. The channel and the environment play an important role in the type of noise and in the behaviour of the ASR system. In this research the problem of highly degraded environments is assesed, where the additive noise is the target unwanted signal. The system then needs an effective algorithm to work in high-noise conditions to achieve a better performance.

1.5 Contribution

The main contribution of this tesis is to obtain a suitable integration of two techniques. The SS and PD-MEMLIN leads to the development of a new scheme that helps to deal with high-noise conditions. The combination of techniques is proposed in order to eliminate the greater amount of noise and approach the noisy space employed to the clean one. This new algorithm is called PD-MEEMLIN (Phoneme-Dependent Multi-Environment Enhanced Models based LInear Normalization).

1.6 Organization

The organization of the document is as follows. In Chapter 2, it is made a description of the ASR architecture employed to obtain the results when the different techniques are applied: SS, PD-MEMLIN and PD-MEEMLIN. Then some algorithms created to improve the performance of the ASR system are mentioned, which had been compared against PD-MEMLIN, that it is applied by this research.

Chapter 3 introduces the SS and PD-MEMLIN techniques and are detailed to explained the work that it is done independently by each. At the end of the chapter, the new architecture technique is described to explain how it works.

In Chapter 4 it is explained the environment employed for the experiments developed. As well as the results obtained for the different noise levels.

Chapter 5 summarizes and conclude the document.

Chapter 2

Speech Recognition and Speech Robustness Techniques

The process of speech recognition in real environmental conditions is affected by additive and convolutional noise. This is the main reason why the performance of the ASR system fails. The development or improvement of techniques to increase robustness during speech recognition is needed. In this thesis just the additive effect of the noise is detailed. The first approach has been done using the basic type of noise known as Gaussian white noise. The second approach details the environmental noise and the compensation needed to improve performance.

Both branches use the same ASR. On the next sections a general explanation of the speech recognition architecture is given and some robustness techniques are detailed.

2.1 Noise

The real condition noise is approached, in the research field, by different environments. Some of them are known as background noise since the human ear can listen to them in the surroundings, but not having the main attention. The human ear is able to extract the important information, using not just the acoustic signal, but also the face image and the language knowledge. However, for the ASR the task should be solved according to robust and enhancement algorithms. The environmental adaptation of the ASR improves the accuracy it can obtain.

The additive noise is defined as

$$y(t) = s(t) + n(t), \tag{2.1}$$

where $n(t)$ correspons to the noise added to the speech signal $s(t)$ in the time domain, obtaining a noisy speech signal $y(t)$. Usually, environment noises correspond to this category. This noise can be linear in the power spectrum domain. Another important characteristic is that additive noise can be stationary or non-stationary. The first one has a power spectral density that does not change over time, and the second one has statistical properties that change over time.

White Gaussian (WG) noise is an example of stationary noise with a Gaussian distribution, a wideband with a constant spectral density. Examples of this noise correspond to the appearing static on the phone. This type of noise is employed in some of the experiments developed in this research and the results are going to be detailed in Chapter 4.

The WG noise is useful as a conceptual entity, because it helps to configure the tools employed on the noise estimation, and corroborate that everything is working correct. When the configuration is done the WG noise is replaced, because almost all the additive noise is non-stationary. Apporaching the environments to real conditions, the non-sationary noises employed are: babble, subway and car.

2.2 Automatic Speech Recognition

The ASR is a pattern clasification process, which its main goal is to classify the given speech signal into pattern sequences previously learned and stored by the acoustic and language models. There are many factors that make difficult the ASR process and its performance.

The speech recognition system is composed as shown in Figure 2.1

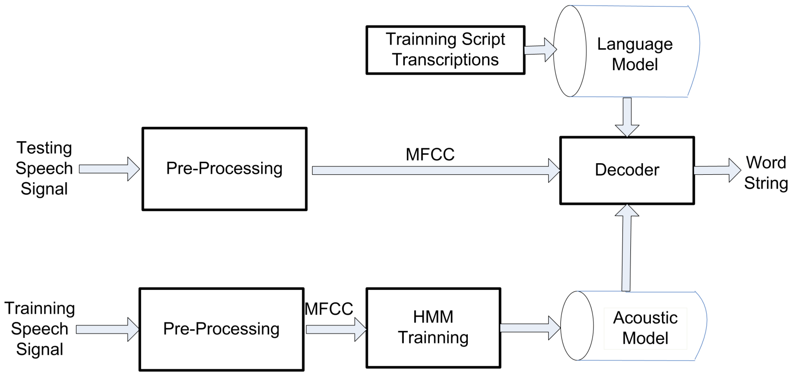

Figure 2.1: General Description of the ASR Architecture.

In a general view, the speech signal pre-processing transforms the signal into a set of feature vectors that makes its manipulation easier for the decoder. The acoustic models represent the information conveyed in the acoustics, phonetics, gender and dialect differences. The language models represent what compose a certain word, which word is possible to occur and what will be the sequence. Finally, the decoder with the help of the language and acoustic models starts to generate the word sequence according to the maximum probability of the feature vectors obtained at the end of the signal processing component [13].

2.2.1 Speech Processing

The ASR obtains a transcription of the words or sentences that the speaker expresses. The general explanation describing the process to define the sequence of phonemes that appear in the transcription is given as follows.

The first module is the pre-processing of the speech signal to obtain the parameters of the feature vector, the process is described in Figure 2.2

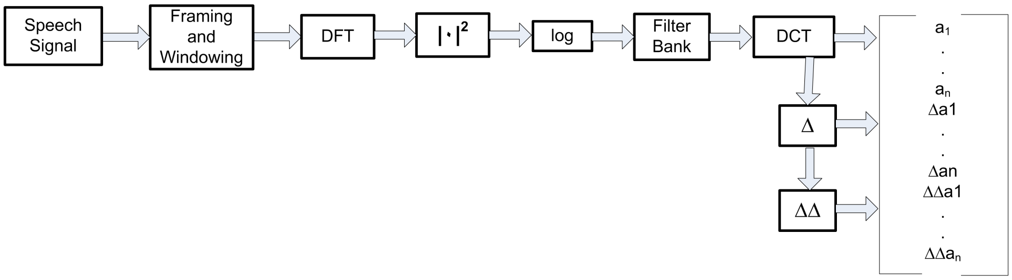

Figure 2.2: Pre-processor.

The clean signal is divided into several short windows, and computes the Mel Frequency Cepstral Coefficients (MFCC) [13]. As a result an n-dimension (usually 12) vector is obtained, where the first values refer to the MFCCs of the waveform and the last one corresponds to the energy coefficient. Afterwards the time derivative (Δ) and the time acceleration ($\Delta\Delta$) are estimated for each parameter of the vector to emphasize the speech dynamical features in time. Once the feature vector is ready it is feeded to either the decoding algorithm or to the training algorithm to calculate the acoustic models. The acoustic models used in this research are based on the computation of the Hidden Markov Model (HMM) [13]. The principal components of a HMM, see Figure 2.3, described in [13] are:

$O = \{O_1, O_2, ..., O_N\}$ Observation sequence (input)
N States representing the state space (number of states)
$A = \{a_{ij}\}$ Transition probability matrix
$B = b_j(O_l)$ Observation probability distribution
$\pi = \pi_i$ Initial state distribution

$Q = q_1, q_2, ..., q_N$ Hidden states.

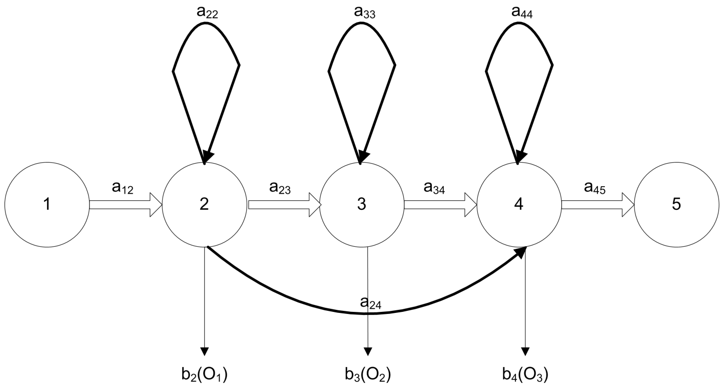

Figure 2.3: Left to right HMM, 1 ... 5 states, a transition probabilities, b output probablitites, O observation sequence.

The compact notation of the HMM is denoted as (A, B, π) [11]. The parameter set N, M, A, B, and ξ is calculated using the training data and it defines a probability measure $Prob(O|\xi)$. The observation probablity distributions of each state of the HMM are commonly represented by a mixture of Gaussian distributions (pdfs), better call Gaussian Mixture Model (GMM), where the means and covariances are the important features. At this point everything is combined and sent to the decoder, where the transcription is obtained.

The pre-procesing showed is used when the environment is free of noise. However, this is going to change, because for this research it is handle a noisy speech signal. With this change it is going to be enhanced the noisy speech signal, approaching the corrupted signal into an undisturbed speech signal. The new description of the procedure is going to be detailed in the next chapter.

2.3 Speech Robustness Techniques

Speech enhancement analyses the corrupted or noisy speech signal in order to improve the quality of the signal to perform different tasks needing a high resolution of voice signal reconstruction. The need of accuracy on certain applications has made that earlier work in speech enhancement influenced the research for robust systems.

The robustness development in speech recognition systems is becoming an important part of the practical applications. Trying to compensate the degradation on the robustness systems several techniques have been developed. A general classification of those techniques is acoustic model adaptation and feature compensation or normalization [11]. The first one tries to improve the system by adapting the speech models to the speaker, channel and task. However a large amount of adaptation data is needed and it takes a large computation time to process the information. The second one modifies the feature vectors, that is why the compensation with the normalization needs lees data and time [10]. In order to have a faster compensation method this research is based on feature compensation techniques.

2.3.1 Continuous Speech Recognition in Noise Using Spectral Subtraction and HMM Adaptation

The research performed by [8] presents a method which includes an integration of the Continuous SS and the Model Adaptation Techniques. In this case a smoothed estimate of the long term spectrum is computed an subtracted. Furthermore a compensation due to the distortion is proposed using PMC. It assumes that $Y(\omega) = S(\omega) + N(\omega)$, where $S(\omega)$ is the clean speech signal spectrum and $N(\omega)$ is the noise spectrum. The method involves the computation of the energy spectrum of each window and then, subtract the estimated energy spectrum of the noise as shown in Equation 2.2.

$$|Y(\omega)| = |S(\omega)| - \alpha|\overline{N(\omega)}| \tag{2.2}$$

$$|Y'(\omega)| = \begin{cases} |\overline{(Y(\omega)}| & \text{if } |\overline{Y(\omega)}| > \beta|\overline{N(\omega)}| \\ \beta|\overline{S(\omega)}| & \text{otherwise.} \end{cases}$$

where α is a factor of over-estimation of the noise and its objective is to reduce the peaks in the spectrum, and β is a flooring value that prevents the decrement of the signal energy to a value less than $\beta|\overline{S(\omega)}|$. With this simple scheme the musical sound is hiden and when the wide band noise is perceived, it is reduced by β.

At the output of this process the signal is enhanced but distorted. If $P(Y)$ is assumed log normal; then the expected value of the enhanced speech is

$$E[Y^D] = E[Y] + \Delta^C_{\mu,\psi} \tag{2.3}$$

where $\Delta^C_{\mu,\psi}$ denotes a correction factor. The first element can be calculated using the SS parameters, the noise estimate, and the signal distribution. The second one can be computed using the expected values of the log normal distribution. Next, Equation 2.3 can be viewed in terms of $E[S_D]$ and an associated correction factor.

$$E[Y^D] = E[S_D] + E[N] + \Lambda^C_{\mu,\psi} \tag{2.4}$$

where $E[N]$and $\Lambda^C_{\mu,\psi}$ are the estimation of the noise espectrum. By using this approach, a HMM compansation can be performed to the output of the CSS. The PMC algorithm is applied directly to the HMM and needs to be done when the background noise is reestimated.

To better understand the algorithm, it is better to detail its full steps.

1. The first one, transforms the cepstral means and variances of speech and noise HMM into its linear domain. (For further details refer to [8].
2. Compute $E[Y]$ and $E[Y^2]$.
3. Calclulate the specific parameters of the expected values of the log normal density function used by $\delta^C_{\mu,\psi}$, and Λ^C_μ
4. Ajust the HMM means acccording to PMC.
5. Get the signal bcak to its cepstral domain.

The results show that the implementation of the integration is an effective method, since it can smooth the effect of the noise, but also they have obtained very good results in very noisey environments (0dB SNR or below).

2.3.2 Cepstrum-Domain Model Combination Based on Decomposition of Speech and Noise for Noisy Speech Recognition

This is a Cepstrum-Domain model combination method for ASR in noisy environments. The algorithm describe by [9] combines separately estimated speech and noise HMMs to obtain a single HMM model that describes the noise corrupted MFCC observation vectors. The method described is composed by two algorithms, the Cepstrum Subtraction Method (CSM) and the Additive Model Combination (AMC).

The CSM compensates MFCC features with respect to additive noise. It is based on the use of the Minimum Mean Square Error-Log Spectral Amplitude (MMSE-LSA), which obtains the estimates the clean speech and noise from the corrupted noisy speech signal. The process followed is a speech enhancement algorithm to obtain the nonlinear transfer function of the frequency dependent gain function. The AMC procedure combines speech and noise models directly in the MFCC domain without converting to the linear spectral domain and while making a minimum of simplifying approximations. AMC procedure for each utterance starts obtaining the noise MFCCs using the CSM procedure described before. Then the mean and variance of single Gaussian representing noise are estimated from the noise speech spectrum.

The combination of AMC and CSM gives a very large performance improvement, thanks to the general tendency of the MMSE-LSA procedure to underestimate the SNR, because it includes SNR as part of its spectral estimate. For that reason, the AMC approach does not require any explicit SNR estimator. Both the AMC and CSM rely on the MMSE-LSA based speech enhancement algorithm to obtain an additive descomposition of speech and noise in the cepstrum domain.

2.3.3 ALGONQUIN

For robust ASR is defined a framework that unifies the noise compensation mechanism and the recognizer. This framework helps to demonstrate the importance of information about the uncertainty of the observations to model the noise and the speech, creating this models with GMM. The framework allows to isolate the effect of retaining or eliminating of the degree of uncertainty of the observations [5].

The definition of the framework mentioned refers to the ALGONQUIN algorithm [6], that it is employed in speech recognition systems, which employ complex speech models like language models and word or phone HMMs for the encoding of the transition probabilites between states. The uncertainty mentioned before is produced by the noise, and it is captured in the variance parameters of the noise model. The speech model used by ALGONQUIN is modeled by GMM, for the noisy model in the log-spectrum domain, with this the joint distribution over noisy speech y, speech x, speech class s^x, noise n, noise class s^n is:

$$p(y, x, n, s^x, s^n) = p(y|x, n)p(s^x)p(x|s^x)p(s^n)p(n|s^n). \tag{2.5}$$

The current noisy speech y frame is estimated with the posterior using a parameterized distribution, q:

$$p(x, n, s^x, s^n|y) \approx q_y(x, n, s^x, s^n), \tag{2.6}$$

this q function is a GMM.

The variational parameters of q are adjusted to have a precised approximation, and then q substitutes the true posterior when computing the estimation of the clean speech features and calculating the soft information.

To find an estimation of the posterior q, the variational inference is employed. The variational inference helps to minimize the relative entropy between q and p. Trying to minimize is a good choice for a cost function, because minimizing the relative entropy is equivalent to maximizing a lower bound on the log-probability of the data.

2.3.4 State Based Piecewise Linear Compensation for Environments (SPLICE)

At the beginning the SPLICE success lies in the requirement to maintain the distortion conditions similar to the part that aids the system to learn the corrections and those that corrupted the data employed to test the system.

For this technique given the observed cepstrum an estimate of undistorted speech is produce. The algorithm does not have an explicit noise model, only needs the noise characteristics embedded between the stereo clean and distorted speech cepstral vectors [2]. The important element here is the piecewise linearity, employed to map the clean and noisy (distorted) signal, because it makes possible to deal with the nonlinearity between the cepstral vectors of the signals mentioned before.

Two important assumptions are made. The first one is that the noisy speech cepstral vector follows the mixture of Gaussians distribution, meaning that the pdf is going to be the sum of the every separate distortion condition and each region obtained through the piecewise linear approximation of the clean cepstral vector with the distorted one. The second one is that the pdf is Gaussian, for the clean vector given the noisy speech vector and the index of the region over which the piecewise linear approximation between clean and distorted vectors.

SPLICE was tested with the AURORA2 database (described in a later chapter), that it is also employed by this research. For that reason it is possible to establish a comparison between this technique with the one developed in this document. Another reason to establish a relation is that PD-MEMLIN had been compared with SPLICE in order to prove that the first one works better under noisy conditions. These evaluations are going to be fully detailed in Chapter 4.

2.3.5 Multivariate Gaussian Based Cepstral Normalization (RATZ)

Another environmental compensation algorithm is RATZ, which assumes that the speech features affected by unknown noise and filtering can be compensated by corrections of Gaussians mixtures components, specifically the mean and variance [4]. RATZ was developed to work with or without stereo data employed to perform the training and testing of all the noisy environments. For this document, stereo RATZ is the one described here.

The development of RATZ for stereo data assumes a multivariate Gaussian mixture distribution to represent the speech statistics, this characteristic is share with SPLICE, because both use of GMM. Besides RATZ defines a model based on additive compensations for mean vectors and covariance matrix of the clean speech as effects from the environment. To better define the algorithm, three stages are defined:

- **Estimation of the statistics of clean speech.** Here the multivariate GMM is employed to model the pdf for the features of the clean speech. The distribution can be written as:

$$x = [x_0 ... x_{p-1} x_p]^T$$
$$p(x) = \Sigma P[k] N_x(\mu_{x,k}, \Sigma_{x,k}) \ .$$

- **Estimation of statistics of noisy speech.** Here the compensation takes place by making the addition of shift parameters learned using the maximum likelihood, which attempts to maximize the probability of the observed noisy data. The model for the effects of the environment is established with the new set of statistics obtained through the process described. During this phase the stereo data is really important, because it helps to find the set of parameters that maximize the likelihood function without an iterative re-estimation needed by EM techniques.

- **Compensation of noisy speech.** To achieve the estimation of the noise, the minimum mean square error is used (MMSE) to maximize the expected value of the unobserved clean speech data.

2.3.6 The Relative Spectral Technique (RASTA)

In the ASR the speech signal reflects the movements of the vocal tract and the non-linguistic components rate is different from the vocal tract rate, this is where RASTA comes important. It takes advantage from the differences that exist within rates and suppresses the spectral components that are changing slowly or quickly than the range of change of speech [7]. RASTA tries to make the speech analysis less sensitive to components with a variation not according to the majority of the speech components.

To support the last paragraph, during the Perceptual Linear Prediction (PLP) speech analysis a spectral estimates is done, in which each frequency channel is band pass filtered with a sharp spectral zero at the zero frequency [7]. Now every constant or slowly varying component is suppressed and the speech analysis is less sensitive to this variations. All this is done to create a model observed in the human speech perception and an approximation on the human hearing (the audible spectrum).

Another characteristic is that RASTA result depends on history, because it uses a larger part of the signal against which the current analysis frame is compared. Employing this on the channel equalization technique is how it can differentiate between the disturbances and the speech components. This history employed effectively enhances transitions between speech segments and the result is dependent on the previous speech segment. It could be a phoneme or a syllable.

The RASTA process is illustrated in Figure 2.4

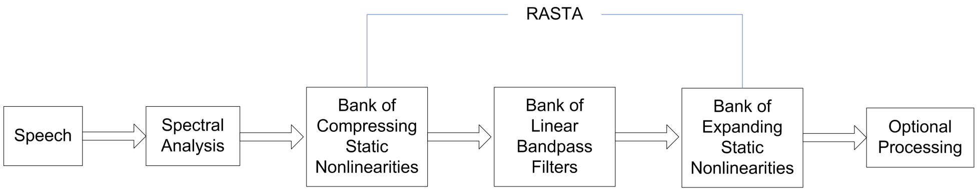
RASTA
Speech
Spectral Analysis
Bank of Compressing Static Nonlinearities
Bank of Linear Bandpass Filters
Bank of Expanding Static Nonlinearities
Optional Processing

Figure 2.4: Speech Processing Technique RASTA [7].

2.3.7 Spectral Subtraction

Spectral Subtraction is a noise suppression technique which reduces the effects of the additive noise in speech [14]. Usually the spectrum noise component is estimated during speech pauses to update noise statistics. This noise estimation is done with the basic spectral subtraction algorithm. Therefore, tracking variations on the noise levels are slow and confined to periods of no speech activity [12].

However, the algorithm employed in this research removes the use of explicit speech pause detection without incrementing computational complexity. For this Spectral Subtraction, the noise calculation is done with the minimum of the subband noise power within a finite window to estimate the noise floor. By using this technique the slow process of tracking the noise levels during speech pauses is avoided.

Algorithm Description

The algorithm is based on the observation of a short time subband power estimate of a noisy speech signal, which exhibits distinct peaks and valleys [12]. The peaks correspond to the speech activity and the valleys are used to obtain an estimate of the subband noise power. A noise reliable estimation is done with a data window large enough to detect any peak of speech activity.

The algorithm modifies the short time spectral magnitude of the noisy speech signal during the process of enhancement. Once the signal is synthesized, it gets close to the clean speech signal. To achieve this result, the spectral magnitude is obtained with the noise power estimate and the subtraction rule. A diagram of the basic spectral subtraction method is shown in Figure 2.5

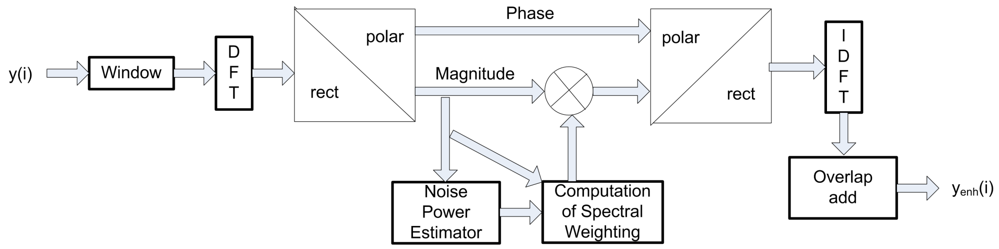

Figure 2.5: Diagram of the Basic SS Method Used [12].

The first consideration made during the process is that the speech which is affected by noise can be expressed as

$$y(i) = x(i) + n(i), \tag{2.7}$$

where $y(i)$ is the speech with noise, $x(i)$ is the clean speech signal, $n(i)$ is the noise signal and i denotes the time index. The next consideration is that $x(i)$ and $n(i)$ are statistically independent, for this reason and thanks to the linearity property

$$E[y^2(i)] = E[x^2(i)] + E[n^2(i)]. \tag{2.8}$$

The statistical properties of the speech signal change over time. For this reason, the spectral processing of the speech signal is done in short time sections called frames. In these short-time segments, speech can be considered stationary. The time data frames are windowed and then converted to frequency domain using the Discrete Fourier Transform (DFT) filter bank with a W_{DFT} subbands and with decimation/interpolation ratio R [12]. This filter bank is an array of band-pass filters that separates the input signal into several components, each one carrying a single frequency subband of the original signal. This filter bank serves to isolate different frequency components in a signal. The window employed is denoted by $h(i)$ and the DFT of the windowed signal $y(i)$ (the disturbed signal) by

$$Y(\lambda, k) = \sum_{\mu}^{W_{DFT}} y(\lambda R + \mu) * h(\mu) * exp(\frac{-j2\pi\mu k}{W_{DFT}}), \tag{2.9}$$

where λ refers to the decimated time index (the part of time where the signal is acquired) and k the DFT frequency bins $\Omega_k = \frac{2\pi k}{W_{DFT}}, k \in 0, 1, ..., W_{DFT} - 1$. Once the subband signals are obtained on the frequency domain, they are converted back to the time domain using the Inverse Discrete Fourier Transform (IDFT). At this point the synthesized improved speech signal is denoted by $y(i)$.

Subtraction Process and Noise Estimation

The proposed minimum subtraction method eliminates the need for a speech activity detector by exploiting the short time characteristics of the speech signals. For stationary noise the performance is similar to the performance of the spectral subtraction with a noise power estimation and speech activity detection. For the algorithm is necessary to estimate the subband noise power $P_n(\lambda, k)$ and the short time signal power $\overline{|Y(\lambda, k)|^2}$. The first step is to obtain the short time signal power, and to achieve this the subsequent magnitude squared input spectra is smoothed with a first order recursive network

$$\overline{|Y(\lambda, k)|^2} = \gamma * \overline{|Y(\lambda - 1, k)|^2} + (1 - \gamma) * |Y(\lambda, k)|^2. \tag{2.10}$$

To subtract spectral magnitudes the proposal of [16] is followed. Here the subtraction is made with an oversubtraction factor $osub(\lambda, k)$ and a limitation of the maximum subtraction by a spectral floor constant ($subf$). The $osub(\lambda, k)$ factor is needed to eliminate the musical noise, but also affects the speech quality. However, the action of this factor is limited. It only needs to be calculated as a function of the subband Signal to Noise Ratio $SNRy(\lambda, k)$ and the frequency bin k. This last line only means that for high SNR conditions and high frequencies less osub factor is needed. For the SNR conditions and low frequencies the $osub$ is lower than what is mentioned before. The $subf$ constant helps to the resultant spectral components from going below a preset minimum level. The $subf$ is express as a fraction of the original noise power spectrum. The relation for the spectral subtraction between $subf$ and $osub$ is defined by

$$\sqrt{subf * P_n(\lambda, k)} \text{ if } |Y(\lambda, k)| * Q(\lambda, k) \leq \sqrt{subf * P_n(\lambda, k)} \tag{2.11}$$

$$|Y(\lambda, k)| * Q(\lambda, k) \text{ where } Q(\lambda, k) = (1 - \sqrt{osub(\lambda, k)\frac{P_n(\lambda, k)}{|Y(\lambda, k)|^2}}). \tag{2.12}$$

The last equations describe the noise subtraction method employed and consist in minimizing the perception of the narrow spectral peaks by decreasing the spectral excursion. This action works over several qualitative aspects of the processed speech signal. These are the levels of the remaining broadband noise, the level of musical noise and the amount of speech distortion. All these effects are controlled thanks to the parameters subf and osub, which are going to be called α and β respectively on the next chapters. The alpha and beta names for the parameters are based on the basic spectral definition of [14].

Now that these parameters are defined, the first step is to calculate the short time subband signal power $P_x(\lambda, k)$ implementing a recursive calculation for smoothed periodograms

$$P_y(\lambda, k) = \zeta * P_y(\lambda - 1, k) + (1 - \alpha) * |Y(\lambda, k)|^2. \tag{2.13}$$

Here ζ represents the smoothing constant to obtain the smoothed periodograms. The last equation allows starting the calculation of the noise model.

As mentioned previously, the basic spectral subtraction algorithm requires a speech activity detector to model the noise. The explicit speech pause detection is not needed for the algorithm employed. To model the noise, the algorithm applies the use of minimum of the subband noise power within a finite window to estimate the noise floor. Where all the valleys corresponding to the noisy speech signal can be used for the estimation of subband noise power.

This estimation lays on two observations. The first one is that the speech and the disturbing noise are usually statistically independent and the second one is that the power of a noisy speech signal frequently decays to the power level of the disturbing noise [15]. Then, even during the speech activity it is possible to derive an accurate noise power spectral density (psd) estimate by tracking the minimum of the noisy signal psd. For this reason the method assume that during speech pause or within small periods between words or syllables the speech energy is close or identical to zero. That is the reason why the minimum power within a finite window must be tracked to identify high power speech segments and estimate with this the noise.

The noise power estimate $P_n(\lambda, k)$ is obtained as a weighted minimum of the short time power estimate $P_y(\lambda, k)$ affected by an overestimation factor omin

$$P_n(\lambda, k) = omin * P_{min}(\lambda, k), \tag{2.14}$$

within a data window of D subband power samples that it is divided into W windows of length M, allowing to update the minimum every M samples without increasing the computation time. The algorithm that helps to reach the noise model from here is described next:

- First the number of signal frames within a subwindow is equal to W and the running minimum estimate is initialized to a preset value.
- A vector is obtained with the estimated minimum power that holds the overall minimum of the length D window. It is updated every time that the number

of signal frames within a subwindow is equal to W, when the present minimum estimate becomes smaller than the estimated minimum power, or when a local minimum is detected.

The separation of the length D window into W subwindows provides the advantage of getting a new minimum estimate after M samples. The advantage is that the noise power is updated instantaneously, independent of the window adjustment. In addition, if the noise power starts to decrease a fast update of the minimum power estimate is achieved. Nevertheless, if the noise power begins to increase the noise is updated with a delay of D+M samples.

The window length D=M*W must be large enough to bridge any peak of speech activity and short enough to follow non-stationary noise variations [12]. This noise estimator combined with the spectral subtraction (speech enhacement algorithm) has the ability to preserve weak speech sounds and provides good intelligibility.

2.3.8 MEMLIN

MEMLIN, as its name conveys, is a multi-environment adaptation technique based on Minimum Mean Squared Error Estimation (MMSE). This technique works with a feature adaptation data and several basic defined environments [10]. The main goal of MEMLIN is to learn the difference between clean and noisy feature vectors represented with Gaussians. The clean model and the noisy model are obtained to represent each basic environment. All mentioned before is used to compensate the mismatch that exists between the clean and noisy vectors of the clean and disturbed speech signal gave to MEMLIN.

The algorithm has been compared to some others techniques, like the ones explained on chapter 2. MEMLIN has obtained important improvements against those techniques, as it is shown in [10]. One reason of the improvement with MEMLIN is that employed several environments. If the environments are well defined and cover the main part of the features space, it is easier to match noisy phrases belonging to several environments and a linear combination of environments is the best to represent a phrase. Another advantage of MEMLIN is the use of the clean and noisy models, which applies a conditional probability model between noisy and clean Gaussians. Defining a transformation vector for each noisy condition, like SPLICE, is not the best path to follow.

MEMLIN has evolved with time, and the technique employed on this research is PD-MEMLIN. PD-MEMLIN has the same basic structure and the same mathematical principle. A detailed explanation of MEMLIN and its changes to become PD-MEMLIN, will be cover along this chapter.

MMSE Estimator

There are several feature vector normalization techniques that assume a prior pdf for the estimation variable, MEMLIN is one of them. It needs to estimate the unknown probabilities to obtain the equations for the process. A Bayesian estimator can be used to estimate the clean feature vector and the optimal estimator for this is the Minimum Mean Square Error (MMSE). According to MEMLIN, first the clean x and the noisy feature y vectors are given. These two vectors help MMSE estimation to calculate the clean estimation vector $\widehat{x}$

$$\widehat{x} = E[x|y] = \int xp(x|y)dx. \tag{2.15}$$

At this point the problem is the computation of the pdf of x given y, $p(x|y)$. To achieve the calculation of the pdf, some approximations are made. MEMLIN suppose that the noisy feature vector will be modeled as a mixture of Gaussians for each environment

$$p_e(y) = \sum_{s_y^e} p(y|s_y^e)p(s_y^e) \tag{2.16}$$

$$p(y|s_y^e) = N(y; \mu_{s_y^e}, \Sigma_{s_y^e}). \tag{2.17}$$

For the equations e represents the environment index, y denotes the correspondent Gaussian of the noisy model for the e environment, $\mu_{s_y^e}$,$\sigma_{s_y^e}$ and $p(y|s_y^e)$. The elements already described, correspond respectively to the mean vector, the diagonal covariance matrix and the weight associated to the correspond Gaussian of the noisy model.

The second assumption made by MEMLIN is that the clean feature vector model is described with a mixture of Gaussians

$$p(x) = \sum_{s_x} p(x|s_x)p(s_x) \tag{2.18}$$

$$p(x|s_x) = N(x; \mu_{s_x}, \Sigma_{s_x}), \tag{2.19}$$

here s_x denotes the correspondent Gaussian of the clean model, and μ_{s_x},σ_{s_x} and $p(x|s_x)$ are the mean, diagonal covariance matrix and the weight associated to s_x.

During the process mentioned, MEMLIN approximates the pdf of x given y, s_y^e and s_x as Gaussian. The covariance matrix of this Gaussian $\sum_{s_x,s_y^e}$ depends on s_x and s_y^e. The mean vector is a linear transformation of the noisy vector that depends on s_x,

s_y^e and α_e (weight associated to each environment). Another important parameter is the transformation vector r_{s_x,s_y^e}, represents the difference between clean and noisy data given a clean and a noisy model Gaussian of an environment [10]

$$p(x \mid y, s_y^e, s_x) = N(x; y - \sum_e \alpha_e r_{s_x,s_y^e}, \Sigma_{s_x,s_y^e}). \tag{2.20}$$

With all the assumptions made, the approximation of the clean feature vector x for the mean of Equation 2.20, following the description of the algorithm, Equation 2.15 can turn into the following equation for MEMLIN

$$\widehat{x_t} \simeq y_t - \sum_{s_x} \sum_e \sum_{s_y^e} \alpha_{e,t} r_{s_x,s_y^e} p(s_y^e|y_t) p(s_x|s_y^e, y_t), \tag{2.21}$$

where t is a temporal index, $p(s_y^e|y_t)$ is the probability of s_y^e given y_t, and $p(s_x|s_y^e, y_t)$ is the probability of the clean model Gaussian given the noisy one and y_t. The calculus of the clean estimation vector is defined, but now is necessary to define the equations of the parameters needed to do the estimation of $\widehat{x_t}$. To calculate this last parameter it is necessary to estimate some elements defined before in some of the equations.

Parameter Estimation

The estimation of certain elements is important in order to define the estimation vector. The necessary parameters are: $\alpha_{e,t}$ and $p(s_y^e|y_t)$, these parameters are dependent of the noisy feature vector and are estimated during recognition. Other parameters needed are: $r_{s_x,s_y^e}, p(s_x|s_t^e, y_t)$, these variables are estimated with a training process with stereo data for each environment. For the calculus of $\alpha_{e,t}$, an iterative solutions is defined. Each t moment, y_t is available and the calculation of the environment weight will be at that instant

$$\alpha_{e,t} = \beta * \alpha_{e,t-1} + (1 - \beta) \frac{p_e(y_t)}{\Sigma_e p_e(y_t)}. \tag{2.22}$$

In the equation β is the memory constant. The value of $p(s_y^e \mid y_t)$ can be calculated using the distribution of mixture of Gaussians of the noisy feature vector and Bayes

$$p(s_y^e|y_t) = \frac{p(y_t|s_y^e)p(s_y^e)}{\Sigma_{s_y^e} p(y_t|s_y^e)p(s_y^e)}. \tag{2.23}$$

The data is acquired for the process and a training process for each environment is made. The available data are the clean $X_e = x_1^e, ..., x_{T_e}^e$, the clean feature vectors and $Y_e = y_1^e, ..., y_{T_e}^e$ for the noisy feature vectors. Now with the Maximum Likelihood algorithm (ML), r_{s_x,s_y^e} and $r_{s_y^e}$ can be obtained

$$L(Y_e) = \sum_{t_e} log(\sum_{sy^e} p(s_y^e) N(y; \mu_{s_y^e} + r_{s_x,s_y^e}, \Sigma_{s_x,s_y^e})). \tag{2.24}$$

However this is an incomplete expression from the clean stereo data, which is difficult to optimize because it contains the log of the sum. To solve the equation and obtain the desired parameters, the Expectation Maximization (EM) algorithm is considered. The optimal solution employing the EM algorithm is

$$r_{s_x,s_y^e} = \frac{\Sigma_{t_e} p(s_x \mid x_{t_e}^e) p(s_y^e \mid y_{t_e}^e)(y_{t_e}^e - x_{t_e}^e)}{\Sigma_{t_e} p(s_x \mid x_{t_e}^e) p(s_y^e \mid y_{t_e}^e)}. \tag{2.25}$$

The above equation performs the expectation and maximization step simultaneously. From here the probability of s_x given the clean feature vector, $p(s_x \mid x_{t_e}^e)$, is calculated like the Equation 2.23.

At last the conditional probability $p(sx \mid s_y^e, y_t)$ is estimated with the training phrases set by relative frequency. The pair of Gaussians that best describes each stereo pair of vectors is obtained. Now, the conditional probability model between Gaussians is described

$$p(s_x \mid s_y^e, y_t) = \frac{C_N(s_x \mid s_y^e)}{N}. \tag{2.26}$$

The upper element of the fraction correspond to the number of times that the most probable pair of Gaussians is s_x and s_y^e. The lower element is the number of times that the most probable Gaussian for noisy vector is s_y^e.

Transition from MEMLIN to PD-MEMLIN

PD-MEMLIN is a feature vector normalization technique which estimates the different compensation linear transformations in a previous training process. The architecture of PD-MEMLIN is described in Figure 2.6:

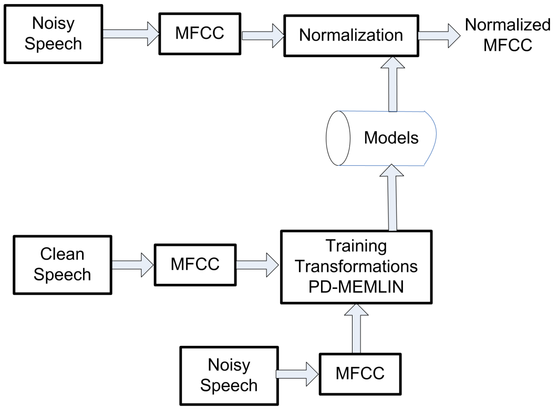

Figure 2.6: PD-MEMLIN Architecture.

MEMLIN and PD-MEMLIN share the same core definition. However PD-MEMLIN differs on the model created by the mixture of Gaussians, because it models also the clean feature space for each phoneme, and the noisy space for each phoneme. All the previous transformations are estimated for a clean phoneme Gaussian and a noisy Gaussian of the same phoneme [11].

For PD-MEMLIN the estimation of the clean feature vector, $\widehat{x}$ according to [11] is

$$\widehat{x} = E[x|y] = \int xp(x|y)dx. \tag{2.27}$$

As well as MEMLIN makes some assumptions, PD-MEMLIN includes the fact that everything is based on certain phoneme. So the three assumptions defined and their changes are

- For the first one, some environments are established for the noisy space, and the noisy feature vectors follow the distribution of GMM for each environment and phoneme.

$$p_{e,ph}(y) = \sum_{s_y^{e,ph}} p(y \mid s_y^{e,ph})p(s_y^{e,ph}) \tag{2.28}$$

$$p(y \mid s_y^{e,ph}) = N(y; \mu_{s_y^{e,ph}}, \Sigma_{s_y^{e,ph}}). \tag{2.29}$$

- The second one is that the clean feature vectors, x, are modeled with the following GMM

$$p_{ph}(x) = \sum_{s_x^{ph}} p(x \mid s_x^{ph})p(s_x^{ph}) \tag{2.30}$$

$$p(x \mid s_x^{ph}) = N(x; \mu_{s_x^{ph}}, \Sigma_{s_x^{ph}}). \tag{2.31}$$

- The third one established that for each time frame t, x is approached as a function of the noisy feature vector, the clean model Gaussians and the noisy environment model Gaussians

$$x \simeq f(y_t, s_x^{ph}, s_y^{e,ph}) = y_t - r_{s_x^{ph}, s_y^{e,ph}}. \tag{2.32}$$

As it can be seen, the difference between the equations defined for MEMLIN, with the ones for PD-MEMLIN, is that the phoneme factor is considered to establish the GMM and the weight associated to that Gaussian.

Following with the process, the same behavior is observed during the MMSE estimation. The same process mentioned for MEMLIN is followed, only changing the phoneme factor that appears for the probabilities calculation. At the end, considering the fact that PD-MEMLIN depends on the combination of the environment along with the phoneme, the elements needed to calculate the estimator vector suffer a change. The equations with the change observed for PD-MEMLIN algorithm are

$$p(e \mid y_t) = \beta * p(e \mid y_{t-1}) + (1-\beta)\frac{\Sigma_{ph} p_{e,ph}(y_t)}{\Sigma_e \Sigma_{ph} p_{e,ph}(y_t)} \tag{2.33}$$

$$p(ph \mid y_t, e) = \frac{p_{e,ph}(y_t)}{\Sigma_{ph} p_{e,ph}(y_t)} \tag{2.34}$$

$$r_{s_x^{ph}, s_y^{e,ph}} = \frac{\Sigma_{t_{e,ph}} p(s_x^{ph} \mid x_{t_{e,ph}}^{e,ph}) p(s_y^{e,ph} \mid y_{t_{e,ph}}^{e,ph})(y_{t_{e,ph}}^{e,ph} - x_{t_{e,ph}}^{e,ph})}{\Sigma_{t_{e,ph}} p(s_x^{ph} \mid x_{t_{e,ph}}^{e,ph}, e, ph) p(s_y^{e,ph} \mid y_{t_{e,ph}}^{e,ph}, e, ph)} \tag{2.35}$$

$$p(s_y^{e,ph} \mid y_t, e, ph) = \frac{p(y_t \mid s_y^{e,ph}) p(s_y^{e,ph})}{\Sigma_{s_y^{e,ph}} p(y_t \mid s_y^{e,ph}) p(s_y^{e,ph})}. \tag{2.36}$$

The above parameters, have the same correspondence to the ones calculated for MEMLIN. But the same consideration as before is made, not only the environment is needed to obtain all the estimations. It is important, to take in consideration, the fact that the phoneme involve during the matching process affects the estimation of the elements to approach the estimator vector as much as it can to the clean model vector.

At the end with the three approximations, the Equation 2.32 leads to obtain the estimation of the clean feature vector normalization [11]

$$\widehat{x}_t \simeq y_t - \sum_e \sum_{ph} \sum_{s_x^{ph}} \sum_{s_y^{e,ph}} r_{s_x^{ph}, s_y^{e,ph}} p(e|y_t) p(ph|y_t, e) p(s_y^{e,ph}|y_t, e, ph) p(s_x^{ph}|y_t, e, ph, s_y^{e,ph}). \tag{2.37}$$

The computation of the different probabilities and the independent term of the linear transformation can be studied in [11].

2.4 Summary

This chapter described the architecture handle for the ASR system employed. Besides gives an introduction to speech robustness, mentioned how the different techniques are classified in two general areas, acoustic model adaptation and feature compensation or normalization. The techniques described here are important for robustness, but there are others. The ones employed in this research are Spectral Subtraction and PD-MEMLIN.

The subtraction method proposed eliminates the need for a speech activity detector and it takes advantage of the short time characteristics of speech signal. The performance obtained with this variation, is very close to the performance of the basic spectral subtraction method with ideal speech activity detection. The key components of the noise calculation and approximation are power spectral density smoothing algorithm, tracking the variance of the smoothed power spectral density in frequency bands and a compensation algorithm for minimum power spectral density estimates. The noise estimator combined with the speech enhancement technique, based on the spectral subtraction, has the advantage of preserve weak speech sounds and gives more accuracy.

In order to increase the performance of the features obtained with the SS method, PD-MEMLIN is added to the process of robustness. Here the PD-MEMLIN algorithm employs the MMSE estimator to define the feature vector estimator to achieve an approximation to the clean model established at the beginning of the algorithm.

The robustness improvement made for the process with the two techniques is fully detailed in chapter 3.

Chapter 3

PD-MEEMLIN

It is well known that the data employed for training and testing conditions, in the ASR system, are different. This leads to a rapidly degradation for the ASR system performance. In order to compensate this mismatch robustness techniques had been developed.

According to the information given in the later chapter, many techniques had been proved. But these algorithms can be combined to have a better performance for the speech recognition system. In the previous chapter, two robustness algorithms employed for the development of a new architecture are presented. The combination of both techniques tries to increase the performance of the speech recognition system, based on the results obtained with every one of the algorithms independently.

The first one is the SS, a well known speech enhancement technique. The second one is PD-MEMLIN, an empirical feature vector normalization technique. SS was selected because of its implementation simplicity and PD-MEMLIN for the good results in comparison with some of the techniques described on the previous chapter.

The new architecture developed for this work is presented. The process followed by the algorithm is detailed and the modules are described.

3.1 PD-MEEMLIN

PD-MEEMLIN is an empirical feature vector normalization which models clean and enhanced-noisy spaces by Gaussian Mixture Models (GMMs). In this algorithm, the probability of the clean model Gaussian, given the enhanced-noisy model and the enhanced-noisy feature vector is a critical point.

PD-MEEMLIN maps the enhanced-noisy feature vector into the clean space. The new calculated enhanced-space is not the clean one, but the estimation gaves an approximation to the original clean space. This approximation can be improved geting the estimation of the enhanced-space closer to the clean one. Applying first the SS to the noisy space starts to improve the estimation of the enhanced-clean space. This enhancement produces an approach between spaces, making the gap smaller among them. It is not possible to achieve this improvement with any technique. It could be possible that the estimated clean space will be worst than the one approximated only with PD-MEMLIN. The SS was considered according to the results obtained in [8], where the SS combined with another one provides high recognition performance for very noisy environments. For that reason, the SS had been selected to support PD-MEMLIN on the approaching of the clean space, to develope the PD-MEEMLIN.

3.1.1 PD-MEEMLIN Architecture

The new architecture obtained from the combination of techniques appears on Figure 3.1

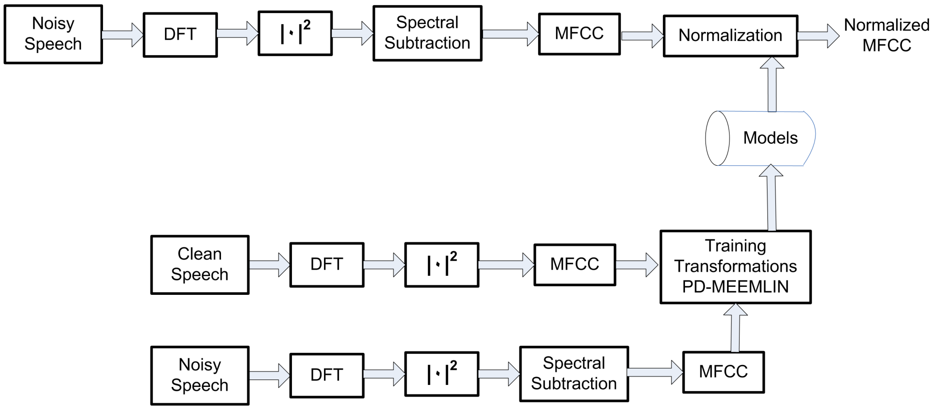

Figure 3.1: PD-MEEMLIN Architecture.

The new architecture combined the advantages of the SS and PD-MEMLIN. The only restriction is that the Signal to Noise Ratio, between the clean speech signal and the noise distorting it, needs to be known to configure the SS method.

Next, the architecture modules are explained:

- Given the SNR, the SS-enhancement of the noisy speech signal is performed.
- Given the clean speech signal and the enhanced noisy speech signal are created the clean and noisy-enhanced model (PD-MEEMLIN).
- In testing, the noisy speech signal is also SS-enhanced and then normalized using PD-MEEMLIN.

- These normalized coefficients are forwareded to the decoder.

3.1.2 Enhanced Estimation

PD-MEEMLIN begins with the enhancement of the noisy speech signal thanks to the SS. The enhanced-noisy speech signal is passed to the MMSE to estimate the enhanced clean feature vector, $\widehat{x_{enh}}$

$$\widehat{x_{enh}} = E[x|y_{enh}] = \int xp(x|y_{enh})dx \tag{3.1}$$

where x corresponds to the clean feature vector, and y_{enh} to the enhanced-noisy feature vector.

The problem observed is how to obtained the pdf of x given y_{enh}, $p(x|y_{enh})$, and how to estimate x. For these approximations, PD-MEEMLIN made some estimations like the noisy space is split into several basic environments, e, and the enhanced-noisy feature vector can be modeled as a mixture of Gaussians for each basic environment, e, and phoneme, ph.

$$p_{e,ph}(y_{enh}) = \sum_{s_{y_{enh}}^{e,ph}} p(y_{enh}|s_{y_{enh}}^{e,ph})p(s_{y_{enh}}^{e,ph}), \tag{3.2}$$

$$p(y_{enh}|s_{y_{enh}}^{e,ph}) = N(y_{enh}; \mu_{s_{y_{enh}}^{e,ph}}, \Sigma_{s_{y_{enh}}^{e,ph}}), \tag{3.3}$$

where $s_{y_{enh}}^{e,ph}$ corresponds to the Gaussian of the enhanced-noisy model for the e environment and ph phoneme. The parameters $\mu_{s_{y_{enh}}^{e,ph}}$, $\Sigma_{s_{y_{enh}}^{e,ph}}$ and $p(s_{y_{enh}}^{e,ph})$ are the mean vector, the diagonal covariance matrix and the weight associated to $s_{y_{enh}}^{e,ph}$.

The second approximation is that the clean feature vectors x, are modeled with the following GMM for each phoneme

$$p_{ph}(x) = \sum_{s_x^{ph}} p(x|s_x^{ph})p(s_x^{ph}), \tag{3.4}$$

$$p(x|s_x^{ph}) = N(x; \mu_{s_x^{ph}}, \Sigma_{s_x^{ph})}, \tag{3.5}$$

which corresponds to the same equations defined before by PD-MEMLIN. These equations remains equal because the clean feature vector does not need to be enhanced by the SS.

The third assumption established that for each time frame t, x is approached as a function of the enhanced-noisy feature vector, the clean model Gaussians and the enhanced-noisy environment model Gaussians

$$x \simeq f(y_{t,enh}, s_x^{ph}, s_{y_{enh}}^{e,ph}) = y_{t,enh} - r_{s_x^{ph}, s_{y_{enh}}^{e,ph}}, \tag{3.6}$$

where $r_{s_x^{ph}, s_{y_{enh}}^{e,ph}}$ is the independent term of the linear transformation, and it depends on each pair of Gaussians, s_x^{ph} and $s_{y_{enh}}^{e,ph}$.

The three approximations lead to obtain the enhanced-clean feature vector normalization, Equation 3.1, as

$$\begin{aligned}\widehat{x_{t,enh}} \simeq y_{t,enh} - \sum_{e} \sum_{ph} \sum_{s_x^{ph}} \sum_{s_{y_{enh}}^{e,ph}} r_{s_x^{ph}, s_{y_{enh}}^{e,ph}} p(e|y_{t,enh}) p(ph|y_{t,enh}, e) \\ p(s_{y_{enh}}^{e,ph}|y_{t,enh}, e, ph) p(s_x^{ph}|y_{t,enh}, e, ph, s_{y_{enh}}^{e,ph}).\end{aligned} \tag{3.7}$$

3.2 Summary

For real applications the clean signal never is present during the noisy process, that is the reason why the clean model needs to be estimated. This leads to a better performance by the ASR system, due to the support provided by this new architecture.

The combination of SS and PD-MEMLIN had developed an algorithm that works better under highly degraded conditions and gives an enhanced space that it is closer to the original clean space. PD-MEEMLIN improves the work done by the SS or by PD-MEMLIN and helps to increase the percentage of recognition for the ASR process.

Chapter 4

Experimental Results

This Chapter presents an evalutation of the new architecture developed for the algorithm PD-MEEMLIN, which its main task is to increase the robustness of the ASR system in high-noise conditions. The evaluations presented are intended to show the potential of the method in theorical application (with the WG noise), and a real-world application with the extra environments defined (subway, babble and car noise). Besides, the new algorithm is going to be compared against SPLICE, because this technique had been one of the principal references in many works in the robustness area, like MEMLIN [10] and PD-MEMLIN [11]. Another reason to establish this comparison is that SPLICE handle a wide range of difficult distorsions, like non-stationary distorsion.

4.1 Database Description

A set of experiments were performed employing the AURORA [17] front-end Database version 2.0. AURORA clean and noisy data is based on TIDigits, it is down-sampled to 8kHz, and filtered with a G712 characteristic. The noise is artificially added at several SNRs (20dB, 15dB, 10dB, 5dB, 0dB, -5dB).

AURORA contains three different sets of noisy speech data. The first set of test speech data is added with subway, babble, car and exhibition hall noise. The second one consists of restaurant, street, airport and train station noise. The last set contains subway and street noise.

The noises presented by AURORA are focused on the representation of realistic scenarios, where the additive noise non-stationary is affecting the ASR process.

4.2 Experimental Results

The first experiments were developed to configure PD-MEEMLIN technique and corroborate the contribution of this new algorithm, that it is to increase the performance of the ASR system in highly degraded conditions. The environment defined for the experiments was with WG noise. This environment was divided into several SNR, from 0db to 20dB.

The models employed by PD-MEEMLIN are obtained from 22 phonemes. The different models are represented with 32 Gaussians each. The feature vectors for the recognition process are built with 12 Normalized MFCC followed by the energy coefficient, the time-derivative and the time-acceleration derivative. The total dimension of the feature vector is 39 coefficients. The 22 phonemes employed are modeled with a three-state HMM and 8 Gaussians per state.

For all the SNR, the results shown that the enhancement produce by the SS helps PD-MEMLIN to increase its robustness during the ASR process. The cascading technique PD-MEEMLIN is better against the case were no techniques are applied, to the SS and to PD-MEMLIN. The results are described in Table 4.1.

The results demonstrate the fact that PD-MEEMLIN exceeds the performance reached by PD-MEMLIN or by SS. The results prove that the SS really helps PD-MEMLIN to get a closer approximation of the estimated clean model to the original clean space, instead of moving it away. Observing this behavior the definition of an approximation to real environments it is done for the second set of experiments.

The second set of experiments employs environments defined according to three noises, with different SNR each: Subway, Babble and Car. The SNR range for each noise goes from -5dB to 20dB (-5, 0, 5, 10, 15 and 20). For every SNR the SS parameters α and β needs to be configured. The parameter α takes values from 1 to 4, and β goes from 0.01 to 0.04.

The models employed by PD-MEEMLIN are obtained from 22 phonemes. The different models are represented with 32 Gaussians each. Besides, two sets of every noise were used because PD-MEEMLIN needs one set to estimate the enhanced-noisy model, and the second one is employed to obtain the normalized coefficients. The feature vectors for the recognition process are builted as it was mentioned before. The 22 phonemes employed are modeled with a three-state HMM and 8 Gaussians per state.

The combined techniques show that for low noise conditions i.e. SNR=10, 15 or 20 dB, the difference between the original noisy space and the one approximated to the clean is similar. Demonstrating how competitive is PD-MEEMLIN against SS, PD-MEMLIN and SPLICE. However, when the SNR is lower (-5dB or 0dB) the SS improves the performance of PD-MEMLIN. Comparing the combination of SS with PD-MEMLIN against the case were no techniques are applied, a significant improvement is seen by both. The results described before are presented in Tables 4.2, 4.3 and 4.4 according to the following format: the column Sent Correct indicates the test utterances percentage correctly recognised, and the column Word Correct indicates the words percentage that were recognised correctly [18].

The results presented suggest that the proposed method PD-MEEMLIN performs well when evaluated by high-noise environment conditions (-5dB and 0dB), even well ahead of some of the methods described in the previous sections, like the results presented by SPLICE in [2]. As it was mentioned before, SPLICE was tested with the same database AURORA2. Allowing to compare these results with the ones obtained by SPLICE.

4.3 Summary

This Chapter described the results obtained with the new architecture PD-MEEMLIN for the stationary and non-stationary addtitive noise. The proposed method derived from SS and PD-MEMLIN by the space enhancement, is shown to perform a comparison with the methods mentioned. Giving a contribution to the state of art in speech robustness.

WG	**ASR**		**ASR+SS**		**ASR+PD-MEMLIN**		**ASR+PD-MEEMLIN**	
	Sent. Corr. %	**Word Corr. %**	**Sent. Corr. %**	**Word Corr. %**	**Sent. Corr. %**	**Word Corr. %**	**Sent. Corr. %**	**Word Corr. %**
SNR 0dB	9.89	28.70	24.48	60.03	38.26	74.60	43.66	79.25
SNR 5dB	22.78	47.91	42.26	76.52	57.14	86.48	62.64	89.11
SNR 10dB	43.46	76.23	56.54	85.73	71.83	92.76	72.03	92.92
SNR 15dB	66.13	90.51	71.23	92.64	80.22	95.35	80.82	95.55
SNR 20dB	77.62	94.94	79.42	95.78	84.52	97.00	86.11	97.34

Table 4.1: Comparative Table for the ASR working with WG noise

Subway	**ASR**		**ASR+SS**		**ASR + PD-MEMLIN**		**SPLICE**		**ASR + PD-MEEMLIN**	
	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %
SNR -5dB	3.40	21.57	10.09	34.22	11.29	37.09	-	35.25	13.29	47.95
SNR 0dB	9.09	29.05	20.18	53.71	27.07	61.88	-	67.76	30.87	69.71
SNR 5dB	17.58	40.45	32.17	70.00	48.15	80.38	-	87.81	51.65	83.40
SNR 10dB	33.07	65.47	50.95	83.23	65.83	90.58	-	93.71	70.13	91.86
SNR 15dB	54.45	84.60	64.84	90.02	78.92	94.98	-	96.78	78.22	94.40
SNR 20dB	72.83	93.40	76.52	94.56	85.91	97.14	-	98.10	86.71	97.30

Table 4.2: Comparative Table for the ASR working with Subway Noise

Babble	**ASR**		**ASR+SS**		**ASR + PD-MEMLIN**		**SPLICE**		**ASR + PD-MEEMLIN**	
	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %
SNR -5dB	4.60	23.08	7.59	29.78	8.49	29.54	-	20.01	6.69	37.79
SNR 0dB	11.29	30.41	15.98	44.49	23.48	55.72	-	53.93	20.08	59.50
SNR 5dB	20.58	44.23	30.37	65.11	48.75	80.55	-	81.80	49.25	83.70
SNR 10dB	40.86	72.85	50.25	80.93	74.93	94.20	-	93.71	69.33	91.48
SNR 15dB	69.03	90.54	69.93	90.56	84.12	96.86	-	97.40	81.32	95.54
SNR 20dB	82.42	96.17	83.52	95.84	88.91	98.09	-	98.43	88.01	97.98

Table 4.3: Comparative Table for the ASR working with Babble Noise

Car	**ASR**		**ASR+SS**		**ASR + PD-MEMLIN**		**SPLICE**		**ASR + PD-MEEMLIN**	
	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %	Sent. Corr. %	Word Corr. %
SNR -5dB	3.10	20.18	10.49	28.87	6.79	25.90	-	30.93	13.89	44.31
SNR 0dB	8.09	26.18	18.58	46.70	23.58	52.67	-	63.91	35.16	70.47
SNR 5dB	14.99	35.34	31.47	66.50	51.95	82.34	-	86.73	58.64	86.30
SNR 10dB	28.77	58.13	54.25	82.72	70.83	92.15	-	94.93	70.93	91.90
SNR 15dB	57.84	84.04	68.03	90.51	82.02	96.16	-	97.85	81.42	95.86
SNR 20dB	78.32	94.61	81.42	95.30	87.01	97.44	-	98.54	87.81	97.77

Table 4.4: Comparative Table for the ASR working with Car Noise

Chapter 5

Conclusions

In this research, speech robustness is being explored in an attempt to improve the performance of the ASR system. Cascading an enhancement technique with an algorithm focused on robustness gives an improvement for the ASR systems. This combination outperforms robustnes techniques developed, as SPLICE and PD-MEMLIN, for highly degraded environments. It was found that the proposed method, PD-MEEMLIN, obtained competitive results for high SNR (from 10 to 20 dB) comparing it with the existing techniques. However for a low SNR (from 0 to -5 dB) the outperformance is observed due to the better results shown for all the environments defined.

The importance of the speech robustness for real environments conditions was defined. Speech robustness is applied to deal with the speech variability among users, in the utterance properties, vocabulary extension and complexity, and environment conditions. The development was focused on the environmental conditions to process and model the noise, and estimate a clean space model closer to the original space. The estimation of the clean space is supported by the proposed speech robustness algorithm to decrement even more the gap between the estimated and the original space. The closer approximation to the clean space ensures a better performance by the integration of the SS and PD-MEMLIN to work in highly degraded environments.

The new algorithm proposed, compared against PD-MEMLIN it is not more complex, because the SS does not employ a probabilistic model to enhance the noisy speech signal used during the process, only needs to obtain the spectrum of the signal to make the subtraction of the power speech signal and noise. This factor along with the SS implementation simplicity were the reason to select this technique to support PD-MEMLIN. For this reason PD-MEEMLIN almost remains with all the characteristics of PD-MEMLIN. The only issue originated by the SS was the configuration of the parameters α and β for every SNR handled for all the environments defined.

The PD-MEEMLIN algorithm, as described in this work, shows a better performance than SS and PD-MEMLIN for a very high degraded speech. This improvement is made by the enhancement of the noisy models employed by PD-MEMLIN, which are close to the original clean model. The gap between the clean and the noisy model, for the very high degraded speech, had been shortened due to the advantages of both techniques. When PD-MEEMLIN is employed the performance is between 11.7% and 24.84% better than PD-MEMLIN, between 11.4% and 34.5% better than SS, and between 6.04% and 22.23% better than SPLICE.

5.1 Future Research Directions

The speech robustnes technique presented in this document shows that the effective improvement of the performance of the ASR system under high-noise conditions is attainable. However, there are some tasks that could be done to improve even more the performance of the proposed method, and to extend the number of environments for the experiments.

5.1.1 Front-end ETSI advanced

Comparing the results against the ones obtained with the front-end ETSI could give a better reference to see how close is the improvement obtained with PD-MEEMLIN to the maximum performance, because the front-end ETSI had been adjust especifically for AURORA. Besides ETSI has been accepting proposals for AURORA to standarize a front-end for speech recognition applications that offers robustnes to noise distorsions.

5.1.2 Environments

The number of environments could be more to have a better reference against more common conditions that are present for the applications. Three environments are being employed at this time, but this number could reach 8 different environments. For the new environments five more noises could be defined under the same filtered noises employed. This number is tentative, because all the noises intended to be used could change the filtering process employed (G712) by anoter one, i.e. MIRS. With this the consideration of a different frequency characteristic need to be considered.

5.1.3 Multi-condition Training

The results obtained for this research had been developed with a clean training for the ASR system. It is possible to change these clean training files by others containing some of the noise employed at this time. This change can give another reference to see how good is the performance of PD-MEEMLIN with the noisy training.

Appendix A

Paper Conference IbPRIA 2007

Igmar Hernández, Juan A. Nolazco, Luis Buera, Eduardo Lleida, and Paola García: Robust Automatic Speech Recognition Using PD-MEEMLIN. Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA) Gerona, Spain, 2007.

Bibliography

[1] S. Boll: Suppression of Acoustic Noise in Speech Using Spectral Subtraction. IEEE Trans ASSP, Vol27, pp. 113–120, 1979.

[2] J. Droppo, L. Deng, and A. Acero: Evaluation of the Splice Algorithm on the Aurora2 Database. In Proc. Eurospeech, Vol. 1, Sep. 2001.

[3] M.J.F. Gales, and S. Young: Cepstral Parameter Compensation for HMM Recognition in Noise. Speech Communication, Vol. 12 Issue 3, pp. 231–239, 1993.

[4] Pedro J. Moreno, Bhiksha Raj, Evandro Gouvea and Richard M. Stern: Multivariate-Gaussian-Based Cepstral Normalization for Robust Speech Recognition. Department of Electrical and Computer Engineering & School of Computer Science. Carnegie Mellon University.

[5] T. Kristjansson, and B.J. Frey: Accounting for Uncertainty in Observations: A new Paradigm for Robust Automatic Speech Recognition. University of Toronto.

[6] B.J. Frey, L. Deng, A. Acero, and T. Kristjansson: ALGONQUIN: Learning dynamic noise models from noisy speech for robust speech recognition. University of Toronto and Speech Technology Group Microsoft Research.

[7] Hynek Hermansky and Nelson Morgan: RASTA Processing of Speech. IEEE Transactions on Speech and Audio Processing, Vol. 2 No. 4, pp. 578–589, October 1994.

[8] J. Nolazco-Flores and S. Young: Continuous Speech Recognition in Noise Using Spectral Subtraction and HMM adaptation. In ICASSP, pages I.409–I.412 (1994).

[9] Hong Kook Kim and Richard C. Rose: Cepstrum-Domain Model Combination Based on Decomposition of Speech and Noise for Noisy Speech Recognition. AT&T Labs-Research, Florham Park, NJ, USA.

[10] L. Buera, E. Lleida, A. Miguel, and A. Ortega: Multienvironment Models Based LInear Normalization for Speech Recognition in Car Conditions. Proc. ICASSP, May 2004.

[11] L. Buera, E. Lleida, A. Miguel, and A. Ortega: Robust Speech Recognition in Cars Using Phoneme Dependent Multienvironment LInear Normalization. In Proceedings of Interspeech. Lisboa, Portugal, 2005, pp. 381-384.

[12] R. Martin: Spectral Subtraction Based on Minimum Statistics. In Proc. Eur. Signal Processing Conf. 1994, pp. 1182-1185.

[13] Xuedong Huang, Alex Acero and Hsiao-Wuen Hon: Spoken Language Processing. Prentice Hall PTR, United States, pp. 504–512, 2001.

[14] Jean-Claude Junqua and Jean-Paul Haton: Robustness in Automatic Speech Recognition, Fundamentals and Applications. 2nd Edition. Kluwer Academic Publishers, Boston Dordrecht London (2000).

[15] R. Martin: Noise Power Spectral Density Estimation Based on Optimal Smoothing and Minimum Statistics. IEEE Transactions on Speech and Audio Processing, Vol. 9, No. 5, July 2000.

[16] M. Berouti, R. Schwartz, and J. Makhoul: Enhancement of Speech Corrupted by Acoustic Noise. Proc. IEEE Conf. ASSP, pp. 208-211, April 1979.

[17] H. G. Hirsch and D. Pearce: The AURORA Experimental Framework for the Performance Evaluations of Speech Recognition Systems Under Noisy Condidions. In ISCA ITRW ASR2000, Automatic Speech Recognition: Challenges for the Next Millennium, Paris, France, September 2000.

[18] HTK-Hidden Markov Model Toolkit home page. http://htk.eng.cam.ac.uk/